SAM L10/L11 Family

Ultra Low-Power, 32-bit Cortex-M23 MCUs with TrustZone, Crypto, and Enhanced PTC

Features

- **Operating Conditions:** 1.62V to 3.63V, -40ºC to +125ºC, DC to 32 MHz
- **Core:** 32 MHz (2.62 CoreMark/MHz and up to 31 DMIPS) ARM® Cortex®-M23 with:
  - Single-cycle hardware multiplier
  - Hardware divider
  - Nested Vector Interrupt Controller (NVIC)
  - Memory Protection Unit (MPU)
  - Stack Limit Checking
  - TrustZone® for ARMv8-M (optional)
- **System**
  - Power-on Reset (POR) and programmable Brown-out Detection (BOD)
  - 8-channel Direct Memory Access Controller (DMAC)
  - 8-channel event system for Inter-peripheral Core-independent Operation
  - CRC-32 generator
- **Memory**
  - 64/32/16 KB Flash
  - 16/8/4 KB SRAM
  - 2 KB Data Flash Write-While-Read (WWR) section for non-volatile data storage
  - 256 bytes TrustRAM with physical protection features
- **Clock Management**
  - Flexible clock distribution optimized for low power
  - 32.768 kHz crystal oscillator
  - 32.768 kHz ultra low-power internal RC oscillator
  - 0.4 to 32 MHz crystal oscillator
  - 16/12/8/4 MHz low-power internal RC oscillator
  - Ultra low-power digital Frequency-Locked Loop (DFLLULP)
  - 48-96 MHz fractional digital Phase-Locked Loop (FDPLL96M)
  - One frequency meter
- **Low Power and Power Management**
  - Active, Idle, Standby with partial or full SRAM retention and off sleep modes:
    - Active mode (< 25 µA/MHz)
    - Idle mode (< 10 µA/MHz) with 1.5 µs wake-up time

Downloaded from Arrow.com.

    - Standby with Full SRAM Retention (0.5 µA) with 5.3 µs wake-up time
    - Off mode (< 100 nA)
  - Static and dynamic power gating architecture
  - Sleepwalking peripherals
  - Two performance levels
  - Embedded Buck/LDO regulator with on-the-fly selection

- **Security**
  - Up to four tamper pins for static and dynamic intrusion detections
  - Data Flash
    - Optimized for secrets storage
    - Data Scrambling with user-defined key (optional)
    - Rapid Tamper erase on scrambling key and on one user-defined row
    - Silent access for side channel attack resistance
  - TrustRAM
    - Address and Data scrambling with user-defined key
    - Chip-level tamper detection on physical RAM to resist microprobing attacks
    - Rapid Tamper Erase on scrambling key and RAM data
    - Silent access for side channel attack resistance
    - Data remanence prevention
  - Peripherals
    - One True Random Generator (TRNG)
    - AES-128, SHA-256, and GCM cryptography accelerators (optional)
    - Secure pin multiplexing to isolate on dedicated I/O pins a secured communication with external devices from the non-secure application (optional)
  - TrustZone for flexible hardware isolation of memories and peripherals (optional)
    - Up to six regions for the Flash
    - Up to two regions for the Data Flash
    - Up to two regions for the SRAM
    - Individual security attribution for each peripheral, I/O, external interrupt line, and Event System Channel
  - Secure Boot with SHA-based authentication (optional)
  - Up to three debug access levels
  - Up to three Chip Erase commands to erase part of or the entire embedded memories
  - Unique 128-bit serial number

- **Advanced Analog and Touch**
  - One 12-bit 1 Msps Analog-to-Digital Converter (ADC) with up to 10 channels
  - Two Analog Comparators (AC) with window compare function
  - One 10-bit 350 kSPS Digital-to-Analog Converter (DAC) with external and internal outputs
  - Three Operational Amplifiers (OPAMP)
  - One enhanced Peripheral Touch Controller (PTC):
    - Up to 20 self-capacitance channels
    - Up to 100 (10 x 10) mutual-capacitance channels
    - Low-power, high-sensitivity, environmentally robust capacitive touch buttons, sliders, and wheels

Downloaded from Arrow.com.

        - Hardware noise filtering and noise signal desynchronization for high conducted immunity
        - Driven Shield Plus for better noise immunity and moisture tolerance
        - Parallel Acquisition through Polarity control
        - Supports wake-up on touch from Standby Sleep mode

- **Communication Interfaces**
    - Up to three Serial Communication Interfaces (SERCOM) that can operate as:
        - USART with full-duplex and single-wire half-duplex configuration
        - I^2C up to 3.4 Mbit/s (High-Speed mode) on one instance and up to 1 Mbit/s (Fast-mode Plus) on the second instance
        - Serial Peripheral Interface (SPI)
        - ISO7816 on one instance
        - RS-485 on one instance
        - LIN Slave on one instance

- **Timers/Output Compare/Input Capture**
    - Three 16-bit Timers/Counters (TC), each configurable as:
        - One 16-bit TC with two compare/capture channels
        - One 8-bit TC with two compare/capture channels
        - One 32-bit TC with two compare/capture channels, by using two TCs
    - 32-bit Real-Time Counter (RTC) with clock/calendar functions
    - Watchdog Timer (WDT) with Window mode

- **Input/Output (I/O)**
    - Up to 25 programmable I/O lines
    - Eight external interrupts (EIC)
    - One non-maskable interrupt (NMI)
    - One Configurable Custom Logic (CCL) that supports:
        - Combinatorial logic functions, such as AND, NAND, OR, and NOR
        - Sequential logic functions, such as Flip-Flop and Latches

- **Qualification and Class-B Support**
    - AEC-Q100 REVH (Grade 1 [-40ºC to +125ºC]) (planned)
    - Class-B safety library, IEC 60730 (future)

- **Debugger Development Support**
    - Two-pin Serial Wire Debug (SWD) programming and debugging interface

- **Packages**

Type	VQFN		TQFP	SSOP	WLCSP(1)
Pin Count	24	32	32	24	32
I/O Pins (up to)	17	25	25	17	25
Contact/Lead Pitch	0.5 mm	0.5 mm	0.8 mm	0.65 mm	0.4 mm
Dimensions	4x4x0.9 mm	5x5x1 mm	7x7x1.2 mm	8.2x5.3x2.0 mm	2.79x2.79x0.482 mm

Note:

1. Contact local sales for availability.

Downloaded from Arrow.com.

Table of Contents

Features 1

1. Configuration Summary 14
2. Ordering Information 16
3. Block Diagram 17
4. Pinouts 18
   - 4.1. Multiplexed Signals 19
   - 4.2. Oscillators Pinout 21
   - 4.3. Serial Wire Debug Interface Pinout 21
   - 4.4. SERCOM Configurations 22
   - 4.5. General Purpose I/O (GPIO) Clusters 23
5. Signal Descriptions List 24
6. Power Considerations 26
   - 6.1. Power Supplies 26
   - 6.2. Power Supply Constraints 26
   - 6.3. Power-On Reset and Brown-Out Detectors 27
   - 6.4. Voltage Regulator 27
   - 6.5. Typical Powering Schematic 27
7. Analog Peripherals Considerations 29
   - 7.1. Reference Voltages 30
   - 7.2. Analog On Demand Feature 30
8. Device Startup 32
   - 8.1. Clocks Startup 32
   - 8.2. Initial Instructions Fetching 32
   - 8.3. I/O Pins 32
   - 8.4. Performance Level Overview 32
9. Product Mapping 34
10. Memories 36
    - 10.1. Embedded Memories 36
    - 10.2. NVM Rows 38
    - 10.3. Serial Number 44
11. Processor and Architecture 45
    - 11.1. Cortex-M23 Processor 45
    - 11.2. Nested Vector Interrupt Controller 47
    - 11.3. High-Speed Bus System 50

Downloaded from Arrow.com.

11.4. SRAM Quality of Service....52

12. Peripherals Configuration Summary....54

13. SAM L11 Security Features....57

13.1. Features....57
13.2. ARM TrustZone Technology for ARMv8-M....58
13.3. Crypto Acceleration....69
13.4. True Random Number Generator (TRNG)....72
13.5. Secure Boot....72
13.6. Secure Pin Multiplexing on SERCOM....72
13.7. Data Flash....72
13.8. TrustRAM (TRAM)....72

14. Boot ROM....73

14.1. Features....73
14.2. Block Diagram....74
14.3. Product Dependencies....74
14.4. Functional Description....74

15. PAC - Peripheral Access Controller....96

15.1. Overview....96
15.2. Features....96
15.3. Block Diagram....96
15.4. Product Dependencies....96
15.5. Functional Description....98
15.6. Register Summary....102
15.7. Register Description....103

16. DSU - Device Service Unit....127

16.1. Overview....127
16.2. Features....127
16.3. Block Diagram....128
16.4. Signal Description....128
16.5. Product Dependencies....128
16.6. Debug Operation....130
16.7. Programming....131
16.8. Security Enforcement....132
16.9. Device Identification....134
16.10. Functional Description....135
16.11. Register Summary....141
16.12. Register Description....143

17. Clock System....172

17.1. Clock Distribution....172
17.2. Synchronous and Asynchronous Clocks....173
17.3. Register Synchronization....174
17.4. Enabling a Peripheral....177

Downloaded from Arrow.com.

17.5. On Demand Clock Requests....177
17.6. Power Consumption vs. Speed....178
17.7. Clocks after Reset....178

18. GCLK - Generic Clock Controller....179
18.1. Overview....179
18.2. Features....179
18.3. Block Diagram....179
18.4. Signal Description....180
18.5. Product Dependencies....180
18.6. Functional Description....181
18.7. Register Summary....187
18.8. Register Description....189

19. MCLK – Main Clock....199
19.1. Overview....199
19.2. Features....199
19.3. Block Diagram....199
19.4. Signal Description....199
19.5. Product Dependencies....199
19.6. Functional Description....201
19.7. Register Summary....206
19.8. Register Description....206

20. FREQM – Frequency Meter....222
20.1. Overview....222
20.2. Features....222
20.3. Block Diagram....222
20.4. Signal Description....222
20.5. Product Dependencies....222
20.6. Functional Description....224
20.7. Register Summary....227
20.8. Register Description....227

21. RSTC – Reset Controller....237
21.1. Overview....237
21.2. Features....237
21.3. Block Diagram....237
21.4. Signal Description....237
21.5. Product Dependencies....237
21.6. Functional Description....239
21.7. Register Summary....241
21.8. Register Description....241

22. PM – Power Manager....243
22.1. Overview....243
22.2. Features....243
22.3. Block Diagram....244

Downloaded from Arrow.com.

22.4. Signal Description ... 244
22.5. Product Dependencies ... 244
22.6. Functional Description ... 245
22.7. Register Summary ... 263
22.8. Register Description ... 263

23. OSCCTRL – Oscillators Controller ... 271
23.1. Overview ... 271
23.2. Features ... 271
23.3. Block Diagram ... 272
23.4. Signal Description ... 272
23.5. Product Dependencies ... 272
23.6. Functional Description ... 274
23.7. Register Summary ... 285
23.8. Register Description ... 286

24. OSC32KCTRL – 32KHz Oscillators Controller ... 319
24.1. Overview ... 319
24.2. Features ... 319
24.3. Block Diagram ... 320
24.4. Signal Description ... 320
24.5. Product Dependencies ... 320
24.6. Functional Description ... 322
24.7. Register Summary ... 327
24.8. Register Description ... 327

25. SUPC – Supply Controller ... 339
25.1. Overview ... 339
25.2. Features ... 339
25.3. Block Diagram ... 340
25.4. Signal Description ... 340
25.5. Product Dependencies ... 340
25.6. Functional Description ... 341
25.7. Register Summary ... 348
25.8. Register Description ... 349

26. WDT – Watchdog Timer ... 366
26.1. Overview ... 366
26.2. Features ... 366
26.3. Block Diagram ... 367
26.4. Signal Description ... 367
26.5. Product Dependencies ... 367
26.6. Functional Description ... 368
26.7. Register Summary ... 374
26.8. Register Description ... 374

27. RTC – Real-Time Counter ... 386
27.1. Overview ... 386

Downloaded from Arrow.com.

27.2. Features....386
27.3. Block Diagram....387
27.4. Signal Description....388
27.5. Product Dependencies....388
27.6. Functional Description....390
27.7. Register Summary - Mode 0 - 32-Bit Counter....402
27.8. Register Description - Mode 0 - 32-Bit Counter....403
27.9. Register Summary - Mode 1 - 16-Bit Counter....426
27.10. Register Description - Mode 1 - 16-Bit Counter....427
27.11. Register Summary - Mode 2 - Clock/Calendar....450
27.12. Register Description - Mode 2 - Clock/Calendar....451

28. DMAC – Direct Memory Access Controller....476
28.1. Overview....476
28.2. Features....476
28.3. Block Diagram....478
28.4. Signal Description....478
28.5. Product Dependencies....478
28.6. Functional Description....480
28.7. Register Summary....500
28.8. Register Description....501
28.9. Register Summary - SRAM....532
28.10. Register Description - SRAM....532

29. EIC – External Interrupt Controller....540
29.1. Overview....540
29.2. Features....540
29.3. Block Diagram....540
29.4. Signal Description....541
29.5. Product Dependencies....541
29.6. Functional Description....543
29.7. Register Summary....550
29.8. Register Description....551

30. NVMCTRL – Nonvolatile Memory Controller....572
30.1. Overview....572
30.2. Features....572
30.3. Block Diagram....573
30.4. Signal Description....573
30.5. Product Dependencies....573
30.6. Functional Description....575
30.7. Register Summary....586
30.8. Register Description....587

31. TRAM - TrustRAM....614
31.1. Overview....614
31.2. Features....614
31.3. Block Diagram....614

Downloaded from Arrow.com.

31.4. Signal Description 614
31.5. Product Dependencies 615
31.6. Functional Description 616
31.7. Register Summary 620
31.8. Register Description 626

32. PORT - I/O Pin Controller 638
32.1. Overview 638
32.2. Features 638
32.3. Block Diagram 639
32.4. Signal Description 639
32.5. Product Dependencies 639
32.6. Functional Description 641
32.7. Register Summary 648
32.8. Register Description 650

33. EVSYS – Event System 685
33.1. Overview 685
33.2. Features 685
33.3. Block Diagram 686
33.4. Product Dependencies 686
33.5. Functional Description 688
33.6. Register Summary 695
33.7. Register Description 698

34. SERCOM – Serial Communication Interface 724
34.1. Overview 724
34.2. Features 724
34.3. Block Diagram 725
34.4. Signal Description 725
34.5. Product Dependencies 725
34.6. Functional Description 727

35. SERCOM USART - SERCOM Synchronous and Asynchronous Receiver and Transmitter 733
35.1. Overview 733
35.2. USART Features 733
35.3. Block Diagram 734
35.4. Signal Description 734
35.5. Product Dependencies 734
35.6. Functional Description 736
35.7. Register Summary 751
35.8. Register Description 752

36. SERCOM SPI – SERCOM Serial Peripheral Interface 778
36.1. Overview 778
36.2. Features 778
36.3. Block Diagram 779

Downloaded from Arrow.com.

36.4. Signal Description ... 779
36.5. Product Dependencies ... 779
36.6. Functional Description ... 781
36.7. Register Summary ... 790
36.8. Register Description ... 791

37. SERCOM I²C – SERCOM Inter-Integrated Circuit ... 811
37.1. Overview ... 811
37.2. Features ... 811
37.3. Block Diagram ... 812
37.4. Signal Description ... 812
37.5. Product Dependencies ... 812
37.6. Functional Description ... 814
37.7. Register Summary - I2C Slave ... 833
37.8. Register Description - I²C Slave ... 833
37.9. Register Summary - I2C Master ... 851
37.10. Register Description - I²C Master ... 852

38. TC – Timer/Counter ... 873
38.1. Overview ... 873
38.2. Features ... 873
38.3. Block Diagram ... 874
38.4. Signal Description ... 874
38.5. Product Dependencies ... 875
38.6. Functional Description ... 876
38.7. Register Description ... 891

39. TRNG – True Random Number Generator ... 965
39.1. Overview ... 965
39.2. Features ... 965
39.3. Block Diagram ... 965
39.4. Signal Description ... 965
39.5. Product Dependencies ... 965
39.6. Functional Description ... 967
39.7. Register Summary ... 969
39.8. Register Description ... 969

40. CCL – Configurable Custom Logic ... 977
40.1. Overview ... 977
40.2. Features ... 977
40.3. Block Diagram ... 978
40.4. Signal Description ... 978
40.5. Product Dependencies ... 978
40.6. Functional Description ... 980
40.7. Register Summary ... 990
40.8. Register Description ... 990

41. ADC – Analog-to-Digital Converter ... 995

Downloaded from Arrow.com.

41.1. Overview ... 995
41.2. Features ... 995
41.3. Block Diagram ... 996
41.4. Signal Description ... 996
41.5. Product Dependencies ... 996
41.6. Functional Description ... 998
41.7. Register Summary ... 1011
41.8. Register Description ... 1012

42. AC – Analog Comparators ... 1040
42.1. Overview ... 1040
42.2. Features ... 1040
42.3. Block Diagram ... 1041
42.4. Signal Description ... 1041
42.5. Product Dependencies ... 1041
42.6. Functional Description ... 1043
42.7. Register Summary ... 1053
42.8. Register Description ... 1053

43. DAC – Digital-to-Analog Converter ... 1071
43.1. Overview ... 1071
43.2. Features ... 1071
43.3. Block Diagram ... 1071
43.4. Signal Description ... 1071
43.5. Product Dependencies ... 1071
43.6. Functional Description ... 1073
43.7. Register Summary ... 1078
43.8. Register Description ... 1078

44. OPAMP – Operational Amplifier Controller ... 1093
44.1. Overview ... 1093
44.2. Features ... 1093
44.3. Block Diagram ... 1094
44.4. Signal Description ... 1094
44.5. Product Dependencies ... 1095
44.6. Functional Description ... 1097
44.7. Register Summary ... 1111
44.8. Register Description ... 1111

45. PTC - Peripheral Touch Controller ... 1120
45.1. Overview ... 1120
45.2. Features ... 1120
45.3. Block Diagram ... 1121
45.4. Signal Description ... 1122
45.5. System Dependencies ... 1122
45.6. Functional Description ... 1124

46. Electrical Characteristics ... 1125

Downloaded from Arrow.com.

46.1. Disclaimer....1125
46.2. Thermal Considerations....1125
46.3. Absolute Maximum Ratings....1126
46.4. General Operating Ratings....1126
46.5. Supply Characteristics....1127
46.6. Maximum Clock Frequencies....1127
46.7. Power Consumption....1129
46.8. Wake-Up Time....1133
46.9. I/O Pin Characteristics....1134
46.10. Injection Current....1135
46.11. Analog Characteristics....1136
46.12. NVM Characteristics....1152
46.13. Oscillators Characteristics....1153
46.14. Timing Characteristics....1160

47. 125°C Electrical Characteristics....1166
47.1. Disclaimer....1166
47.2. General Operating Ratings....1166
47.3. Power Consumption....1166
47.4. Analog Characteristics....1170
47.5. Oscillators Characteristics....1183
47.6. Timing Characteristics....1186

48. AC and DC Characteristics Graphs....1192
48.1. Typical Power Consumption over Temperature in Sleep Modes - 85°C....1192
48.2. Typical Power Consumption over Temperature in Sleep Modes - 125°C....1194

49. Packaging Information....1196
49.1. Package Marking Information....1196
49.2. Package Drawings....1197
49.3. Soldering Profile....1204

50. Schematic Checklist....1205
50.1. Introduction....1205
50.2. Power Supply....1205
50.3. External Analog Reference Connections....1207
50.4. External Reset Circuit....1209
50.5. Unused or Unconnected Pins....1210
50.6. Clocks and Crystal Oscillators....1210
50.7. Programming and Debug Ports....1212
50.8. Peripherals Considerations....1215

51. Conventions....1216
51.1. Numerical Notation....1216
51.2. Memory Size and Type....1216
51.3. Frequency and Time....1216
51.4. Registers and Bits....1217

Downloaded from Arrow.com.

52. Acronyms and Abbreviations....1218

53. Datasheet Revision History....1221

53.1. Rev A - 09/2017....1221

53.2. Rev B - 6/2018....1221

The Microchip Web Site....1222

Customer Change Notification Service....1222

Customer Support....1222

Product Identification System....1223

Microchip Devices Code Protection Feature....1223

Legal Notice....1223

Trademarks....1224

Quality Management System Certified by DNV....1224

Worldwide Sales and Service....1226

Downloaded from Arrow.com.

1. Configuration Summary

Table 1-1. SAM L10/L11 Device-specific Features

Device	Flash + Data Flash Memory (KB)	SRAM (KB)	Pins	SERCOM	ADC Channels	Analog Comparators Inputs	PTC Self-capacitance/ Mutual-capacitance Channels	I/O Pins	Tamper Pins	Packages
SAML10D14	16+2	4	24	2	5	2	16/64	17	3	VQFN, SSOP
SAML10D15	32+2	8								
SAML10D16	64+2	16								
SAML10E14	16+2	4	32	3	10	4	20/100	25	4	VQFN, TQFP, WLCSP
SAML10E15	32+2	8								
SAML10E16	64+2	16								
SAML11D14	16+2	8	24	2	5	2	16/64	17	3	VQFN, SSOP
SAML11D15	32+2	8								
SAML11D16	64+2	16								
SAML11E14	16+2	8	32	3	10	4	20/100	25	4	VQFN, TQFP, WLCSP
SAML11E15	32+2	8								
SAML11E16	64+2	16								

Table 1-2. SAM L10/L11 Family Features

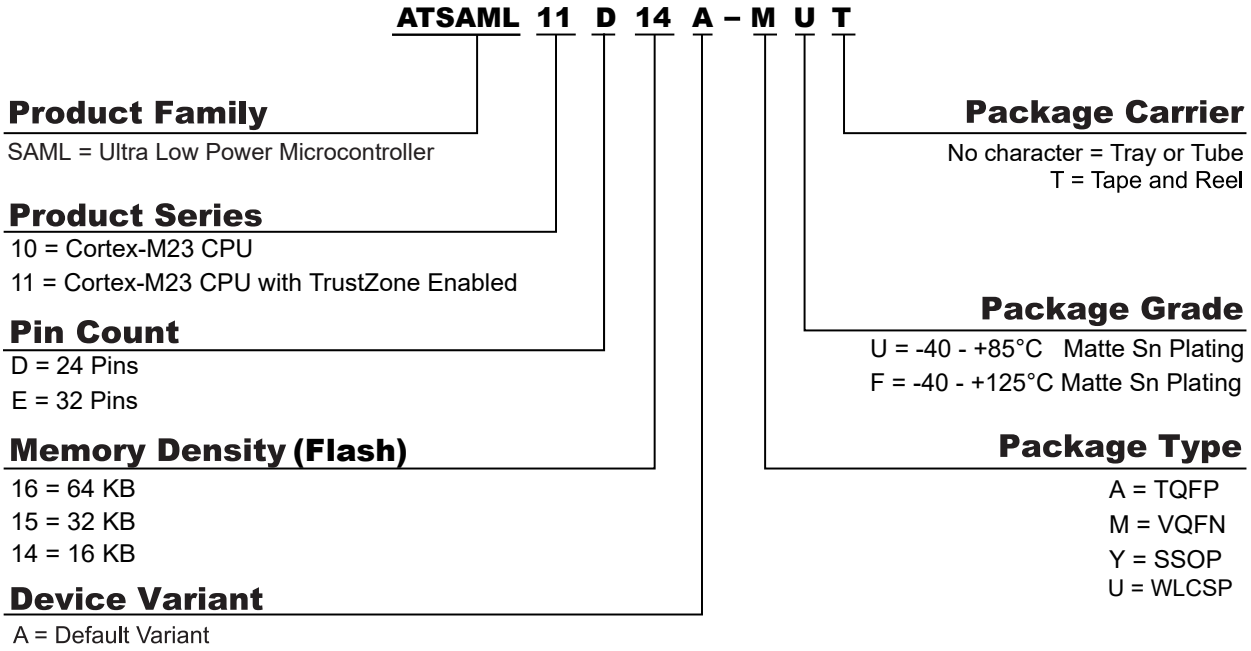
Feature	SAM L10 Family	SAM L11 Family
MPU	1	2
TrustZone for ARMv8-M	No	Yes
Secure Boot	No	Yes
TrustRAM (Bytes)	256	256
DMA Channels	8	8
Data Scrambling	TrustRAM	TrustRAM, Data Flash
Event System Channels	8	8
External Interrupt Lines/NMI	8/1	8/1
Brown-out Detection	V_{DDIO} and V_{DDCORE}	V_{DDIO} and V_{DDCORE}
Secure Pin Multiplexing (on SERCOM)	No	Yes
TC/Compare	3	3
RTC	1	1
Watchdog	1	1
DAC Channels	1	1
OPAMP	3	3
CCL Look-up Tables	2	2
Frequency Meter	1	1
Crypto Accelerators	No	Yes
TRNG	Yes	Yes

Downloaded from Arrow.com.

Feature	SAM L10 Family	SAM L11 Family
CRC	Yes	Yes
Debug Access Levels (DAL)	2	3

Downloaded from Arrow.com.

2. Ordering Information



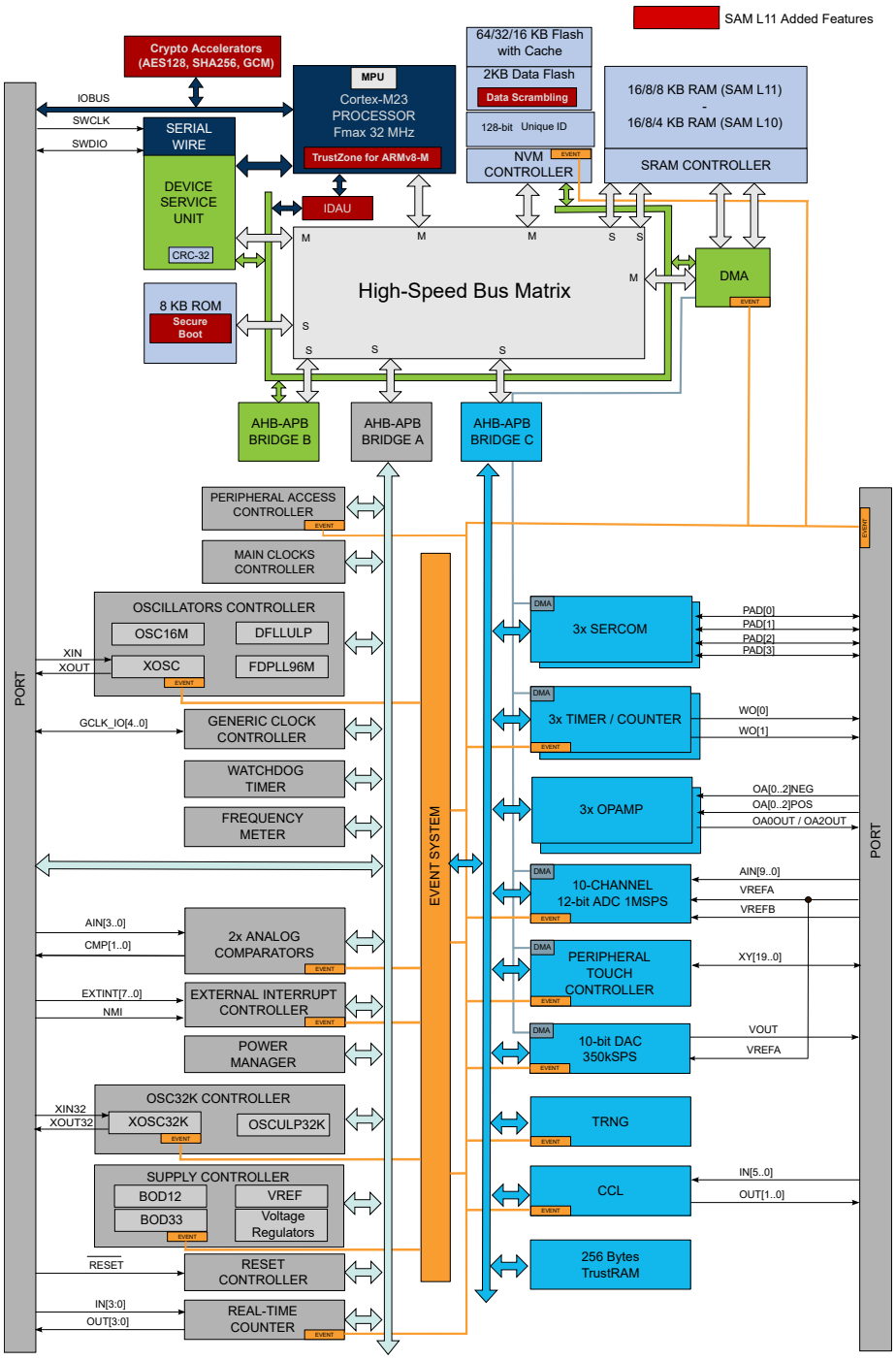
Note:

1. Devices in the WLCSP package include a factory programmed bootloader. Contact your local Microchip sales office for more information.
2. Devices can be factory programmed with securely key provisioned software. Contact your local Microchip sales office for more information.

Downloaded from Arrow.com.

3. Block Diagram

Figure 3-1. SAM L10/L11 Block Diagram



Note: Number of SERCOM instances, PTC/ADC channels, Tamper input pins, and Analog Compare inputs differ on the packages pinout.

Downloaded from Arrow.com.

4. Pinouts

Figure 4-1. SAM L10/L11 24-pin VQFN Pinout

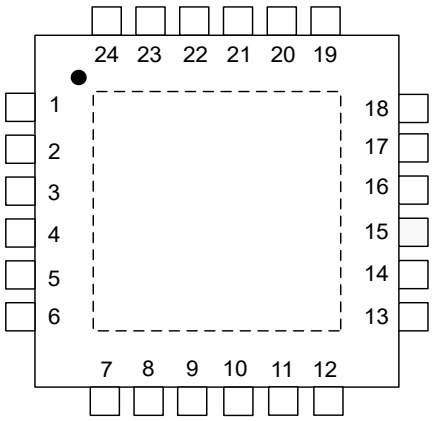


Figure 4-2. SAM L10/L11 24-pin SSOP Pinout

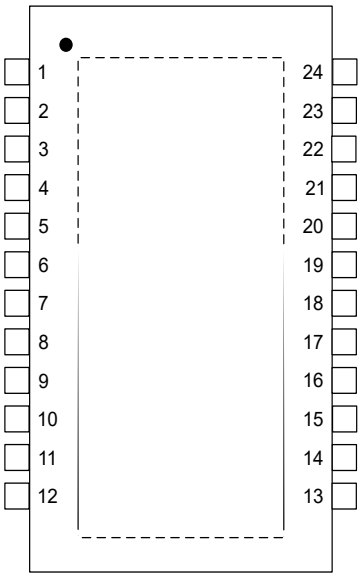
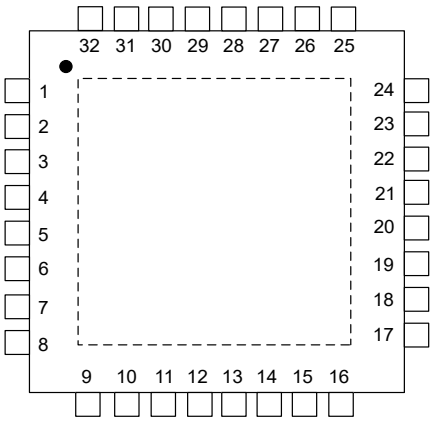
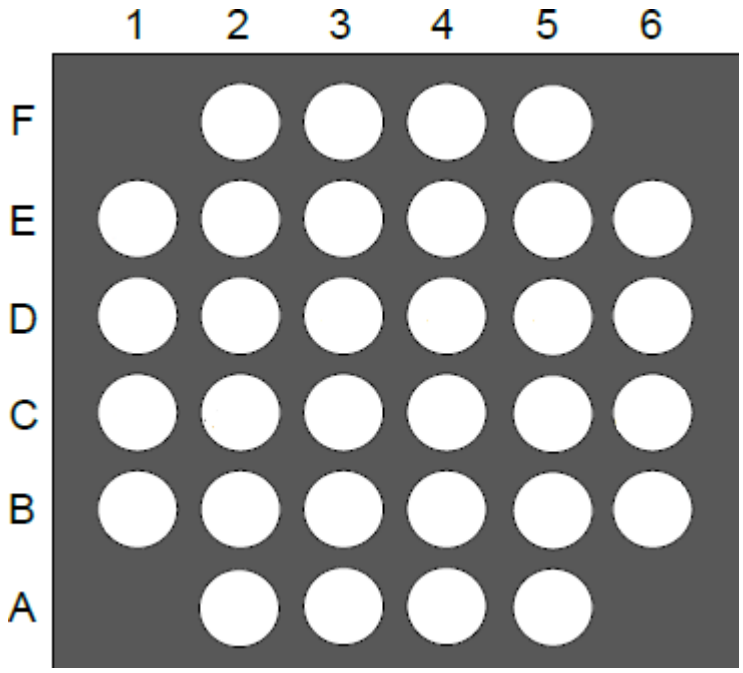


Figure 4-3. SAM L10/L11 32-pin VQFN and TQFP Pinout



Downloaded from Arrow.com.

Figure 4-4. SAM L10/L11 32-pin WLCSP Pinout



4.1 Multiplexed Signals

Each pin is controlled by the I/O Pin Controller (PORT) as a general purpose I/O and alternatively can be assigned to one of the peripheral functions: A, B, C, D, E, G, H, or I.

The following table describes the peripheral signals multiplexed to the PORT I/O pins.

The column “Reset State” indicates the reset state of the line with mnemonics:

- "I/O" or "Function" indicates whether the I/O pin resets in I/O mode or in peripheral function mode.
- “I” / ”O” / "Hi-Z" indicates whether the I/O is configured as an input, output or is tri-stated.
- “PU” / “PD” indicates whether pullup, pulldown or nothing is enabled.

Table 4-1. Pinout Multiplexing

Pin				Pin Name	Supply	A	B(1)						C(2)(3)	D(2)(3)	E	G	H	I	Reset State
SSOP24	VQFN24	WLCSP32	TQFP32 / VQFN32			EIC	REF	ADC	AC	PTC	DAC	OPAMP	SERCOM	SERCOM ALTERNATIVE	TC	RTC/ Debug	AC/ GCLK	CCL	
5	2	A2	1	PA00 / XIN32	VDDANA	EXTINT[0]				XY[0]		OA1NEG		SERCOM1/ PAD[0]	TC2/ WO[0]				I/O, Hi-Z
6	3	A3	2	PA01 / XOUT32	VDDANA	EXTINT[1]				XY[1]		OA1POS		SERCOM1/ PAD[1]	TC2/ WO[1]				I/O, Hi-Z
7	4	A4	3	PA02	VDDANA	EXTINT[2]		AIN[0]		XY[2]	VOUT	OA0NEG		SERCOM0/ PAD[2]					I/O, Hi-Z
8	5	B3	4	PA03	VDDANA	EXTINT[3]	VREFA	AIN[1]		XY[3]		OA2NEG		SERCOM0/ PAD[3]					I/O, Hi-Z
9	6	B4	5	PA04	VDDANA	EXTINT[4]	VREFB	AIN[2]	AIN[0]			OA2OUT		SERCOM0/ PAD[0]	TC0/ WO[0]			IN[0]	I/O, Hi-Z
10	7	A5	6	PA05	VDDANA	EXTINT[5]		AIN[3]	AIN[1]	XY[4]		OA2POS		SERCOM0/ PAD[1]	TC0/ WO[1]			IN[1]	I/O, Hi-Z

Downloaded from Arrow.com.

Pin				Pin Name	Supply	A	B[1]						C[2][3]	D[2][3]	E	G	H	I	Reset State
SSOP24	VQFN24	WLCSP32	TQFP32 / VQFN32			EIC	REF	ADC	AC	PTC	DAC	OPAMP	SERCOM	SERCOM ALTERNATIVE	TC	RTC/ Debug	AC/ GCLK	CCL	
		C4	7	PA06	VDDANA	EXTINT[6]		AIN[4]	AIN[2]	XY[5]		OA0POS		SERCOM0/ PAD[2]	TC1/ WO[0]			IN[2]	I/O, Hi-Z
		B5	8	PA07	VDDANA	EXTINT[7]		AIN[5]	AIN[3]			OA0OUT		SERCOM0/ PAD[3]	TC1/ WO[1]			OUT[0]	I/O, Hi-Z
11	8	B6	9	VDDANA															-
12	9	C6	10	GNDANA															-
13	10	D4	11	PA08	VDDIO	NMI		AIN[6]		XY[6]			SERCOM1/ PAD[0]	SERCOM2/ PAD[0]		RTC/ IN[0]		IN[3]	I/O, Hi-Z
		D6	12	PA09	VDDIO	EXTINT[0]		AIN[7]		XY[7]			SERCOM1/ PAD[1]	SERCOM2/ PAD[1]		RTC/ IN[1]		IN[4]	I/O, Hi-Z
		C5	13	PA10	VDDIO	EXTINT[1]		AIN[8]		XY[8]			SERCOM1/ PAD[2]	SERCOM2/ PAD[2]			GCLK_IO[4]	IN[5]	I/O, Hi-Z
		D5	14	PA11	VDDIO	EXTINT[2]		AIN[9]		XY[9]			SERCOM1/ PAD[3]	SERCOM2/ PAD[3]			GCLK_IO[3]	OUT[1]	I/O, Hi-Z
14	11	E6	15	PA14 / XOSC	VDDIO	EXTINT[3]				XY[10]			SERCOM2/ PAD[2]	SERCOM0/ PAD[2]	TC0/ WO[0]		GCLK_IO[0]		I/O, Hi-Z
15	12	E5	16	PA15 / XOUT	VDDIO	EXTINT[4]				XY[11]			SERCOM2/ PAD[3]	SERCOM0/ PAD[3]	TC0/ WO[1]		GCLK_IO[1]		I/O, Hi-Z
16	13	D3	17	PA16[4]	VDDIO	EXTINT[5]				XY[12]			SERCOM1/ PAD[0]	SERCOM0/ PAD[0]		RTC/ IN[2]	GCLK_IO[2]	IN[0]	I/O, Hi-Z
17	14	F5	18	PA17[4]	VDDIO	EXTINT[6]				XY[13]			SERCOM1/ PAD[1]	SERCOM0/ PAD[1]		RTC/ IN[3]	GCLK_IO[3]	IN[1]	I/O, Hi-Z
18	15	E4	19	PA18	VDDIO	EXTINT[7]				XY[14]			SERCOM1/ PAD[2]	SERCOM0/ PAD[2]	TC2/ WO[0]	RTC/ OUT[0]	AC/ CMP[0]	IN[2]	I/O, Hi-Z
19	16	E3	20	PA19	VDDIO	EXTINT[0]				XY[15]			SERCOM1/ PAD[3]	SERCOM0/ PAD[3]	TC2/ WO[1]	RTC/ OUT[1]	AC/ CMP[1]	OUT[0]	I/O, Hi-Z
20	17	F4	21	PA22[4]	VDDIO	EXTINT[1]				XY[16]			SERCOM0/ PAD[0]	SERCOM2/ PAD[0]	TC0/ WO[0]	RTC/ OUT[2]	GCLK_IO[2]		I/O, Hi-Z
21	18	F3	22	PA23[4]	VDDIO	EXTINT[2]				XY[17]			SERCOM0/ PAD[1]	SERCOM2/ PAD[1]	TC0/ WO[1]	RTC/ OUT[3]	GCLK_IO[1]		I/O, Hi-Z
		F2	23	PA24	VDDIO	EXTINT[3]							SERCOM0/ PAD[2]	SERCOM2/ PAD[2]	TC1/ WO[0]				I/O, Hi-Z
		E2	24	PA25	VDDIO	EXTINT[4]							SERCOM0/ PAD[3]	SERCOM2/ PAD[3]	TC1/ WO[1]				I/O, Hi-Z
		D2	25	PA27	VDDIO	EXTINT[5]											GCLK_IO[0]		I/O, Hi-Z
22	19	C2	26	$\overline{\text{RESET}}$	VDDIO														I, PU
23	20	E1	27	VDDCORE															-
24	21	D1	28	GND															-
1	22	C1	29	VDDOUT															-
2	23	B1	30	VDDIO															-

Downloaded from Arrow.com.

Pin				Pin Name	Supply	A	B(1)						C(2)(3)	D(2)(3)	E	G	H	I	Reset State
SSOP24	VQFN24	WLCSP32	TQFP32 / VQFN32			EIC	REF	ADC	AC	PTC	DAC	OPAMP	SERCOM	SERCOM ALTERNATIVE	TC	RTC/ Debug	AC/ GCLK	CCL	
3	24	B2	31	PA30 / SWCLK	VDDIO	EXTINT[6]				XY[18]				SERCOM1/ PAD[2]	TC1/ WO[0]	SWCLK	GCLK_IO[0]	IN[3]	SWCLK, I, PU
4	1	C3	32	PA31 / SWDIO(4)	VDDIO	EXTINT[7]				XY[19]				SERCOM1/ PAD[3]	TC1/ WO[1]			OUT[1]	I/O, Hi-Z

1. All analog pin functions are on the peripheral function B. The peripheral function B must be selected to disable the digital control of the pin.
2. Refer to SERCOM Configurations to get the list of the supported features for each SERCOM instance.
3. 24-pin packages only have two SERCOM instances: SERCOM0 and SERCOM1.
4. The following pins are High Sink pins and have different properties than standard pins: PA16, PA17, PA22, PA23 and PA31.

4.2 Oscillators Pinout

The oscillators are not mapped to the I/O Pin Controller (PORT) functions and their multiplexing is controlled by the Oscillators Controller (OSCCTRL) and 32 kHz Oscillators Controller (OSC32KCTRL) registers.

Table 4-2. Oscillator Pinout

Oscillator	Supply	Signal	I/O pin
XOSC	VDDIO	XIN	PA14
		XOUT	PA15
XOSC32K	VDDANA	XIN32	PA00
		XOUT32	PA01

To improve the cycle-to-cycle jitter of the XOSC32 oscillator, it is recommended to keep the neighboring pins of XIN32 and the following pins of XOUT32 as static as possible:

Table 4-3. XOSC32 Jitter Minimization

Package Pin Count	Static Signal Recommended
32	PA02, PA03
24	PA02, PA03

4.3 Serial Wire Debug Interface Pinout

The SWCLK pin is by default assigned to the SWCLK peripheral function G to allow debugger probe detection.

A debugger probe detection (cold-plugging or hot-plugging) will automatically switch the SWDIO I/O pin to the SWDIO function, as long as the SWLCK peripheral function is selected.

Downloaded from Arrow.com.

Table 4-4. Serial Wire Debug Interface Pinout

Signal	Supply	I/O pin
SWCLK	VDDIO	PA30
SWDIO	VDDIO	PA31

4.4 SERCOM Configurations

The following table lists the supported features for each SERCOM instance:

Table 4-5. SERCOM Features Summary

	SERCOM Instance		
Protocol	**SERCOM0**	**SERCOM1**	**SERCOM2**
SPI	Yes	Yes	Yes
I^2C (1)	Yes High-speed mode (≤ 3,4 Mbit/s)	Yes Fast plus Mode (≤ 1 Mbit/s)	No
USART	Yes including: Hardware Handshaking IrDA	Yes including: Hardware Handshaking IrDA	Yes including: Hardware Handshaking IrDA RS-485 Auto-baud mode LIN Slave ISO7816
USART/SPI Receive Buffer Size	Two-level	Four-level	Two-level
Secure Pin Multiplexing (**SAM L11 only**)	No	Yes	No

Note:

1. I^2C is not supported on all SERCOM pins. Refer to the SERCOM I^2C Pins table for more details.

4.4.1 SERCOM I2C Pins

The following table lists the SERCOM pins which support I^2C:

Table 4-6. SERCOM I^2C Pins

Pin Name	SERCOM0 I^2C Pad Name	SERCOM1 I^2C Pad Name
PA16	SERCOM0/PAD[0]	SERCOM1/PAD[0]
PA17	SERCOM0/PAD[1]	SERCOM1/PAD[1]
PA22	SERCOM0/PAD[0]	N/A
PA23	SERCOM0/PAD[1]	N/A

4.4.2 Secure Pin Multiplexing (on SERCOM) Pins

The Secure Pin Multiplexing feature can be used on dedicated SERCOM I/O pins to isolate a secure communication with external devices from the non-secure application.

Refer to 13.6 Secure Pin Multiplexing on SERCOM for more details.

The following table lists the SERCOM pins that support the Secure Pin Multiplexing feature:

Downloaded from Arrow.com.

Table 4-7. Secure Pin Multiplexing on SERCOM Pins

Pin Name	Secure Pin Multiplexing Pad Name
PA16	SERCOM1/PAD[0]
PA17	SERCOM1/PAD[1]
PA18	SERCOM1/PAD[2]
PA19	SERCOM1/PAD[3]

4.5 General Purpose I/O (GPIO) Clusters

Table 4-8. GPIO Clusters

Package	Cluster	GPIO	Supply Pins Connected to the Cluster
32-pin	1	PA00 PA01 PA02 PA03 PA04 PA05 PA06 PA07	V_{DDANA}/GNDANA
	2	PA08 PA09 PA10 PA11 PA14 PA15 PA16 PA17 PA18 PA19 PA22 PA23 PA24 PA25 PA27 PA30 PA31	V_{DDIO}/GND
24-pin	1	PA00 PA01 PA02 PA03 PA04 PA05	V_{DDANA}/GND
	2	PA08 PA14 PA15 PA16 PA17 PA18 PA19 PA22 PA23 PA30 PA31	V_{DDIO}/GND

Downloaded from Arrow.com.

5. Signal Descriptions List

The following table provides details on signal names classified by peripherals.

Table 5-1. Signal Descriptions List

Signal Name	Function	Type
Generic Clock Generator - GCLK		
GCLK_IO[4:0]	Generators Clock Source (Input) or Generic Clock Signal (Output)	Digital I/O
Oscillators Control - OSCCTRL		
XIN	Crystal Oscillator or External Clock Input	Analog Input (Crystal Oscillator)/Digital Input (External Clock)
XOUT	Crystal Oscillator Output	Analog Output
32 kHz Oscillators Control - OSC32KCTRL		
XIN32	32.768 kHz Crystal Oscillator or External Clock Input	Analog Input (Crystal Oscillator)/Digital Input (External Clock)
XOUT32	32.768 kHz Crystal Oscillator Output	Analog Output
Serial Communication Interface - SERCOMx		
PAD[3:0]	General SERCOM Pins	Digital I/O
Timer Counter - TCx		
WO[1:0]	Capture Inputs or Waveform Outputs	Digital I/O
Real Timer Clock - RTC		
IN[3:0]	Tamper Detection Inputs	Digital Input
OUT[3:0]	Tamper Detection Outputs	Digital Output
Analog Comparators - AC		
AIN[3:0]	AC Comparator Inputs	Analog Input
CMP[1:0]	AC Comparator Outputs	Digital Output
Analog Digital Converter - ADC		
AIN[9:0]	ADC Input Channels	Analog Input
VREFA(1)	ADC External Reference Voltage A	Analog Input
VREFB	ADC External Reference Voltage B	Analog Input
Digital Analog Converter - DAC		
VOUT	DAC Voltage Output	Analog Output
VREFA(1)	DAC External Reference Voltage A	Analog Input
Operational Amplifier - OPAMP		
OA[2:0]NEG	OPAMP Negative Inputs	Analog Input
OA[2:0]POS	OPAMP Positive Inputs	Analog Input
OA0OUT / OA2OUT	OPAMP Outputs	Analog Output
Peripheral Touch Controller - PTC		
XY[19:0]	X-lines and Y-lines	Digital Output (X-line) /Analog I/O (Y-line)
Custom Control Logic - CCL		
IN[5:0]	Inputs to lookup table	Digital Output
OUT[1:0]	Outputs from lookup table	Digital Input

Downloaded from Arrow.com.

Signal Name	Function	Type
External Interrupt Controller - EIC		
EXTINT[7:0]	External Interrupts Pins	Digital Input
NMI	Non-Maskable Interrupt Pin	Digital Input
General Purpose I/O - PORT		
PA11-PA00 / PA19-PA14 / PA25-PA22 / PA27 / PA31-PA30	General Purpose I/O Pin in Port A	Digital I/O
Reset Controller - RSTC		
$\overline{\text{RESET}}$	External Reset Pin (Active Level: LOW)	Digital Input
Debug Service Unit - DSU		
SWCLK	Serial Wire Clock	Digital Input
SWDIO	Serial Wire Bidirectional Data Pin	Digital I/O

1. VREFA is shared between the ADC and DAC peripherals.

Downloaded from Arrow.com.

6. Power Considerations

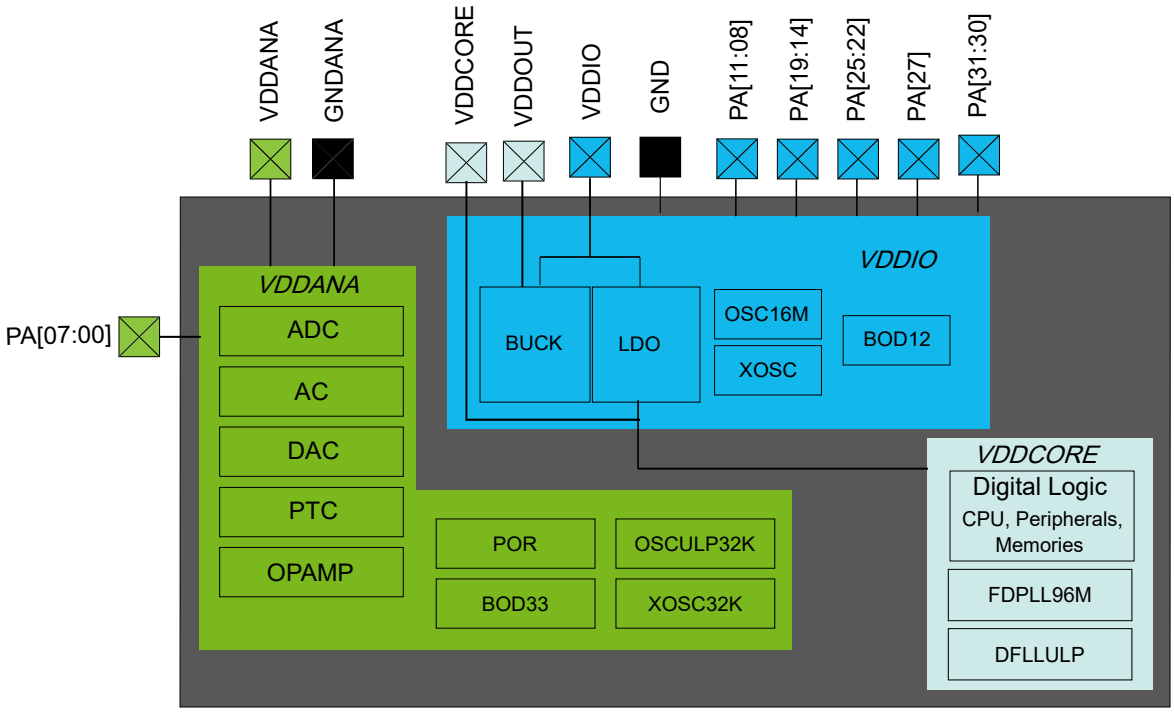
6.1 Power Supplies

The SAM L10/L11 have three different power supply pins:

Table 6-1. SAM L10/L11 Power Supplies

Name	Associated Ground	Powers
VDDIO	GND	OSC16M, XOSC, the internal voltage regulator and BOD12 I/O lines: PA[11:08], PA[19:14], PA[25:22], PA[27] and PA[31:30] Voltage range, nominal: 1.62V - 3.63V, 3.3V
VDDANA	GNDANA	OSCULP32K, XOSC32K, the POR/BOD33, the analog peripherals (ADC, AC, DAC, PTC, OPAMP) I/O lines: PA[07:00] Voltage range, nominal: 1.62V - 3.63V, 3.3V
VDDCORE	GND	Core, embedded memories, peripherals, the FDPLL96M and the DFLLULP Voltage range: 0.9V - 1.2V

Figure 6-1. Power Domain Overview



6.2 Power Supply Constraints

The same voltage source must be applied to both VDDIO and VDDANA.

Note: This common voltage is referred to as VDD in the Data Sheet.

The maximum supply falling and rising rates of the different power supplies must not exceed the values described in the *Supply Characteristics* section of the *Electrical Characteristics* chapters.

Downloaded from Arrow.com.

6.3 Power-On Reset and Brown-Out Detectors

The SAM L10/L11 embed three features to monitor, warn and reset the device:

- A Power-on Reset (POR) on V_{DD} (VDDANA and VDDIO):
  - Monitoring is always activated, including during device startup or during any sleep modes.
  - Having V_{DD} below a fixed threshold voltage will reset the whole device.

  Note: Refer to 46.11.2 Power-On Reset (POR) Characteristics for the rising and falling threshold voltages.
- A Brown-out Detector (BOD33) on V_{DD} (VDDANA and VDDIO):
  - The BOD33 can monitor VDD continuously (continuous mode) or periodically (sampled mode) with a programmable sample frequency in active mode as in any sleep modes.
  - A programmable threshold loaded from the NVM User Row is used to trigger an interrupt and/or reset the whole device.
- A Brown-out Detector (BOD12) on VDDCORE.
  Note: BOD12 is calibrated in production and its calibration parameters are stored in the NVM User Row. These data must not be changed to ensure correct device behavior.

6.4 Voltage Regulator

The embedded voltage regulator is used to provide VDDCORE to the device.

The SAM L10/L11 Voltage Regulator has three modes:

- Linear (LDO) mode: The default mode after reset.
- Switching (BUCK) mode: The most power efficient mode when the CPU and peripherals are running (Active mode).
  Note: In Active mode, the voltage regulator can be selected on the fly between LDO (low-dropout) type regulator and Buck converter using the Supply Controller (SUPC)
- Low-Power mode (LPVREG): The default mode, used when the device is in Standby Sleep mode.

6.5 Typical Powering Schematic

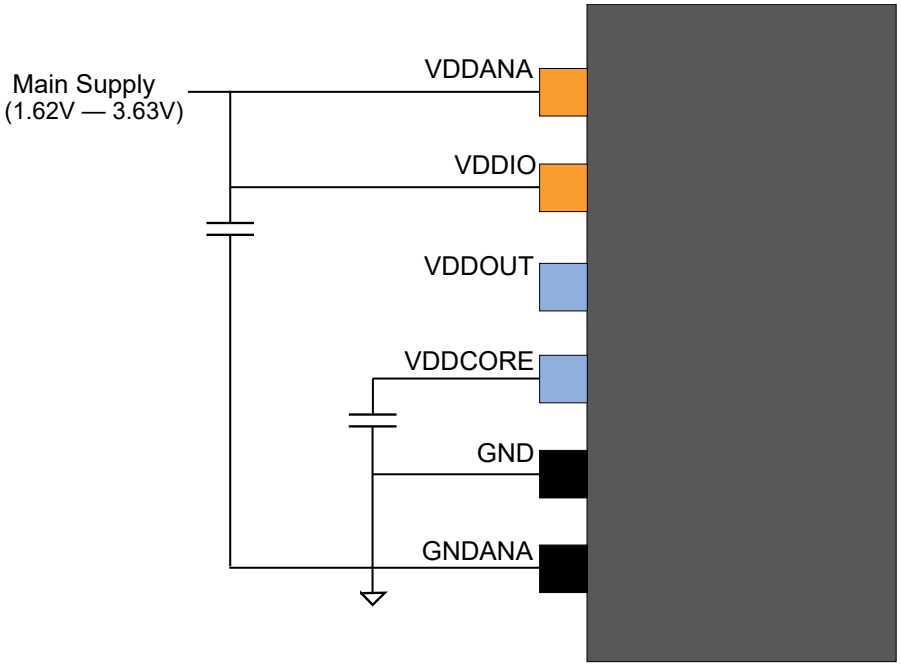
The SAM L10/L11 requires a single supply from 1.62V to 3.63V.

The following figures show the recommended power supply connections for two voltage regulators use cases:

- LDO mode only
- LDO/BUCK modes
  Note: By default the LDO voltage regulator is enabled after any reset. Switching to BUCK mode is then required to benefit from its power efficiency.

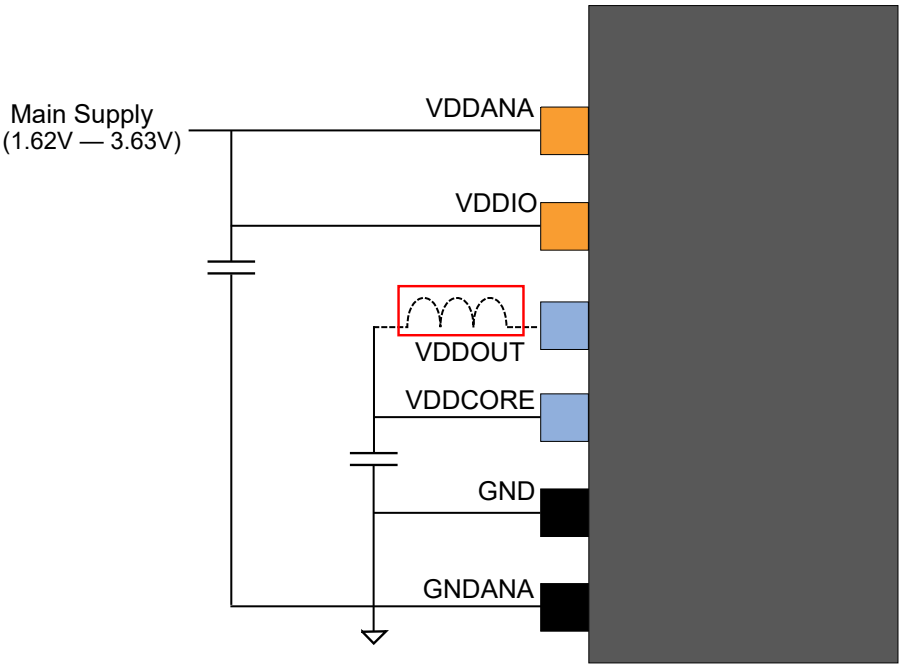
Downloaded from Arrow.com.

Figure 6-2. Power Supply Connections for Linear (LDO) Mode Only



Note: Refer to "Schematic Checklist" chapter for additional information.

Figure 6-3. Power Supply Connections for Switching (BUCK) / Linear (LDO) Modes



Note: Refer to "Schematic Checklist" chapter for additional information.

Downloaded from Arrow.com.

7. Analog Peripherals Considerations

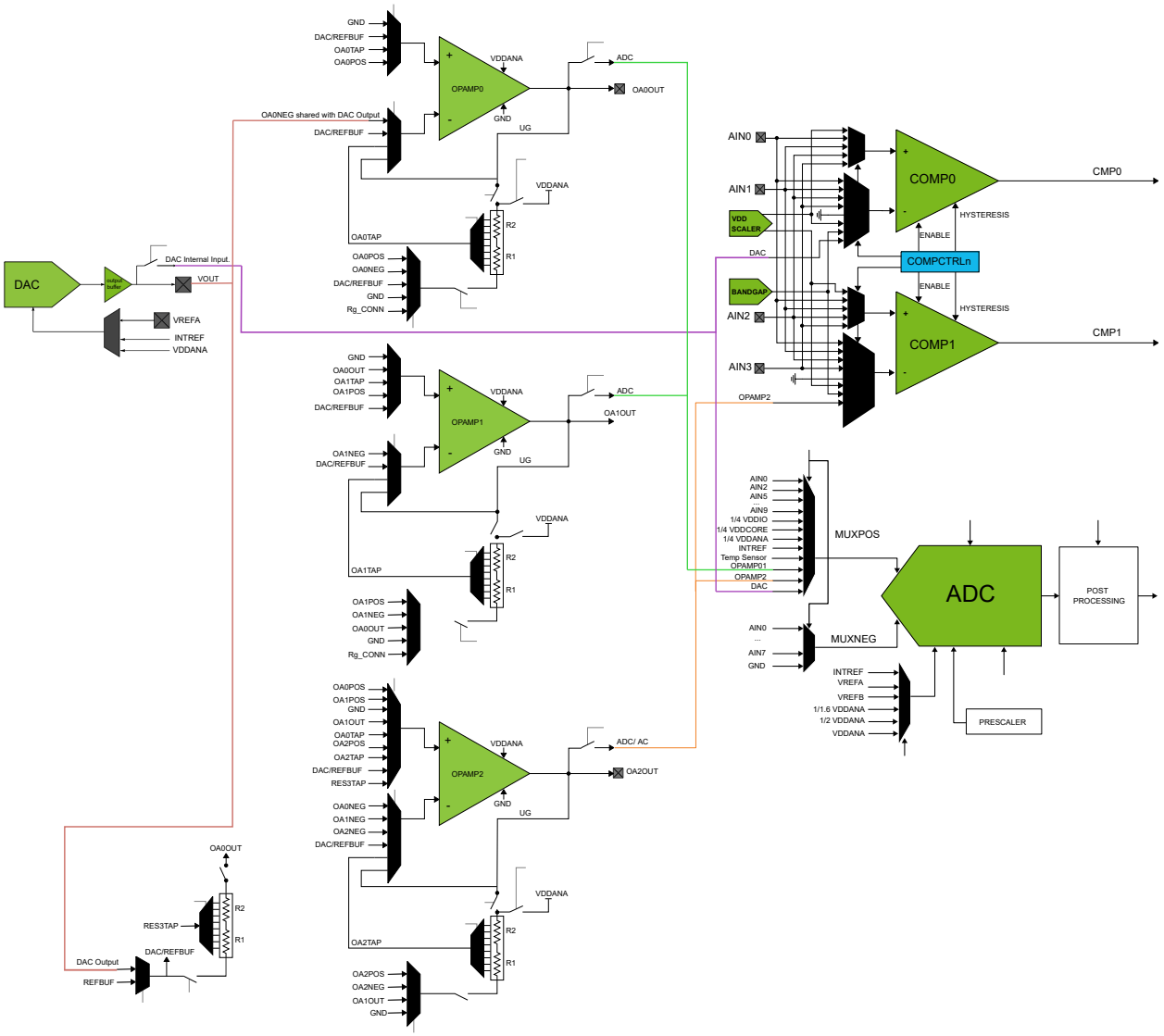
This chapter provides a global view of the analog system, which is composed of the following analog peripherals: AC, ADC, DAC, OPAMP.

The analog peripherals can be connected to each other as illustrated in the following block diagram.

Important:
When an analog peripheral is enabled, each analog output of the peripheral will be prevented from using the alternative functions of the output pads. This is also true even when the peripheral is used for internal purposes.

Analog inputs do not interfere with alternate pad functions.

Figure 7-1. Analog Signal Components Interconnections



Downloaded from Arrow.com.

Note: Some OPAMP Outputs (OAxOUT) can be connected directly to specific Analog Comparator or ADC Inputs (AINx) if they share the same pad: as an example, OA0OUT can be connected to the Analog Comparator AIN3 or ADC AIN5 input (PA07 pin).

7.1 Reference Voltages

Some analog peripherals require a reference voltage for proper operation.

Apart from external voltages (that is, V_{DDANA} or V_{REFx}), the device has a DETREF module that provides two different internal voltage references:

- BANDGAP: A stable voltage reference, fixed at 1.1V.
- INTREF: A variable voltage reference, configured by the Voltage References System Control register in the Supply Controller (SUPC.VREF).

The respective reference voltage source must be selected within each dedicated analog peripheral register:

- ADC: Reference Control register (ADC.REFCTRL)
- DAC: Reference Selection bits in the Control B register (DAC.CTRLB.REFSEL)
  Note: AC has a fixed reference voltage to BANDGAP value.

7.2 Analog On Demand Feature

The Analog On Demand feature allows the ADC and the AC analog peripherals to automatically enable the OPAMPx only when it is needed, thereby allowing a reduction in power consumption. It also allows the ADC analog block to be powered-off when a conversion is completed.
Note: The Analog On Demand is independent from the On Demand Clock request feature, which is used by peripherals to automatically request a source clock which was previously stopped.

OPAMP case
The Analog On Demand feature of the OPAMPx is activated by writing a '1' to the OPAMP.OPAMPCTRLx.ONDEMAND bit.

In that case, the OPAMPx is automatically enabled when the ADC or the AC requests it (as an input) and is automatically disabled when no more requests are coming from these peripherals.

⚠ CAUTION The Analog On Demand feature is not fully supported on cascaded OPAMPs. If several OPAMPs are cascaded together, only the OPAMPx that is connected to the ADC or AC can be enabled/disabled automatically. Upstream OPAMPs will not benefit from this feature.

In Standby Sleep mode, the Analog On Demand feature is still supported if OPAMP.OPAMPCTRLx.RUNSTDBY=1.

If OPAMP.OPAMPCTRLx.RUNSTDBY=0, the OPAMPx will be disabled entering this Sleep mode.

ADC case
For the ADC peripheral, Analog On Demand feature is enabled by writing the ADC.CTRLA.ONDEMAND bit to '1'.

When this feature is activated, the analog block is powered-off when the conversion is complete.

In Sleep mode, when an ADC start request is detected, the analog block is powered-on again and the ADC starts a new conversion after the start-up time delay.

Downloaded from Arrow.com.

Note: If the OPAMPx is set to accept Analog On Demand requests but the ADC is not, the ADC will send continuous requests to the OPAMPx keeping it enabled until the ADC is switching on another input.

AC case

For the AC peripheral,there is no explicit ONDEMAND bit.

Analog On Demand requests are issued either when the AC is used in Single-Shot mode, or when comparisons are triggered by events from the Event System.

Related Links

44. OPAMP – Operational Amplifier Controller
41. ADC – Analog-to-Digital Converter
42. AC – Analog Comparators

Downloaded from Arrow.com.

8. Device Startup

This section summarizes the SAM L10/L11 device startup sequence which starts after device power-up.

After power-up, the device is kept in reset until the power has stabilized throughout the device.

Once VDDIO/VDDANA and VDDCORE voltages reach a stable value, the internal reset is released.

8.1 Clocks Startup

The device selects the OSC16M oscillator which is enabled by default after reset and configured at 4MHz.

This 4MHz clock is also the default time base for the Generic Clock Generator 0 which provides the main clock (CLK_MAIN) to the system through the GLCK_MAIN clock.
Note: Other generic clocks are disabled to optimize power consumption.

Some synchronous clocks require also to be active after startup.

Note: These active synchronous clocks also receive the 4MHz clock from Generic Clock Generator 0.

Refer to the *Clock Mask Register* section in the *Main Clock (MCLK)* chapter to obtain the list of clocks that are running by default.

8.2 Initial Instructions Fetching

After reset is released, the CPU starts fetching from the Boot ROM.

Unless a debugger is connected and places the Boot ROM in a specific mode called Boot Interactive mode, the CPU will jump to the Flash memory loading the Program Counter (PC) and Stack Pointer (SP) values and start fetching flash user code. Before jumping to the Flash, the Boot ROM resets the two first 2kB of SRAM. The Clocks remain unchanged.
Note: SAM L10/L11 Boot Interactive mode allows a debugger to perform several actions on the device such as NVM areas integrity check, chip erase, etc. Refer to 14. Boot ROM for more information.

In addition, the SAM L11 Boot ROM has extra security features, such as device integrity checks, memories and peripherals security attributions, and secure boot that can be executed before jumping to the Flash in Secure state.

8.3 I/O Pins

After reset, the I/O pins are tri-stated except PA30 pin (configured as an input with pull-up enabled) which is by default assigned to the SWCLK peripheral function to allow debugger probe detection.

8.4 Performance Level Overview

The SAM L10/L11 support two different performance levels: PL0 and PL2.

The default performance level after reset is PL0. This performance level is aiming for the lowest power consumption by limiting logic speeds and CPU frequency. As a consequence, some peripherals and clock sources will work with limited capabilities.

Full device functionality and performance will be ensured with PL2 mode.

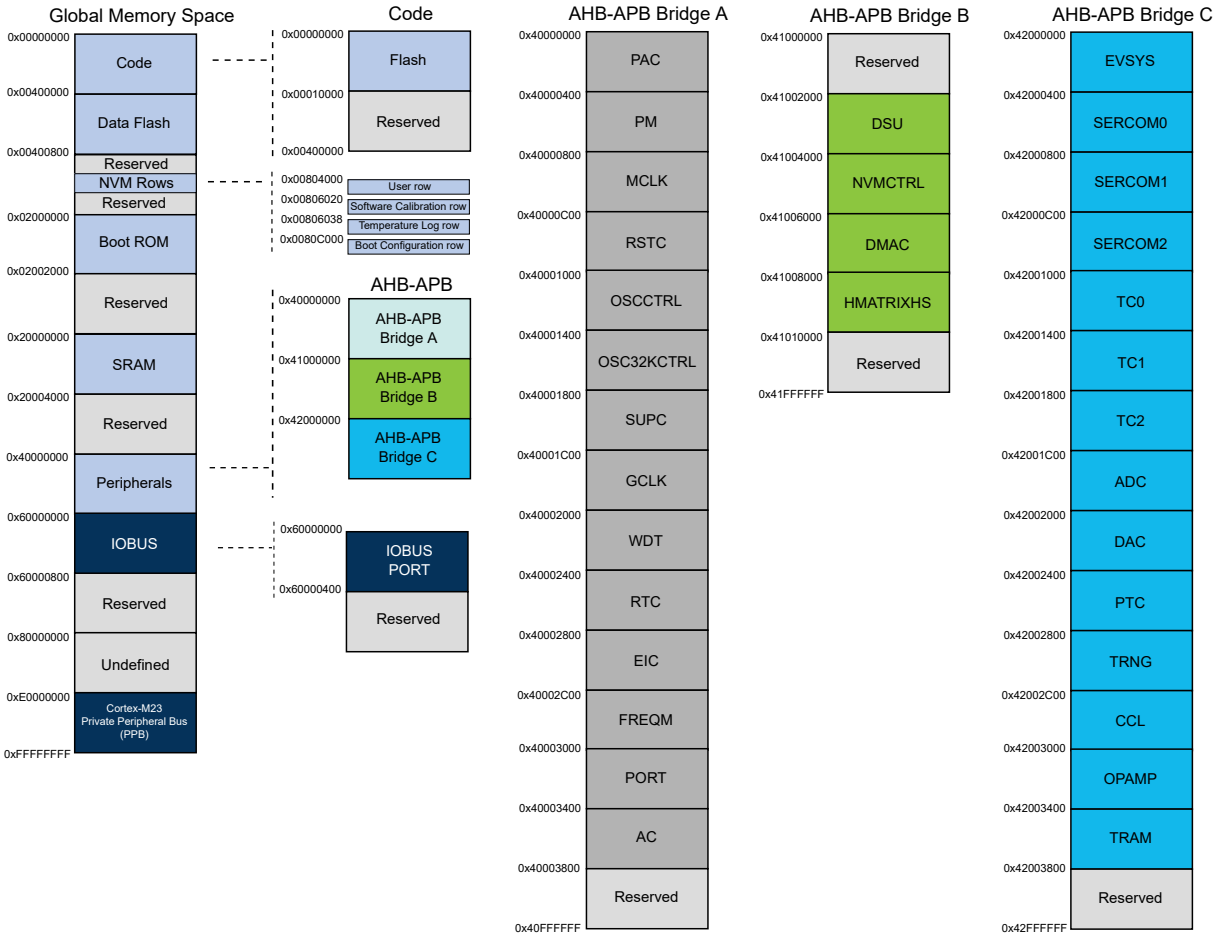
Downloaded from Arrow.com.

Please refer to the Electrical Characteristics sections for more information.

Downloaded from Arrow.com.

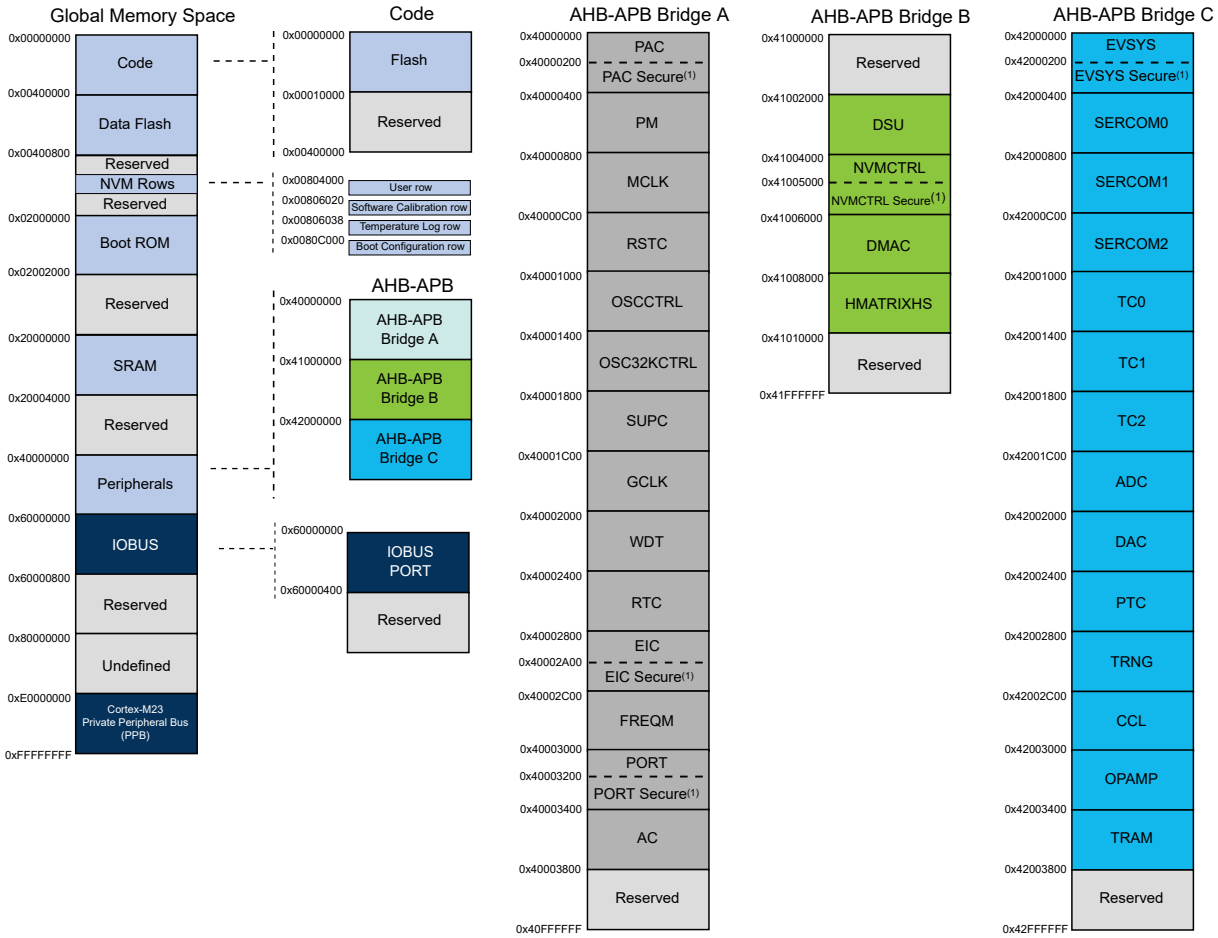
9. Product Mapping

Figure 9-1. SAM L10 Product Mapping



Downloaded from Arrow.com.

Figure 9-2. SAM L11 Product Mapping



Note:

1. This peripheral secure memory region will only appear if the peripheral is secured using PAC. Refer to Mix-Secure Peripherals for details.

Downloaded from Arrow.com.

10. Memories

10.1 Embedded Memories

The 32-bit physical memory address space is mapped as follows:

Table 10-1. Memory Sizes

Memory	Base Address	Size [KB]			
		SAM L11x16(1) SAM L10x16(1)	SAM L11x15(1) SAM L10x15(1)	SAM L11x14 (1)	SAM L10x14 (1)
Flash	0x00000000	64	32	16	16
Data Flash	0x00400000	2	2	2	2
SRAM	0x20000000	16	8	8	4
Boot ROM	0x02000000	8	8	8	8

Note: 1. x = E or D.

10.1.1 Flash

SAM L10/L11 devices embed 16 KB, 32 KB or 64 KB of internal Flash mapped at address 0x0000 0000.

The Flash has a 512-byte (64 lines of 8 bytes) direct-mapped cache which is disabled by default after power up.

The Flash is organized into rows, where each row contains four pages. The Flash has a row-erase and a page-write granularity.

Table 10-2. Flash Memory Parameters

Device	Memory Size [KB]	Number of Rows	Row size [Bytes]	Number of Pages	Page size [Bytes]
SAM L11x16 / SAM L10x16 (1)	64	256	256	1024	64
SAM L11x15 / SAM L10x15 (1)	32	128	256	512	64
SAM L11x14 / SAM L10x14 (1)	16	64	256	256	64

Note:

1. x = E or D.

The Flash is divided in different regions. Each region has a dedicated lock bit preventing from writing and erasing pages on it. Refer to the NVM Memory Organization figures in the *NVMCTRL* chapter to get the different regions definition.

Note: The regions size is configured by the Boot ROM at device startup by reading the NVM Boot Configuration Row (BOCOR). Please refer to the 14. Boot ROM chapter for more information.

Table 10-3. Flash Lock Regions Parameters

Device	SAM L10	SAM L11
Number of FLASH Lock Regions	2	4
Regions Name	Flash Boot / Flash Application	Flash Boot Secure / Flash Boot Non-Secure Flash Application Secure / Flash Application Non-Secure

Downloaded from Arrow.com.

10.1.2 Data Flash

SAM L10/L11 devices embed 2 KB of internal Data Flash with Write-While-Read (WWR) capability mapped at address 0x0040 0000.

The Data Flash can be programmed or erased while reading the Flash memory. It is not possible to read the Data Flash while writing or erasing the Flash.
Note: The Data Flash memory can be executable but requires more cycles to be read which may affect system performance.

The Data Flash cannot be cached.

The Data Flash is organized into rows, where each row contains four pages. The Data Flash has a row-erase and a page-write granularity.

Table 10-4. Data Flash Memory Parameters

Device	Memory Size [KB]	Number of Rows	Row size [Bytes]	Number of Pages	Page size [Bytes]
SAM L10/L11	2	8	256	32	64

The Data Flash is divided into one or two regions. Each region has a dedicated lock bit preventing from writing and erasing pages on it. Refer to the NVM Memory Organization figures in the *NVMCTRL* chapter to obtain the definitions of the different regions.

Note: The regions size is configured by the Boot ROM at device startup by reading the NVM Boot Configuration Row (BOCOR).

Table 10-5. Data Flash Lock Regions Parameters

Device	SAM L10	SAM L11
Number of Data FLASH Lock Regions	1	2
Regions Name	Data Flash	Data Flash Secure / Data Flash Non-Secure

10.1.3 SRAM

SAM L10/L11 devices embed 4 KB, 8 KB, or 16 KB of internal SRAM mapped at address 0x2000 0000.

Table 10-6. SRAM Memory Parameters

Device	Memory Size [KB]
SAM L11x16 / SAM L10x16 (1)	16
SAM L11x15 / SAM L10x15 (1)	8
SAM L11x14 (1)	8
SAM L10x14 (1)	4

Note:

1. x = E or D.

SRAM is composed of 4KB sub-blocks which can be retained or not in STANDBY Low-Power mode to optimize power consumption.

By default, all sub-blocks are retained, but it is possible to switch them off using the Power Manager (PM).

SRAM retention is guaranteed for Watchog, External and System Reset resets. However, the two first 2kB of SRAM are reset by the Boot ROM.

Downloaded from Arrow.com.

Important: SRAM retention is not guaranteed after Power Supply Resets (POR, BOD12 and BOD33).

10.1.4 Boot ROM

SAM L10/L11 devices embed 8 KB of internal ROM mapped at address 0x0200 0000.

Note: Please refer to 14. Boot ROM for more details.

10.2 NVM Rows

SAM L10 and SAM L11 have different Non Volatile Memory (NVM) rows which contain device configuration data that can be used by the system:

Table 10-7. NVM Rows Mapping

NVM Rows	Address
User Row (UROW)	0x00804000
Software Calibration Row	0x00806020
Temperature Log Row	0x00806038
Boot Configuration Row (BOCOR)	0x0080C000

10.2.1 NVM User Row (UROW)

The Non Volatile Memory User Row (UROW) contains device configuration data that are automatically read at device power-on.

This row can be updated using the NVMCTRL peripheral.

When writing to the NVM User Row, the new values are not loaded by the other peripherals on the device until a device reset occurs.

The NVM User Row can be read at the address 0x00804000.

SAM L10 and SAM L11 have different NVM User Row mappings.

Related Links

30. NVMCTRL – Nonvolatile Memory Controller
25. SUPC – Supply Controller
25.8.5 BOD33
26. WDT – Watchdog Timer
26.8.1 CTRLA
26.8.2 CONFIG
26.8.3 EWCTRL

10.2.1.1 SAM L10 User Row

Table 10-8. SAM L10 UROW Bitfields Definition

Bit Pos.	Name	Usage	Factory Setting	Related Peripheral Register
2:0	Reserved	Reserved	Reserved	Reserved
5:3	NSULCK	NVM UnLock Bits	0x7	NVMCTRL.NSULCK

Downloaded from Arrow.com.

Bit Pos.	Name	Usage	Factory Setting	Related Peripheral Register
6	Reserved	Reserved	Reserved	Reserved
12:7	BOD33 Level	BOD33 threshold level at power-on	0x6	SUPC.BOD33
13	BOD33 Disable	BOD33 Disable at power-on	0x0	SUPC.BOD33
15:14	BOD33 Action	BOD33 Action at power-on	0x1	SUPC.BOD33
24:16	BOD12 Calibration Parameters	**DO NOT CHANGE(1)**	**0x08F**	Reserved
25	WDT_RUNSTDBY	WDT Runstdby at power-on	0x0	WDT.CTRLA
26	WDT_ENABLE	WDT Enable at power-on	0x0	WDT.CTRLA
27	WDT_ALWAYSON	WDT Always-On at power-on	0x0	WDT.CTRLA
31:28	WDT_PER	WDT Period at power-on	0xB	WDT.CONFIG
35:32	WDT_WINDOW	WDT Window mode time-out at power-on	0xB	WDT.CONFIG
39:36	WDT_EWOFFSET	WDT Early Warning Interrupt Time Offset at power-on	0xB	WDT.EWCTRL
40	WDT_WEN	WDT Timer Window Mode Enable at power-on	0x0	WDT.CTRLA
41	BOD33_HYST	BOD33 Hysteresis configuration at power-on	0x0	SUPC.BOD33
255:42	Reserved	Reserved	Reserved	Reserved

⚠ CAUTION

1. BOD12 is calibrated in production and its calibration parameters must not be changed to ensure the correct device behavior.

Table 10-9. SAM L10 UROW Mapping

Offset	Bit Pos.	Name							
0x00	7:0	BOD33 Level	-	NSULCK			Reserved		
0x01	15:8	BOD33 Action		BOD33 Disable	BOD33 Level				
0x02	23:16	BOD12 Calibration Parameters							
0x03	31:24	WDT_PER				WDT_ALWAYSON	WDT_ENABLE	WDT_RUNSTDBY	BOD12 Calibration Parameters
0x04	39:32	WDT_EWOFFSET				WDT_WINDOW			
0x05	47:40	Reserved						BOD33_HYST	WDT_WEN
0x06-0x1F	255:48	Reserved							

10.2.1.2 SAM L11 User Row

Table 10-10. SAM L11 UROW Bitfields Definition

Bit Pos.	Name	Usage	Factory Setting	Related Peripheral Register
2:0	SULCK	NVM Secure Region UnLock Bits	0x7	NVMCTRL.SULCK
5:3	NSULCK	NVM Non-Secure Region UnLock Bits	0x7	NVMCTRL.NSULCK
6	Reserved	Reserved	Reserved	Reserved
12:7	BOD33 Level	BOD33 threshold level at power-on.	0x6	SUPC.BOD33
13	BOD33 Disable	BOD33 Disable at power-on	0x0	SUPC.BOD33
15:14	BOD33 Action	BOD33 Action at power-on	0x1	SUPC.BOD33
24:16	BOD12 Calibration Parameters	**Do not change(See Note 1 under Caution)**	**0x08F**	Reserved

Downloaded from Arrow.com.

Bit Pos.	Name	Usage	Factory Setting	Related Peripheral Register
25	WDT_RUNSTDBY	WDT Runstdby at power-on	0x0	WDT.CTRLA
26	WDT_ENABLE	WDT Enable at power-on	0x0	WDT.CTRLA
27	WDT_ALWAYSON	WDT Always-On at power-on	0x0	WDT.CTRLA
31:28	WDT_PER	WDT Period at power-on	0xB	WDT.CONFIG
35:32	WDT_WINDOW	WDT Window mode time-out at power-on	0xB	WDT.CONFIG
39:36	WDT_EWOFFSET	WDT Early Warning Interrupt Time Offset at power-on	0xB	WDT.EWCTRL
40	WDT_WEN	WDT Timer Window Mode Enable at power-on	0x0	WDT.CTRLA
41	BOD33_HYST	BOD33 Hysteresis configuration at power-on	0x0	SUPC.BOD33
42	Reserved	Reserved	Reserved	Reserved
43	RXN	RAM is eXecute Never	0x1	IDAU.SECCTRL
44	DXN	Data Flash is eXecute Never	0x1	NVMCTRL.SECCTRL
63:45	Reserved	Reserved	Reserved	Reserved
71:64	AS	Flash Application Secure Size = AS*0x100	0xFF	IDAU.SCFGA
77:72	ANSC	Flash Application Non-Secure Callable Size = ANSC*0x20	0x0	IDAU.SCFGA
79:78	Reserved	Reserved	Reserved	Reserved
83:80	DS	Data Flash Secure Size = DS*0x100	0x8	IDAU.SCFGA
87:84	Reserved	Reserved	Reserved	Reserved
94:88	RS	RAM Secure Size = RS*0x80	0x7F	IDAU.SCFGR
95	Reserved	Reserved	Reserved	Reserved
96	URWEN	User Row Write Enable	0x1	NVMCTRL.SCFGAD
127:97	Reserved	Reserved	Reserved	Reserved
159:128	NONSECA(1)	Peripherals Non-Secure Status Fuses for Bridge A	0x0000_0000	PAC.NONSECA
191:160	NONSECB(2, 3)	Peripherals Non-Secure Status Fuses for Bridge B	0x0000_0000	PAC.NONSECB
223:192	NONSECC	Peripherals Non-Secure Status Fuses for Bridge C	0x0000_0000	PAC.NONSECC
255:224	USERCRC	CRC of NVM User Row bits 223:64	0x8433651E	Boot ROM

Note:

1. The PAC Peripheral is always secured regardless of its bit value
2. The IDAU and NVMCTRL peripherals are always secured regardless of their bit values.
3. The DSU peripheral is always non-secured regardless of its bit value.

⚠ CAUTION

1. BOD12 is calibrated in production and its calibration parameters must not be changed to ensure the correct device behavior.

Table 10-11. SAM L11 UROW Mapping

Offset	Bit Pos.	Name			
0x00	7:0	BOD33 Level	-	NSULCK	SULCK
0x01	15:8	BOD33 Action		BOD33 Disable	BOD33 Level
0x02	23:16	BOD12 Calibration Parameters			

Downloaded from Arrow.com.

Offset	Bit Pos.	Name					
0x03	31:24	WDT_PER		WDT_ALWAYSON	WDT_ENABLE	WDT_RUNSTDBY	BOD12 Calibration Parameters
0x04	39:32	WDT_EWOFFSET		WDT_WINDOW			
0x05	47:40	Reserved	DXN	RXN	Reserved	BOD33_HYST	WDT_WEN
0x06	55:48	Reserved					
0x07	63:56	Reserved					
0x08	71:64	AS					
0x09	79:72	Reserved	ANSC				
0x0A	87:80	Reserved		DS			
0x0B	95:88	Reserved	RS				
0x0C	103:96	Reserved					URWEN
0x0D-0xF	127:104	Reserved					
0x10-0x13	159:128	NONSECA					
0x14-0x17	191:160	NONSECB					
0x18-0x1B	223:192	NONSECC					
0x1C-0x1F	255:224	USERCRC					

10.2.2 NVM Software Calibration Area

The NVM Software Calibration Area contains calibration data that can be used by some peripherals, such as the ADC.

Note: Calibration data are determined and written during production test and cannot be written.

The NVM Software Calibration Area can be read at address 0x00806020.

Table 10-12. NVM Software Calibration Bitfields Definition

Bit Position	Name	Description
2:0	ADC LINEARITY	ADC Linearity Calibration. Should be written to CALIB register.
5:3	ADC BIASCAL	ADC Bias Calibration. Should be written to CALIB register.
8:6	DFLLULP Division Factor in PL0	DFLLULP Division Factor in PL0. Should be written to DFLLULPCTRL register.
11:9	DFLLULP Division Factor in PL2	DFLLULP Division Factor in PL2. Should be written to DFLLULPCTRL register.
127:12	Reserved	Reserved

Table 10-13. NVM Software Calibration Row Mapping

Offset	Bit Pos.	Name					
0x00	7:0	DFLLULP Division Factor in PL0	ADC BIASCAL		ADC LINEARITY		
0x01	15:8	Reserved		DFLLULP Division Factor in PL2			DFLLULP Division Factor in PL0
0x02-0xF	127:16	Reserved					

10.2.3 NVM Temperature Log Row

The NVM Temperature Log Row contains calibration data that are determined and written during production test and cannot be written.

These calibration values are required for calculating the temperature from measuring the temperature sensor in the Supply Controller (SUPC) by the ADC.

Downloaded from Arrow.com.

The NVM Temperature Log Row can be read at address 0x00806038.

Table 10-14. Temperature Log Row Bitfields Definition

Bit Position	Name	Description
7:0	ROOM_TEMP_VAL_INT	Integer part of room temperature in °C
11:8	ROOM_TEMP_VAL_DEC	Decimal part of room temperature
19:12	HOT_TEMP_VAL_INT	Integer part of hot temperature in °C
23:20	HOT_TEMP_VAL_DEC	Decimal part of hot temperature
31:24	ROOM_INT1V_VAL	2's complement of the internal 1V reference drift at room temperature (versus a 1.0 centered value)
39:32	HOT_INT1V_VAL	2's complement of the internal 1V reference drift at hot temperature (versus a 1.0 centered value)
51:40	ROOM_ADC_VAL	Temperature sensor 12bit ADC conversion at room temperature
63:52	HOT_ADC_VAL	Temperature sensor 12bit ADC conversion at hot temperature

Important: Hot temperature corresponds to the max operating temperature +/- 5%, so 85°C +/- 5% (package grade 'U') or 125°C +/- 5% (package grade 'F').

Table 10-15. Temperature Log Row Mapping

Offset	Bit Pos.	Name	
0x00	7:0	ROOM_TEMP_VAL_INT	
0x01	15:8	HOT_TEMP_VAL_INT	ROOM_TEMP_VAL_DEC
0x02	23:16	HOT_TEMP_VAL_DEC	HOT_TEMP_VAL_INT
0x03	31:24	ROOM_INT1V_VAL	
0x04	39:32	HOT_INT1V_VAL	
0x05	47:40	ROOM_ADC_VAL	
0x06	55:48	HOT_ADC_VAL	ROOM_ADC_VAL
0x07	63:56	HOT_ADC_VAL	

10.2.4 NVM Boot Configuration Row (BOCOR)

The Non-Volatile Memory Boot Configuration Row (BOCOR) contains device configuration data that are automatically read by the Boot ROM program at device startup.

This row can be updated using the NVMCTRL peripheral.

When writing to the NVM Boot Configuration Row, the new values are not loaded by the other peripherals on the device until a device reset occurs.

The NVM Boot Configuration Row can be read at address 0x0080C000.

SAM L10 and SAM L11 have different NVM Boot Configuration Row mappings.

10.2.4.1 SAM L10 Boot Configuration Row

Table 10-16. SAM L10 BOCOR Bitfields Definition

Bit Pos.	Name	Usage	Factory Setting	Related Peripheral Register
31:0	Reserved	Reserved	Reserved	Reserved
39:32	BOOTPROT	Boot Protection size = BOOTPROT*0x100	0x00	Boot ROM

Downloaded from Arrow.com.

Bit Pos.	Name	Usage	Factory Setting	Related Peripheral Register
95:40	Reserved	Reserved	Reserved	Reserved
127:96	ROMVERSION	ROM Code Version	Device Dependent	Boot ROM
511:128	Reserved	Reserved	Reserved	Reserved
639:512	CRCKEY	CRC Key	All '1's	Boot ROM
2047:640	Reserved	Reserved	Reserved	Reserved

Table 10-17. SAM L10 BOCOR Mapping

Offset	Bit Pos.	Name
0x00-0x03	31:0	Reserved
0x04	39:32	BOOTPROT
0x05-0x0B	95:40	Reserved
0x0C-0x0F	127:96	ROMVERSION
0x10-0x3F	511:128	Reserved
0x40-0x4F	639:512	CRCKEY
0x50-0xFF	2047:640	Reserved

10.2.4.2 SAM L11 Boot Configuration Row

Table 10-18. SAM L11 BOCOR Bitfields Definition

Bit Pos.	Name	Usage	Factory Setting	Related Peripheral Register
7:0	Reserved	Reserved	Reserved	Reserved
15:8	BS	Boot Flash Secure Size = BS*0x100	0x00	IDAU.SCFGB
21:16	BNSC	Boot Flash Non-Secure Callable Size = BNSC*0x20	0x00	IDAU.SCFGB
23:22	Reserved	Reserved	Reserved	Reserved
31:24	BOOTOPT	Boot Option	0xA0	Boot ROM
39:32	BOOTPROT	Boot Protection size = BOOTPROT*0x100	0x00	IDAU.SCFGB
47:40	Reserved	Reserved	Reserved	Reserved
48	BCWEN	Boot Configuration Write Enable	0x1	NVMCTRL.SCFGB
49	BCREN	Boot Configuration Read Enable	0x1	NVMCTRL.SCFGB
63:50	Reserved	Reserved	Reserved	Reserved
95:64	BOCORCRC	Boot Configuration CRC for bit 63:0	0xC1D7ECC3	Boot ROM
127:96	ROMVERSION	ROM Code Version	0x0000003A	Boot ROM
255:128	CEKEY0	Chip Erase Key 0	All 1s	Boot ROM
383:256	CEKEY1	Chip Erase Key 1	All 1s	Boot ROM
511:384	CEKEY2	Chip Erase Key 2	All 1s	Boot ROM
639:512	CRCKEY	CRC Key	All 1s	Boot ROM
895:640	BOOTKEY	Secure Boot Key	All 1s	Boot ROM
1791:896	Reserved	Reserved	Reserved	Reserved
2047:1792	BOCORHASH	Boot Configuration Row Hash	All 1s	Boot ROM

Downloaded from Arrow.com.

Table 10-19. SAM L11 BOCOR Mapping

Offset	Bit Pos.	Name			
0x00	7:0	Reserved			
0x01	15:8	BS			
0x02	23:16	Reserved	BNSC		
0x03	31:24	BOOTOPT			
0x04	39:32	BOOTPROT			
0x05	47:40	Reserved			
0x06	55:48	Reserved		BCREN	BCWEN
0x07	63:56	Reserved			
0x08-0x0B	95:64	BOCORCRC			
0x0C-0x0F	127:96	ROMVERSION			
0x10-0x1F	255:128	CEKEY0			
0x20-0x2F	383:256	CEKEY1			
0x30-0x3F	511:384	CEKEY2			
0x40-0x4F	639:512	CRCKEY			
0x50-0x6F	895:640	BOOTKEY			
0x70-0xDF	1791:896	Reserved			
0xE0-0xFF	2047:1792	BOCORHASH			

10.3 Serial Number

Each device has a unique 128-bit serial number which is a concatenation of four 32-bit words contained at the following addresses of the NVM Rows memory space:

- Word 0: 0x0080A00C
- Word 1: 0x0080A040
- Word 2: 0x0080A044
- Word 3: 0x0080A048

Note: The uniqueness of the serial number is only guaranteed when considering all 128 bits.

Downloaded from Arrow.com.

11. Processor and Architecture

11.1 Cortex-M23 Processor

The SAM L10/L11 implement the ARM® Cortex®-M23 processor, based on the ARMv8-M Baseline Architecture, which is the smallest and most energy efficient ARM processor with ARM TrustZone® security technology.

TrustZone® for ARMv8-M provides hardware-enforced security isolation between trusted and the untrusted resources on a Cortex™-M23 based device, while maintaining the efficient exception handling.

The implemented ARM Cortex-M23 is revision **r1p0**.

The ARM Cortex-M23 core has two bus interfaces:

- Single 32-bit AMBA®-5 AHB-Lite system interface that provides connections to peripherals and memories.
- Single 32-bit I/O port bus interfacing to the PORT and Crypto Accelerator peripherals with 1-cycle load and store.

Note: For more information refer to http://www.arm.com

11.1.1 Cortex-M23 Configuration

This table gives the configuration for the ARM Cortex-M23 processor.

Table 11-1. SAM L10/L11 Cortex-M23 Configuration

Features	Cortex-M23 Configurable Options	SAM L10 Implementation	SAM L11 Implementation
Memory Protection Unit (MPU)	Not present, 4, 8, 12, or 16 regions	One MPU with 4 regions	Two MPUs with 4 regions each (one Secure / one Non-Secure)
Security Attribute Unit (SAU)	Absent, 4-region, or 8-region	Absent	Absent
Implementation Defined Attribution Unit (IDAU)	Present or Absent	Absent	Present
SysTick timer(s)	Absent, 1 timer or 2 timers (one Secure and one Non-Secure)	One SysTick timer	Two timers (One Secure / One Non-Secure)
Vector Table Offset Register	Present or absent	Present (one Vector table)	Present (two Vector tables)
Reset all registers	Present or absent	Absent	Absent
Multiplier	Fast (one cycle) or slow (32 cycles)	Fast (one cycle)	Fast (one cycle)
Divider	Fast (17 cycles) or slow (34 cycles)	Fast (17 cycles)	Fast (17 cycles)
Interrupts	External interrupts 0-240	43(1)	45(1)
Instruction fetch width	16-bit only or 32-bit	32-bit	32-bit
Single-cycle I/O port	Present or absent	Present	Present
Architectural clock gating present	Present or absent	Present	Present
Data endianness	Little-endian or big-endian	Little-endian	Little-endian
Halt debug support	Present or absent	Absent	Absent
Wake-up interrupt controller (WIC)	Present or absent	Absent	Absent
Number of breakpoint comparators	0, 1, 2, 3, 4	4	4
Number of watchpoint comparators	0, 1, 2, 3, 4	2	2

Downloaded from Arrow.com.

Features	Cortex-M23 Configurable Options	SAM L10 Implementation	SAM L11 Implementation
Cross Trigger Interface (CTI)	Present or absent	Absent	Absent
Micro Trace Buffer (MTB)	Present or absent	Absent	Absent
Embedded Trace Macrocell (ETM)	Present or absent	Absent	Absent
JTAGnSW debug protocol	Selects between JTAG or Serial-Wire interfaces for the DAP	Serial-Wire	Serial-Wire
Multi-drop for Serial Wire	Present or absent	Absent	Absent

Note:

1. Refer to Table 11-3 for more information.

For more details, refer to the ARM Cortex-M23 Processor Technical Reference Manual (http://www.arm.com).

11.1.2 Cortex-M23 Core Peripherals

The processor has the following core peripheral:

- System Timer (SysTick)
  - The System Timer is a 24-bit timer clocked by the core frequency.

> **Important:** On SAM L11 devices, there are two System timers, one for Secure state and one for Non-secure state.

- Nested Vectored Interrupt Controller (NVIC)
  - The NVIC is an embedded interrupt controller that supports low latency interrupt processing.

> **Important:** On SAM L11 devices, there are two Vector tables: the Secure Vector table and the Non-Secure Vector table.

- System Control Block (SCB)
  - The System Control Block (SCB) provides system implementation information and system control that includes configuration, control, and reporting of system exceptions
- Memory Protection Unit (MPU)
  - The MPU improves system reliability by defining the memory attributes for different memory regions.

> **Important:** On SAM L11 devices, there are two MPUs: one for the Secure state and one for the Non-secure state. Each MPU can define memory access permissions and attributes independently.

- Security Attribution Unit (SAU)
  - The SAU improves system security by defining security attributes for different regions.

Downloaded from Arrow.com.

Important: The SAU is absent from SAM L10 and SAM L11 devices.

For more details, refer to the ARM Cortex-M23 Processor Technical Reference Manual (http://www.arm.com).

Table 11-2. Cortex-M23 Core Peripherals Address Map

Core Peripherals	Base Address (SAM L10 and SAM L11)	(Non-Secure) Alias Base Address (SAM L11 only)
System Timer (SysTick)	0xE000E010	0xE002E010
Nested Vectored Interrupt Controller (NVIC)	0xE000E100	0xE002E100
System Control Block (SCB)	0xE000ED00	0xE002ED00
Memory Protection Unit (MPU)	0xE000ED90	0xE002ED90

11.1.3 Single Cycle I/O Port

The device allows direct access to PORT registers. Accesses to the AMBA® AHB-Lite™ and the single cycle I/O interface can be made concurrently, so the Cortex-M23 processor can fetch the next instructions while accessing the I/Os. This enables single cycle I/O access to be sustained for as long as necessary.
Note: The Crypto Accelerator peripheral also benefits from this port. Refer to the 13.3 Crypto Acceleration section for more information.

11.2 Nested Vector Interrupt Controller

11.2.1 Overview

The Nested Vectored Interrupt Controller (NVIC) in the SAM L10/L11 supports up to 45 interrupt lines with four different priority levels + 1 Non Maskable Interrupt (NMI) line.

For more details, refer to the Cortex-M23 Technical Reference Manual (http://www.arm.com).

11.2.2 Interrupt Line Mapping

Each interrupt line is connected to one peripheral instance, as shown in the table below. Each peripheral can have one or more interrupt flags, located in the peripheral's Interrupt Flag Status and Clear (INTFLAG) register.

An interrupt flag is set when the interrupt condition occurs. Each interrupt in the peripheral can be individually enabled by writing a 1 to the corresponding bit in the peripheral's Interrupt Enable Set (INTENSET) register, and disabled by writing 1 to the corresponding bit in the peripheral's Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated from the peripheral when the interrupt flag is set and the corresponding interrupt is enabled.

The interrupt requests for one peripheral are ORed together on system level, generating one interrupt request for each peripheral. An interrupt request will set the corresponding interrupt pending bit in the NVIC interrupt pending registers (SETPEND/CLRPEND bits in ISPR/ICPR).

For the NVIC to activate the interrupt, it must be enabled in the NVIC interrupt enable register (SETENA/CLRENA bits in ISER/ICER). The NVIC interrupt priority registers IPR0-IPR7 provide a priority field for each interrupt.

Downloaded from Arrow.com.

Table 11-3. Interrupt Line Mapping

Module	Source	NVIC line
EIC NMI – External Interrupt Controller	NMI	NMI
PM – Power Manager	PLRDY	0
MCLK - Main Clock	CKRDY	
OSCCTRL - Oscillators Controller	XOSCRDY	
	XOSCFAIL	
	OSC16MRDY	
	DFLLULPRDY	
	DFLLULPLOCK	
	DFLLULPNOLOCK	
	DPLLLCKR	
	DPLLLCKF	
	DPLLLTO	
	DPLLLDRTO	
OSC32KCTRL - 32KHz Oscillators Controller	XOSC32KRDY	
	CLKFAIL	
SUPC - Supply Controller	BOD33RDY	
	BOD33DET	
	B33SRDY	
	VREGRDY	
	VCORERDY	
	ULPVREFRDY	
WDT – Watchdog Timer	EW	1
RTC – Real Time Counter	CMP0	2
	CMP1	
	OVF	
	PER0	
	PER1	
	PER2	
	PER3	
	PER4	
	PER5	
	PER6	
	PER7	
	TAMPER	
EIC – External Interrupt Controller	EXTINT 0	3
	EXTINT 1	4
	EXTINT 2	5
	EXTINT 3	6
	EXTINT 4..7	7
	NSCHK(1)	
FREQM - Frequency Meter	DONE	8
NVMCTRL – Non-Volatile Memory Controller	DONE	9
	PROGE	

Downloaded from Arrow.com.

Module	Source	NVIC line
	LOCKE	
	NVME	
	KEYE	
	NSCHK(1)	
PORT - I/O Pin Controller	NSCHK(1)	10
DMAC - Direct Memory Access Controller	SUSP 0	11
	TERR 0	
	TCMPL 0	
	SUSP 1	12
	TERR 1	
	TCMPL 1	
	SUSP 2	13
	TERR 2	
	TCMPL 2	
	SUSP 3	14
	TERR 3	
	TCMPL 3	
	SUSP 4..7	15
	TERR 4..7	
	TCMPL 4..7	
EVSYS – Event System	EVD 0	16
	OVR 0	
	EVD 1	17
	OVR 1	
	EVD 2	18
	OVR 2	
	EVD 3	19
	OVR 3	
	NSCHK(1)	20
PAC - Peripheral Access Controller	ERR	21
SERCOM0 – Serial Communication Interface 0 (Interrupt Sources vary depending on SERCOM mode)	Interrupt Bit 0	22
	Interrupt Bit 1	23
	Interrupt Bit 2	24
	Interrupt Bits 3..6	25
SERCOM1 – Serial Communication Interface 1 (Interrupt Sources vary depending on SERCOM mode)	Interrupt Bit 0	26
	Interrupt Bit 1	27
	Interrupt Bit 2	28
	Interrupt Bit 3..6	29
SERCOM2 – Serial Communication Interface 2 (Interrupt Sources vary depending on SERCOM mode)	Interrupt Bit 0	30
	Interrupt Bit 1	31
	Interrupt Bit 2	32
	Interrupt Bits 3..6	33
TC0 – Timer Counter 0	ERR A	34
	MC 0	

Downloaded from Arrow.com.

Module	Source	NVIC line
	MC 1	
	OVF	
TC1 – Timer Counter 1	ERR A	35
	MC 0	
	MC 1	
	OVF	
TC2 – Timer Counter 2	ERR A	36
	MC 0	
	MC 1	
	OVF	
ADC – Analog-to-Digital Converter	OVERRUN	37
	WINMON	
	RESRDY	38
AC – Analog Comparator	COMP 0	39
	COMP 1	
	WIN 0	
DAC – Digital-to-Analog Converter	UNDERRUN	40
	EMPTY	41
PTC – Peripheral Touch Controller	EOC	42
	WCOMP	
TRNG - True Random Number Generator	DATARDY	43
TRAM - Trust RAM	DRP	44
	ERR	

Note:

1. NSCHK interrupt sources will not generate any interrupts for SAM L10 devices.

11.3 High-Speed Bus System

11.3.1 Features

The High-Speed Bus Matrix has the following features:

- 32-bit data bus
- Allows concurrent accesses from different masters to different slaves
- Operation at a one-to-one clock frequency with the bus masters

11.3.2 Configuration

There are two master-to-slave connections to optimize system bandwidth:

- Multi-Slave Masters which are connected through the AHB bus matrix

Table 11-4. AHB Multi-Slave Masters

AHB Multi-Slave Masters
Cortex-M23 Processor
DSU - Device Service Unit
DMAC - Direct Memory Access Controller / Data Access

Downloaded from Arrow.com.

Table 11-5. AHB Slaves

AHB Slaves
Flash Memory
AHB-APB Bridge A
AHB-APB Bridge B
AHB-APB Bridge C
SRAM Port 0 - Cortex-M23 Access
SRAM Port 1 - DMAC Access
SRAM Port 2 - DSU Access
Boot ROM

- Privileged SRAM-access Masters which have a direct access to some dedicated SRAM ports

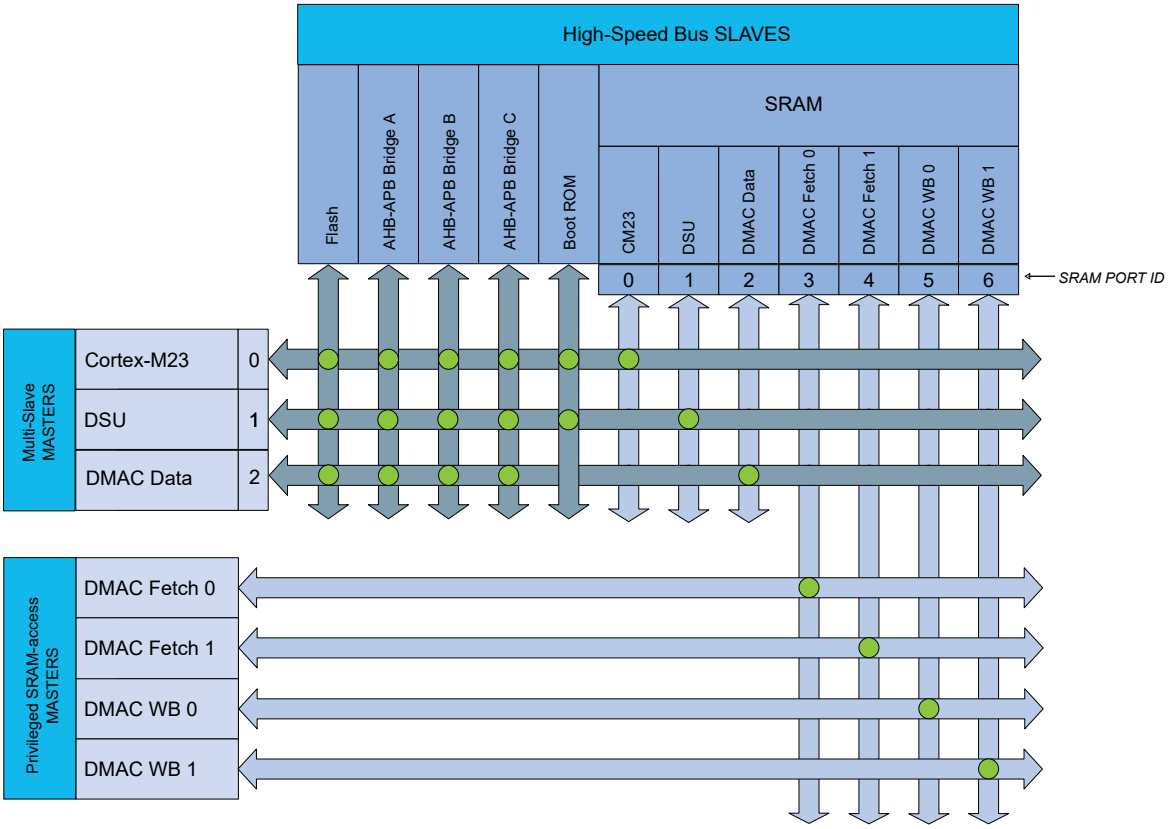
Table 11-6. Privileged SRAM-access Masters

Privileged SRAM-access Masters
DMAC - Fetch 0 Access
DMAC - Fetch 1 Access
DMAC - Write Back 0 Access
DMAC - Write Back 1 Access

Note: Privileged SRAM-access Masters rely on SRAM quality of service to define priority levels (SRAM Port ID). Refer to 11.4 SRAM Quality of Service for more information.

Downloaded from Arrow.com.

Figure 11-1. Master-to-Slave Access



11.4 SRAM Quality of Service

To ensure that masters with latency requirements get sufficient priority when accessing RAM, priority levels can be assigned to the masters for different types of access.

The Quality of Service (QoS) level is independently selected for each master accessing the RAM. For any access to the RAM, the RAM also receives a QoS level. The QoS levels and their corresponding bit values are shown in the following table.

Table 11-7. Quality of Service

Value	Name	Description
0x0	DISABLE	Background (no sensitive operation)
0x1	LOW	Sensitive Bandwidth
0x2	MEDIUM	Sensitive Latency
0x3	HIGH	Critical Latency

Note: If a master is configured with QoS level DISABLE (0x0) or LOW (0x1), there will be a minimum latency of one cycle to get RAM access.

Downloaded from Arrow.com.

The priority order for concurrent accesses are decided by two factors:

- As first priority, the QoS level for the master.
- As a second priority, a static priority given by the port ID. The lowest port ID has the highest static priority.

See the tables below for more details.

Table 11-8. HS SRAM Port Connections QoS

HS SRAM Port Connection	Port ID	Connection Type	QoS	default QoS
DMAC - Direct Memory Access Controller - Write-Back 1 Access	6	Direct	DMAC QOSCTRL.WRBQOS	0x2
DMAC - Direct Memory Access Controller - Write-Back 0 Access	5	Direct	DMAC QOSCTRL.WRBQOS	0x2
DMAC - Direct Memory Access Controller - Fetch 1 Access	4	Direct	DMAC QOSCTRL.FQOS	0x2
DMAC - Direct Memory Access Controller - Fetch 0 Access	3	Direct	DMAC QOSCTRL.FQOS	0x2
DMAC - Direct Memory Access Controller - Data Access	2	Bus Matrix	DMAC QOSCTRL.DQOS	0x2
DSU - Device Service Unit	1	Bus Matrix	DSU CFG.LQOS	0x2
CM23 - Cortex M23 Processor	0	Bus Matrix	0x41008114, bits[1:0](1)	0x3

Note:

1. The CPU QoS level can be written/read, using 32-bit access only.

Downloaded from Arrow.com.

12. Peripherals Configuration Summary

Table 12-1. Peripherals Configuration Summary

Peripheral name	Base address	IRQ line	AHB clock		APB clock		Generic Clock	PAC		Events			DMA	Power domain
			Index	Enabled at Reset	Index	Enabled at Reset	Index	Index	Write Protection at Reset	User	Generator	Sleep Walking	Index	Name
AHB-APB Bridge A	0x40000000	—	0	Y	—	—	—	—	—	—	—	N/A	—	PDSW
PAC	0x40000000	21: ERR	6	Y	0	Y	—	0	N	—	49: ERR	Y	—	PDSW
PM	0x40000400	0: PLRDY	—	—	1	Y	—	1	N	—	—	N/A	—	PDAO
MCLK	0x40000800	0: CKRDY	—	—	2	Y	—	2	N	—	—	N/A	—	PDSW
RSTC	0x40000C00	—	—	—	3	Y	—	3	N	—	—	N/A	—	PDAO
OSCCTRL	0x40001000	0: XOSCRDY, XOSCFAIL, OSC16MRDY, DFLLULPRDY, DFLLULPLOCK, DFLLULPNOLOCK, DPLLLCKR, DPLLLCKF, DPLLLTO, DPLLLDRTO	—	—	4	Y	0: FDPLL96M clk source 1: FDPLL96M 32 kHz 2: DFLLULP reference	4	N	0: TUNE	1: CFD	Y	—	PDSW
OSC32KCTRL	0x40001400	0: XOSC32KRDY, CLKFAIL	—	—	5	Y	—	5	N	—	2: CFD	Y	—	PDAO
SUPC	0x40001800	0: BOD33RDY,BOD33DET, B33SRDY, VREGRDY, VCORERDY, ULPVREFRDY	—	—	6	Y	—	6	N	—	3: BOD33DET	Y	—	PDAO
GCLK	0x40001C00	—	—	—	7	Y	—	7	N	—	—	N/A	—	PDSW
WDT	0x40002000	1: EW	—	—	8	Y	—	8	N	—	—	N/A	—	PDSW
RTC	0x40002400	2: CMP0-1, TAMPER, OVF, PER0-7	—	—	9	Y	—	9	N	1: TAMPER	4-11 : PER0-7 12-13 : CMP0-1 14 : TAMPER 15 : OVF 16 : PERD	Y	1: TIMESTAMP	PDAO
EIC	0x40002800	3: EXTINT0 4: EXTINT1 5: EXTINT2 6: EXTINT3 7: EXTINT4-7, NSCHK NMI	—	—	10	Y	3	10	N	—	17-24: EXTINT0-7	Y	—	PDAO
FREQM	0x40002C00	8: DONE	—	—	11	Y	4: FREQM_MSR 5: FREQM_REF	11	N	—	—	N/A	—	PDSW
PORT	0x40003000	10: NSCHK	—	—	12	Y	—	12	N	3-6 : EV0-3	—	Y	—	PDAO
AC	0x40003400	39: COMP0-1, WIN0	—	—	13	Y	17	13	N	16-17: SOC0-1	40-41: COMP0-1 42: WIN0	Y	—	PDAO

Downloaded from Arrow.com.

Peripheral name	Base address	IRQ line	AHB clock		APB clock		Generic Clock	PAC		Events			DMA	Power domain
			Index	Enabled at Reset	Index	Enabled at Reset	Index	Index	Write Protection at Reset	User	Generator	Sleep Walking	Index	Name
AHB-APB Bridge B	0x41000000	—	1	Y	—	—	—	—	—	—	—	N/A	—	PDSW
DSU	0x41002000	—	4	Y	1	Y	—	1	Y	—	—	N/A	2-3: DCC0-1	PDSW
NVMCTRL	0x41004000	9: DONE, PROGE, LOCKE, NVME, KEYE, NSCHK	7	Y	2	Y	—	2	N	2: AUTOW	—	Y	—	PDSW
DMAC	0x41006000	11: SUSP0, TERR0, TCMPL0 12: SUSP1, TERR1, TCMPL1 13: SUSP2, TERR2, TCMPL2 14: SUSP3, TERR3, TCMPL3 15: SUSP4-7, TERR4-7, TCMPL4-7	3	Y	—	—	—	3	—	7-10: CH0-3	25-28: CH0-3	Y	—	PDSW
HMATRIXHS	0x41008000	—	5	Y	—	—	—	4	N	—	—	N/A	—	PDSW
AHB-APB Bridge C	0x42000000	—	2	Y	—	—	—	—	—	—	—	N/A	—	PDSW
EVSYS	0x42000000	16: EVD0, OVR0 17: EVD1, OVR1 18: EVD2, OVR2 19: EVD3, OVR3 20: NSCHK	—	—	0	Y	6: CH0 7: CH1 8: CH2 9: CH3	0	N	—	—	N/A	—	PDSW
SERCOM0	0x42000400	22: bit 0 23: bit 1 24: bit 2 25: bit 3-6	—	—	1	Y	11: CORE 10: SLOW	1	N	—	—	N/A	4: RX 5: TX	PDSW
SERCOM1	0x42000800	26: bit 0 27: bit 1 28: bit 2 29: bit 3-6	—	—	2	Y	12: CORE 10: SLOW	2	N	—	—	N/A	6: RX 7: TX	PDSW
SERCOM2	0x42000C00	30: bit 0 31: bit 1 32: bit 2 33: bit 3-6	—	—	3	Y	13: CORE 10: SLOW	3	N	—	—	N/A	8: RX 9: TX	PDSW
TC0	0x42001000	34: ERR, MC0, MC1, OVF	—	—	4	Y	14	4	N	11: EVU	29: OVF 30-31: MC0-1	Y	10: OVF 11-12: MC0-1	PDSW
TC1	0x42001400	35: ERR, MC0, MC1, OVF	—	—	5	Y	14	5	N	12: EVU	32: OVF 33-34: MC0-1	Y	13: OVF 14-15: MC0-1	PDSW
TC2	0x42001800	36: ERR, MC0, MC1, OVF	—	—	6	Y	15	6	N	13: EVU	35: OVF 36-37: MC0-1	Y	16: OVF 17-18: MC0-1	PDSW
ADC	0x42001C00	37: OVERRUN, WINMON 38:RESRDY	—	—	7	Y	16	7	N	14: START 15: SYNC	38: RESRDY 39: WINMON	Y	19: RESRDY	PDSW
DAC	0x42002000	40: UNDERRU	—	—	8	Y	18	8	N	18: START	43: EMPTY	Y	20: EMPTY	PDSW

Downloaded from Arrow.com.

Peripheral name	Base address	IRQ line	AHB clock		APB clock		Generic Clock	PAC		Events			DMA	Power domain
			Index	Enabled at Reset	Index	Enabled at Reset	Index	Index	Write Protection at Reset	User	Generator	Sleep Walking	Index	Name
		N 41: EMPTY												
PTC	0x42002400	42: EOC, WCOMP	—	—	9	Y	19	9	N	19 : STCONV 20 : DSEQR	44: EOC 45: WCOMP	Y	21 : EOC 22 : SEQ 23 : WCOMP	PDSW
TRNG	0x42002800	43: DATARDY	—	—	10	Y	—	10	N	—	46 : READY	Y	—	PDSW
CCL	0x42002C00	—	—	—	11	Y	20	11	N	21 : LUTIN0 22 : LUTIN1	47 : LUTOUT0 48 : LUTOUT1	Y	—	PDSW
OPAMP	0x42003000	—	—	—	12	Y	—	12	N	—	—	N/A	—	PDSW
TRAM	0x42003400	44: DRP, ERR	12	Y	—	—	—	13	—	—	—	N/A	—	PDSW

Downloaded from Arrow.com.

13. SAM L11 Security Features

This chapter provides an overview of the SAM L11 security features.

13.1 Features

Security features can be split in two main categories.

The first category relates to the ARM TrustZone for Cortex-M technology features:

- Flexible hardware isolation of memories and peripherals:
  - Up to six regions for the Flash
  - Up to two regions for the Data Flash
  - Up to two regions for the SRAM
  - Individual security attribution (secure or non-secure) for each peripheral using the Peripheral Access Controller (PAC)
  - Mix-secure peripherals which support both secure and non-secure security attributions
- Three debug access levels allowing:
  - The highest debug level with no restrictions in term of memory and peripheral accesses
  - A restricted debug level with non-secure memory regions access only
  - The lowest debug level where no access is authorized except with a debugger using a Boot ROM-specific mode
- Different chip erase support according to security settings
- Security configuration is fully stored in Flash and safely auto-loaded at startup during Boot ROM execution using CRC checks

Important: Debug access levels, such as Chip Erase support are described in the Boot ROM chapter.

The second category relates to the extra security features which are not related to ARM TrustZone for Cortex-M technology support:

- Built-in cryptographic accelerator accessible through cryptographic libraries stored in ROM
  - Supporting AES-128 encryption/decryption, SHA-256 authentication, GCM encryption and authentication
  - Cryptographic libraries are especially designed for side channel and fault injection attacks prevention
- One True Random Generator (TRNG)
- Secure Boot, which performs integrity check on a configurable portion of the Flash (BS memory area)
- Secure Pin Multiplexing to isolate on dedicated SERCOM I/O pins a secured communication with external devices from the non secure application
- Data Flash
  - Optimized for secrets storage
  - Data Scrambling with user-defined key (optional)

Downloaded from Arrow.com.

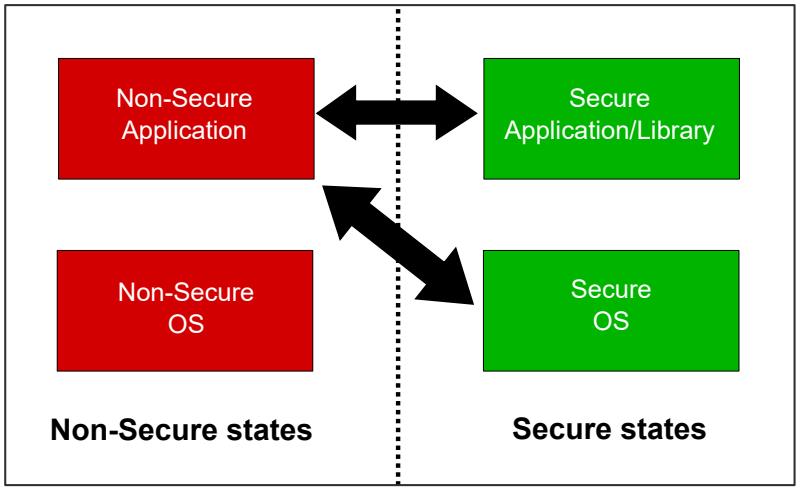
  - Rapid Tamper erase on scrambling key and on one user-defined row
  - Silent access for side channel attack resistance
- TrustRAM
  - Address and Data scrambling with user-defined key
  - Chip-level tamper detection on physical RAM to resist microprobing attacks
  - Rapid Tamper Erase on scrambling key and RAM data
  - Silent access for side channel attack resistance
  - Data remanence prevention
- Unique 128-bit serial number

13.2 ARM TrustZone Technology for ARMv8-M

ARM TrustZone for Cortex-M technology is an optional core extension, which enables the system and the software to be partitioned into Secure and Non-secure domains.

Secure software can access both Secure and Non-secure memories and resources, while Non-Secure software can only access Non-secure memories and resources.

Figure 13-1. TrustZone for ARMv8-M



If the TrustZone is implemented (SAM L11 devices), the system starts up in Secure state by default.

The security state of the processor can be either Secure or Non-secure.

Important: For more details, please refer to *TrustZone Technology for ARMv8-M Architecture*, which is available on the ARM web site (www.arm.com).

13.2.1 Memory System and Memory Partitioning

The memory space is partitioned into Non-Secure and Secure memory regions:

- Non-Secure (NS): Non-Secure addresses are used for memory and peripherals accessible by all software, that is, running on the device.
- Secure (S): Secure addresses are used for memory and peripherals accessible only by Secure software or masters.

Downloaded from Arrow.com.

- Non-Secure Callable (NSC): NSC is a special type of Secure memory location. It allows software to transition from Non-Secure to Secure state.

The Cortex-M23 provides two ways for managing the security configurations of the device.

The first solution consists in using the Cortex-M23 SAU (Security Attribution Unit), which is a Memory Protection Unit (MPU) like hardware embedded in the core. The role of the SAU is to manage all the Secure and Non-Secure transactions coming from the core. However, using the SAU implies that the security configuration must be propagated somewhere else in the MCU architecture for security awareness.

The second approach, which is the one used for SAM L11 devices, is articulated around a centralized Implementation Defined Attribution Unit (IDAU), which is a hardware unit external to the core.

For SAM L11 devices, the IDAU is coupled to the Cortex-M23 and manages all the security configurations related to the core. In addition, the IDAU propagates all the security configurations to the memory controllers. The IDAU, Flash, Data Flash and SRAM embedded memories can be split in sub-regions, which are reserved either for the Secure or for the Non-Secure application. Therefore, the SAU is not required and is absent from SAM L10/L11 devices.

The peripherals security attribution is managed by the Peripherals Access Controller (PAC). The PAC and each peripheral can be allocated either to the Secure or to the Non-Secure application, with the exception of the PAC, NVMCTRL, and DSU.

Note:

1. The PAC and NVMCTRL peripherals are always secured.
2. The DSU peripheral is always non-secured.

Both IDAU and PAC security configurations are stored in NVM fuses, which are read after each reset during Boot ROM execution and are loaded after Boot ROM verifications into their respective registers.

The peripherals security attribution (using PAC) is locked before exiting the Boot ROM execution sequence, that is, it is not possible to change a peripheral's configuration (Secure or Non-Secure) during application execution. However, the security attribution of each peripheral, excluding the PAC, NVMCTRL, and DSU, can be modified using the NONSECx NVM fused from the User Row (UROW) during application execution, hence it can be considered after any reset.

13.2.2 Memories Security Attribution

The IDAU is used to indicate the processor if a particular memory region is Secure (S), Non-secure Callable (NSC), or Non-secure (NS). It can also mark a memory region to be exempted from security checking.

Table 13-1. IDAU Memory Attribution Definition

Attribute	Description
Non-Secure	Memory can be accessed in Secure or Non-Secure state.
Secure	Memory can only be accessed in Secure state. It cannot be called from Non-Secure state.
Non-Secure callable	Memory can only be accessed in Secure state, but can be called from Non-Secure state.
Exempt	No attribution check will be done, and the operation will take place on the bus

Note: Refer to " SAM L11 Security Attribution" chapter for the detailed SAM L11 memories and peripherals security attribution description.

Downloaded from Arrow.com.

The Cortex-M23 will search each access (fetch or data) in the IDAU, which returns the privilege information about that specific address. If the access is not permitted, the CPU enters a HardFault exception.

The IDAU memory region's attributes are partly hardwired and partly set by NVM configuration fuses, and are loaded into the IDAU by the Boot ROM before application execution. The IDAU memory region's attributes are blocked for further writes from the application, but their current state can still be read through dedicated IDAU registers.

Note: Refer to the "SAM L11 IDAU Memory Mapping Registers".

13.2.2.1 Flash

The SAM L11 Flash can be split in up to six regions:

- The first three regions are called BOOT regions and can be configured to support a first-level bootloader for the application.
- The other regions are called APPLICATION regions and relate to the application itself.

The total size of the BOOT regions is defined by the BOOTPROT fuses from the NVM Boot Configuration Row (BOCOR), the APPLICATION regions total size being the remaining available size of the Flash.

Each of these BOOT / APPLICATION global regions can be split in up to three sub-regions:

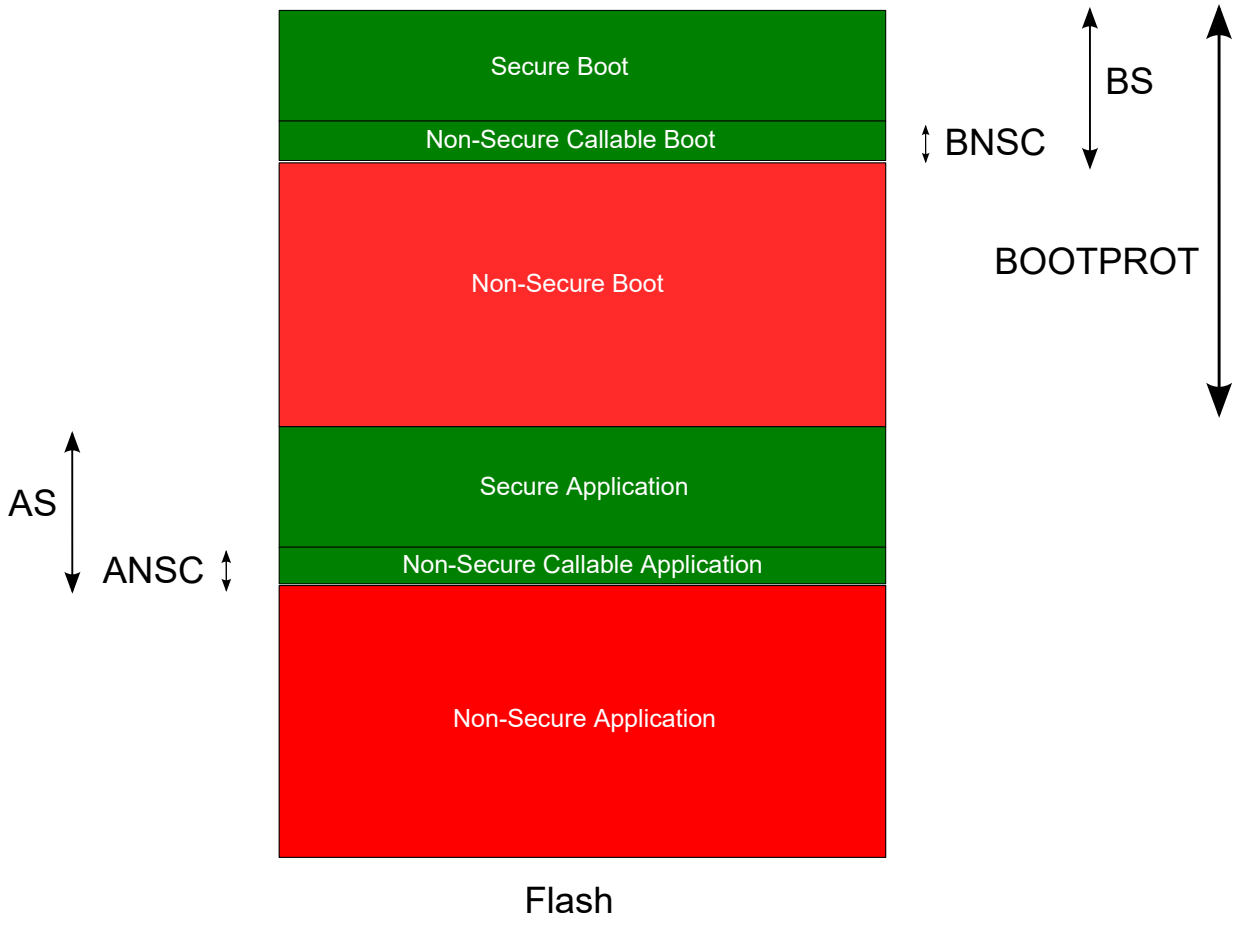
- The Secure sub-region
- The Non-Secure Callable (NSC) sub-region
- The Non-Secure (NS) sub-region

Each sub-region size can be configured using dedicated fuses from the NVM Boot Configuration Row (BOCOR):

- BS fuse corresponds to the size of the Secure + NSC sub-regions of the BOOT region
- BNSC fuse corresponds to the NSC sub-region size of the BOOT region
- AS fuse corresponds to the size of the Secure + NSC sub-regions of the APPLICATION region
- ANSC fuse corresponds to the NSC sub-region size of the APPLICATION region

Downloaded from Arrow.com.

Figure 13-2. SAM L11 Flash Memory Mapping

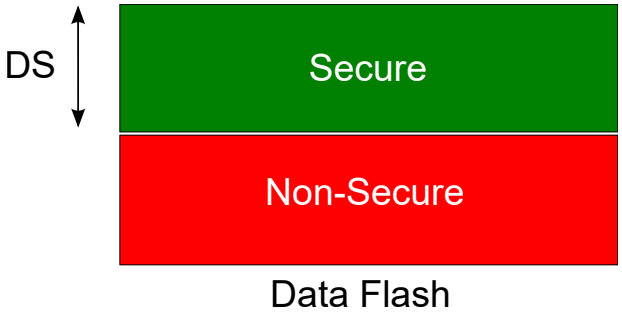


13.2.2.2 Data Flash

The SAM L11 Data Flash can be split in up to two regions:

- The Secure Data Flash region, with a size defined by the DS fuse from the NVM Boot Configuration Row (BOCOR)
- The Non-Secure (NS) Data Flash region

Figure 13-3. SAM L11 Data Flash Memory Mapping



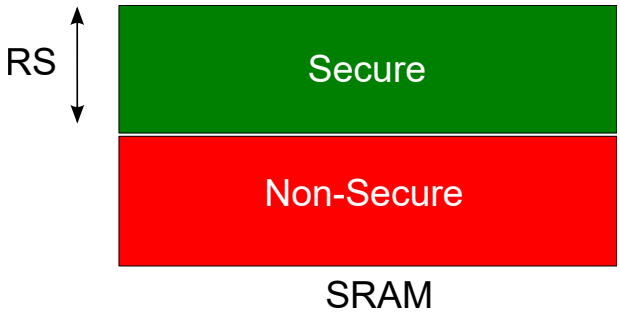
13.2.2.3 SRAM

The SAM L11 SRAM can be split in up to two regions:

Downloaded from Arrow.com.

- The Secure SRAM region, with a size defined by the RS fuse from the NVM Boot Configuration Row (BOCOR)
- The Non-Secure (NS) SRAM region

Figure 13-4. SAM L11 SRAM Memory Mapping



13.2.3 SAM L11 Memory Mapping Configuration Summary

The table below summarizes the mapping of the SAM L11 memory regions.

Table 13-2. SAM L11 Memory Regions Mapping

Memory region	Base address	Size
Flash Boot area	0x00000000	BOOTPROT * 256Bytes
Flash Secure Boot	0x00000000	BS*256Bytes - BNSC*32Bytes
Flash Non-Secure Callable Boot	Contiguous to Flash Secure Boot	BNSC * 32Bytes
Flash Non-Secure Boot (1)	BS * 256Bytes	Flash Boot remaining size (BOOTPROT * 256Bytes - BS*256Bytes)
Flash Appliation area	BOOTPROT * 256Bytes	Flash size - BOOTPROT * 256Bytes (Flash Boot area)
Flash Secure Application	BOOTPROT * 256Bytes	AS*256Bytes-ANSC*32Bytes
Flash Non-Secure Callable Application	Contiguous to Flash Secure APPLICATION	ANSC * 32Bytes
Flash Non-Secure Application	(BOOTPROT+AS) * 256Bytes	Flash remaining size (Flash size - BOOTPROT*256Bytes - AS*256Bytes)
Secure Data Flash	0x00400000	DS * 256Bytes
Non-Secure Data Flash	Contiguous to Secure Data Flash	2KB - Secure Data Flash size
Secure SRAM	0x20000000	RS * 128Bytes
Non-Secure SRAM	Contiguous to Secure SRAM	SRAM size - Secure SRAM size

Note:

1. Flash Secure Boot size cannot be null if a Flash Non-Secure Boot size is defined.

Here is a typical configuration for a 64KB of Flash, 2KB of Data Flash and 16KB of SRAM:

- Flash boot area:
  - Total Boot area size = 8KB => BOOTPROT = 8192 / 256 = 0x20
  - Flash Secure + Non-Secure Callable Boot size = 1KB => BS = 1024 / 256 = 0x4
  - Flash Non-Secure Callable Boot size = 32Bytes => BNSC = 32 / 32 = 0x1
  - Flash Non-Secure Boot size = 8KB - 1KB = 7KB
- Flash Application area:
  - Total Application area size = 64 KB - 8KB = 56KB

Downloaded from Arrow.com.

  - Flash Secure + Non-Secure Callable Application size = 16KB => AS = 16 * 1024 / 256 = 0x40
  - Flash Non-Secure Callable Application size = 32Bytes => ANSC = 32 / 32 = 0x1
  - Flash Non-Secure Application size = 56KB - 16KB = 40KB
- Data Flash area:
  - Data Flash Secure size = 2KB => DS = 2048 / 256 = 0x8
  - Data Flash Non-Secure size = 2 KB - 2 KB = 0KB
- SRAM Flash area
  - SRAM Secure size = 4KB => RS = 4096 / 128 = 0x20
  - SRAM Non-Secure size = 16KB - 4KB = 12KB

13.2.4 SAM L11 IDAU Memory Mapping Registers

The tables below summarizes the mapping of the SAM L11 IDAU memory regions.

Table 13-3. SAM L11 IDAU Memory Register Address

Registers	Address
SECCTRL	0x41000001
SCFGB	0x41000004
SCFGA	0x41000008
SCFGR	0x4100000C

Table 13-4. SAM L11 IDAU SECCTRL Register (8-bit)

Bit Position	Name		
7:0	Reserved	RXN (Bit 2)	Reserved

Table 13-5. SAM L11 IDAU SCFGB Register (32-bit)

Bit Position	Name	
7:0	BS	
15:8	Reserved	BNSC (bit 8-13)
23:16	BOOTPROT	
31:24	Reserved	

Table 13-6. SAM L11 IDAU SCFGA Register (32-bit)

Bit Position	Name	
7:0	AS	
15:8	Reserved	ANSC (bit 8-13)
23:16	Reserved	DS (bit 16-19)
31:24	Reserved	

Table 13-7. SAM L11IDAU SCFGR Register (8-bit)

Bit Position	Name	
7:0	Reserved	RS (bit 0-6)

13.2.5 Peripherals Security Attribution

In addition to generic protection features, the Peripheral Access Controller (PAC) configures the security privileges for each individual peripheral in the system.

Downloaded from Arrow.com.

Each peripheral can only be configured either in Secure or in Non-Secure mode.

The PAC NONSECx registers (Read Only) contain one bit per peripheral for that purpose, which is the image of the NONSECx fuses from the NVM User row (UROW).

During Boot ROM execution, the NONSECx fuses from the NVM User row are copied in the PAC peripheral NONSECx registers so that they can be read by the application.

All peripherals are marked as "exempt" in the memory map, meaning that all bus transactions are propagated. As a consequence, any illegal accesses are reported back to the PAC and trigger an interrupt if enabled.

The security configuration (Secure or Non-Secure) is propagated to each individual peripheral, thus it is the responsibility of the peripheral to grant or not the access with the following rules:

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0), a PAC error is triggered

Important: These rules do not apply to the specific peripherals called Mix-Secure peripherals.

Note: The Secure application will usually provide an API for the Non-Secure application using the Non-Secure Callable region (NSC) to allow the Non-Secure application to request specific resources.

Table 13-8. Peripheral PAC Security Attribution (Excluding Mix-Secure Peripherals)

Mode	Secure Master Access	Non-Secure Master Access
Non-Secure	Read / Write	Read / Write
Secure	Read / Write	Discarded (Write ignored / Read 0x0) PAC Error is generated

13.2.5.1 SAM L11 Peripherals Configuration Example

Below is a typical configuration examples where all peripherals except the ADC, TC0, and Event System (EVSYS) are reserved to the Secure application:

- Secure/Non-Secure Peripherals PAC configuration:
  - PAC.NONSECA=PAC.NONSECB=0x0000_0000
  - PAC.NONSECC=0x0000_00091 (ADC, TC0 and EVSYS available for the Non-Secure application)

13.2.6 SAM L11 Memory Space Security Attribution

This table provides the security attributions of the SAM L11 memory space:

Downloaded from Arrow.com.

Table 13-9. SAM L11 Memory Space Security Attributions

Memory region	Attribute
Flash Secure Boot	Secure
Flash Non-Secure Callable Boot	Non-secure callable
Flash Non-Secure Boot	Non-secure
Flash Secure Application	Secure
Flash Non-Secure Callable Application	Non-secure callable
Flash Non-Secure Application	Non-secure
Secure Data Flash	Secure
Non-Secure Data Flash	Non-secure
NVM User Rows	Secure (R/W access) Non-Secure (Discarded for BOCOR, Read-only for the others)
Boot ROM	Secure Execute-only for CRYA functions Hardfault generated if not
Secure SRAM	Secure
Non-Secure SRAM	Non-secure
Peripherals	Exempt
IOBUS	Exempt
Others (Reserved, Undefined...)	Secure

13.2.7 Cortex-M23 Test Target Instructions

Software may check the privilege state of a memory location by using the Cortex-M23 Test Target instructions: TT, TTT, TTA, and TTAT.

The memory location is referenced using the Cortex-M23 IREGION bitfield, which specifies the IDAU region number (see the *ARMv8-M Architecture Reference Manual* for more information).

Table 13-10. SAM L11 IDAU Region Number for TT, TTT, TTA and TTAT Cortex-M23 Instructions

Memory Region	IDAU Region Number for TTx Instructions (IREGION bits)
Flash Secure BOOT	0x01
Flash Non-Secure Callable BOOT	0x02
Flash Non-Secure BOOT	0x03
Flash Secure APPLICATION	0x04
Flash Non-Secure Callable APPLICATION	0x05
Flash Non-Secure APPLICATION	0x06
Secure Data Flash	0x07
Non-Secure Data Flash	0x08
NVM User Rows	0x00 (invalid)
Boot ROM	0x09
Secure SRAM	0x0A
Non-Secure SRAM	0x0B
Peripherals	0x00 (invalid)

Downloaded from Arrow.com.

Memory Region	IDAU Region Number for TTx Instructions (IREGION bits)
IOBUS	0x00 (invalid)
Others (Reserved, Undefined...)	0x00 (invalid)

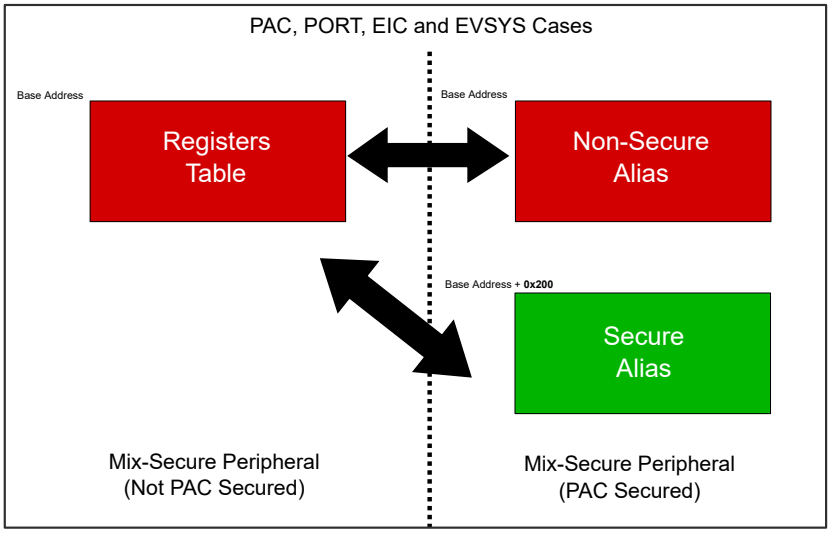
13.2.8 Mix-Secure Peripherals

There are five Mix-Secure peripherals that allow internal resources to be shared between the Secure and Non-Secure applications:

- The PAC controller which manages peripherals security attribution (Secure or Non-Secure).
- The Flash memory controller (NVMCTRL) which supports Secure and Non-Secure Flash regions programming.
- The I/O controller (PORT) which allows to individually allocate each I/O to the Secure or Non-Secure applications.
- The External Interrupt Controller (EIC) which allows to individually assign each external interrupt to the Secure or Non-Secure applications.
- The Event System (EVSYS) allows to individually assign each event channel to the Secure or Non-Secure applications.

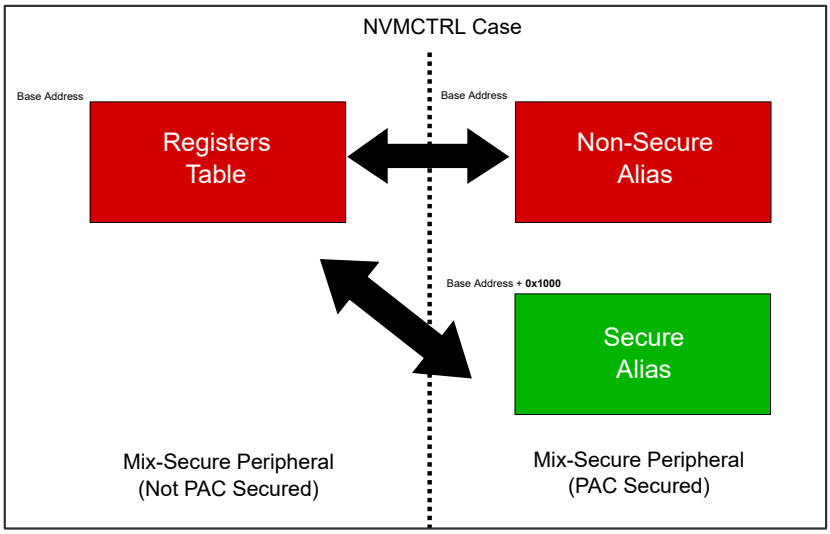
When a Mix-Secure peripheral is configured as Secure in the PAC, its register map is automatically duplicated in a Secure and Non-Secure alias:

- The Non-Secure alias is at the peripheral base address.
- The Secure alias is located at the peripheral base address:
  - + 0x200 offset for the PAC, EIC, PORT and EVSYS peripherals



  - + 0x1000 offset for the NVMCTRL peripheral.

Downloaded from Arrow.com.



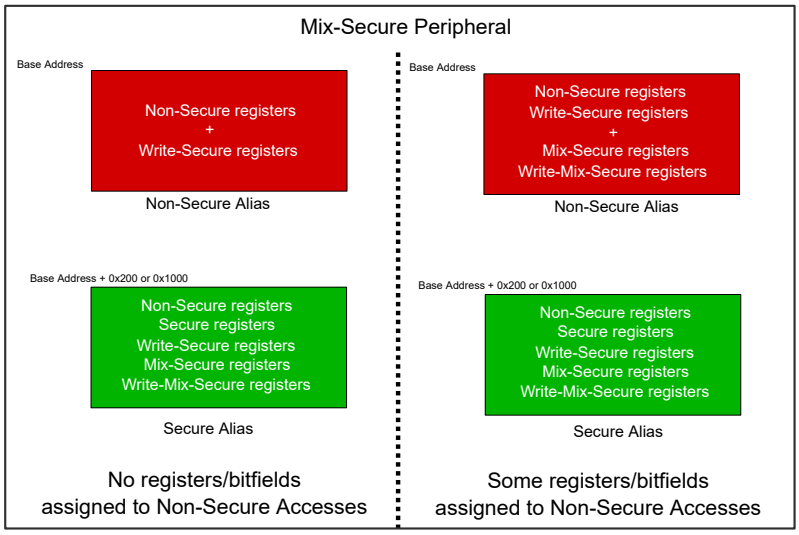
The Secure alias has the following characteristics:

- All of the peripheral registers are available for the Secure application through the Secure alias
- When an internal resource becomes available to the Non-Secure application, the corresponding registers (called Mix-Secure registers) or bitfields in registers are still accessible through this Secure alias by the Secure application
- Non-Secure accesses to this Secure alias are discarded (Write is ignored, Read is 0x0) and a PAC error is triggered

The Non-Secure alias has the following characteristics:

- Only a restricted set of registers are available for the Non-Secure application through the Non-Secure alias
- It is the responsibility of the Secure application to assign some resources to the Non-Secure application. This is done by setting the corresponding bits in the NONSECx registers of the Mix-Secure peripheral.
  - When an internal resource becomes available for the Non-Secure application, the corresponding registers (called Mix-Secure and Write-Mix-Secure registers) or bitfields in the registers are accessible through the Non-Secure alias by the Non-Secure application
  - Non-Secure accesses to Secure resources (registers, bitfields) are silently discarded (Write is ignored, Read is 0x0) and no error is generated
- Secure accesses to the Non-Secure alias are silently discarded (Write is ignored, Read is 0x0) and no error is generated

Downloaded from Arrow.com.



Mix-Secure peripherals have always the following registers:

- NONSEC register is a generic register that tells the Non-Secure application which resources inside a Mix-Secure peripheral can be used
- NSCHK register is a register allowing the Non-Secure application to be notified when the security configuration of a Mix-Secure peripheral is being modified during application execution

Important: It is recommended that the Non-Secure application first copy the content of NONSEC register inside NSCHK register, and then enable the NSCHK interrupt flags. Once done, any changes to the NONSEC register by the Secure application will trigger an interrupt so that Non-Secure application can take appropriate actions. This mechanism allows the Secure application to dynamically change the security attribution of a Mix-Secure peripheral and avoid illegal accesses from the Non-Secure application. The interrupt handler should always copy the NONSEC register to NSCHK register before exiting it.

Mix-Secure peripherals can have five type of registers:

- **Non-Secure**: these registers will always be available in both the Secure and Non-Secure aliases
- **Secure**: these registers will never be available in the Non-Secure alias and always available in the Secure alias
- **Write-Secure**: these are registers than can:
  - Be written or read by the Secure application only in the Secure alias
  - Only read by the Non-Secure application in Non-Secure alias. Write is forbidden.
- **Mix-Secure** registers : these ones are used when a resource can be allocated to either the Secure and Non-Secure alias
  - Note that, in some cases, the Mix-Secure properties apply to a bitfield only (like one I/O bit in the PORT peripheral register)
- **Write-Mix-Secure** registers (NVMCTRL peripheral only): these are Mix-Secure registers, which:
  - can be written or read by the Secure application only in the Secure alias
  - can only be read by the Non-Secure application in Non-Secure alias **except** if Non-Secure writes are authorized in NVMCTRL.NONSEC register

Downloaded from Arrow.com.

Table 13-11. SAM L11 Mix-Secure Peripheral Registers Access

Mix-Secure Peripheral Register	Secure Master Access		Non-Secure Master Access	
	Secure Alias	Non-Secure Alias	Secure Alias	Non-Secure Alias
Non-Secure	Read / Write	Discarded (Write ignored / Read 0x0) No Error is generated	Discarded (Write ignored / Read 0x0) PAC Error is generated	Read / Write
Secure				Discarded (Write ignored / Read 0x0) No Error is generated
Write-Secure				Read-only (Write ignored) No Error is generated
Mix-Secure				Read/Write if the resource is available for the Non-Secure Application Discarded if not (Write ignored / Read 0x0) and no error is generated
Write-Mix-Secure				Read /Write if the resource is available for the Non-Secure Application Read-only if not (Write ignored) and no error is generated

13.3 Crypto Acceleration

13.3.1 Overview

The SAM L11 product embeds a hardware/software cryptographic accelerator (CRYA) which supports Advanced Encryption Standard (AES) encryption and decryption, Secure Hash Algorithm 2 (SHA-256) authentication, and Galois Counter Mode (GCM) encryption and authentication through a set of APIs, which are only accessible once the Boot ROM has completed.

Note: The CRYA cryptographic accelerator is mapped as a slave on the IOBUS port and is driven by the CPU using assembly code (located in ROM).

The Advanced Encryption Standard (AES) is compliant with the American FIPS (Federal Information Processing Standard) Publication 197 specification. The AES operates on a 128-bit block of input data. The key size used for an AES cipher specifies the number of repetitions of transformation rounds that convert the input plaintext, into the final output, called the ciphertext. The AES works on a symmetric-key algorithm, meaning the same key is used for both encrypting and decrypting the data.

The SHA-256 is a cryptographic hash function that creates a 256-bit hash of a data block. The data block is processed in chunks of 512 bits.

The GCM is a mode of operation for AES that combines the CTR (Counter) mode of operation with an authentication hash function.

For detail algorithm specification, refer to following standards and specification:

- AES: FIPS Publication 197, Advanced Encryption Standard (AES)
- SHA: FIPS Pub 180-4, The Secure Hash Standard
- GCM: NIST Special Publication 800-38D Recommendation

13.3.2 Features

- Advanced Encryption Standard (AES), compliant with FIPS Publication 197
  - Encryption with 128-bit cryptographic key
  - Decryption with 128-bit cryptographic key

Downloaded from Arrow.com.

  - Countermeasures against side-channel attacks for AES
- Secure Hash Algorithm 2 (SHA-256), compliant with FIPS Pub 180-4
  - Accelerates message schedule and inner compression loop
- Galois Counter Mode (GCM) encryption using AES engine and authentication
  - Accelerates the GF(2128) multiplication for AES-GCM hash function

13.3.3 CRYA APIs

The CRYA APIs which are located in a dedicated Boot ROM area are only accessible from the user application after the Boot ROM has completed. This area is an execute-only area, meaning the CPU cannot do any loads, but can call the APIs. The Boot ROM memory space is a secure area, only the secure application can directly call these APIs.

Table 13-12. CRYA APIs Addresses

CRYA API	Address
AES Encryption	0x02001904
AES Decryption	0x02001908
SHA Process	0x02001900
GCM Process	0x0200190C

13.3.3.1 AES API

The AES software has two function routines to do encryption and decryption on a 128 bit block of input data.

The AES encryption function entry point is located at the Boot ROM address 0x02001904 and the encryption function parameters are:

- Src[in] : a pointer to a 128-bit data block to be encrypted
- Dst[out]: a pointer to 128 bit encrypted data
- Keys[in]: a pointer to 128 bit key
- Length[in]: Number of 32-bit words comprising the Key, 4 for 128 bits key

The AES decryption function entry point is located at the Boot ROM address 0x02001908 and the decryption function parameters are:

- Src[in] : a pointer to a 128-bit data block to be decrypted
- Dst[out]: a pointer to 128 bit decrypted data
- Keys[in]: a pointer to 128 bit key
- Length[in]: Number of 32-bit words comprising the Key, 4 for 128 bits key

The APIs are:

```
/* Type definition for CRYA AES functions. */
typedef void (*crya_aes_encrypt_t) (const uint8_t *keys, uint32_t key_len, const
uint8_t *src, uint8_t *dst);
typedef void (*crya_aes_decrypt_t) (const uint8_t *keys, uint32_t key_len, const
uint8_t *src, uint8_t *dst);
```

```
/* AES encrypt function
* \param keys[in]: A pointer to 128-bit key
* \param key_len[in]: Number of 32-bit words comprising the key, 4 for 128-bit key
* \param src[in]: A pointer to a 128-bit data block to be encrypted
```


Downloaded from Arrow.com.

```
* \param dst[out]: A pointer to a 128-bit encrypted data
*/
#define secure_crya_aes_encrypt ((crya_aes_encrypt_t ) (0x02001904 | 0x1))
```

```
/* AES decrypt function
* \param keys[in]: A pointer to 128-bit key
* \param key_len[in]: Number of 32-bit words comprising the key, 4 for 128-bit key
* \param src[in]: A pointer to a 128-bit data block to be decrypted
* \param dst[out]: A pointer to a 128-bit decrypted data
*/
#define secure_crya_aes_decrypt ((crya_aes_decrypt_t ) (0x02001908 | 0x1))
```

13.3.3.2 SHA API

The SHA software function can update the hash value based on the 512-bit data.

It is assumed that the message is already preprocessed properly for the SHA algorithm, so that the SHA software can work directly on 512-bit portions.

The SHA function entry point is located at the Boot ROM address 0x02001900 and has three parameters:

- [In/out]: A pointer to a hash location (hash input and output)
- [In]: A pointer to a 512-bit data block
- [In]: A pointer to a RAM buffer (256B needed for internal algorithm)

The updated hash value is put at first parameter after the function exit.

The API is:

```
/* Type definition for CRYA SHA function. */
typedef void (*crya_sha_process_t) (uint32_t hash_in_out[8], const uint8_t data[64], uint32_t
ram_buf[64]);
```

```
/* CRYA SHA function
* \param hash_in_out[In/out]: A pointer to a hash location (hash input and output)
* \param data[In]: A pointer to a 512 bit data block
* \param ram_buf[In]: A pointer to a RAM buffer (256B needed for internal algorithm)
*/
#define crya_sha_process ((crya_sha_process_t ) (0x02001900 | 0x1))
```

Code example of using CRYA SHA software:

```
void sha256_process(uint32_t hash[8], const uint8_t data[64])
{
   uint32_t ram_buf[64]; /* 256 bytes needed for message schedule table */

   /* Pointer to CRYA SHA function in ROM */
   static void (*crya_sha_process)(uint32_t hash_in_out[8], const uint8_t data[64],
               uint32_t ram_buf[64]);

   crya_sha_process = (void (*)(uint32_t *, const uint8_t *, uint32_t *))
                      *((uint32_t*)0x02001900);

   crya_sha_process (hash, data, ram_buf);
}
```

13.3.3.3 GCM API

The GCM function entry point is is located at the Boot ROM address 0x0200190C and the function parameters are:

- Block1[in]: a pointer to 128-bit data blocks that are to be multiplied
- Block2[in]: a pointer to 128-bit data blocks that are to be multiplied
- dst[out]: a pointer to a location for storing the result

Downloaded from Arrow.com.

The API is:

```
/* Type definition for GF(2^128) multiplication */
typedef void (*crya_gf_mult128_t) (const uint32_t *block1, const uint32_t *block2, uint32_t
*dst);
```

```
/* GF(2^128) multiplication.
*
* \param block1[In]: A pointer to 128-bit data blocks that are to be multiplied
* \param block2[In]: A pointer to 128-bit data blocks that are to be multiplied
* \param dst[out]: A pointer to a location for storing the result
*/
#define secure_crya_gf_mult128 ((crya_gf_mult128_t ) (0x0200190C | 0x1))
```

13.4 True Random Number Generator (TRNG)

Refer to TRNG - True Random Number Generator for more information.

13.5 Secure Boot

A Secure Boot with SHA-based authentication on a configurable portion on the Flash (BS memory area) is available with verification mechanisms allowing to reset and restart the authentication process in case of a failure.

Refer to 14. Boot ROM for more information.

13.6 Secure Pin Multiplexing on SERCOM

The Secure Pin Multiplexing feature can be used on dedicated SERCOM I/O pins to isolate a secured communication with external devices from the non-secure application.

To benefit from this feature, the security attribution of the SERCOM must be set as secured using the PAC peripheral.

When this operation occurs:

- The secured SERCOM instances becomes mapped only on a specific set of I/Os
- All of the alternate I/O pins of the secured SERCOM instance are kept in a Hi-Z configuration
- The PTC cannot enable PTC lines mapped to any of the secured SERCOM instance I/O pins
- The CCL I/Os mapped to the secured SERCOM instance I/O pins are set to '0'

Refer to 4.4.2 Secure Pin Multiplexing (on SERCOM) Pins to obtain the list of pins supporting that feature.

13.7 Data Flash

Refer to 30. NVMCTRL – Nonvolatile Memory Controller to get all security features related to the Data Flash.

13.8 TrustRAM (TRAM)

Refer to TRAM - TrustRAM to get all security features related to the TrustRAM.

Downloaded from Arrow.com.

14. Boot ROM

The Boot ROM allows to ensure the integrity of the device at boot.

The Boot ROM features Boot Interactive mode, which allows the user to perform several actions on the device, such as NVM areas integrity check and chip erase via a debugger connection.

Unless a debugger is connected and places the Boot ROM in Boot Interactive mode, the CPU will jump to the Flash memory, loading the Program Counter (PC) and Stack Pointer (SP) values, and will start fetching Flash user code.
Note: Before jumping to the Flash, the Boot ROM resets the two first 2kB of SRAM. The Clocks remain unchanged.

In addition, the SAM L11 Boot ROM has extra security features, such as device integrity checks, memories/peripherals security attributions, and secure boot, which can be executed before jumping to the Flash in Secure state.

For security reasons, while the Boot ROM is executing, no debug is possible except when entering a specific Boot ROM mode called CPU Park mode.

Related Links
13.1 Features

14.1 Features

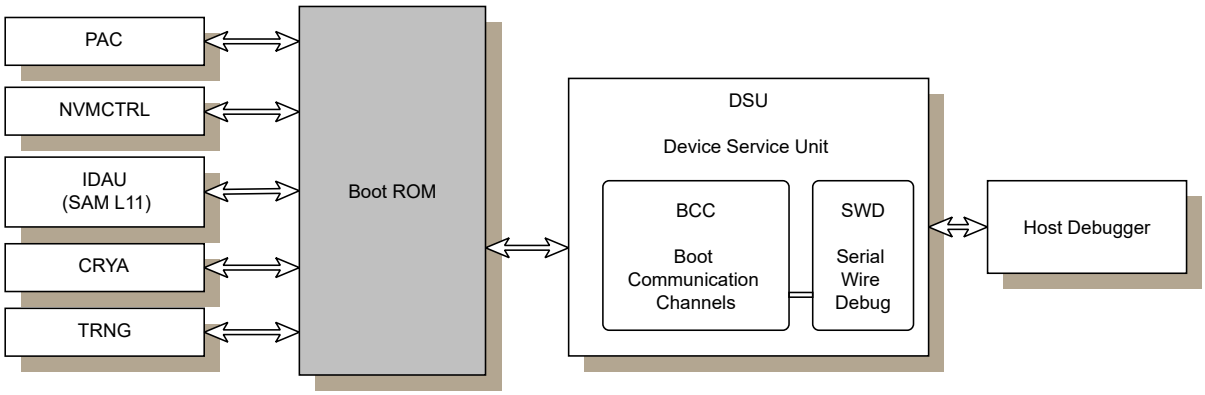
- Command interface for the host debugger supporting:
  - Chip erase commands to provide secure transitions between the different Debug Access Levels (DAL)
  - Device integrity check of the NVM memory regions
  - Debugger read access of the NVM rows
- CPU Park mode to get access for a debugger to the resources of the device depending on Debug Access Level (DAL)
- SAM L11 Added features:
  - Device integrity checks
  - Memory and peripheral security attributions from user configuration stored in NVM rows
  - Secure Boot on Flash BS Memory Area

Related Links
13.1 Features

Downloaded from Arrow.com.

14.2 Block Diagram

Figure 14-1. Boot ROM Block Diagram



Related Links

13.1 Features

14.3 Product Dependencies

In order to use this module, other parts of the system must be configured correctly, as described below.

Related Links

13.1 Features

14.3.1 Clocks

The device selects the OSC16M oscillator which is enabled by default after reset and configured at 4 MHz.

14.3.2 NVM User (UROW) and Boot Configuration (BOCOR) rows

The Boot ROM reads the different NVM rows during its execution.

The relevant fuses must be set appropriately by any configuration tools supported the device in order to operate correctly.

Refer to the 10.2 NVM Rows section for additional information.

14.3.3 Debug Operations

For security reasons, no debug is possible during the Boot ROM execution except when entering the Boot ROM CPU Park mode.

14.4 Functional Description

Related Links

13.1 Features

14.4.1 SAM L10 Boot ROM Flow

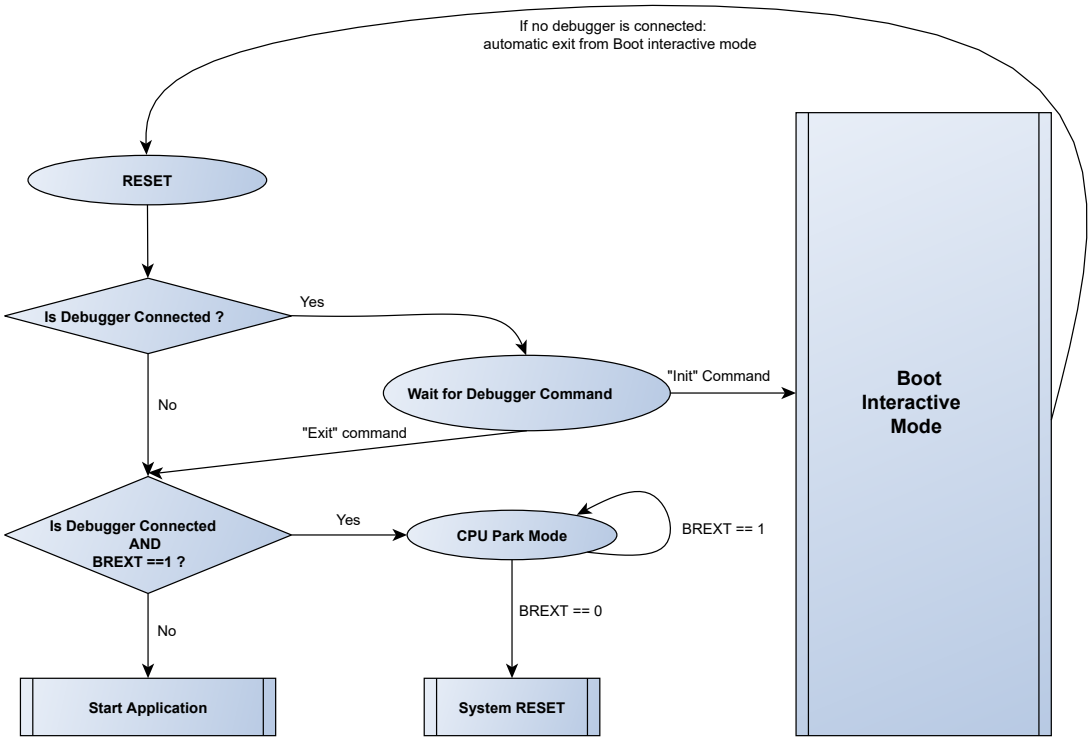
The SAM L10 Boot ROM checks firstly if a debugger is present to enter the Boot Interactive mode which allows the user to perform specific tasks via a debugger connection.

Downloaded from Arrow.com.

Before jumping to the application, the Boot ROM can also enter in a specific mode called CPU Park to allow the debugger to get access to the resources of the device depending on Debug Access Level (DAL).

Note: Boot Interactive and CPU Park modes are described later on.

Figure 14-2. SAM L10 Boot ROM Flow



14.4.1.1 Typical Boot Timings

The delay is given from the release of the CPU reset to the execution of the first instruction of the user code:

Table 14-1. SAM L10 Typical Boot Timing

Time to reach User Code
1.33 ms

14.4.2 SAM L11 Boot ROM Flow

The SAM L11 Boot ROM sequence consists in performing several security tasks (integrity checks, memories and peripherals security attribution, secure boot...) before starting the application.

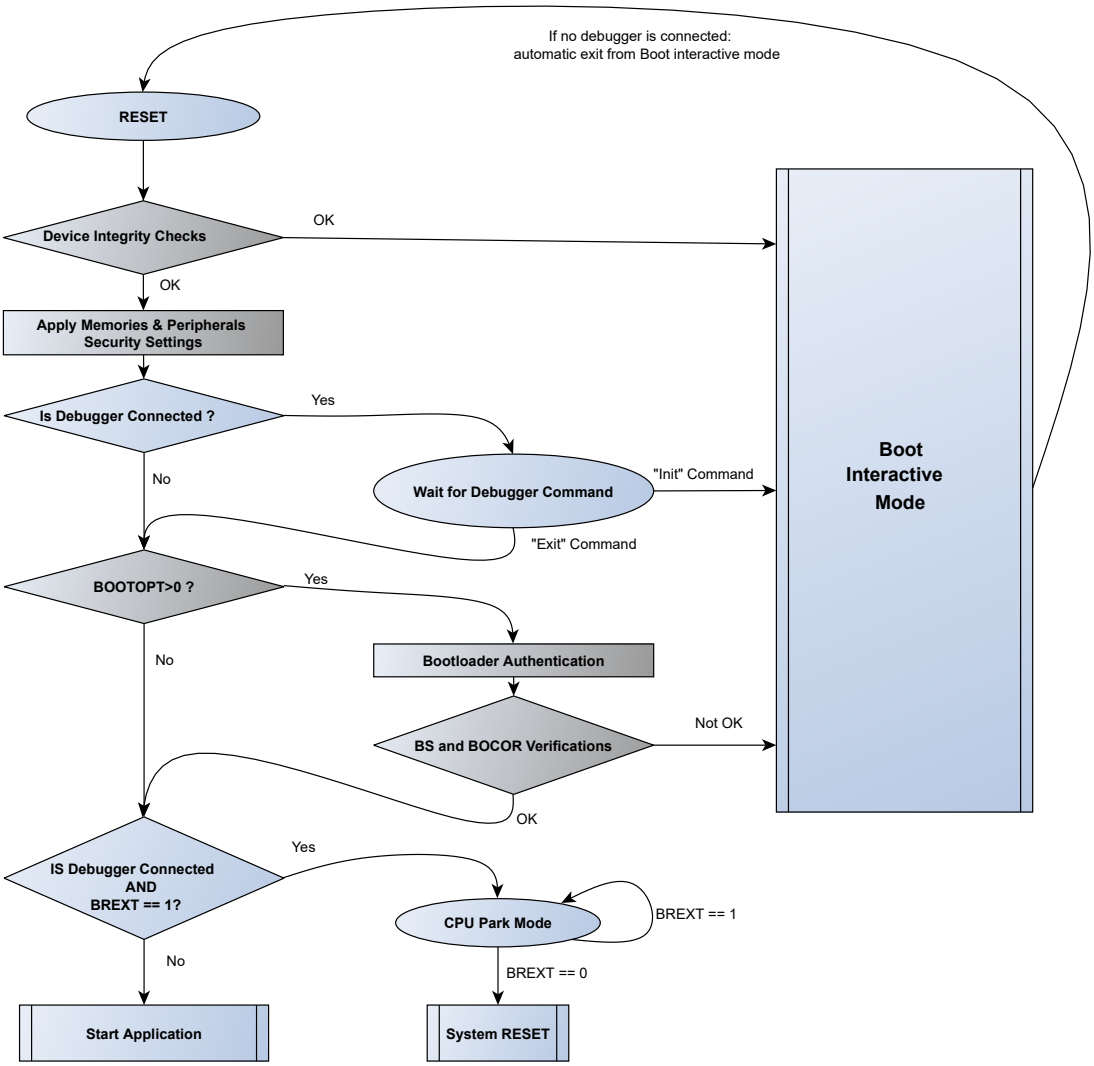
The Boot ROM checks firstly if a debugger is present to enter the Boot Interactive mode which allows the user to perform specific tasks via a debugger connection.

Before jumping to the application in Secure state, the Boot ROM can also enter in a specific mode called CPU Park to allow the debugger to get access to the resources of the device depending on Debug Access Level (DAL).

Note: Boot Interactive and CPU Park modes are described later on.

Downloaded from Arrow.com.

Figure 14-3. SAM L11 Boot ROM Flow



14.4.2.1 Device Integrity Checks

For SAM L11 devices, the Boot ROM performs security checks on two CRCs:

- The User Row CRC (USERCRC) which is located in the NVM User Row (UROW) at: [0x80401C: 0x80401F]:

UROW Offset	Bit Position	Name
0x1C-0x1F	255:224	USERCRC

- The Boot Configuration Row CRC (BOCORCRC) which is located in the NVM Boot Configuration Row (BOCOR) at: [0x80C008:0x80C00B]:

BCOR Offset	Bit Position	Name
0x08-0x0B	95:64	BOCORCRC

14.4.2.1.1 User Row CRC (USER CRC)

USERCRC allows to check the following fuses parameters integrity:

- AS, ANSC, DS, RS

Downloaded from Arrow.com.

- URWEN
- NONSECA, NONSECB, NONSECC

USERCRC is the CRC of the NVM User row area which starts from 0x00804008 and finish at 0x0080401B (bit 64 to bit 223):

Table 14-2. SAM L11 UROW Area Computed in USERCRC

Offset	Bit Pos.	Name							
0x08	71:64	AS							
0x09	79:72	Reserved					ANSC		
0x0A	87:80	Reserved						DS	
0x0B	95:88	Reserved		RS					
0x0C	103:96	Reserved							URWEN
0x0D-0xF	127:104	Reserved							
0x10-0x13	159:128	NONSECA							
0x14-0x17	191:160	NONSECB							
0x18-0x1B	223:192	NONSECC							

14.4.2.1.2 Boot Configuration Row CRC (BOCORCRC)

BOCORCRC allows to check the following fuses parameters integrity:

- BS, BNSC
- BOOTOPT
- BOOTPROT, BCWEN, BCREN

BOCORCRC is the CRC of the NVM Boot Configuration row area, which starts from 0x0080C000 and finish at 0x00800C007 (bit 0 to bit 63).

Table 14-3. SAM L11 BOCOR Area Computed in BOCORCRC

Offset	Bit Pos.	Name							
0x00	7:0	Reserved							
0x01	15:8	BS							
0x02	23:16	Reserved					BNSC		
0x03	31:24	BOOTOPT							
0x04	39:32	BOOTPROT							
0x05	47:40	Reserved							
0x06	55:48	Reserved						BCREN	BCWEN
0x07	63:56	Reserved							

If one of the checks fails, the Boot ROM will report the error to the DSU peripheral and will enter the Boot Interactive mode:

- This will allow, if a debugger is connected, to put the device in the highest debug access level mode (DAL = 2) by issuing a Chip Erase command . Once in that mode, it is possible for a programming tool to reprogram the NVM Rows.
- When the check fails and no debugger is connected, the part will reset and restart the check sequence again.

Note: Boot Interactive mode is described later in this chapter.

Downloaded from Arrow.com.

14.4.2.1.3 CRC Computation and Programming

The CRCs needs to be recalculated and updated in their respective NVM row as soon as a data from any of the checked regions is changed.

Important: USERCRC and BOCORCRC CRCs programming must be done by any programming tool supporting the SAM L11 devices.

The algorithm is a CRC-32 module embedded in the DSU peripheral and that uses for both CRC calculation with the following parameters:

- Width = 32 bits
- Polynomial = 0x04C11DB7 (Poly)
- Initial Value = 0xFFFFFFFF (Init)
- Input Data is reflected (RefIn)
- Output Data is reflected (RefOut)
- No XOR is performed on the output CRC (XorOut)

Example: the DSU CRC of 0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37, 0x38, 0x39 is 0x340BC6D9

14.4.2.2 Memories and Peripherals Configurations Initialization

For SAM L11 devices, memories and peripherals security attributions are done by reading the different fuses values from the NVM User (UROW) and Boot Configuration (BOCOR) rows.

The Boot ROM is responsible for setting these attributions on the different concerned memory and peripheral controllers:

- Set memory security attribution according to AS, ANSC, DS, RS, BS, BSNC and BOOTPROT fuses
- Set peripherals security attribution according to NONSECA, NONSECB and NONSECC fuses

Important: The Boot ROM does not perform any consistency checks on the configured memory attributions (e.g setting BS>BOOTPROT will not trigger any errors during Boot ROM execution).

14.4.2.3 Secure Boot

Depending on the BOOTOPT fuse value (from BOCOR NVM row), the following secure boot integrity checks will be performed on:

- The Flash BS memory area which is composed by:
  - The Flash Secure BOOT memory region
  - The Flash Non-Secure Callable BOOT memory region
- And the NVM Boot Configuration row (BOCOR)

Table 14-4. Secure Boot Options

BOOTOPT	Verified Areas	Verification Method
0	None	-
1	Flash BS Memory Region + NVM BOCOR row	SHA-256

Downloaded from Arrow.com.

BOOTOPT	Verified Areas	Verification Method
2 or 3	Flash BS Memory Region + NVM BOCOR row	SHA-256 with BOOTKEY (defined in BOCOR)
Other Values	None	-

If the verification fails, the Boot ROM will report the error to the DSU peripheral and will enter the Boot Interactive mode. This will allow, if a debugger is connected, to put the device in the highest debug level access mode (DAL = 2) by issuing a Chip Erase command. Once in that mode, it is possible for a programming tool to reprogram the different memory regions and/or NVM rows.

When verification fails and no debugger is connected, the part will reset and restart the integrity checks sequences again.

14.4.2.3.1 Hash algorithm (SHA-256) Verification Method

The verifications are done using the standard SHA256 hash algorithm.

Both Flash BS region and NVM BOCOR row hashes are computed on the defined memory/row area and compared to their expected reference hash value.

Note: The hash consists of 256 bits, i.e. 32 bytes.

SHA256 with BOOTKEY Variant

To prevent unauthorized change of the bootloader code, the hash computation can be slightly modified to require a key to produce a valid hash.

When SHA with BOOTKEY is selected (BOOTOPT=2 or =3), the hash computation (for both Flash BS region and NVM BOCOR row) starts by processing the secure boot key (BOOTKEY) data twice, then proceeds with the rest of data.

This secure boot key (BOOTKEY) is located in the NVM Boot Configuration row (BOCOR) at [0x80C0050:0x80C006F]:

BOCOR Offset	Bit Position	Name
0x50-0x6F	895:640	BOOTKEY

14.4.2.3.2 BS Verification

When BOOTOPT>0, the bootloader authentication starts allowing a secure bootloader code to be protected against inadvertent or malicious changes.

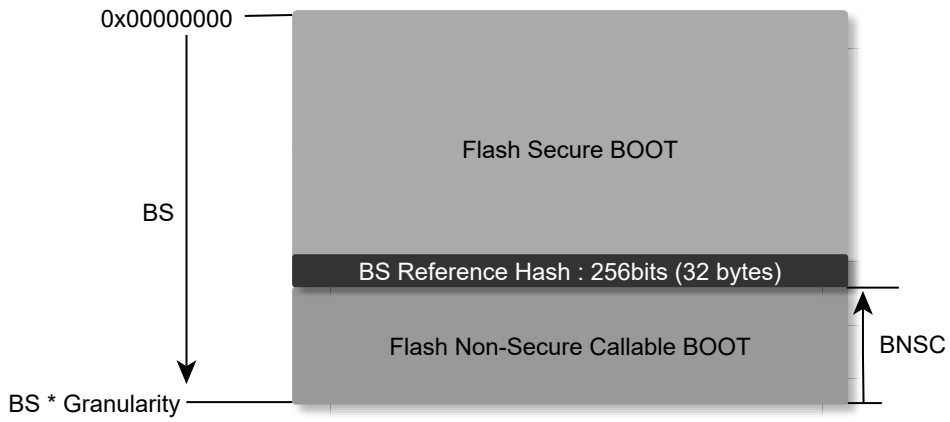
The hash is computed on the Flash Secure BOOT and Flash Non-Secure Callable BOOT (BNSC) regions.

The hash reference value for this area is stored at the end of the Secure BOOT area, just before the Non-Secure Callable BOOT (BNSC) one.

Note: The last 256 bits where the hash is stored are not included in the hash computation.

Downloaded from Arrow.com.

Figure 14-4. BS Hash location in BS memory area



Important: The Non-Secure BOOT region as well as Secure or Non-Secure APPLICATION regions are not part of the Secure Boot verification. So if an authentication of one of these memory regions is required, it must be handled by the user code itself.

14.4.2.3.3 BOCOR Verification

When BOOTOPT>0, the hash for the NVM BOCOR row is computed on the whole NVM BOCOR row excluding BOCORHASH fuse value which is the fuse where to store the hash reference value [0x80C00E0:0x80C00FF]:

BOCOR Offset	Bit Position	Name
0xE0-0xFF	2047:1792	BOCORHASH

14.4.2.4 Typical Boot Timings

Depending on the boot authentication options, the Boot ROM will require a certain time to complete its different tasks.

The delay is given from the release of the CPU reset to the execution of the first instruction of the user code.

Table 14-5. SAM L11 Typical Boot Timings

Boot options	Time to reach User Code
BOOTOPT=0	2.30 ms
BOOTOPT=1, BS=0x40	207 ms
BOOTOPT=1, BS=0x80	409 ms
BOOTOPT=2, BS=0x40	209 ms
BOOTOPT=2, BS=0x80	411 ms

14.4.3 Debug Access Levels

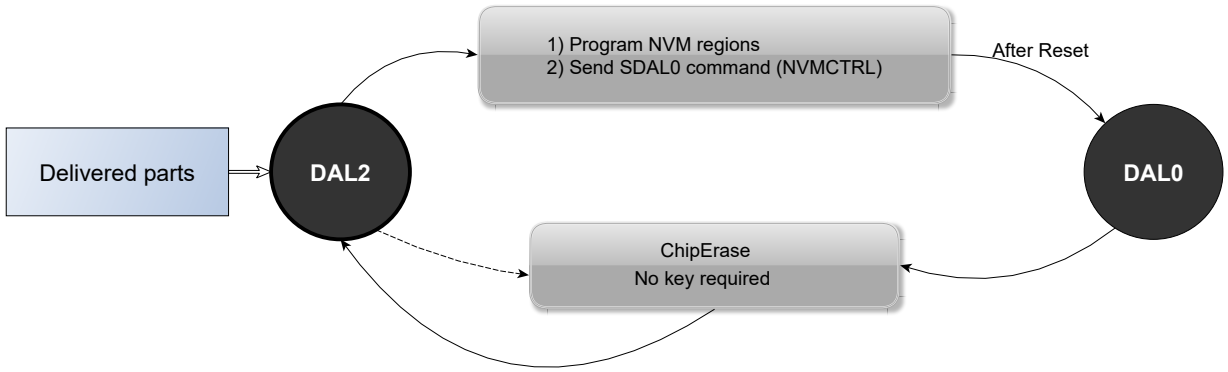
The SAM L10 has only two debug access levels (DAL):

- DAL2: Highest debug level access with no restrictions in term of memory and peripheral accesses.
- DAL0: No access is authorized except with a debugger using the Boot ROM Interactive mode.

Downloaded from Arrow.com.

The possible transitions between each debug access level are described below:

Figure 14-5. SAM L10 Debug Access Levels Transitions

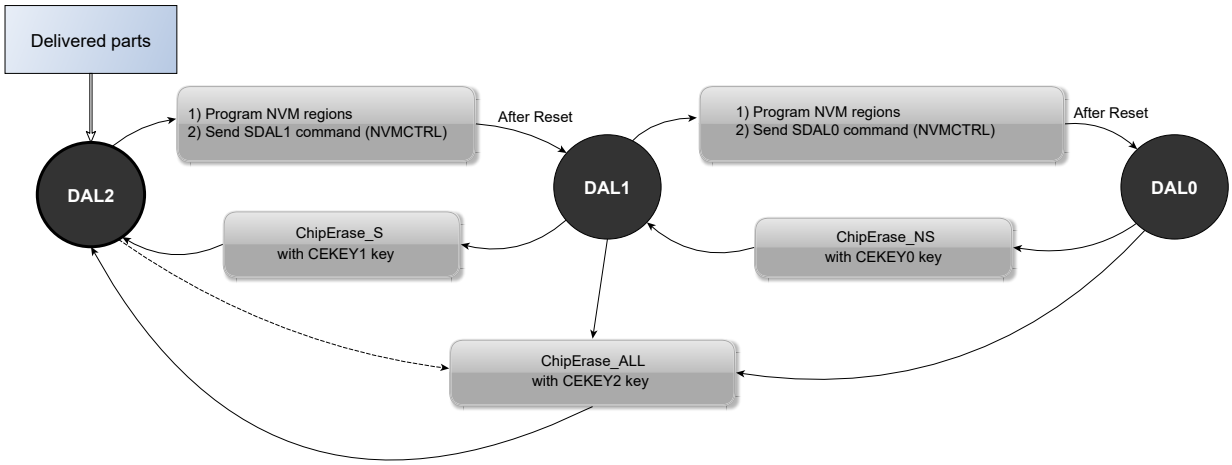


The SAM L11 has three possible debug access levels (DAL):

- DAL2: Highest debug level access with no restrictions in term of memory and peripheral accesses.
- DAL1: Access is limited to the Non-Secure memory regions. Secure memory regions accesses are forbidden.
- DAL0: No access is authorized except with a debugger using the Boot ROM Interactive mode.

The possible transitions between each debug access level are described below:

Figure 14-6. SAM L11 Debug Access Levels Transitions



Decreasing the Debug Level Access is done using the NVMCTRL peripheral command from the debugger or the CPU.

Note: Refer to 30. NVMCTRL – Nonvolatile Memory Controller for more information.

For security reasons, increasing the Debug Level Access is only possible during Boot ROM execution and will be always preceded by a specific chip erase depending on the Debug Access Level.

14.4.4 Chip Erase

The chip erase commands allow to erase memories of the device and provide secure transitions between the different Debug Access Levels.

Downloaded from Arrow.com.

Important: Chip Erase commands are only issued using the Boot ROM Interactive mode (CMD_CE0, CMD_CE1, CMD_CE2 and CMD_CHIPERASE commands).

For SAM L10, the chip erase command does not require a key.

For SAM L11, the chip erase commands are protected with keys (CEKEYx) defined in the NVM BOCOR row.

Note: The chip erase keys can only be read if BOCOR.BCREN=1.

By default, the devices are delivered with these keys set at “All 1s”.

Important: If the key is set at “All 0s”, the corresponding chip erase command is disabled and it will be impossible for the debugger to use it.

The following table gives the effect of the Chip Erase commands on the different memories:

Table 14-6. Chip Erase Commands Effects

	SAM L11			SAM L10
Boot ROM Command	ChipErase_NS (CE0)	ChipErase_S (CE1)	ChipErase_ALL (CE2)	ChipErase (CHIPERASE)
Key Requirement	Yes (CEKEY0)	Yes (CEKEY1)	Yes (CEKEY2)	No
Flash BOOT area BOOTPROT (BS+BNSC+BNS)	No	No	Yes	No
Flash Secure APPLICATION (AS)	No	Yes	Yes	Yes
Flash Non-secure APPLICATION	Yes	Yes	Yes	-
Secure Data Flash (DS)	No	Yes	Yes	Yes
Non-Secure Data Flash	Yes	Yes	Yes	-
NVM User Row (UROW)	No	No	Yes	No
NVM Boot Configuration Row (BOCOR)	No	No	Yes	No
Volatile Memories	Yes	Yes	Yes	Yes
Debugger Access Level after reset	2 (if DAL was 2) else 1	2 (if DAL was 2 or BS==0) else 1	2	2

Note: Only the ChipErase_ALL (CE2) command affects rows belonging to the BOOT area (BOOTPROT fuse bits)

14.4.5 Boot ROM Interactive Mode

The interactive mode allows the user to perform several actions on the device during the Boot ROM execution via a debugger connection.

The debugger communicates with the device using the DSU Boot Communication Channels (BCC). This communication is bi-directional and allows the debugger to post commands and receive status from the Boot ROM.

Note: Refer to Device Service Unit for more information on BCC.

Downloaded from Arrow.com.

14.4.5.1 Enter Interactive Mode (CMD_INIT)

This command allows launching the Boot Interactive command mode of the Boot ROM.

To reach interactive mode, the debugger will trigger a “cold plugging” sequence as described in DSU chapter.

Important: Debugger must not clear DSU.STATUSA.BREXT bit before clearing DSU.STATUSA.CRSTEXT bit.

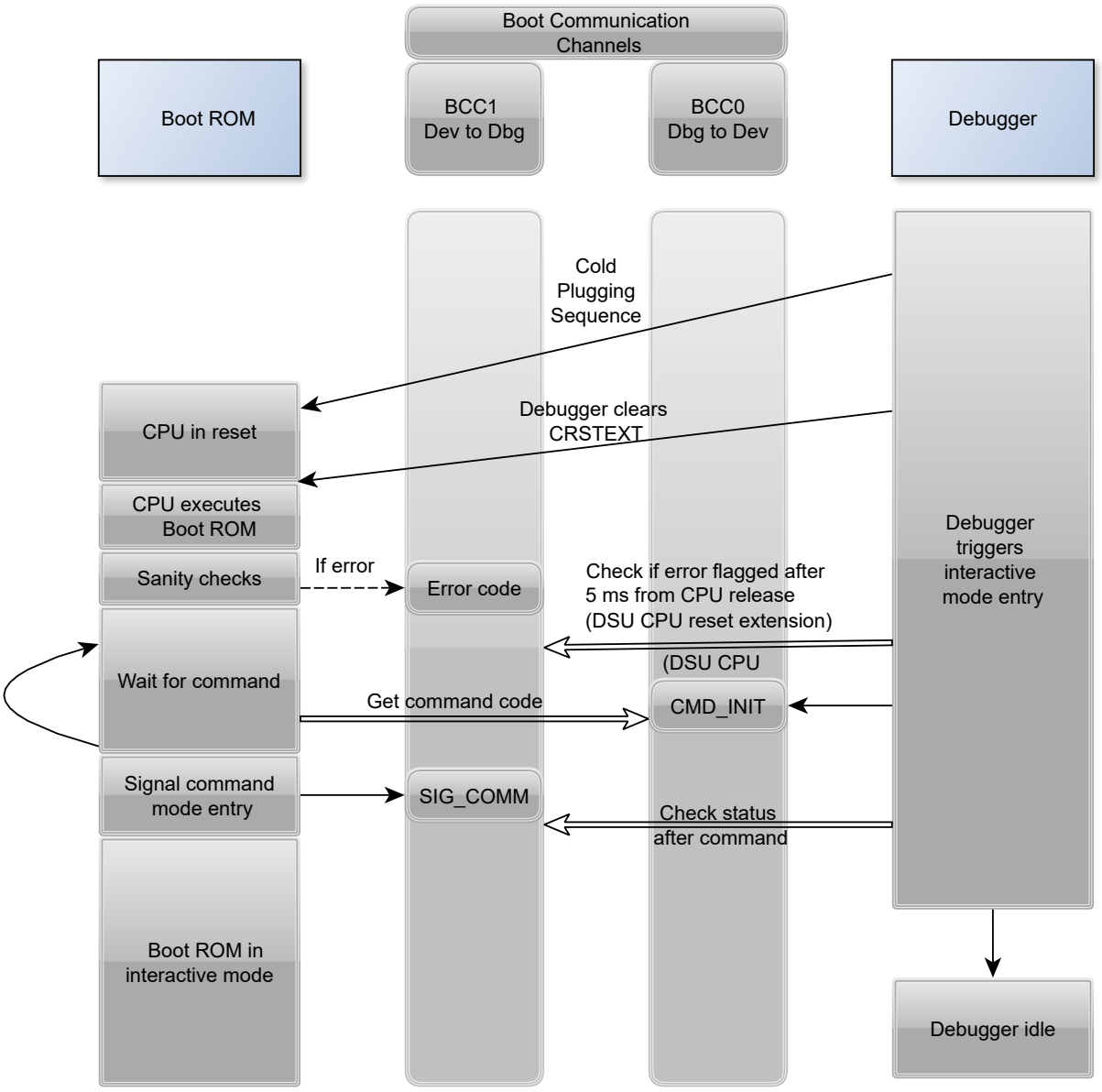
When CRSTEXT is cleared, CPU starts Boot ROM Interactive mode execution. After a small delay (5ms advised), the debugger must check if the Boot ROM has not flagged any errors by checking the BCC1D bit in DSU.STATUSB register.

If no error is reported, the debugger writes the CMD_INIT command to DSU.BCC0 register to request Boot ROM Interactive mode entry. When command is successful, Boot ROM will place the “SIG_COMM” status in DSU.BCC1 register.

Downloaded from Arrow.com.

14.4.5.1.1 CMD_INIT

Figure 14-7. CMD_INIT Flow diagram



14.4.5.2 Exit Interactive Mode (CMD_EXIT)

This command allows exiting the Boot Interactive mode.

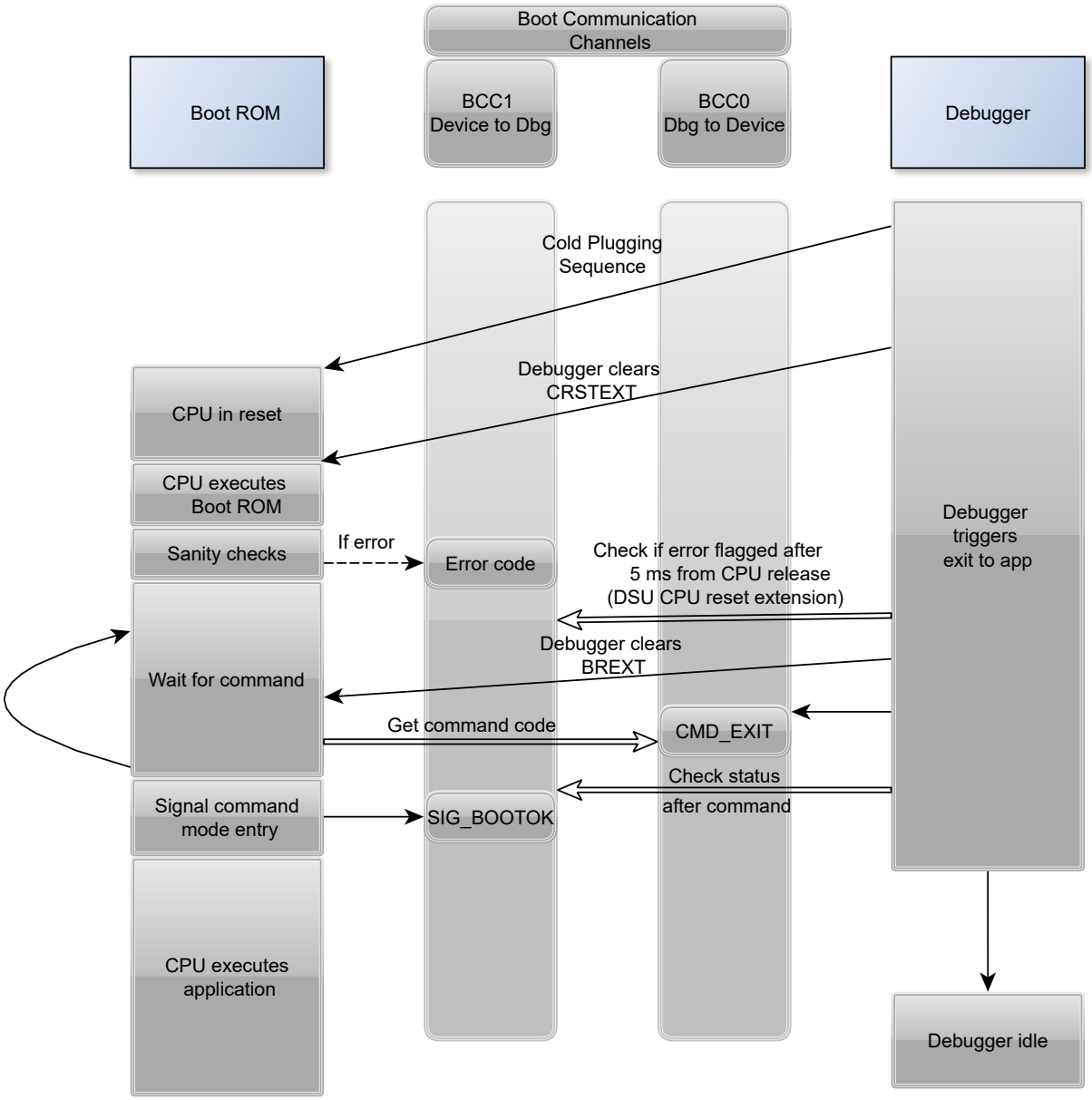
Exiting the Boot Interactive mode allows to jump to one of the following:

- The Application
- The CPU Park Mode

Downloaded from Arrow.com.

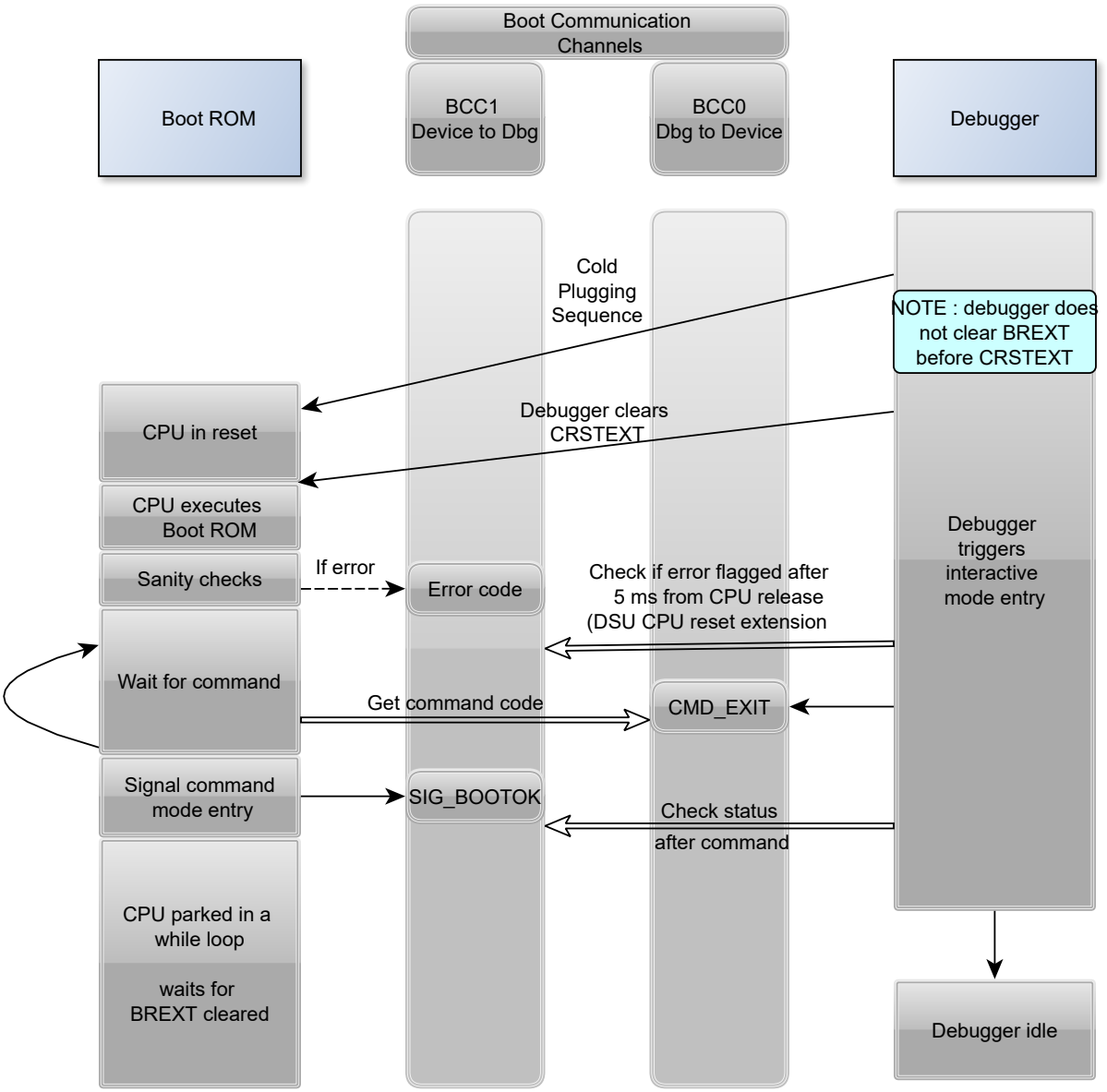
14.4.5.2.1 CMD_EXIT

Figure 14-8. CMD_EXIT to APP flow diagram



Downloaded from Arrow.com.

Figure 14-9. CMD_EXIT to Park mode flow diagram



14.4.5.3 System Reset Request (CMD_RESET)

This command allows resetting the system using a system reset request. Since the reset is executed immediately after receiving the command, no reply is sent to the debugger.

After reset, the CPU executes the Boot ROM code from the beginning

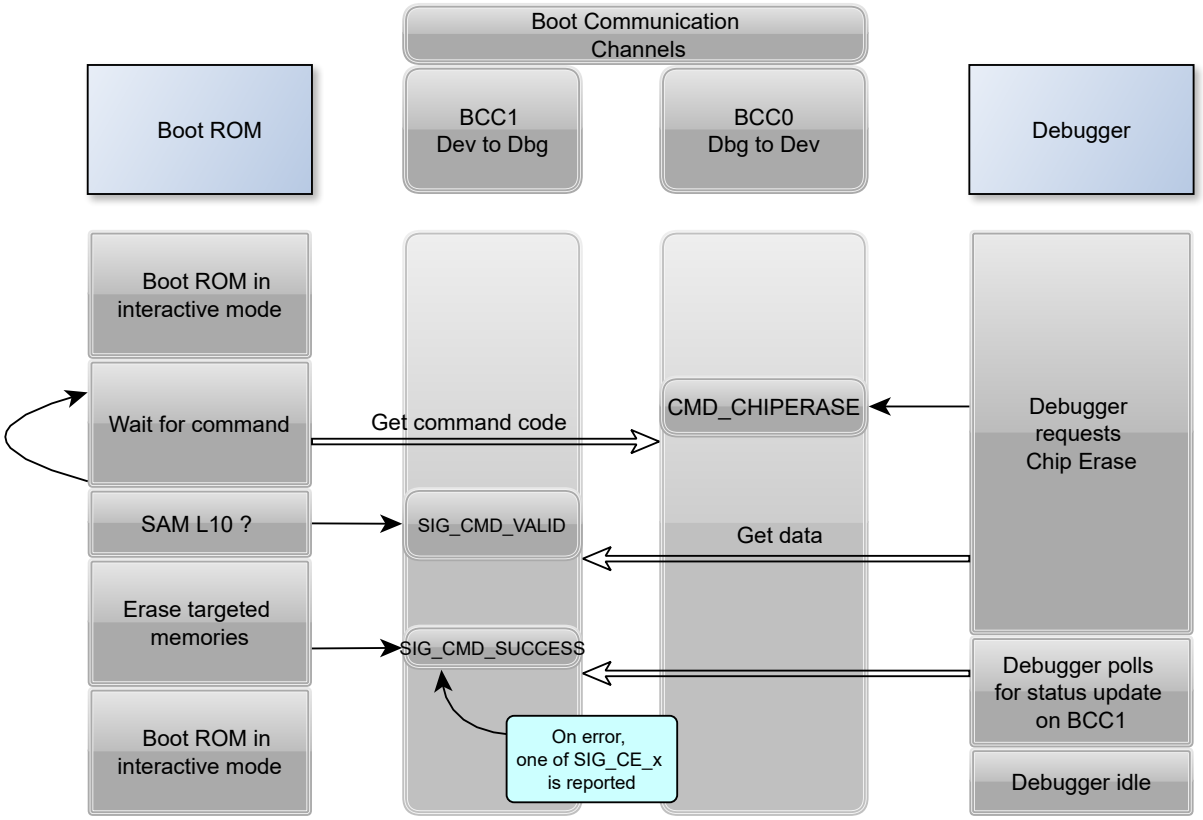
14.4.5.4 Chip Erase (CMD_CHIPERASE) - SAM L10 only

CMD_CHIPERASE command erases the entire device except BOOT area, and reverts to Debug Access Level 2.

Downloaded from Arrow.com.

14.4.5.4.1 CMD_CHIPERASE (SAM L10 only)

Figure 14-10. CMD_CHIPERASE Flow diagram



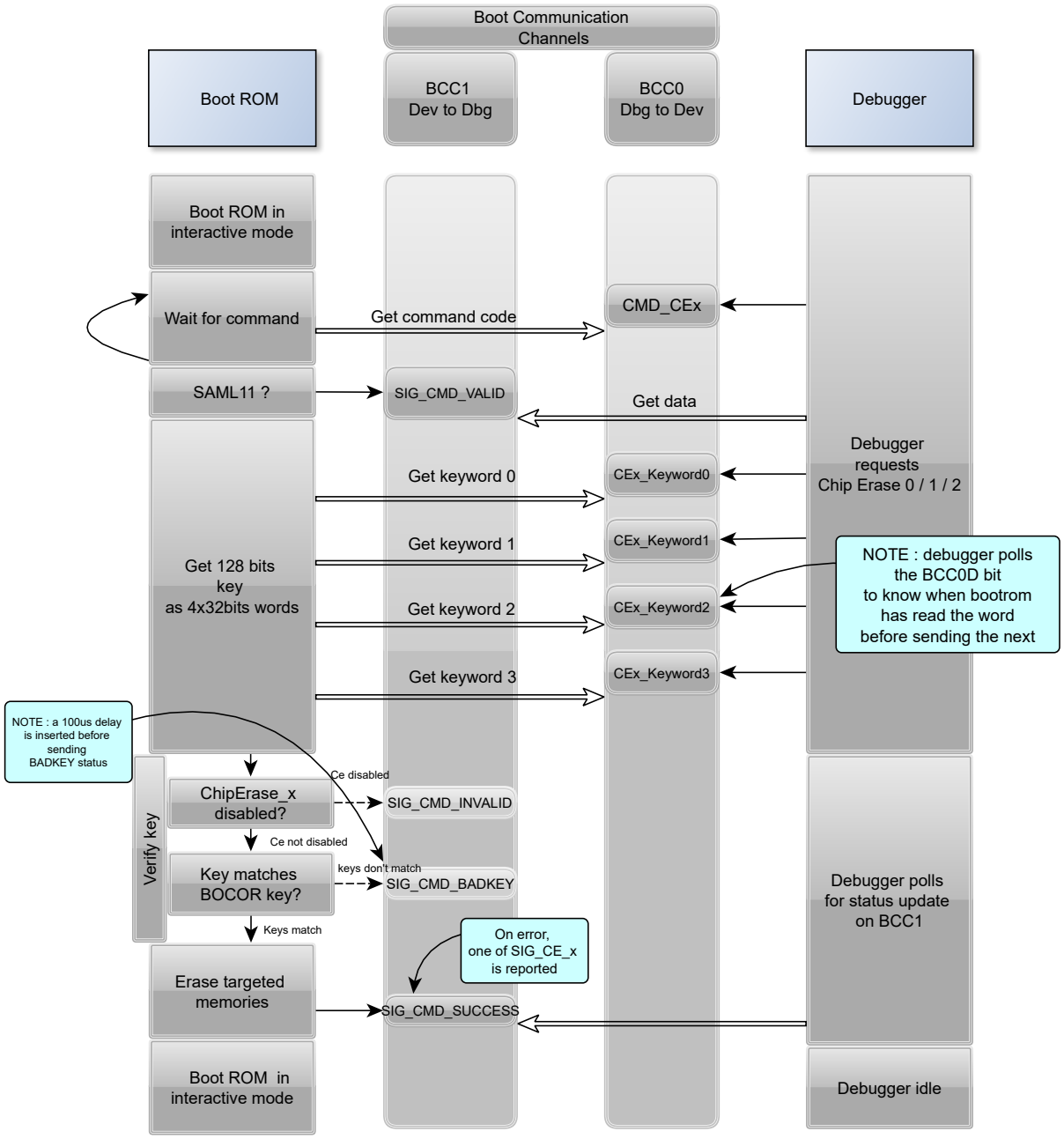
14.4.5.5 Chip Erase (CMD_CEx) - SAM L11 only

CMD_CEx commands are used to erase specific part of the device and to increase the Debug Access Level.

Downloaded from Arrow.com.

14.4.5.5.1 CMD_CEx (SAM L11 only)

Figure 14-11. CMD_CEx Flow diagram



14.4.5.6 NVM Memory Regions Integrity Checks (CMD_CRC)

The Boot ROM provides a way to check the integrity of the embedded non-volatile memories which may be of interest in case of a failure analysis.

This requires the user to place tables describing the memory area to be checked with their expected CRC values.

Note: During this integrity check process, the debugger sends the CRC table address to the device.

Downloaded from Arrow.com.

Important: The table(s) must be programmed by the programming tool in addition to the application binaries.

14.4.5.6.1 CRC Table format

Table 14-7. CRC Table Fields Description

Description	Header	Start Address (1)	Size in bytes (2)	Expected value (3)
Field	HDR	ADDR	SIZE	REFVAL
Offset	0x0	0x4	0x8	0xC
Value	0x43524349	0x00000000	0x100	0xAABBCCDD

Note 1: ADDR must be a multiple of 4 (Only ADDR[31:2] are used).

Note 2: SIZE must be a multiple of 4 (Only SIZE[31:2] are used).

Note 3: The expected value is the computed CRC32 value of the memory target.

14.4.5.6.2 Requirements

- Each table occupies 16 bytes in memory.
- The table must start at a 16byte aligned address. (i.e. 0xXXXXXXX0)
- The table must be placed in the same memory region as its target memory range. (i.e. a table placed in the Secure APPLICATION region can only target Secure APPLICATION memory addresses).
  Note: There are two exceptions to this rule:
  - For SAM L10: all non-volatile memories are considered as a single region (e.g. a table located in Data Flash can target main array)
  - For SAM L11: ANSC and BNSC regions are considered to belong to the same region as their “parent” region: AS for ANSC and BS for BNSC.

14.4.5.6.3 CRC Command Key

The CRC command (CMD_ CRC) requires an access key (CRCKEY) which is in the NVM BOCOR row at: [0x80C040:0x80C04F]:

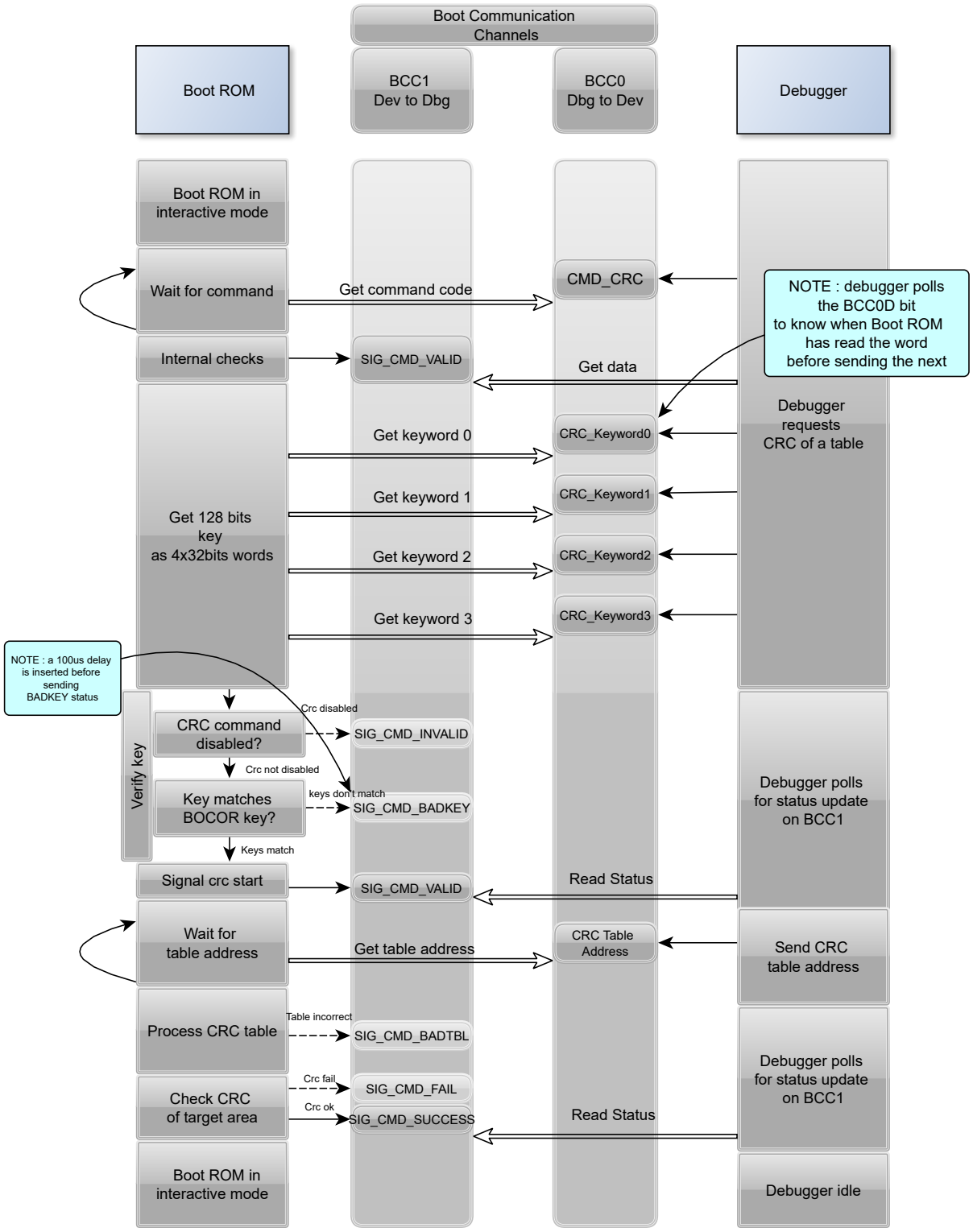
BOCOR Offset	Bit Position	Name
0x40-0x4F	639:512	CRCKEY

Just like the ChipErase keys, the key can be set to all 0s to prevent any access to the command.

Downloaded from Arrow.com.

14.4.5.6.4 CMD_CRC

Figure 14-12. CMD_CRC Flow diagram



Downloaded from Arrow.com.

14.4.5.7 Random Session Key Generation (CMD_DCEK) - SAM L11 only

This command allows using a challenge-response scheme to prevent exposure of the keys in clear text on the debugger communication lines.

The different keys sent by the debugger during the Boot ROM for Chip Erase (CMD_CEx) and CRC (CMD_CRC) commands execution are:

- CRCKEY for CMD_CRC command
- CEKEYx for CMD_CEx commands

Note: The CMD_DCEK command has no effect on the SAM L10, the key derivation will not be enabled.

The random challenge value is generated using the TRNG of the device. It is generated once the CMD_DCEK is received and communicated to the debugger.

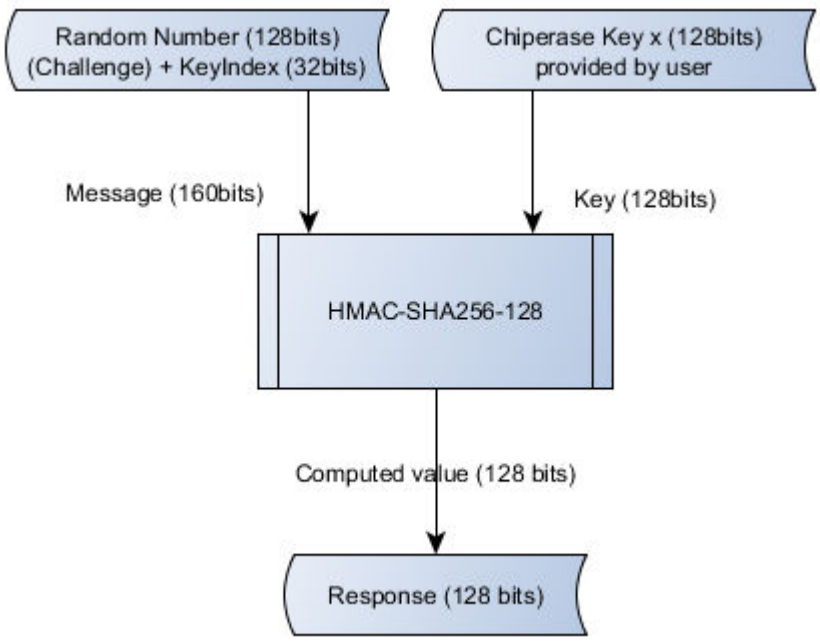
The next CMD_CEx or CMD_CRC commands will expect the key value to be replaced by the computed response corresponding to the challenge.

The challenge value is valid only for the next CMD_CEx/ CMD_CRC command.

Before sending a new CMD_CEx/ CMD_CRC command, a CMD_DCEK shall be used to re-enable the challenge-response scheme a get a new challenge value.

On the debugger side, the response shall be computed using the following algorithm:

Figure 14-13. Debugger Algorithm



Where KeyIndex is:

- 0 for ChipErase_NS
- 1 for ChipErase_S
- 2 for ChipErase_ALL
- 3 for CRC Command

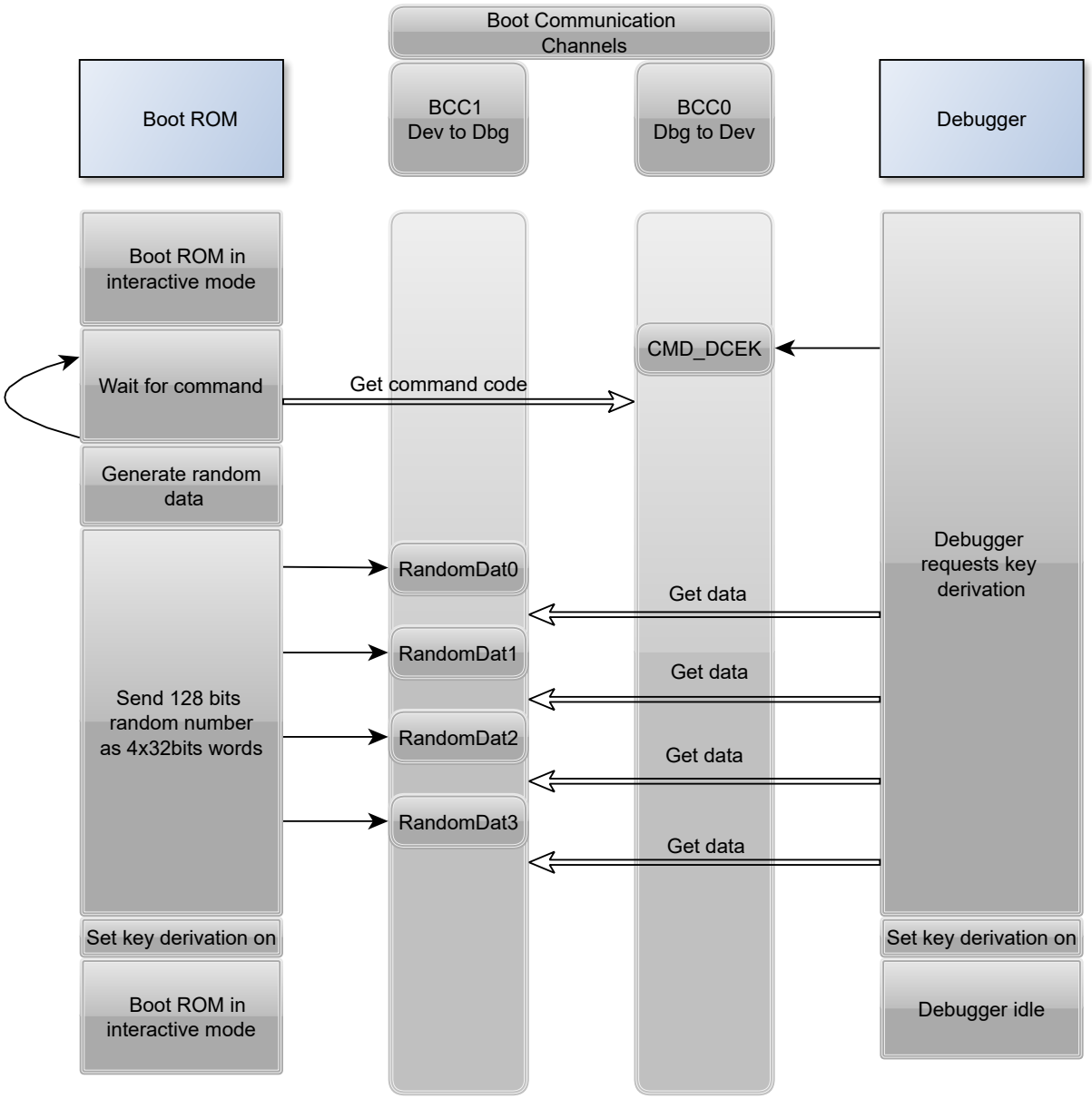
Note:

Downloaded from Arrow.com.

- HMAC is described in FIPS PUB 198-1.
- The hash used for HMAC is SHA256.
- The output of the HMAC-SHA256 is truncated to obtain an HMAC-SHA256-128 as explained in RFC4868.

14.4.5.7.1 CMD_DCEK (SAM L11 only)

Figure 14-14. CMD_DCEK Flow diagram



14.4.5.8 NVM Rows Content Checks (CMD_RAUX)

The Boot ROM provides a way to check the content of the NVM rows.

When device is secured (DAL0), the fuse configuration can still be read by the debugger using the Read Auxiliary command (CMD_RAUX).

Downloaded from Arrow.com.

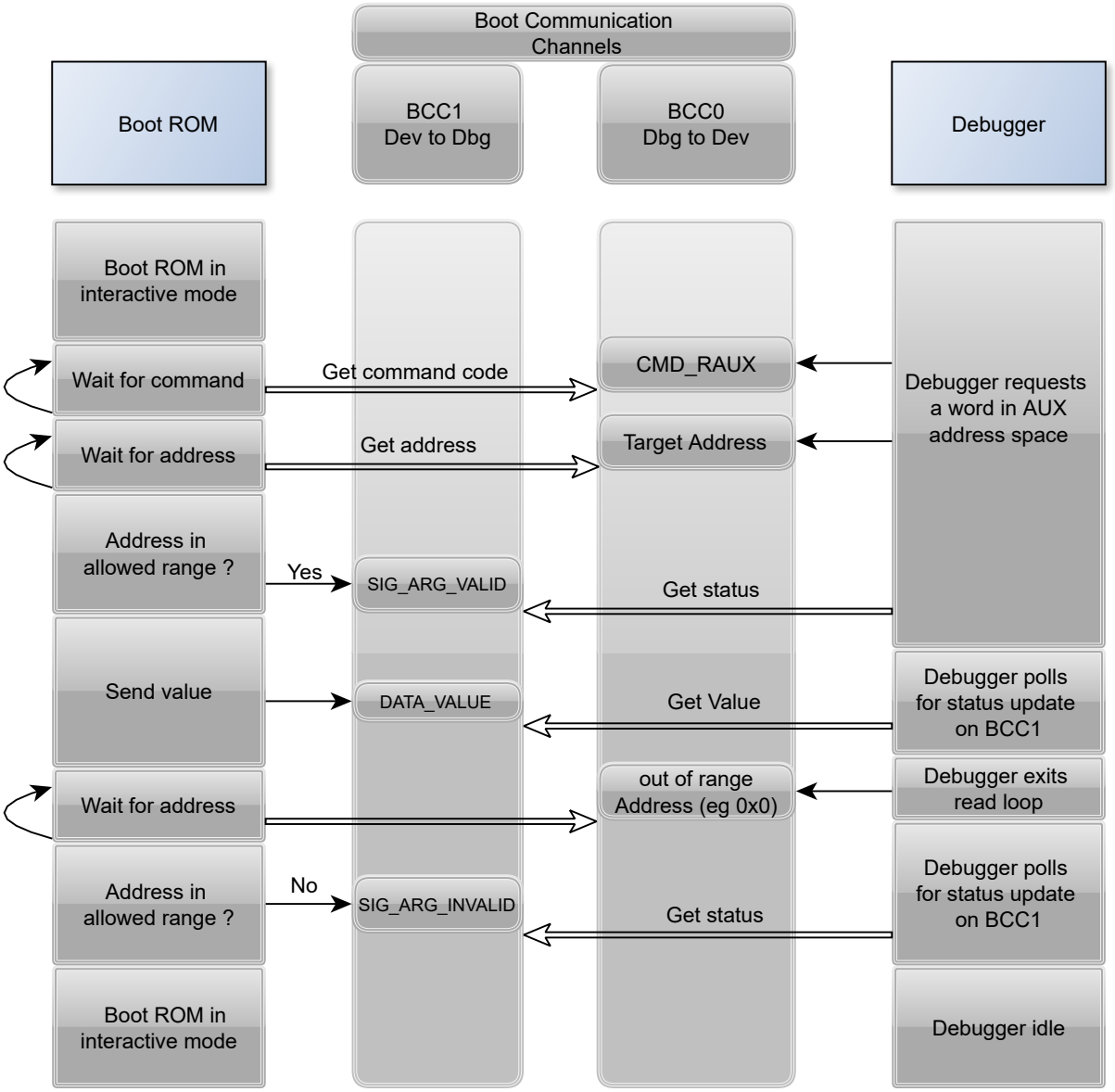
The following areas are accessible:

Table 14-8. Accessible Memory Range by Read Auxiliary Row Command

Area	Start address	End address
User row (UROW)	0x00804000	0x0080401F
Software Calibration row	0x00806020	0x0080602F
Temperature Log row	0x00806038	0x0080603F
Boot Configuration row (BOCOR)	0x0080C000	0x0080C0FF

14.4.5.8.1 CMD_RAUX

Figure 14-15. CMD_RAUX Flow diagram



Note: After the CMD_RAUX is sent, the debugger can read multiple data, the read loop is exit when an out of range address is sent.

Downloaded from Arrow.com.

14.4.5.9 Boot Interactive Mode Commands

Table 14-9. Boot Interactive Mode Commands

Command Name	Description	Command prefix	Command
CMD_INIT	Entering Interactive Mode	0x444247	55
CMD_EXIT	Exit Interactive Mode	0x444247	AA
CMD_RESET	System Reset Request	0x444247	52
CMD_CE0	ChipErase_NS for SAM L11	0x444247	E0
CMD_CE1	ChipErase_S for SAM L11	0x444247	E1
CMD_CE2	ChipErase_ALL for SAM L11	0x444247	E2
CMD_CHIPERASE	ChipErase for SAM L10	0x444247	E3
CMD_CRC	NVM Memory Regions Integrity Checks	0x444247	C0
CMD_DCEK	Random Session Key Generation for SAM L11	0x444247	44
CMD_RAUX	NVM Rows Integrity Checks	0x444247	4C

14.4.5.10 Boot Interactive Mode Status

Table 14-10. Boot Interactive Mode Status

Status Name	Description	Status prefix	Status coding
SIG_NO	No Error	0xEC0000	00
SIG_SAN_FFF	Fresh from factory error	0xEC0000	10
SIG_SAN_UROW	UROW checksum error	0xEC0000	11
SIG_SAN_SECEN	SECEN parameter error	0xEC0000	12
SIG_SAN_BOCOR	BOCOR checksum error	0xEC0000	13
SIG_SAN_BOOTPROT	BOOTPROT parameter error	0xEC0000	14
SIG_SAN_NOSECREG	No secure register parameter error	0xEC0000	15
SIG_COMM	Debugger start communication command	0xEC0000	20
SIG_CMD_SUCCESS	Debugger command success	0xEC0000	21
SIG_CMD_FAIL	Debugger command fail	0xEC0000	22
SIG_CMD_BADKEY	Debugger bad key	0xEC0000	23
SIG_CMD_VALID	Valid command	0xEC0000	24
SIG_CMD_INVALID	Invalid command	0xEC0000	25
SIG_ARG_VALID	Valid argument	0xEC0000	26
SIG_ARG_INVALID	Invalid argument	0xEC0000	27
SIG_CE_CVM	Chip erase error: CVM	0xEC0000	30
SIG_CE_ARRAY_ERASEFAIL	Chip erase error: array erase fail	0xEC0000	31
SIG_CE_ARRAY_NVME	Chip erase error: array NVME	0xEC0000	32
SIG_CE_DATA_ERASEFAIL	Chip erase error: data erase fail	0xEC0000	33
SIG_CE_DATA_NVME	Chip erase error: data NVME	0xEC0000	34
SIG_CE_BCUR	Chip erase error: BOCOR, UROW	0xEC0000	35
SIG_CE_BC	Chip erase error: BC check	0xEC0000	36
SIG_BOOT_OPT	BOOTOPT parameter error	0xEC0000	40

Downloaded from Arrow.com.

Status Name	Description	Status prefix	Status coding
SIG_BOOT_ERR	Boot image hash verify fail	0xEC0000	41
SIG_BOCOR_HASH	BOCOR hash error	0xEC0000	42
SIG_CRC_BADTBL	Bad CRC table	0xEC0000	50
SIG_SECEN0_ERR	PAC or IDAU cfg check failure	0xEC0000	60
SIG_SECEN1_ERR	PAC or IDAU cfg check failure	0xEC0000	61
SIG_EXIT_ERR	Exit: BC or check error	0xEC0000	70
SIG_HARDFAULT	Hardfault error	0xEC0000	F0
SIG_BOOTOK	Boot ROM ok to exit	0xEC0000	39

14.4.6 CPU Park mode

This mode allows the debugger to get access to the resources of the device during Boot ROM execution while the CPU is trapped in a while loop. The debug access level when entering that mode corresponds to the DAL value which is programmed in the device.

Important: This mode is the recommended way to enter a debugging session in a safe way even if it is also possible to launch a debug session when the application is running.

This mode is reached by sending the Exit command (CMD_EXIT) without clearing the DSU.STATUSA.BREXT bit to the Boot ROM.

As soon as the BREXT bit is cleared, the device exits this state and performs a system reset.

At this point, the MPU is still enabled and prevents software execution elsewhere than in Boot ROM region.

If the host needs to run software on the device, MPU shall be disabled by accessing the Cortex-M23 MPU CTRL register with the debugger.

Downloaded from Arrow.com.

15. PAC - Peripheral Access Controller

15.1 Overview

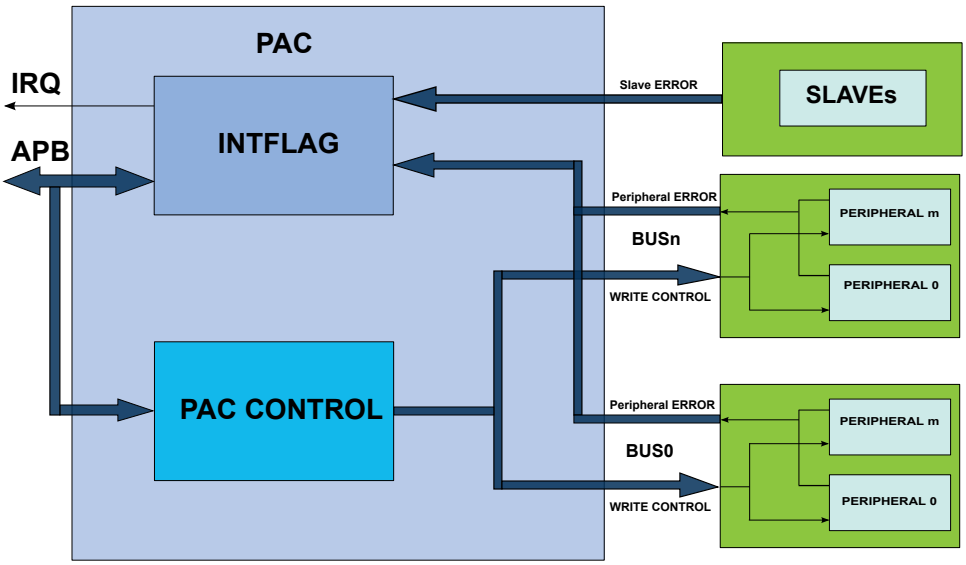
The Peripheral Access Controller provides an interface for the locking and unlocking and for managing security attribution of peripheral registers within the device. It reports all violations that could happen when accessing a peripheral: write protected access, illegal access, enable protected access, access when clock synchronization or software reset is on-going. These errors are reported in a unique interrupt flag for a peripheral. The PAC module also reports errors occurring at the slave bus level, when an access to a non-existing address is detected.

15.2 Features

- Manages write protection access and reports access errors for the peripheral modules or bridges.
- Manages security attribution for the peripheral modules **(SAM L11)**

15.3 Block Diagram

Figure 15-1. PAC Block Diagram



15.4 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

15.4.1 IO Lines

Not applicable.

15.4.2 Power Management

The PAC can continue to operate in any Sleep mode where the selected source clock is running. The PAC interrupts can be used to wake up the device from Sleep modes. The events can trigger other operations in the system without exiting sleep modes.

Downloaded from Arrow.com.

Related Links

22. PM – Power Manager

15.4.3 Clocks

The PAC bus clock (CLK_PAC_APB) can be enabled and disabled in the Main Clock module. The default state of CLK_PAC_APB can be found in the related links.

Related Links

19. MCLK – Main Clock

19.6.2.6 Peripheral Clock Masking

15.4.4 DMA

Not applicable.

15.4.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. Using the PAC interrupt requires the Interrupt Controller to be configured first.

Table 15-1. Interrupt Lines

Instances	NVIC Line
PAC	ERR

15.4.6 Events

The events are connected to the Event System, which may need configuration.

Related Links

33. EVSYS – Event System

15.4.7 Debug Operation

When the CPU is halted in Debug mode, write protection of all peripherals is disabled and the PAC continues normal operation.

15.4.8 Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Write Control (WRCTRL) register
- AHB Slave Bus Interrupt Flag Status and Clear (INTFLAGAHB) register
- Peripheral Interrupt Flag Status and Clear n (INTFLAG A/B/C...) registers

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

15.4.9 SAM L11 TrustZone Specific Register Access Protection

All PAC registers can only be accessed in the secure alias, with the following exceptions:

- Write Control (WRCTRL) register is also accessible in the Non-Secure Alias, but only for write protection requests on non-secured peripherals.
- Peripheral Write Protection Status (STATUSn) registers are also accessible in the Non-Secure Alias, but they will only report information on non-secured peripherals.

Downloaded from Arrow.com.

Note: Refer to the *Mix-Secure Peripherals* section in the *SAM L11 Security Features* chapter for more information.

15.5 Functional Description

15.5.1 Principle of Operation

The Peripheral Access Control module allows the user to set a write protection or security attribution on peripheral modules and generate an interrupt in case of a peripheral access violation. The peripheral's protection can be set, cleared or locked at the user discretion. A set of Interrupt Flag and Status registers informs the user on the status of the violation in the peripherals. In addition, slaves bus errors can be also reported in the cases where reserved area is accessed by the application.

15.5.2 Basic Operation

15.5.2.1 Initialization

After reset, the PAC is enabled.

15.5.2.2 Initialization, Enabling and Resetting

The PAC is always enabled after reset.

Only a hardware reset will reset the PAC module.

15.5.2.3 Operations

The PAC module allows the user to set, clear or lock the write protection status and security attribution of all peripherals on all Peripheral Bridges.

If a peripheral register violation occurs, the Peripheral Interrupt Flag n registers (INTFLAGn) are updated to inform the user on the status of the violation in the peripherals connected to the Peripheral Bridge n (n = A,B,C ...). The corresponding Peripheral Write Control Status n register (STATUSn) gives the state of the write protection for all peripherals connected to the corresponding Peripheral Bridge n. The corresponding Peripheral Non-Secure Status n register (NONSECn) gives the state of the security attribution for all peripherals connected to the corresponding Peripheral Bridge n. Refer to 15.5.2.4 Peripheral Access Errors for details.

The PAC module also report the errors occurring at slave bus level when an access to reserved area is detected. AHB Slave Bus Interrupt Flag register (INTFLAGAHB) informs the user on the status of the violation in the corresponding slave. Refer to the 15.5.2.9 AHB Slave Bus Errors for details.

15.5.2.4 Peripheral Access Errors

The following events will generate a Peripheral Access Error:

- Protected write: To avoid unexpected writes to a peripheral's registers, each peripheral can be write protected. Only the registers denoted as "PAC Write-Protection" in the module's datasheet can be protected. If a peripheral is not write protected, write data accesses are performed normally. If a peripheral is write protected and if a write access is attempted, data will not be written and peripheral returns an access error. The corresponding interrupt flag bit in the INTFLAGn register will be set.
- Illegal access: Access to an unimplemented register within the module.
- Synchronized write error: For write-synchronized registers an error will be reported if the register is written while a synchronization is ongoing.

When any of the INTFLAGn registers bit are set, an interrupt will be requested if the PAC interrupt enable bit is set.

Downloaded from Arrow.com.

15.5.2.5 Write Access Protection Management

Peripheral access control can be enabled or disabled by writing to the WRCTRL register.

The data written to the WRCTRL register is composed of two fields; WRCTRL.PERID and WRCTRL.KEY. The WRCTRL.PERID is an unique identifier corresponding to a peripheral. The WRCTRL.KEY is a key value that defines the operation to be done on the control access bit. These operations can be “clear protection”, “set protection” and “set and lock protection bit”.

The “clear protection” operation will remove the write access protection for the peripheral selected by WRCTRL.PERID. Write accesses are allowed for the registers in this peripheral.

The “set protection” operation will set the write access protection for the peripheral selected by WRCTRL.PERID. Write accesses are not allowed for the registers with write protection property in this peripheral.

The “set and lock protection” operation will set the write access protection for the peripheral selected by WRCTRL.PERID and locks the access rights of the selected peripheral registers. The write access protection will only be cleared by a hardware reset.

The peripheral access control status can be read from the corresponding STATUSn register.

15.5.2.6 Write Access Protection Management Errors

Only word-wise writes to the WRCTRL register will effectively change the access protection. Other type of accesses will have no effect and will cause a PAC write access error. This error is reported in the INTFLAGn.PAC bit corresponding to the PAC module.

PAC also offers an additional safety feature for correct program execution with an interrupt generated on double write clear protection or double write set protection. If a peripheral is write protected and a subsequent set protection operation is detected then the PAC returns an error, and similarly for a double clear protection operation.

In addition, an error is generated when writing a “set and lock” protection to a write-protected peripheral or when a write access is done to a locked set protection. This can be used to ensure that the application follows the intended program flow by always following a write protect with an unprotect and conversely. However in applications where a write protected peripheral is used in several contexts, e.g. interrupt, care should be taken so that either the interrupt can not happen while the main application or other interrupt levels manipulates the write protection status or when the interrupt handler needs to unprotect the peripheral based on the current protection status by reading the STATUS register.

The errors generated while accessing the PAC module registers (eg. key error, double protect error...) will set the INTFLAGn.PAC flag.

15.5.2.7 SAM L11 Security Attribution Management

The peripheral security attribution status can be read from the corresponding NONSECn register.

15.5.2.8 SAM L11 Security Attribution Management Errors

The errors generated while accessing the PAC module registers (e.g., key error, double protect error...) will set the INTFLAGn.PAC flag.

15.5.2.9 AHB Slave Bus Errors

The PAC module reports errors occurring at the AHB Slave bus level. These errors are generated when an access is performed at an address where no slave (bridge or peripheral) is mapped or where non-secure accesses are prohibited. These errors are reported in the corresponding bits of the INTFLAGAHB register.

Downloaded from Arrow.com.

15.5.2.10 Generating Events

The PAC module can also generate an event when any of the Interrupt Flag registers bit are set. To enable the PAC event generation, the control bit EVCTRL.ERREO must be set a '1'.

15.5.3 DMA Operation

Not applicable.

15.5.4 Interrupts

The PAC has the following interrupt source:

- Error (ERR): Indicates that a peripheral access violation occurred in one of the peripherals controlled by the PAC module, or a bridge error occurred in one of the bridges reported by the PAC
  - This interrupt is a synchronous wake-up source.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAGAHB and INTFLAGn) registers is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register. An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the PAC is reset. All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. The user must read the INTFLAGAHB and INTFLAGn registers to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated.

Related Links

22.6.3.3 Sleep Mode Controller

15.5.5 Events

The PAC can generate the following output event:

- Error (ERR): Generated when one of the interrupt flag registers bits is set

Writing a '1' to an Event Output bit in the Event Control Register (EVCTRL.ERREO) enables the corresponding output event. Writing a '0' to this bit disables the corresponding output event.

15.5.6 Sleep Mode Operation

In Sleep mode, the PAC is kept enabled if an available bus master (CPU, DMA) is running. The PAC will continue to catch access errors from the module and generate interrupts or events.

15.5.7 SAM L11 Secure and Non-Secure Read/Write Accesses

Non-Secure write to EVCTRL, INTENCLR, INTENSET, INTFLAGAHB, INTFLAGx, and NONSECx registers is prohibited.

Non-Secure read to EVCTRL, INTENCLR, INTENSET, INTFLAGAHB, and INTFLAGx registers will return zero with no error resulting.

Non-secure write to a bit of STATUSx registers (by writing to the WRCTRL register) is prohibited if the corresponding bit in NONSECx is zero.

STATUSx bits relating to secure peripherals (i.e., the corresponding bits in NONSECx are zero), read as zero in Non-Secure mode, with no error resulting.

Downloaded from Arrow.com.

15.5.8 Synchronization

Not applicable.

Downloaded from Arrow.com.

15.6 Register Summary

Important:
For **SAM L11**, the PAC register map is automatically duplicated in a Secure and Non-Secure alias:

- The Non-Secure alias is at the peripheral base address
- The Secure alias is located at the peripheral base address + 0x200

Refer to *Mix-Secure Peripherals* for more information on register access rights

Offset	Name	Bit Pos.								
0x00	WRCTRL	7:0	PERID[7:0]							
		15:8	PERID[15:8]							
		23:16	KEY[7:0]							
		31:24								
0x04	EVCTRL	7:0								ERREO
0x05 ... 0x07	Reserved									
0x08	INTENCLR	7:0								ERR
0x09	INTENSET	7:0								ERR
0x0A ... 0x0F	Reserved									
0x10	INTFLAGAHB	7:0	BROM	HSRAMDSU	HSRAMDMAC	HSRAMCPU	HPB2	HPB1	HPB0	FLASH
		15:8								
		23:16								
		31:24								
0x14	INTFLAGA	7:0	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM	PAC
		15:8			AC	PORT	FREQM	EIC	RTC	WDT
		23:16								
		31:24								
0x18	INTFLAGB	7:0				Reserved	DMAC	NVMCTRL	DSU	IDAU
		15:8								
		23:16								
		31:24								
0x1C	INTFLAGC	7:0	ADC	TC2	TC1	TC0	SERCOM2	SERCOM1	SERCOM0	EVSYS
		15:8			TRAM	OPAMP	CCL	TRNG	PTC	DAC
		23:16								
		31:24								
0x20 ... 0x33	Reserved									

Downloaded from Arrow.com.

Offset	Name	Bit Pos.								
0x34	STATUSA	7:0	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM	PAC
		15:8			AC	PORT	FREQM	EIC	RTC	WDT
		23:16								
		31:24								
0x38	STATUSB	7:0				Reserved	DMAC	NVMCTRL	DSU	IDAU
		15:8								
		23:16								
		31:24								
0x3C	STATUSC	7:0	ADC	TC2	TC1	TC0	SERCOM2	SERCOM1	SERCOM0	EVSYS
		15:8			TRAM	OPAMP	CCL	TRNG	PTC	DAC
		23:16								
		31:24								
0x40 ... 0x53	Reserved									
0x54	NONSECA	7:0	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM	PAC
		15:8			AC	PORT	FREQM	EIC	RTC	WDT
		23:16								
		31:24								
0x58	NONSECB	7:0				HMATRIXHS	DMAC	NVMCTRL	DSU	IDAU
		15:8								
		23:16								
		31:24								
0x5C	NONSECC	7:0	ADC	TC2	TC1	TC0	SERCOM2	SERCOM1	SERCOM0	EVSYS
		15:8			TRAM	OPAMP	CCL	TRNG	PTC	DAC
		23:16								
		31:24								

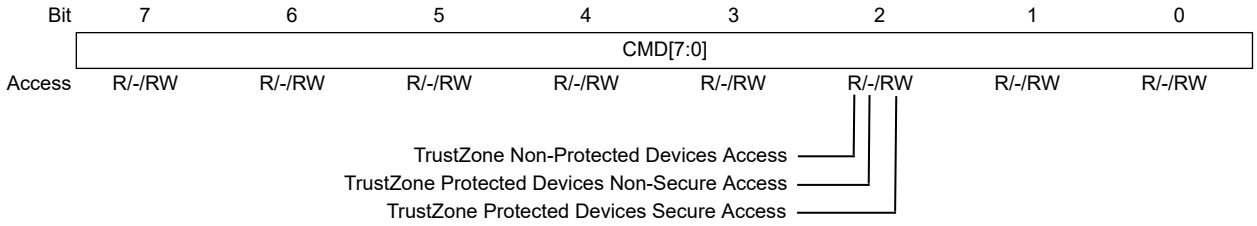
15.7 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to the related links.

On SAM L11 devices, the Mix-Secure peripheral has different types of registers (Non-Secure, Secure, Write-Secure, Mix-Secure, and Write-Mix-Secure) with different access permissions for each bitfield. Refer to *Mix-Secure Peripherals* for more details. In the following register descriptions, the access permissions are specified as shown in the following figure.

Downloaded from Arrow.com.



Downloaded from Arrow.com.

15.7.1 Write Control

Name: WRCTRL
Offset: 0x00
Reset: 0x00000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	KEY[7:0]							
Access	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PERID[15:8]							
Access	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PERID[7:0]							
Access	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – KEY[7:0] Peripheral Access Control Key
These bits define the peripheral access control key:

Value	Name	Description
0x0	OFF	No action
0x1	CLEAR	Clear the peripheral write control
0x2	SET	Set the peripheral write control
0x3	LOCK	Set and lock the peripheral write control until the next hardware reset

Bits 15:0 – PERID[15:0] Peripheral Identifier
The PERID represents the peripheral whose control is changed using the WRCTRL.KEY. The Peripheral Identifier is calculated by the following formula:

$$PERID = 32^{*}\,\mathrm{BridgeNumber} + \mathrm{N}$$

Where BridgeNumber represents the Peripheral Bridge Number (0 for Peripheral Bridge A, 1 for Peripheral Bridge B, etc.). N represents the peripheral index from the respective Peripheral Bridge Number, which can be retrieved in the *Peripherals Configuration Summary* table:

Downloaded from Arrow.com.

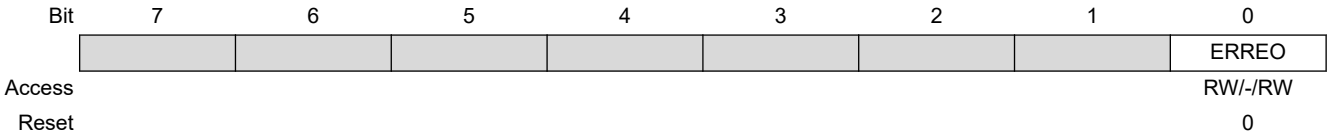
Table 15-2. PERID Values

Peripheral Bridge Name	BridgeNumber	PERID Values
A	0	0+N
B	1	32+N
C	2	64+N

Downloaded from Arrow.com.

15.7.2 Event Control

Name: EVCTRL
Offset: 0x04
Reset: 0x00
Property: Secure



Bit 0 – ERREO Peripheral Access Error Event Output
This bit indicates if the Peripheral Access Error Event Output is enabled or disabled. When enabled, an event will be generated when one of the interrupt flag registers bits (INTFLAGAHB, INTFLAGn) is set:

Value	Description
0	Peripheral Access Error Event Output is disabled.
1	Peripheral Access Error Event Output is enabled.

Downloaded from Arrow.com.

15.7.3 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x08
Reset: 0x00
Property: PAC Write-Protection, Secure

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Bit	7	6	5	4	3	2	1	0
								ERR
Access								RW/-/RW
Reset								0

Bit 0 – ERR Peripheral Access Error Interrupt Disable
This bit indicates that the Peripheral Access Error Interrupt is enabled and an interrupt request will be generated when one of the interrupt flag registers bits (INTFLAGAHB, INTFLAGn) is set:

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Peripheral Access Error interrupt Enable bit and disables the corresponding interrupt request.

Value	Description
0	Peripheral Access Error interrupt is disabled.
1	Peripheral Access Error interrupt is enabled.

Downloaded from Arrow.com.

15.7.4 Interrupt Enable Set

Name: INTENSET
Offset: 0x09
Reset: 0x00
Property: PAC Write-Protection, Secure

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENCLR).

Bit	7	6	5	4	3	2	1	0
								ERR
Access								RW/-/RW
Reset								0

Bit 0 – ERR Peripheral Access Error Interrupt Enable
This bit indicates that the Peripheral Access Error Interrupt is enabled and an interrupt request will be generated when one of the interrupt flag registers bits (INTFLAGAHB, INTFLAGn) is set:

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Peripheral Access Error interrupt Enable bit and enables the corresponding interrupt request.

Value	Description
0	Peripheral Access Error interrupt is disabled.
1	Peripheral Access Error interrupt is enabled.

Downloaded from Arrow.com.

15.7.5 AHB Slave Bus Interrupt Flag Status and Clear

Name: INTFLAGAHB
Offset: 0x10
Reset: 0x000000
Property: Secure

This flag is cleared by writing a '1' to the flag.

This flag is set when an access error is detected by the SLAVE n, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the corresponding INTFLAGAHB interrupt flag.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	BROM	HSRAMDSU	HSRAMDMAC	HSRAMCPU	HPB2	HPB1	HPB0	FLASH
Access	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW
Reset	0	0	0	0	0	0	0	0

Bit 7 – BROM Interrupt Flag for Boot ROM

Bit 6 – HSRAMDSU Interrupt Flag for SLAVE HS SRAM Port 2 - DSU Access

Bit 5 – HSRAMDMAC Interrupt Flag for SLAVE HS SRAM Port 1 - DMAC Access

Bit 4 – HSRAMCPU Interrupt Flag for SLAVE HS SRAM Port 0 - CPU Access

Bit 3 – HPB2 Interrupt Flag for SLAVE AHB-APB Bridge C

Bit 2 – HPB1 Interrupt Flag for SLAVE AHB-APB Bridge B

Bit 1 – HPB0 Interrupt Flag for SLAVE AHB-APB Bridge A

Bit 0 – FLASH Interrupt Flag for SLAVE FLASH

Downloaded from Arrow.com.

15.7.6 Peripheral Interrupt Flag Status and Clear A

Name: INTFLAGA
Offset: 0x14
Reset: 0x000000
Property: Secure

This flag is cleared by writing a one to the flag.

This flag is set when a Peripheral Access Error occurs while accessing the peripheral associated with the respective INTFLAGA bit, and will generate an interrupt request if INTENCLR/SET.ERR is one.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the corresponding INTFLAGA interrupt flag.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			AC	PORT	FREQM	EIC	RTC	WDT
Access			RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM	PAC
Access	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW
Reset	0	0	0	0	0	0	0	0

Bit 13 – AC Interrupt Flag for AC

Bit 12 – PORT Interrupt Flag for PORT

Bit 11 – FREQM Interrupt Flag for FREQM

Bit 10 – EIC Interrupt Flag for EIC

Bit 9 – RTC Interrupt Flag for RTC

Bit 8 – WDT Interrupt Flag for WDT

Bit 7 – GCLK Interrupt Flag for GCLK

Bit 6 – SUPC Interrupt Flag for SUPC

Downloaded from Arrow.com.

Bit 5 – OSC32KCTRL Interrupt Flag for OSC32KCTRL

Bit 4 – OSCCTRL Interrupt Flag for OSCCTRL

Bit 3 – RSTC Interrupt Flag for RSTC

Bit 2 – MCLK Interrupt Flag for MCLK

Bit 1 – PM Interrupt Flag for PM

Bit 0 – PAC Interrupt Flag for PAC

Downloaded from Arrow.com.

15.7.7 Peripheral Interrupt Flag Status and Clear B

Name: INTFLAGB
Offset: 0x18
Reset: 0x000000
Property: Secure

This flag is cleared by writing a '1' to the flag.

This flag is set when a Peripheral Access Error occurs while accessing the peripheral associated with the respective INTFLAGB bit, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the corresponding INTFLAGB interrupt flag.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				Reserved	DMAC	NVMCTRL	DSU	IDAU
Access				RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW
Reset				0	0	0	0	0

Bit 4 – Reserved Reserved

Bit 3 – DMAC Interrupt Flag for DMAC

Bit 2 – NVMCTRL Interrupt Flag for NVMCTRL

Bit 1 – DSU Interrupt Flag for DSU

Bit 0 – IDAU Interrupt Flag for IDAU

Downloaded from Arrow.com.

15.7.8 Peripheral Interrupt Flag Status and Clear C

Name: INTFLAGC
Offset: 0x1C
Reset: 0x000000
Property: Secure

This flag is cleared by writing a one to the flag.

This flag is set when a Peripheral Access Error occurs while accessing the peripheral associated with the respective INTFLAGC bit, and will generate an interrupt request if INTENCLR/SET.ERR is one.

Writing a zero to this bit has no effect.

Writing a one to this bit will clear the corresponding INTFLAGC interrupt flag.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			TRAM	OPAMP	CCL	TRNG	PTC	DAC
Access			RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADC	TC2	TC1	TC0	SERCOM2	SERCOM1	SERCOM0	EVSYS
Access	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW
Reset	0	0	0	0	0	0	0	0

Bit 13 – TRAM Interrupt Flag for TRAM

Bit 12 – OPAMP Interrupt Flag for OPAMP

Bit 11 – CCL Interrupt Flag for CCL

Bit 10 – TRNG Interrupt Flag for TRNG

Bit 9 – PTC Interrupt Flag for PTC

Bit 8 – DAC Interrupt Flag for DAC

Bit 7 – ADC Interrupt Flag for ADC

Bits 4, 5, 6 – TC Interrupt Flag for TCn [n = 2..0]

Downloaded from Arrow.com.

Bits 1, 2, 3 – SERCOM Interrupt Flag for SERCOMn [n = 2..0]

Bit 0 – EVSYS Interrupt Flag for EVSYS

Downloaded from Arrow.com.

15.7.9 Peripheral Write Protection Status A

Name: STATUSA
Offset: 0x34
Reset: 0x000000
Property: Mix-Secure

Reading STATUSA register returns peripheral write protection status:

Value	Description
0	Peripheral is not write protected.
1	Peripheral is write protected.

Important: For **SAM L11 Non-Secure** accesses, read accesses (R*) are allowed only if the peripheral security attribution for the corresponding peripheral is set as Non-Secured in the NONSECx register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			AC	PORT	FREQM	EIC	RTC	WDT
Access			R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM	PAC
Access	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R
Reset	0	0	0	0	0	0	0	0

Bit 13 – AC Peripheral AC Write Protection Status

Bit 12 – PORT Peripheral PORT Write Protection Status

Bit 11 – FREQM Peripheral FREQM Write Protection Status

Bit 10 – EIC Peripheral EIC Write Protection Status

Bit 9 – RTC Peripheral RTC Write Protection Status

Downloaded from Arrow.com.

Bit 8 – WDT Peripheral WDT Write Protection Status

Bit 7 – GCLK Peripheral GCLK Write Protection Status

Bit 6 – SUPC Peripheral SUPC Write Protection Status

Bit 5 – OSC32KCTRL Peripheral OSC32KCTRL Write Protection Status

Bit 4 – OSCCTRL Peripheral OSCCTRL Write Protection Status

Bit 3 – RSTC Peripheral RSTC Write Protection Status

Bit 2 – MCLK Peripheral MCLK Write Protection Status

Bit 1 – PM Peripheral PM Write Protection Status

Bit 0 – PAC Peripheral PAC Write Protection Status

Downloaded from Arrow.com.

15.7.10 Peripheral Write Protection Status B

Name: STATUSB
Offset: 0x38
Reset: 0x000000
Property: Mix-Secure

Reading the STATUSB register returns the peripheral write protection status:

Value	Description
0	Peripheral is not write protected.
1	Peripheral is write protected.

Important: For **SAM L11 Non-Secure** accesses, read accesses (R*) are allowed only if the peripheral security attribution for the corresponding peripheral is set as Non-Secured in the NONSECx register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				Reserved	DMAC	NVMCTRL	DSU	IDAU
Access				R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R
Reset				0	0	0	0	0

Bit 4 – Reserved Reserved

Bit 3 – DMAC Peripheral DMAC Write Protection Status

Bit 2 – NVMCTRL Peripheral NVMCTRL Write Protection Status

Bit 1 – DSU Peripheral DSU Write Protection Status

Bit 0 – IDAU Peripheral IDAU Write Protection Status

Downloaded from Arrow.com.

15.7.11 Peripheral Write Protection Status C

Name: STATUSC
Offset: 0x3C
Reset: 0x000000
Property: Mix-Secure

Reading the STATUSC register returns the peripheral write protection status:

Value	Description
0	Peripheral is not write protected.
1	Peripheral is write protected.

Important: For **SAM L11 Non-Secure** accesses, read accesses (R*) are allowed only if the peripheral security attribution for the corresponding peripheral is set as Non-Secured in the NONSECx register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			TRAM	OPAMP	CCL	TRNG	PTC	DAC
Access			R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADC	TC2	TC1	TC0	SERCOM2	SERCOM1	SERCOM0	EVSYS
Access	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R
Reset	0	0	0	0	0	0	0	0

Bit 13 – TRAM Peripheral TRAM Write Protection Status

Bit 12 – OPAMP Peripheral OPAMP Write Protection Status

Bit 11 – CCL Peripheral CCL Write Protection Status

Bit 10 – TRNG Peripheral TRNG Write Protection Status

Bit 9 – PTC Peripheral PTC Write Protection Status

Downloaded from Arrow.com.

Bit 8 – DAC Peripheral DAC Write Protection Status

Bit 7 – ADC Peripheral ADC Write Protection Status

Bits 4, 5, 6 – TC Peripheral TCn Write Protection Status [n = 2..0]

Bits 1, 2, 3 – SERCOM Peripheral SERCOMn Write Protection Status [n = 2..0]

Bit 0 – EVSYS Peripheral EVSYS Write Protection Status

Downloaded from Arrow.com.

15.7.12 Peripheral Non-Secure Status - Bridge A

Name: NONSECA
Offset: 0x54
Reset: x initially determined from NVM User Row after reset
Property: Write-Secure

Important: This register is only available for **SAM L11** and has no effect for **SAM L10**.

Reading NONSEC register returns peripheral security attribution status:

Value	Description
0	Peripheral is secured.
1	Peripheral is non-secured.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			AC	PORT	FREQM	EIC	RTC	WDT
Access			R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R
Reset			x	x	x	x	x	x

Bit	7	6	5	4	3	2	1	0
	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM	PAC
Access	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R
Reset	x	x	x	x	x	x	x	0

Bit 13 – AC Peripheral AC Non-Secure

Bit 12 – PORT Peripheral PORT Non-Secure

Bit 11 – FREQM Peripheral FREQM Non-Secure

Bit 10 – EIC Peripheral EIC Non-Secure

Bit 9 – RTC Peripheral RTC Non-Secure

Downloaded from Arrow.com.

Bit 8 – WDT Peripheral WDT Non-Secure

Bit 7 – GCLK Peripheral GCLK Non-Secure

Bit 6 – SUPC Peripheral SUPC Non-Secure

Bit 5 – OSC32KCTRL Peripheral OSC32KCTRL Non-Secure

Bit 4 – OSCCTRL Peripheral OSCCTRL Non-Secure

Bit 3 – RSTC Peripheral RSTC Non-Secure

Bit 2 – MCLK Peripheral MCLK Non-Secure

Bit 1 – PM Peripheral PM Non-Secure

Bit 0 – PAC Peripheral PAC Non-Secure
The PAC Peripheral is always secured.

Downloaded from Arrow.com.

15.7.13 Peripheral Non-Secure Status - Bridge B

Name: NONSECB
Offset: 0x58
Reset: x initially determined from NVM User Row after reset
Property: Write-Secure

Important: This register is only available for **SAM L11** and has no effect for **SAM L10**.

Reading NONSEC register returns peripheral security attribution status:

Value	Description
0	Peripheral is secured.
1	Peripheral is non-secured.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				HMATRIXHS	DMAC	NVMCTRL	DSU	IDAU
Access				R/R/R	R/R/R	R/R/R	R/R/R	R/R/R
Reset				x	x	0	1	0

Bit 4 – HMATRIXHS Peripheral HMATRIXHS Non-Secure

Bit 3 – DMAC Peripheral DMAC Non-Secure

Bit 2 – NVMCTRL Peripheral NVMCTRL Non-Secure
The NVMCTRL Peripheral is always secured.

Bit 1 – DSU Peripheral DSU Non-Secure
The DSU Peripheral is always non-secured.

Downloaded from Arrow.com.

Bit 0 – IDAU Peripheral IDAU Non-Secure
The IDAU Peripheral is always secured.

Downloaded from Arrow.com.

15.7.14 Peripheral Non-Secure Status - Bridge C

Name: NONSECC
Offset: 0x5C
Reset: x initially determined from NVM User Row after reset
Property: Write-Secure

Important: This register is only available for **SAM L11** and has no effect for **SAM L10**.

Reading NONSEC register returns peripheral Security Attribution status:

Value	Description
0	Peripheral is secured.
1	Peripheral is non-secured.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			TRAM	OPAMP	CCL	TRNG	PTC	DAC
Access			R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R
Reset			x	x	x	x	x	x

Bit	7	6	5	4	3	2	1	0
	ADC	TC2	TC1	TC0	SERCOM2	SERCOM1	SERCOM0	EVSYS
Access	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R
Reset	x	x	x	x	x	x	x	x

Bit 13 – TRAM Peripheral TRAM Non-Secure

Bit 12 – OPAMP Peripheral OPAMP Non-Secure

Bit 11 – CCL Peripheral CCL Non-Secure

Bit 10 – TRNG Peripheral TRNG Non-Secure

Bit 9 – PTC Peripheral PTC Non-Secure

Downloaded from Arrow.com.

Bit 8 – DAC Peripheral DAC Non-Secure

Bit 7 – ADC Peripheral ADC Non-Secure

Bits 4, 5, 6 – TC Peripheral TCn Non-Secure [n = 2..0]

Bits 1, 2, 3 – SERCOM Peripheral SERCOMn Non-Secure [n = 2..0]

Bit 0 – EVSYS Peripheral EVSYS Non-Secure

Downloaded from Arrow.com.

16. DSU - Device Service Unit

16.1 Overview

The Device Service Unit (DSU) provides a means to detect debugger probes. This enables the ARM Debug Access Port (DAP) to have control over multiplexed debug pads and CPU reset. The DSU also provides system-level services to debug adapters in an ARM debug system. It implements a CoreSight Debug ROM that provides device identification as well as identification of other debug components within the system. Hence, it complies with the ARM Peripheral Identification specification. The DSU also provides system services to applications that need memory testing, as required for IEC60730 Class B compliance, for example. The DSU can be accessed simultaneously by a debugger and the CPU, as it is connected on the High-Speed Bus Matrix. It implements communication channels between the device and external tools which can be used at boot time to make use of Boot ROM services. For security reasons, some of the DSU features will be limited or unavailable when the Debug Access Level (DAL) is less than 0x2.

Related Links

30. NVMCTRL – Nonvolatile Memory Controller

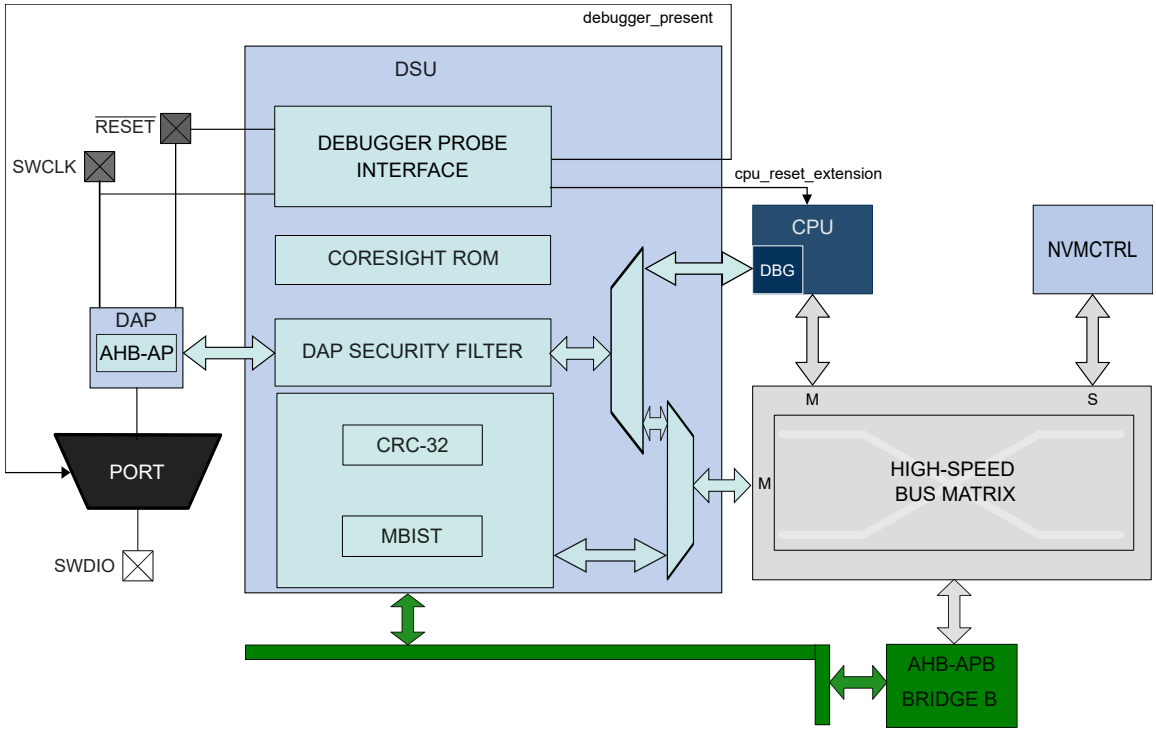
16.2 Features

- CPU reset extension
- Debugger probe detection (Cold- and Hot-Plugging)
- 32-bit cyclic redundancy check (CRC32) of any memory accessible through the bus matrix
- ARM® CoreSight™ compliant device identification
- Two debug communications channels
- Two Boot communications channels
- Debug access port security filter
- Onboard memory built-in self-test (MBIST)

Downloaded from Arrow.com.

16.3 Block Diagram

Figure 16-1. DSU Block Diagram



16.4 Signal Description

The DSU uses three signals to function.

Signal Name	Type	Description
$\overline{\text{RESET}}$	Digital Input	External reset
SWCLK	Digital Input	SW clock
SWDIO	Digital I/O	SW bidirectional data pin

16.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

16.5.1 I/O Lines

The SWCLK pin is by default assigned to the DSU module to allow debugger probe detection and to stretch the CPU reset phase. For more information, refer to 16.6.3 Debugger Probe Detection. The Hot-Plugging feature depends on the PORT configuration. If the SWCLK pin function is changed in the PORT or if the PORT_MUX is disabled, the Hot-Plugging feature is disabled until a power-reset or an external reset is performed.

16.5.2 Power Management

The DSU will continue to operate in Idle mode.

Downloaded from Arrow.com.

Related Links

22. PM – Power Manager

16.5.3 Clocks

The DSU bus clocks (CLK_DSU_APB and CLK_DSU_AHB) can be enabled and disabled by the Main Clock Controller.

Related Links

22. PM – Power Manager

19. MCLK – Main Clock

19.6.2.6 Peripheral Clock Masking

16.5.4 DMA

The DMA request lines are connected to the DMA Controller (DMAC). To use DMA requests with this peripheral, the DMAC must be configured first. Refer to 28. DMAC – Direct Memory Access Controller for details. The CFG.DCCDMALEVEL bitfield must be configured depending on the DMA channels access modes (read or write for DCC0 and DCC1).

16.5.5 Interrupts

Not applicable.

16.5.6 Events

Not applicable.

16.5.7 Register Access Protection

Registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC), except for the following:

- Debug Communication Channel 0 register (DCC0)
- Debug Communication Channel 1 register (DCC1)
- Boot Communication Channel 0 register (BCC0)
- Boot Communication Channel 1 register (BCC1)

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

Write-protection does not apply for accesses through an external debugger.

Related Links

15. PAC - Peripheral Access Controller

16.5.8 SAM L11 TrustZone Specific Register Access Protection

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Downloaded from Arrow.com.

Refer to *Peripherals Security Attribution* for more information.

16.5.9 Analog Connections

Not applicable.

16.6 Debug Operation

16.6.1 Principle of Operation

The DSU provides basic services to allow on-chip debug using the ARM Debug Access Port and the ARM processor debug resources:

- CPU reset extension
- Debugger probe detection
- Boot Communication Channels

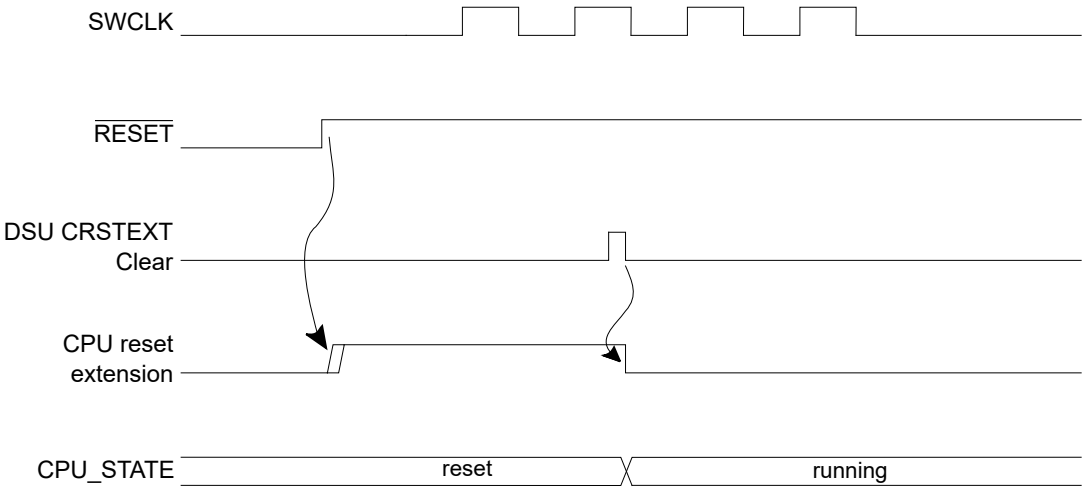
For more details on the ARM debug components, refer to the ARM Debug Interface v5 Architecture Specification.

16.6.2 CPU Reset Extension

“CPU reset extension” refers to the extension of the reset phase of the CPU core after the external reset is released. This ensures that the CPU is not executing code at startup while a debugger connects to the system. It is detected on a $\overline{\text{RESET}}$ release event when SWCLK is low. At startup, SWCLK is internally pulled up to avoid false detection of a debugger if SWCLK is left unconnected. When the CPU is held in the reset extension phase, the CPU Reset Extension bit of the Status A register (STATUSA.CRSTEXT) is set. To release the CPU, write a '1' to STATUSA.CRSTEXT. STATUSA.CRSTEXT will then be set to zero. Writing a '0' to STATUSA.CRSTEXT has no effect. Releasing the "CPU reset extension" is possible for all DAL levels. The CPU then executes the Boot ROM that offers basic failure analysis services and security checks. It is not possible to access the bus system until the Boot ROM has performed these security checks.

Note: Refer to 14. Boot ROM for more information.

Figure 16-2. Typical CPU Reset Extension Set and Clear Timing Diagram



Downloaded from Arrow.com.

16.6.3 Debugger Probe Detection

16.6.3.1 Cold Plugging

Cold-Plugging is the detection of a debugger when the system is in reset. Cold-Plugging is detected when the CPU reset extension is requested, as described above.

16.6.3.2 Hot Plugging

Hot-Plugging is the detection of a debugger probe when the system is not in reset. Hot-Plugging is not possible under reset because the detector is reset when POR or $\overline{\text{RESET}}$ are asserted. Hot-Plugging is active when a SWCLK falling edge is detected. The SWCLK pad is multiplexed with other functions and the user must ensure that its default function is assigned to the debug system. If the SWCLK function is changed, the Hot-Plugging feature is disabled until a power-reset or external reset occurs. Availability of the Hot-Plugging feature can be read from the Hot-Plugging Enable bit of the Status B register (STATUSB.HPE).

Figure 16-3. Hot-Plugging Detection Timing Diagram

The presence of a debugger probe is detected when either Hot-Plugging or Cold-Plugging is detected. Once detected, the Debugger Present bit of the Status B register (STATUSB.DBGPRES) is set. For security reasons, Hot-Plugging is not available when DAL equals to 0x0.

This detection requires that pads are correctly powered. Thus, at cold startup, this detection cannot be done until POR is released. If DAL equals 0x0, Cold-Plugging is the only way to detect a debugger probe, and so the external reset timing must be longer than the POR timing. If external reset is de-asserted before POR release, the user must retry the procedure above until it gets connected to the device.

Related Links

30. NVMCTRL – Nonvolatile Memory Controller

16.6.4 Boot Communication Channels

Boot Communication Channels allow communication between a debug adapter and the CPU executing the Boot ROM at startup. The Boot ROM implements system level commands. Refer to 14. Boot ROM for more information.

16.7 Programming

Programming the Flash or RAM memories is only possible when the debugger access level is sufficient to access the desired resource:

If DAL is equal to:

- 0x2: debugger can access secured and non-secure areas
- 0x1 (**SAM L11 only**): debugger can access only non-secure areas, refer to Table 16-4.

Downloaded from Arrow.com.

- 0x0: debugger can only access the DSU external address space making it possible to communicate with the Boot ROM after reset.

A typical programming procedure when DAL=0x2 is as follows:

1. At power up, $\overline{\text{RESET}}$ is driven low by a debugger. The on-chip regulator holds the system in a POR state until the input supply is above the POR threshold. The system continues to be held in this static state until the internally regulated supplies have reached a safe operating state.
2. The Power Manager (PM) starts, clocks are switched to the slow clock (Core Clock, System Clock, Flash Clock and any Bus Clocks that do not have clock gate control). Internal resets are maintained due to the external reset.
3. The debugger maintains a low level on SWCLK. $\overline{\text{RESET}}$ is released, resulting in a debugger Cold-Plugging procedure.
4. The debugger generates a clock signal on the SWCLK pin, the Debug Access Port (DAP) receives a clock.
5. The CPU executes the Boot ROM.
6. It is recommended to issue a Chip-Erase (supported by the Boot ROM) to ensure that the Flash is fully erased prior to programming.
7. If the operation issued above was accepted and has completed successfully then DAL equals 0x2 thus programming is available through the AHB-AP.
8. After the operation is completed, the chip can be restarted either by asserting $\overline{\text{RESET}}$, toggling power, or sending a command to the Boot ROM to jump to the NVM code. Make sure that the SWCLK pin is high when releasing $\overline{\text{RESET}}$ to prevent entering again the cold-plugging procedure with the Boot ROM stalling the CPU.

Related Links

30. NVMCTRL – Nonvolatile Memory Controller

16.8 Security Enforcement

Security enforcement aims at protecting intellectual property, which includes:

- Restricts access to internal memories from external tools depending on the debugger access level.
- Restricts access to a portion of the DSU address space from non-secure AHB masters depending on the debugger access level.

The DAL setting can be locked or reverted using Boot ROM commands depending on the Boot ROM user configuration. When DAL is equal to 0x0, read/write accesses using the AHB-AP are limited to the DSU external address range and DSU commands are restricted. When issuing a Boot ROM Chip-Erase, sensitive information is erased from volatile memory and Flash. Refer to 14. Boot ROM more information about the Boot ROM features.

The DSU implements a security filter that monitors the AHB transactions generated by the ARM AHB-AP inside the DAP. If DAL=0x0, then AHB-AP read/write accesses outside the DSU external address range are discarded, causing an error response that sets the ARM AHB-AP sticky error bits (refer to the "ARM Debug Interface v5 Architecture Specification", which is available for download at http://www.arm.com).

For security reasons, DSU features have limitations when used from a debug adapter. To differentiate external accesses from internal ones, the first 0x100 bytes of the DSU register map have been replicated at offset 0x100:

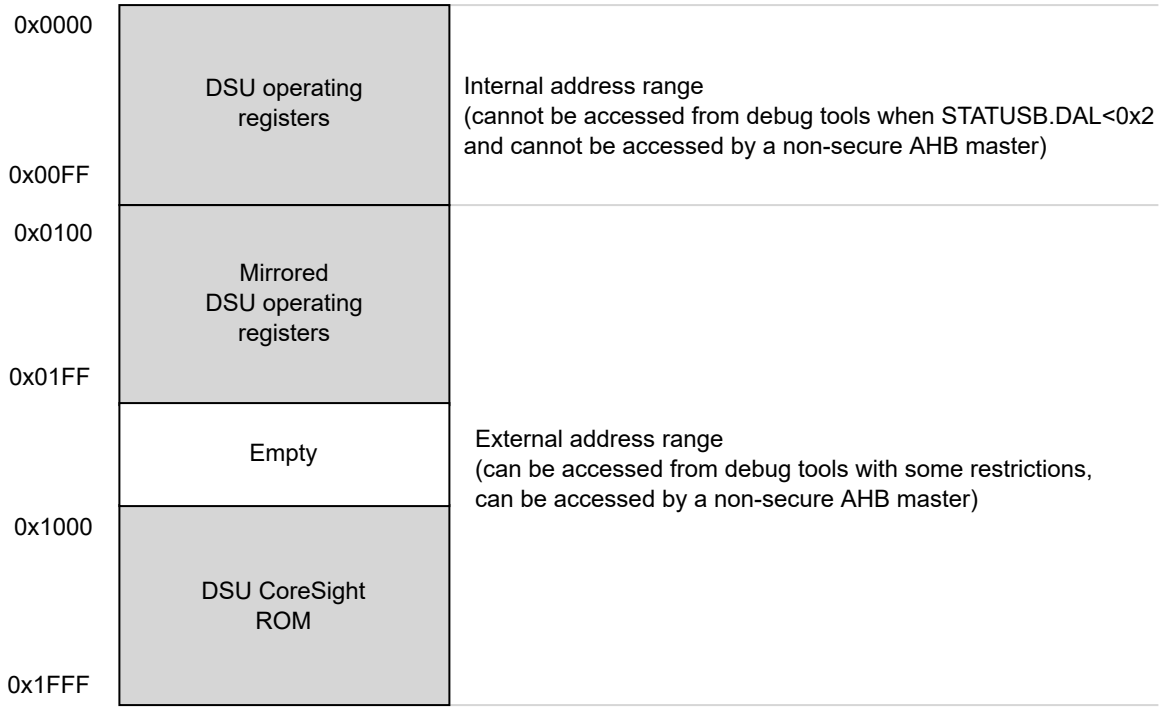
- The first 0x100 bytes form the internal address range
- The next 0x1F00 bytes form the external address range

Downloaded from Arrow.com.

When the device is protected, the DAP can only issue MEM-AP accesses in the DSU address range limited to the 0x100- 0x2000 offset range.

The DSU operating registers are located in the 0x00-0xFF area and mirrored to 0x100-0x1FF to differentiate accesses coming from a debugger and the CPU. If the device is protected and an access is issued in the region 0x100-0x1FF, it is subject to security restrictions. For more information, refer to the Table 16-2.

Figure 16-4. APB Memory Mapping



The DSU filters-out DAP transactions depending on the DAL setting and routes DAP transactions:

- In the PPB or IOBUS space to the CPU debug port
- Outside the PPB space and outside the IOBUS space to the DSU master port

Table 16-1. DAP access rights depending on DAL:

DAP access to	SAM L11			SAM L10	
	DAL=0	DAL=1	DAL=2	DAL=0	DAL=2
PPB or IOBUS	No	Yes (see **Note 1**)	Yes	No	Yes
DSU internal address space	No	No (see **Note 2**)	Yes	No	Yes
DSU external address space	Yes	Yes	Yes	Yes	Yes
Other secure areas	No	No	Yes	No	Yes
Other non-secure areas	No	Yes	Yes	No	Yes

Note:

1. Refer to ARMv8-M debug documentation for detailed information on PPB and IOBUS access restrictions.

Downloaded from Arrow.com.

2. When DAL=1 DAP transfers are always non-secure. The DSU internal address space can only be accessed by secure masters.

Some features not activated by APB transactions are not available when the device is protected:

Table 16-2. Feature Availability Under Protection

Features	Availability when DAL equals to		
	0x0	0x1 (SAM L11 only)	0x2
CPU Reset Extension	Yes	Yes	Yes
Clear CPU Reset extension	Yes	Yes	Yes
Debugger Cold-Plugging	Yes	Yes	Yes
Debugger Hot-Plugging	No	Yes	Yes

16.9 Device Identification

Device identification relies on the ARM CoreSight component identification scheme, which allows the chip to be identified as a SAM device implementing a DSU. The DSU contains identification registers to differentiate the device.

16.9.1 CoreSight Identification

A system-level ARM® CoreSight™ ROM table is present in the device to identify the vendor and the chip identification method. Its address is provided in the MEM-AP BASE register inside the ARM Debug Access Port. The CoreSight ROM implements a 64-bit conceptual ID composed as follows from the PID0 to PID7 CoreSight ROM Table registers:

Figure 16-5. Conceptual 64-bit Peripheral ID

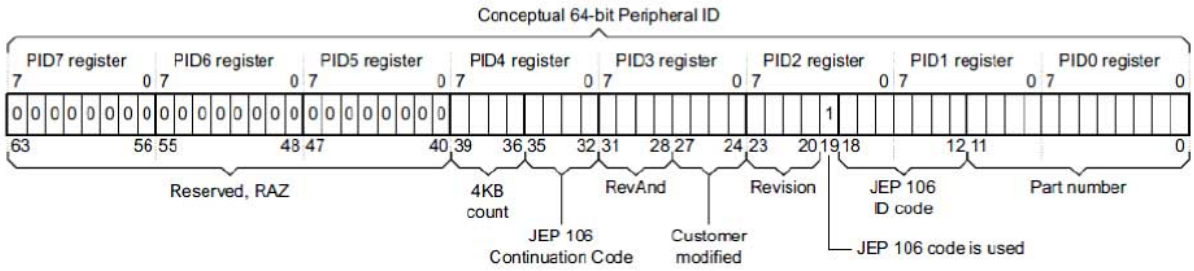


Table 16-3. Conceptual 64-Bit Peripheral ID Bit Descriptions

Field	Size	Description	Location
JEP-106 CC code	4	Continuation code: 0x0	PID4
JEP-106 ID code	7	Device ID: 0x1F	PID1+PID2
4KB count	4	Indicates that the CoreSight component is a ROM: 0x0	PID4
RevAnd	4	Not used; read as 0	PID3
CUSMOD	4	Not used; read as 0	PID3

Downloaded from Arrow.com.

Field	Size	Description	Location
PARTNUM	12	Contains 0xCD0 to indicate that DSU is present	PID0+PID1
REVISION	4	DSU revision (starts at 0x0 and increments by 1 at both major and minor revisions). Identifies DSU identification method variants. If 0x0, this indicates that device identification can be completed by reading the Device Identification register (DID)	PID2

For more information, refer to the ARM Debug Interface Version 5 Architecture Specification.

16.9.2 Chip Identification Method

The DSU DID register identifies the device by implementing the following information:

- Processor identification
- Product family identification
- Product series identification
- Device select

16.10 Functional Description

16.10.1 Principle of Operation

The DSU provides memory services, such as CRC32 or MBIST that require almost the same interface. Hence, the Address, Length and Data registers (ADDR, LENGTH, DATA) are shared. These shared registers must be configured first; then a command can be issued by writing the Control register. When a command is ongoing, other commands are discarded until the current operation is completed. Hence, the user must wait for the STATUSA.DONE bit to be set prior to issuing another one.

16.10.2 Basic Operation

16.10.2.1 Initialization

The module is enabled by enabling its clocks. For more details, refer to 16.5.3 Clocks. The DSU registers can be PAC write-protected.

Related Links

15. PAC - Peripheral Access Controller

16.10.2.2 Operation From a Debug Adapter

Debug adapters should access the DSU registers in the external address range [0x100 – 0x1FFF].

If STATUSB.DAL is equal to 0x0, accessing the first 0x100 bytes causes the DSU security filter to return an error to the DAP.

(**SAM L11 only**): If STATUSB.DAL is equal to 0x1, debug accesses will go through the DSU security filter but will be forced as non-secure, therefore the DSU internal address space will not be accessible and any access in this case is discarded (writes are ignored, reads return 0) and raise STATUSA.PERR.

Downloaded from Arrow.com.

Table 16-4. DAP transaction authorizations and error response types

DAL	Debugger Access Type	DAP transaction allowed?			
		DSU internal address space	DSU external address space	Other than PPB	PPB
0	Secure	No (Bus Error)	Yes	No (Bus Error)	No (Bus Error)
0	Non-Secure	No (Bus Error)	Yes	No (Bus Error)	No (Bus Error)
1 (**SAM L11 only**)	Secure	No (PERR)	Yes	Yes (NS,PERR)	Yes (ARMv8M)
1 (**SAM L11 only**)	Non-Secure	No (PERR)	Yes	Yes (NS,PERR)	Yes (ARMv8M)
2	Secure	Yes	Yes	Yes	Yes
2	Non-Secure	No (PERR)	Yes	Yes	Yes

Bus Error: A Bus Error is sent back to the DAP setting its sticky bit error.

PERR: No bus error, STATUSA.PERR rises, writes are discarded, reads always return 0

NS, PERR: Access forced to non-secure, secure violations are reported in STATUSA.PERR.

Note: Refer to the *ARM Debug Interface Architecture Specification* for details.

Related Links

30. NVMCTRL – Nonvolatile Memory Controller

16.10.2.3 Operation From the CPU

Only secure masters can access the DSU internal address space. Attempting to access the internal address space from a non-secure AHB master will report a PAC error, such accesses are discarded. The external address space can be accessed by either secure or non-secure AHB masters. The user should access DSU registers in the internal address range (0x0 – 0xFF) to avoid external security restrictions. Refer to 16.8 Security Enforcement.

16.10.3 32-bit Cyclic Redundancy Check CRC32

The DSU unit provides support for calculating a cyclic redundancy check (CRC32) value for a memory area (including Flash and AHB RAM).

When the CRC32 command is issued from:

- The internal range from a secure AHB master, the CRC32 can be operated at any memory location
- The external range, the CRC32 operation is not available

The algorithm employed is the industry standard CRC32 algorithm using the generator polynomial 0xEDB88320 (reversed representation).

16.10.3.1 Starting CRC32 Calculation

CRC32 calculation for a memory range is started after writing the start address into the Address register (ADDR) and the size of the memory range into the Length register (LENGTH). Both must be word-aligned.

The initial value used for the CRC32 calculation must be written to the Data register (DATA). This value will usually be 0xFFFFFFFF, but can be, for example, the result of a previous CRC32 calculation if generating a common CRC32 of separate memory blocks.

Downloaded from Arrow.com.

Once completed, the calculated CRC32 value can be read out of the Data register. The read value must be complemented to match standard CRC32 implementations or kept non-inverted if used as starting point for subsequent CRC32 calculations.

The actual test is started by writing a '1' in the 32-bit Cyclic Redundancy Check bit of the Control register (CTRL.CRC). A running CRC32 operation can be canceled by resetting the module (writing '1' to CTRL.SWRST).

Related Links

30. NVMCTRL – Nonvolatile Memory Controller

16.10.3.2 Interpreting the Results

The user should monitor the Status A register. When the operation is completed, STATUSA.DONE is set. Then the Bus Error bit of the Status A register (STATUSA.BERR) must be read to ensure that no bus error occurred.

16.10.4 Debug Communication Channels

The Debug Communication Channels (DCCO and DCC1) consist of a pair of registers with associated handshake logic, accessible by both CPU and debugger with no security restriction. The registers can be used to exchange data between the CPU and the debugger, during run time as well as in debug mode. This enables the user to build a custom debug protocol using only these registers.

The DCC0 and DCC1 registers are always accessible from the external address space. When the device starts with the cold-plugging procedure, a specific Boot ROM command is needed to exit the Boot ROM main routine.

Important: This command is allowed only when DAL=0x2, otherwise the device must be reset to leave the cold plugging state to let the CPU exit the Boot ROM routine and execute the user code.

Two Debug Communication Channel status bits in the Status B registers (STATUS.DCCDx) indicate whether a new value has been written in DCC0 or DCC1. These bits, DCC0D and DCC1D, are located in the STATUSB registers. They are automatically set on write and cleared on read.

Note: The DCC0 and DCC1 registers are shared with the on-board memory testing logic (MBIST). Accordingly, DCC0 and DCC1 must not be used while performing MBIST operations.

Note: The DCC0 and DCC1 registers are shared with the BCC0 and BCC1 registers therefore mixing DCC and BCC communication is not recommended.

Related Links

30. NVMCTRL – Nonvolatile Memory Controller

16.10.5 Boot Communication Channels

The Boot Communication Channels (BCC0 and BCC1) consist of a pair of registers with associated handshake logic, accessible by both CPU and debugger with no security restriction. The registers are intended to communicate with the CPU while executing the Boot ROM which implements security and failure analysis commands and therefore must not be used for another purpose.

Note: The BCC0 and BCC registers values are not reset except in case of POR or BOD resets.

Downloaded from Arrow.com.

Two Boot Communication Channel status bits in the Status B registers (STATUS.BCCDx) indicate whether a new value has been written in BCC0 or BCC1. These bits, BCC0D and BCC1D, are located in the STATUSB registers. They are automatically set on write and cleared on read.

Note: The DCC0 and DCC1 registers are shared with the BCC0 and BCC1 registers therefore using DCC is not recommended while the Boot ROM is being executed.

16.10.6 Testing of On-Board Memories MBIST

The DSU implements a feature for automatic testing of memory also known as MBIST (memory built-in self test). This is primarily intended for production test of on-board memories. MBIST cannot be operated from the external address range when DAL<0x2. If an MBIST command is issued when the device is protected, it is filtered-out, a protection error is reported in the Protection Error bit in the Status A register (STATUSA.PERR) and STATUSA.DONE don't rise.

1. Algorithm
   The algorithm used for testing is a type of March algorithm called "March LR". This algorithm is able to detect a wide range of memory defects, while still keeping a linear run time. The algorithm is:

   1.1. Write entire memory to '0', in any order.
   1.2. Bit for bit read '0', write '1', in descending order.
   1.3. Bit for bit read '1', write '0', read '0', write '1', in ascending order.
   1.4. Bit for bit read '1', write '0', in ascending order.
   1.5. Bit for bit read '0', write '1', read '1', write '0', in ascending order.
   1.6. Read '0' from entire memory, in ascending order.

   The specific implementation used has a run time which depends on the CPU clock frequency and the number of bytes tested in the RAM. The detected faults are:

   - Address decoder faults
   - Stuck-at faults
   - Transition faults
   - Coupling faults
   - Linked Coupling faults

2. Starting MBIST
   To test a memory, you need to write the start address of the memory to the ADDR.ADDR bit field, and the size of the memory into the Length register.

   For best test coverage, an entire physical memory block should be tested at once. It is possible to test only a subset of a memory, but the test coverage will then be somewhat lower.

   The actual test is started by writing a '1' to CTRL.MBIST. A running MBIST operation can be canceled by writing a '1' to CTRL.SWRST.

3. Interpreting the Results
   The tester should monitor the STATUSA register. When the operation is completed, STATUSA.DONE is set. There are two different modes:

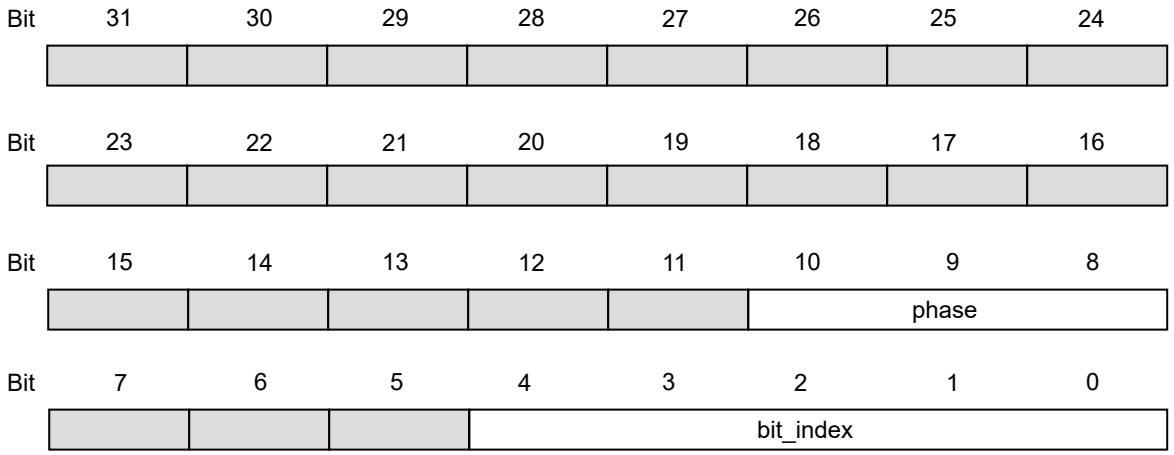
   - ADDR.AMOD=0: exit-on-error (default)
     In this mode, the algorithm terminates either when a fault is detected or on successful completion. In both cases, STATUSA.DONE is set. If an error was detected, STATUSA.FAIL will be set. User then can read the DATA and ADDR registers to locate the fault.
   - ADDR.AMOD=1: pause-on-error

Downloaded from Arrow.com.

In this mode, the MBIST algorithm is paused when an error is detected. In such a situation, only STATUSA.FAIL is asserted. The state machine waits for user to clear STATUSA.FAIL by writing a '1' in STATUSA.FAIL to resume. Prior to resuming, user can read the DATA and ADDR registers to locate the fault.

4. Locating Faults
If the test stops with STATUSA.FAIL set, one or more bits failed the test. The test stops at the first detected error. The position of the failing bit can be found by reading the following registers:
   – ADDR: Address of the word containing the failing bit
   – DATA: contains data to identify which bit failed, and during which phase of the test it failed. The DATA register will in this case contains the following bit groups:

Figure 16-6. DATA bits Description When MBIST Operation Returns an Error



- bit_index: contains the bit number of the failing bit
- phase: indicates which phase of the test failed and the cause of the error, as listed in the following table.

Table 16-5. MBIST Operation Phases

Phase	Test actions
0	Write all bits to zero. This phase cannot fail.
1	Read '0', write '1', increment address
2	Read '1', write '0'
3	Read '0', write '1', decrement address
4	Read '1', write '0', decrement address
5	Read '0', write '1'
6	Read '1', write '0', decrement address
7	Read all zeros. bit_index is not used

Downloaded from Arrow.com.

Table 16-6. AMOD Bit Descriptions for MBIST

AMOD[1:0]	Description
0x0	Exit on Error
0x1	Pause on Error
0x2, 0x3	Reserved

Related Links

30. NVMCTRL – Nonvolatile Memory Controller

16.10.7 System Services Availability when Accessed Externally

External access: Access performed in the DSU address offset 0x100-0x1FFF range.

Internal access: Access performed in the DSU address offset 0x0-0xFF range.

Table 16-7. Available Features when Operated From The External Address Range and Device is Protected

Features	Availability From The External Address Range when DAL<2
CRC32	No
CoreSight Compliant Device identification	Yes
Debug communication channels	Yes
Boot communication channels	Yes
Testing of onboard memories (MBIST)	No

Downloaded from Arrow.com.

16.11 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRL	7:0					MBIST	CRC		SWRST
0x01	STATUSA	7:0			BREXT	PERR	FAIL	BERR	CRSTEXT	DONE
0x02	STATUSB	7:0	BCCDx	BCCDx	DCCDx	DCCDx	HPE	DBGPRES	DAL[1:0]	
0x03	Reserved									
0x04	ADDR	7:0	ADDR[5:0]						AMOD[1:0]	
		15:8	ADDR[13:6]							
		23:16	ADDR[21:14]							
		31:24	ADDR[29:22]							
0x08	LENGTH	7:0	LENGTH[5:0]							
		15:8	LENGTH[13:6]							
		23:16	LENGTH[21:14]							
		31:24	LENGTH[29:22]							
0x0C	DATA	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x10	DCC0	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x14	DCC1	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x18	DID	7:0	DEVSEL[7:0]							
		15:8	DIE[3:0]				REVISION[3:0]			
		23:16	FAMILY[0:0]		SERIES[5:0]					
		31:24	PROCESSOR[3:0]				FAMILY[4:1]			
0x1C	CFG	7:0					DCCDMALEVEL[1:0]		LQOS[1:0]	
		15:8								
		23:16								
		31:24								
0x20	BCC0	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x24	BCC1	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x28 ... 0x0FFF	Reserved									

Downloaded from Arrow.com.

Offset	Name	Bit Pos.								
0x1000	ENTRY0	7:0							FMT	EPRES
		15:8	ADDOFF[3:0]							
		23:16	ADDOFF[11:4]							
		31:24	ADDOFF[19:12]							
0x1004	ENTRY1	7:0							FMT	EPRES
		15:8	ADDOFF[3:0]							
		23:16	ADDOFF[11:4]							
		31:24	ADDOFF[19:12]							
0x1008	END	7:0	END[7:0]							
		15:8	END[15:8]							
		23:16	END[23:16]							
		31:24	END[31:24]							
0x100C ... 0x1FCB	Reserved									
0x1FCC	MEMTYPE	7:0								SMEMP
		15:8								
		23:16								
		31:24								
0x1FD0	PID4	7:0	FKBC[3:0]				JEPCC[3:0]			
		15:8								
		23:16								
		31:24								
0x1FD4 ... 0x1FDF	Reserved									
0x1FE0	PID0	7:0	PARTNBL[7:0]							
		15:8								
		23:16								
		31:24								
0x1FE4	PID1	7:0	JEPIDCL[3:0]				PARTNBH[3:0]			
		15:8								
		23:16								
		31:24								
0x1FE8	PID2	7:0	REVISION[3:0]				JEPU	JEPIDCH[2:0]		
		15:8								
		23:16								
		31:24								
0x1FEC	PID3	7:0	REVAND[3:0]				CUSMOD[3:0]			
		15:8								
		23:16								
		31:24								
0x1FF0	CID0	7:0	PREAMBLEB0[7:0]							
		15:8								
		23:16								
		31:24								

Downloaded from Arrow.com.

Offset	Name	Bit Pos.								
0x1FF4	CID1	7:0	CCLASS[3:0]				PREAMBLE[3:0]			
		15:8								
		23:16								
		31:24								
0x1FF8	CID2	7:0	PREAMBLEB2[7:0]							
		15:8								
		23:16								
		31:24								
0x1FFC	CID3	7:0	PREAMBLEB3[7:0]							
		15:8								
		23:16								
		31:24								

16.12 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to 16.5.7 Register Access Protection.

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

Downloaded from Arrow.com.

16.12.1 Control

Name: CTRL
Offset: 0x0000
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
					MBIST	CRC		SWRST
Access					W	W		W
Reset					0	0		0

Bit 3 – MBIST Memory Built-In Self-Test
Writing a '0' to this bit has no effect.

Writing a '1' to this bit starts the memory BIST algorithm.

Bit 2 – CRC 32-bit Cyclic Redundancy Check
Writing a '0' to this bit has no effect.

Writing a '1' to this bit starts the cyclic redundancy check algorithm.

Bit 0 – SWRST Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets the module.

Downloaded from Arrow.com.

16.12.2 Status A

Name: STATUSA
Offset: 0x0001
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
			BREXT	PERR	FAIL	BERR	CRSTEXT	DONE
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – BREXT Boot ROM Phase Extension
Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Boot ROM Phase Extension bit.

This bit is set when a debug adapter Cold-Plugging is detected, which extends the Boot ROM phase.

Bit 4 – PERR Protection Error
Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Protection Error bit.

This bit is set upon access to:

- A reserved address
- CTRL, ADDR, LENGTH, DATA, CFG from the external address space when DAL<2
- The internal address space with a Non-Secure access (security violation) (SAM L11 only)

Bit 3 – FAIL Failure
Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Failure bit.

This bit is set when a DSU operation failure is detected.

Bit 2 – BERR Bus Error
Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Bus Error bit.

This bit is set when a bus error is detected.

Bit 1 – CRSTEXT CPU Reset Phase Extension
Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the CPU Reset Phase Extension bit.

This bit is set when a debug adapter Cold-Plugging is detected, which extends the CPU reset phase.

Bit 0 – DONE Done
Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Done bit.

Downloaded from Arrow.com.

This bit is set when a DSU operation is completed.

Downloaded from Arrow.com.

16.12.3 Status B

Name: STATUSB
Offset: 0x0002
Reset: 0xX
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	BCCDx	BCCDx	DCCDx	DCCDx	HPE	DBGPRES	DAL[1:0]	
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	1	0	0	x

Bits 7,6 – BCCDx BOOT Communication Channel x Dirty [x=1..0]
Writing a '0' to this bit has no effect.

Writing a '1' to this bit has no effect.

This bit is set when BCCx is written.

This bit is cleared when BCCx is read.

Bits 5,4 – DCCDx Debug Communication Channel x Dirty [x=1..0]
Writing a '0' to this bit has no effect.

Writing a '1' to this bit has no effect.

This bit is set when DCCx is written.

This bit is cleared when DCCx is read.

Bit 3 – HPE Hot-Plugging Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit has no effect.

This bit is set when Hot-Plugging is enabled.

This bit is cleared when Hot-Plugging is disabled. This is the case when the SWCLK function is changed. Only a power-reset or a external reset can set it again.

Bit 2 – DBGPRES Debugger Present
Writing a '0' to this bit has no effect.

Writing a '1' to this bit has no effect.

This bit is set when a debugger probe is detected.

This bit is never cleared.

Bits 1:0 – DAL[1:0] Debugger Access Level
Indicates the debugger access level:

- 0x0: Debugger can only access the DSU external address space.
- 0x1: Debugger can access only Non-Secure regions (SAM L11 only).
- 0x2: Debugger can access secure and Non-Secure regions.

Downloaded from Arrow.com.

Writing in this bitfield has no effect.

Downloaded from Arrow.com.

16.12.4 Address

Name: ADDR
Offset: 0x0004
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	ADDR[29:22]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDR[21:14]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ADDR[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[5:0]						AMOD[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:2 – ADDR[29:0] Address
Initial word start address needed for memory operations.

Bits 1:0 – AMOD[1:0] Address Mode
The functionality of these bits is dependent on the operation mode.

Bit description when testing on-16.10.6 Testing of On-Board Memories MBISTboard memories (MBIST): refer to

Downloaded from Arrow.com.

16.12.5 Length

Name: LENGTH
Offset: 0x0008
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	LENGTH[29:22]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	LENGTH[21:14]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	LENGTH[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LENGTH[5:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bits 31:2 – LENGTH[29:0] Length
Length in words needed for memory operations.

Downloaded from Arrow.com.

16.12.6 Data

Name: DATA
Offset: 0x000C
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0] Data
Memory operation initial value or result value.

Downloaded from Arrow.com.

16.12.7 Debug Communication Channel 0

Name: DCC0
Offset: 0x0010
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0] Data
Data register.

Downloaded from Arrow.com.

16.12.8 Debug Communication Channel 1

Name: DCC1
Offset: 0x0014
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0] Data
Data register.

Downloaded from Arrow.com.

16.12.9 Device Identification

Name: DID
Offset: 0x0018
Reset: see related links
Property: PAC Write-Protection

The information in this register is related to the 2. Ordering Information.

Bit	31	30	29	28	27	26	25	24
	PROCESSOR[3:0]				FAMILY[4:1]			
Access	R	R	R	R	R	R	R	R
Reset	p	p	p	p	f	f	f	f

Bit	23	22	21	20	19	18	17	16
	FAMILY[0:0]		SERIES[5:0]					
Access	R		R	R	R	R	R	R
Reset	f		s	s	s	s	s	s

Bit	15	14	13	12	11	10	9	8
	DIE[3:0]				REVISION[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	d	d	d	d	r	r	r	r

Bit	7	6	5	4	3	2	1	0
	DEVSEL[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bits 31:28 – PROCESSOR[3:0] Processor
The value of this field defines the processor used on the device.

Bits 27:23 – FAMILY[4:0] Product Family
The value of this field corresponds to the Product Family part of the ordering code.

Bits 21:16 – SERIES[5:0] Product Series
The value of this field corresponds to the Product Series part of the ordering code.

Bits 15:12 – DIE[3:0] Die Number
Identifies the die family.

Bits 11:8 – REVISION[3:0] Revision Number
Identifies the die revision number. 0x0=rev.A, 0x1=rev.B etc.

Note: The device variant (last letter of the ordering number) is independent of the die revision (DSU.DID.REVISION): The device variant denotes functional differences, whereas the die revision marks evolution of the die.

Downloaded from Arrow.com.

Bits 7:0 – DEVSEL[7:0] Device Selection
This bit field identifies a device within a product family and product series. Refer to 2. Ordering Information for device configurations and corresponding values for Flash memory density, pin count, and device variant.

Downloaded from Arrow.com.

16.12.10 Configuration

Name: CFG
Offset: 0x001C
Reset: 0x0000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					DCCDMALEVEL[1:0]		LQOS[1:0]	
Access					RW	RW	RW	RW
Reset					0	0	0	2

Bits 3:2 – DCCDMALEVEL[1:0] DMA TriggerLevel
0x0X: DCC1 trigger is the image of STATUSB.DCC1D, this signals to the DMA that a data is available for read, this is the correct configuration for a channel that reads DCC1.

0x1X: DCC1 trigger is the image of STATUSB.DCC1D inverted, this signals to the DMA that DCC1 is ready for write, this is the correct configuration for a channel that writes DCC1

0xX0: DCC0 trigger is the image of STATUSB.DCC0D, this signals to the DMA that a data is available for read, this is the correct configuration for a channel that reads DCC0.

0xX1: DCC0 trigger is the image of STATUSB.DCC0D inverted, this signals to the DMA that DCC0 is ready for write, this is the correct configuration for a channel that writes DCC0

Bits 1:0 – LQOS[1:0] Latency Quality Of Service
Defines the latency quality of service required when accessing the RAM:

0: Background Transfers

1: Bandwidth Sensitive

2: Latency sensitive

3: Latency critical

Downloaded from Arrow.com.

16.12.11 Boot Communication Channel 0

Name: BCC0
Offset: 0x0020
Reset: N/A
Property: -

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0] Data
Data register.

Downloaded from Arrow.com.

16.12.12 Boot Communication Channel 1

Name: BCC1
Offset: 0x0024
Reset: N/A
Property: -

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0] Data
Data register.

Downloaded from Arrow.com.

16.12.13 CoreSight ROM Table Entry 0

Name: ENTRY0
Offset: 0x1000
Reset: 0xXXXXX00X
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	ADDOFF[19:12]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bit	23	22	21	20	19	18	17	16
	ADDOFF[11:4]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bit	15	14	13	12	11	10	9	8
	ADDOFF[3:0]							
Access	R	R	R	R				
Reset	x	x	x	x				

Bit	7	6	5	4	3	2	1	0
							FMT	EPRES
Access							R	R
Reset							1	x

Bits 31:12 – ADDOFF[19:0] Address Offset
The base address of the component, relative to the base address of this ROM table.

Bit 1 – FMT Format
Always reads as '1', indicating a 32-bit ROM table.

Bit 0 – EPRES Entry Present
This bit indicates whether an entry is present at this location in the ROM table.

This bit is set at power-up if the device is not protected indicating that the entry is not present.

This bit is cleared at power-up if the device is not protected indicating that the entry is present.

Downloaded from Arrow.com.

16.12.14 CoreSight ROM Table Entry 1

Name: ENTRY1
Offset: 0x1004
Reset: 0xXXXXX00X
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	ADDOFF[19:12]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bit	23	22	21	20	19	18	17	16
	ADDOFF[11:4]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bit	15	14	13	12	11	10	9	8
	ADDOFF[3:0]							
Access	R	R	R	R				
Reset	x	x	x	x				

Bit	7	6	5	4	3	2	1	0
							FMT	EPRES
Access							R	R
Reset							1	x

Bits 31:12 – ADDOFF[19:0] Address Offset
The base address of the component, relative to the base address of this ROM table.

Bit 1 – FMT Format
Always read as '1', indicating a 32-bit ROM table.

Bit 0 – EPRES Entry Present
This bit indicates whether an entry is present at this location in the ROM table.

This bit is set at power-up if the device is not protected indicating that the entry is not present.

This bit is cleared at power-up if the device is not protected indicating that the entry is present.

Downloaded from Arrow.com.

16.12.15 CoreSight ROM Table End

Name: END
Offset: 0x1008
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	END[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	END[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	END[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	END[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – END[31:0] End Marker
Indicates the end of the CoreSight ROM table entries.

Downloaded from Arrow.com.

16.12.16 CoreSight ROM Table Memory Type

Name: MEMTYPE
Offset: 0x1FCC
Reset: 0x0000000x
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								SMEMP
Access								R
Reset								x

Bit 0 – SMEMP System Memory Present
This bit indicates whether system memory is present on the bus that connects to the ROM table.

This bit is set at power-up if the device is not protected, indicating that the system memory is accessible from a debug adapter.

This bit is cleared at power-up if the device is protected, indicating that the system memory is not accessible from a debug adapter.

Downloaded from Arrow.com.

16.12.17 Peripheral Identification 4

Name: PID4
Offset: 0x1FD0
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	FKBC[3:0]				JEPCC[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:4 – FKBC[3:0] 4KB Count
These bits will always return zero when read, indicating that this debug component occupies one 4KB block.

Bits 3:0 – JEPCC[3:0] JEP-106 Continuation Code
These bits will always return zero when read.

Downloaded from Arrow.com.

16.12.18 Peripheral Identification 0

Name: PID0
Offset: 0x1FE0
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	PARTNBL[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – PARTNBL[7:0] Part Number Low
These bits will always return 0xD0 when read, indicating that this device implements a DSU module instance.

Downloaded from Arrow.com.

16.12.19 Peripheral Identification 1

Name: PID1
Offset: 0x1FE4
Reset: 0x000000FC
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	JEPIDCL[3:0]				PARTNBH[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	1	1	1	1	1	1	0	0

Bits 7:4 – JEPIDCL[3:0] Low part of the JEP-106 Identity Code
These bits will always return 0xF when read (JEP-106 identity code is 0x1F).

Bits 3:0 – PARTNBH[3:0] Part Number High
These bits will always return 0xC when read, indicating that this device implements a DSU module instance.

Downloaded from Arrow.com.

16.12.20 Peripheral Identification 2

Name: PID2
Offset: 0x1FE8
Reset: 0x00000019
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	REVISION[3:0]				JEPU	JEPIDCH[2:0]		
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	1	1	0	0	1

Bits 7:4 – REVISION[3:0] Revision Number
Revision of the peripheral. Starts at 0x0 and increments by one at both major and minor revisions.

Bit 3 – JEPU JEP-106 Identity Code is used
This bit will always return one when read, indicating that JEP-106 code is used.

Bits 2:0 – JEPIDCH[2:0] JEP-106 Identity Code High
These bits will always return 0x1 when read, indicating an Atmel device (Atmel JEP-106 identity code is 0x1F).

Downloaded from Arrow.com.

16.12.21 Peripheral Identification 3

Name: PID3
Offset: 0x1FEC
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	REVAND[3:0]				CUSMOD[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:4 – REVAND[3:0] Revision Number
These bits will always return 0x0 when read.

Bits 3:0 – CUSMOD[3:0] ARM CUSMOD
These bits will always return 0x0 when read.

Downloaded from Arrow.com.

16.12.22 Component Identification 0

Name: CID0
Offset: 0x1FF0
Reset: 0x0000000D
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	PREAMBLEB0[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	1	1	0	1

Bits 7:0 – PREAMBLEB0[7:0] Preamble Byte 0
These bits will always return 0x0000000D when read.

Downloaded from Arrow.com.

16.12.23 Component Identification 1

Name: CID1
Offset: 0x1FF4
Reset: 0x00000010
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CCLASS[3:0]				PREAMBLE[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	1	0	0	0	0

Bits 7:4 – CCLASS[3:0] Component Class
These bits will always return 0x1 when read indicating that this ARM CoreSight component is ROM table (refer to the ARM Debug Interface v5 Architecture Specification at http://www.arm.com).

Bits 3:0 – PREAMBLE[3:0] Preamble
These bits will always return 0x00 when read.

Downloaded from Arrow.com.

16.12.24 Component Identification 2

Name: CID2
Offset: 0x1FF8
Reset: 0x00000005
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	PREAMBLEB2[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	1	0	1

Bits 7:0 – PREAMBLEB2[7:0] Preamble Byte 2
These bits will always return 0x00000005 when read.

Downloaded from Arrow.com.

16.12.25 Component Identification 3

Name: CID3
Offset: 0x1FFC
Reset: 0x000000B1
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	PREAMBLEB3[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	1	0	1	1	0	0	0	1

Bits 7:0 – PREAMBLEB3[7:0] Preamble Byte 3
These bits will always return 0x000000B1 when read.

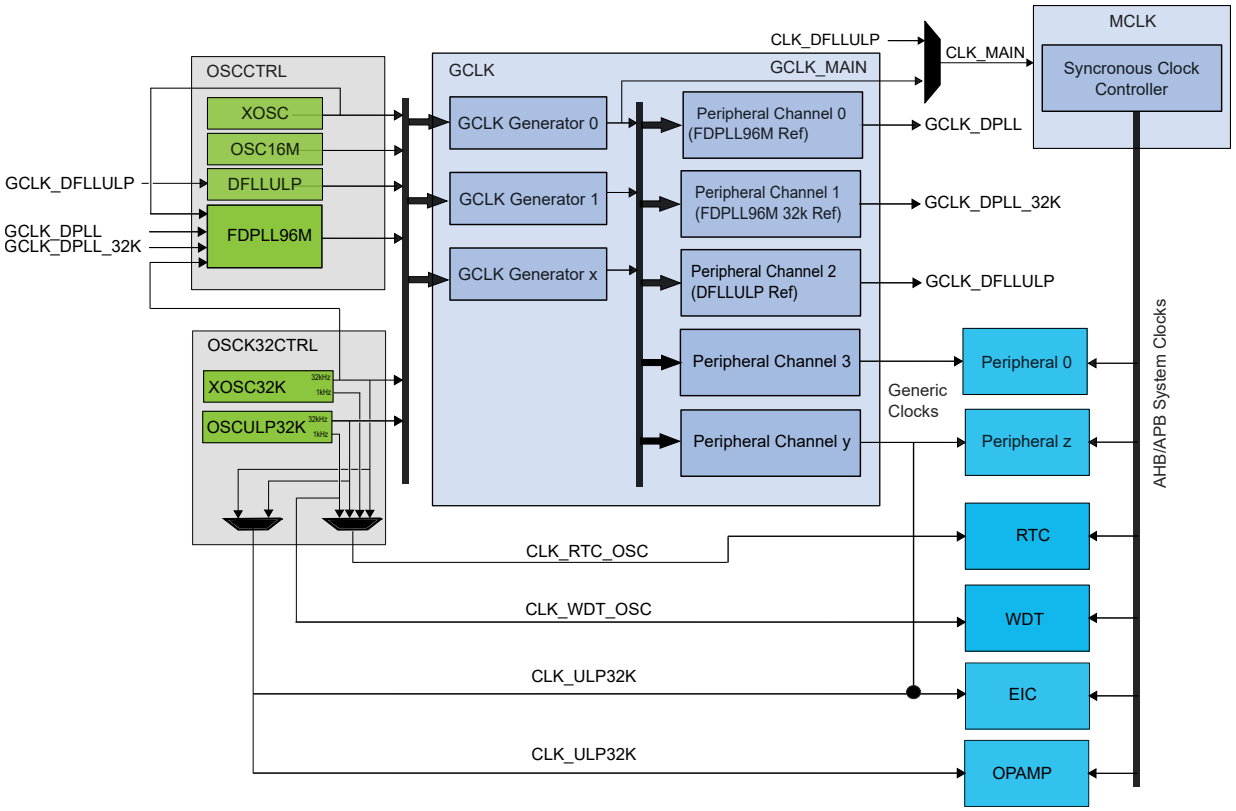
Downloaded from Arrow.com.

17. Clock System

This chapter summarizes the clock distribution and terminology in the SAM L10/L11 device. This document will not explain every detail of its configuration, hence for in-depth details, refer to the respective peripherals descriptions and the *Generic Clock* documentation.

17.1 Clock Distribution

Figure 17-1. Clock Distribution



The SAM L10/L11 clock system consists of these features:

- *Clock sources*, that is oscillators controlled by OSCCTRL and OSC32KCTRL
  - A clock source provides a time base that is used by other components, such as Generic Clock Generators. Example clock sources are the internal 16MHz oscillator (OSC16M), external crystal oscillator (XOSC) and the Fractional Digital Phase Locked Loop (FDPLL96M).
- *Generic Clock Controller (GCLK),* which generates, controls and distributes the asynchronous clock consisting of:
  - *Generic Clock Generators:* These are programmable prescalers that can use any of the system clock sources as a time base. The Generic Clock Generator 0 generates the clock signal GCLK_MAIN, which is used by the Power Manager and the Main Clock (MCLK) module, which in turn generates synchronous clocks.
  - *Generic Clocks:* These are clock signals generated by Generic Clock Generators and output by the Peripheral Channels, and serve as clocks for the peripherals of the system. Multiple instances of a peripheral will typically have a separate Generic Clock for each instance.

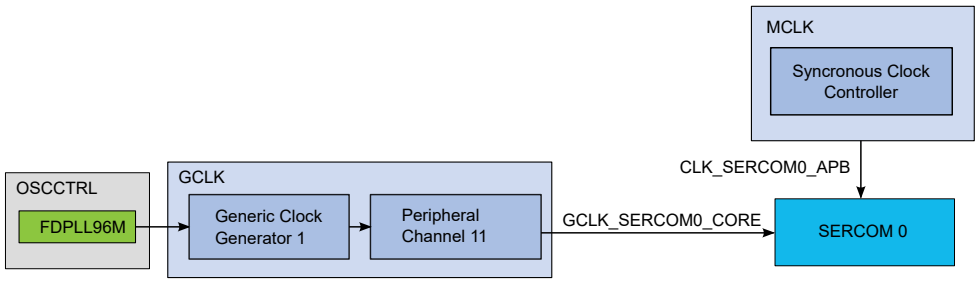
Downloaded from Arrow.com.

Generic Clock 0 serves as the clock source for the FDPLL96M clock input (when multiplying another clock source).

- *Main Clock Controller (MCLK)*
  - The MCLK generates and controls the synchronous clocks on the system. This includes the CPU, bus clocks (APB, AHB) as well as the synchronous (to the CPU) user interfaces of the peripherals. It contains clock masks that can turn on/off the user interface of a peripheral as well as prescalers for the CPU and bus clocks.

The figure below illustrates an example, where SERCOM0 is clocked by the FDPLL96M in Open Loop mode. The FDPLL96M is enabled, the Generic Clock Generator 1 uses the FDPLL96M as its clock source and feeds into Peripheral Channel 11. The Generic Clock 10, also called GCLK_SERCOM0_CORE, is connected to SERCOM0. The SERCOM0 interface, clocked by CLK_SERCOM0_APB, has been unmasked in the APBC Mask register in the MCLK.

Figure 17-2. Example of SERCOM Clock



To customize the clock distribution, refer to these registers and bit fields:

- The source oscillator for a generic clock generator 'n' is selected by writing to the Source bit field in the Generator Control n register (GCLK.GENCTRLn.SRC).
- A Peripheral Channel m can be configured to use a specific Generic Clock Generator by writing to the Generic Clock Generator bit field in the respective Peripheral Channel m register (GCLK.PCHCTRLm.GEN)
- The Peripheral Channel number, *m*, is fixed for a given peripheral. See the Mapping table in the description of GCLK.PCHCTRLm.
- The AHB clocks are enabled and disabled by writing to the respective bit in the AHB Mask register (MCLK.AHBMASK).
- The APB clocks are enabled and disabled by writing to the respective bit in the APB x Mask registers (MCLK.APBxMASK).

17.2 Synchronous and Asynchronous Clocks

As the CPU and the peripherals can be in different clock domains, that is, they are clocked from different clock sources and with different clock speeds, some peripheral accesses by the CPU need to be synchronized. In this case the peripheral includes a Synchronization Busy (SYNCBUSY) register that can be used to check if a sync operation is in progress.

For a general description, see 17.3 Register Synchronization. Some peripherals have specific properties described in their individual “Synchronization” sub-sections.

Downloaded from Arrow.com.

In the data sheet, references to Synchronous Clocks are referring to the CPU and bus clocks (MCLK), while asynchronous clocks are generated by the Generic Clock Controller (GCLK).

17.3 Register Synchronization

17.3.1 Overview

All peripherals are composed of one digital bus interface connected to the APB or AHB bus and running from a corresponding clock in the Main Clock domain, and one peripheral core running from the peripheral Generic Clock (GCLK).

Communication between these clock domains must be synchronized. This mechanism is implemented in hardware, so the synchronization process takes place even if the peripheral generic clock is running from the same clock source and on the same frequency as the bus interface.

All registers in the bus interface are accessible without synchronization.

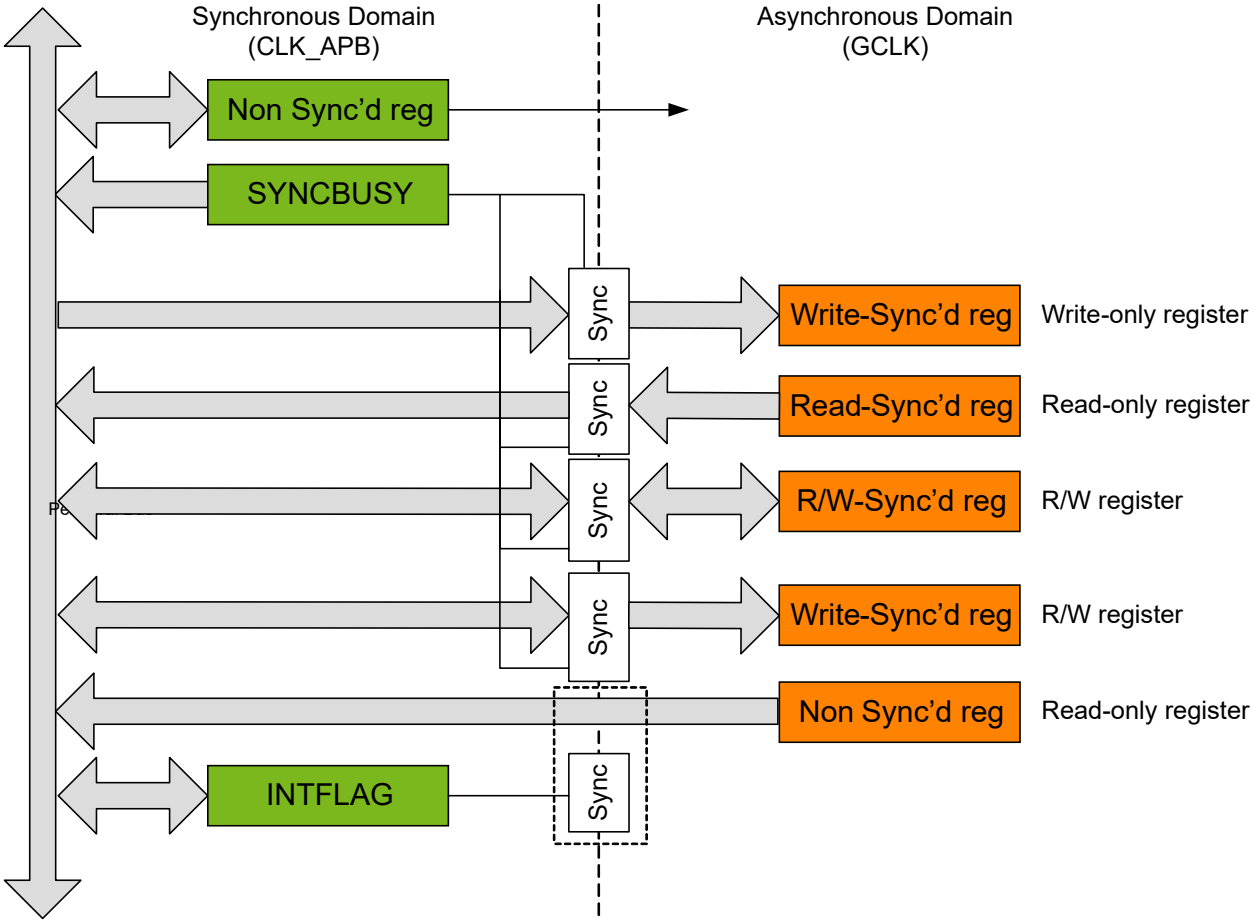
All registers in the peripheral core are synchronized when written. Some registers in the peripheral core are synchronized when read.

Each individual register description will have the properties "Read-Synchronized" and/or "Write-Synchronized" if a register is synchronized.

As shown in the figure below, each register that requires synchronization has its individual synchronizer and its individual synchronization status bit in the Synchronization Busy register (SYNCBUSY).
Note: For registers requiring both read- and write-synchronization, the corresponding bit in SYNCBUSY is shared.

Downloaded from Arrow.com.

Figure 17-3. Register Synchronization Overview



17.3.2 General Write Synchronization

Write-Synchronization is triggered by writing to a register in the peripheral clock domain (GCLK). The respective bit in the Synchronization Busy register (SYNCBUSY) will be set when the write-synchronization starts and cleared when the write-synchronization is complete. Refer also to17.3.7 Synchronization Delay.

When write-synchronization is ongoing for a register, any subsequent write attempts to this register will be discarded, and an error will be reported though the Peripheral Access Controller (PAC).

Example:

REGA, REGB are 8-bit core registers. REGC is a 16-bit core register.

Offset	Register
0x00	REGA
0x01	REGB
0x02	REGC
0x03	

Synchronization is per register, so multiple registers can be synchronized in parallel. Consequently, after REGA (8-bit access) was written, REGB (8-bit access) can be written immediately without error.

Downloaded from Arrow.com.

REGC (16-bit access) can be written without affecting REGA or REGB. If REGC is written to in two consecutive 8-bit accesses without waiting for synchronization, the second write attempt will be discarded and an error is generated through the PAC.

A 32-bit access to offset 0x00 will write all three registers. Note that REGA, REGB and REGC can be updated at different times because of independent write synchronization.

17.3.3 General Read Synchronization

Read-synchronized registers are synchronized each time the register value is updated but the corresponding SYNCBUSY bits are not set. Reading a read-synchronized register does not start a new synchronization, it returns the last synchronized value.
Note: The corresponding bits in SYNCBUSY will automatically be set when the device wakes up from sleep because read-synchronized registers need to be synchronized. Therefore reading a read-synchronized register before its corresponding SYNCBUSY bit is cleared will return the last synchronized value before sleep mode.
Moreover, if a register is also write-synchronized, any write access while the SYNCBUSY bit is set will be discarded and generate an error.

17.3.4 Completion of Synchronization

In order to check if synchronization is complete, the user can either poll the relevant bits in SYNCBUSY or use the Synchronisation Ready interrupt (if available). The Synchronization Ready interrupt flag will be set when all ongoing synchronizations are complete, i.e. when all bits in SYNCBUSY are '0'.

17.3.5 Write Synchronization for CTRLA.ENABLE

Setting the Enable bit in a module's Control A register (CTRLA.ENABLE) will trigger write-synchronization and set SYNCBUSY.ENABLE.

CTRLA.ENABLE will read its new value immediately after being written.

SYNCBUSY.ENABLE will be cleared by hardware when the operation is complete.

The Synchronization Ready interrupt (if available) cannot be used to enable write-synchronization.

17.3.6 Write-Synchronization for Software Reset Bit

Setting the Software Reset bit in CTRLA (CTRLA.SWRST=1) will trigger write-synchronization and set SYNCBUSY.SWRST. When writing a ‘1’ to the CTRLA.SWRST bit it will immediately read as ‘1’.

CTRL.SWRST and SYNCBUSY.SWRST will be cleared by hardware when the peripheral has been reset.

Writing a '0' to the CTRL.SWRST bit has no effect.

The Ready interrupt (if available) cannot be used for Software Reset write-synchronization.

Note: Not all peripherals have the SWRST bit in the respective CTRLA register.

17.3.7 Synchronization Delay

The synchronization will delay write and read accesses by a certain amount. This delay *D* is within the range of:

$5 \times P_{GCLK} + 2 \times P_{APB} < D < 6 \times P_{GCLK} + 3 \times P_{APB}$

Where P_{GCLK} is the period of the generic clock and P_{APB} is the period of the peripheral bus clock. A normal peripheral bus register access duration is $2 \times P_{APB}$.

Downloaded from Arrow.com.

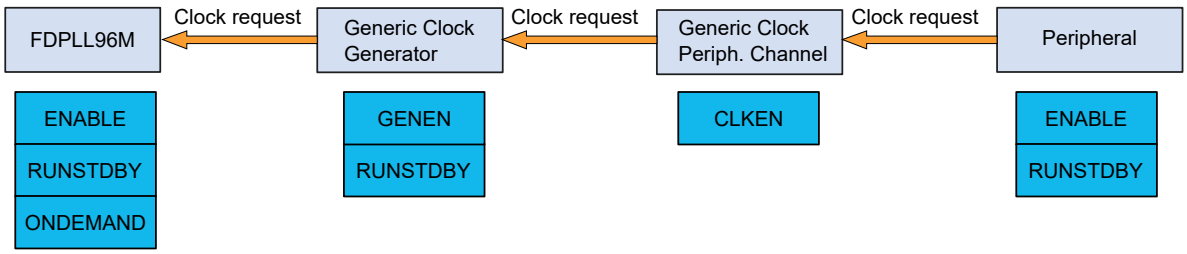
17.4 Enabling a Peripheral

In order to enable a peripheral that is clocked by a Generic Clock, the following parts of the system needs to be configured:

- A running Clock Source
- A clock from the Generic Clock Generator must be configured to use one of the running Clock Sources, and the Generator must be enabled.
- The Peripheral Channel that provides the Generic Clock signal to the peripheral must be configured to use a running Generic Clock Generator, and the Generic Clock must be enabled.
- The user interface of the peripheral needs to be unmasked in the Main Clock Controller (MCLK). If this is not done the peripheral registers will read all '0's and any writing attempts to the peripheral will be discarded.

17.5 On Demand Clock Requests

Figure 17-4. Clock Request Routing



All clock sources in the system can be run in an on-demand mode: the clock source is in a stopped state unless a peripheral is requesting the clock source. Clock requests propagate from the peripheral, via the GCLK, to the clock source. If one or more peripheral is using a clock source, the clock source will be started/kept running. As soon as the clock source is no longer needed and no peripheral has an active request, the clock source will be stopped until requested again.

The clock request can reach the clock source only if the peripheral, the generic clock and the clock from the Generic Clock Generator in-between are enabled. The time taken from a clock request being asserted to the clock source being ready is dependent on the clock source startup time, clock source frequency as well as the divider used in the Generic Clock Generator. The total startup time T_{start} from a clock request until the clock is available for the peripheral is between:

T_{start_max} = Clock source startup time + 2 × clock source periods + 2 × divided clock source periods

T_{start_min} = Clock source startup time + 1 × clock source period + 1 × divided clock source period

The time between the last active clock request stopped and the clock is shut down, T_{stop}, is between:

T_{stop_min} = 1 × divided clock source period + 1 × clock source period

T_{stop_max} = 2 × divided clock source periods + 2 × clock source periods

The On-Demand function can be disabled individually for each clock source by clearing the ONDEMAND bit located in each clock source controller. Consequently, the clock will always run whatever the clock request status is. This has the effect of removing the clock source startup time at the cost of power consumption.

The clock request mechanism can be configured to work in standby mode by setting the RUNSDTBY bits of the modules (see Figure 17-4).

Downloaded from Arrow.com.

17.6 Power Consumption vs. Speed

When targeting for either a low-power or a fast acting system, some considerations have to be taken into account due to the nature of the asynchronous clocking of the peripherals:

If clocking a peripheral with a very low clock, the active power consumption of the peripheral will be lower. At the same time the synchronization to the synchronous (CPU) clock domain is dependent on the peripheral clock speed, and will take longer with a slower peripheral clock. This will cause worse response times and longer synchronization delays.

17.7 Clocks after Reset

On any Reset the synchronous clocks start to their initial state:

- OSC16M is enabled and configured to run at 4MHz
- Generic Clock Generator 0 uses OSC16M as source and generates GCLK_MAIN and CLK_MAIN
- CPU and BUS clocks are undivided

On a Power-on Reset, the 32KHz clock sources are reset and the GCLK module starts to its initial state:

- All Generic Clock Generators are disabled except Generator 0
- All Peripheral Channels in GCLK are disabled.

On a User Reset the GCLK module starts to its initial state, except for:

- Generic Clocks that are write-locked, i.e., the according WRTLOCK is set to 1 prior to Reset

Downloaded from Arrow.com.

18. GCLK - Generic Clock Controller

18.1 Overview

Depending on the application, peripherals may require specific clock frequencies to operate correctly. The Generic Clock controller (GCLK) features 5 Generic Clock Generators 0..4 that can provide a wide range of clock frequencies.

Generators can be set to use different external and internal oscillators as source. The clock of each Generator can be divided. The outputs from the Generators are used as sources for the Peripheral Channels, which provide the Generic Clock (GCLK_PERIPH) to the peripheral modules, as shown in Figure 18-2. The number of Peripheral Clocks depends on how many peripherals the device has.
Note: The Generator 0 is always the direct source of the GCLK_MAIN signal.

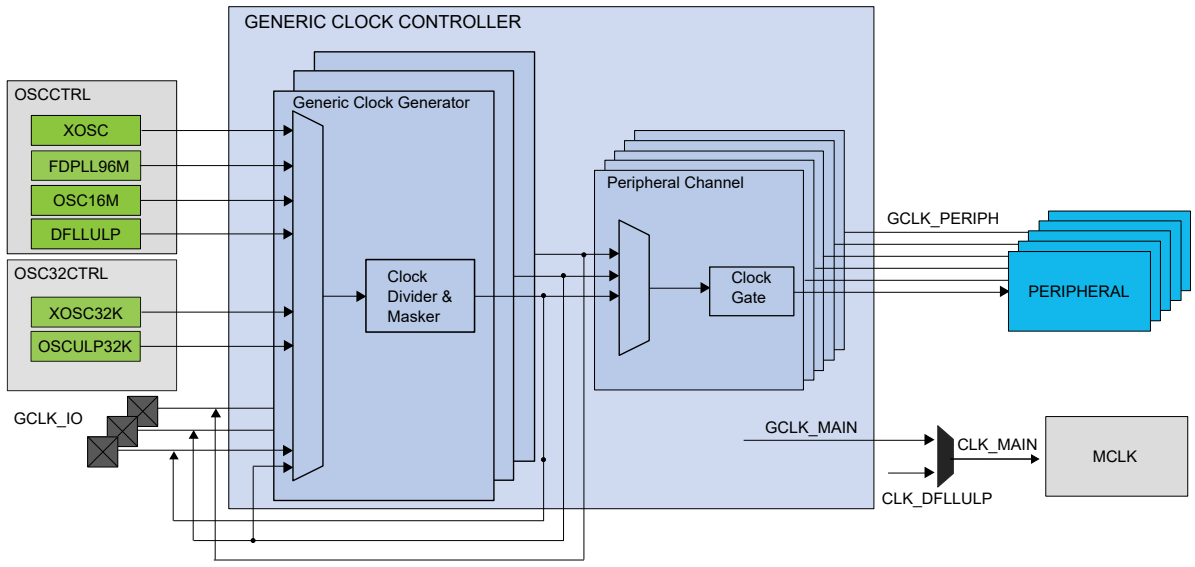
18.2 Features

- Provides a device-defined, configurable number of Peripheral Channel clocks
- Wide frequency range:
  - Various clock sources
  - Embedded dividers

18.3 Block Diagram

The generation of Peripheral Clock signals (GCLK_PERIPH) and the Main Clock (GCLK_MAIN) can be seen in Device Clocking Diagram.

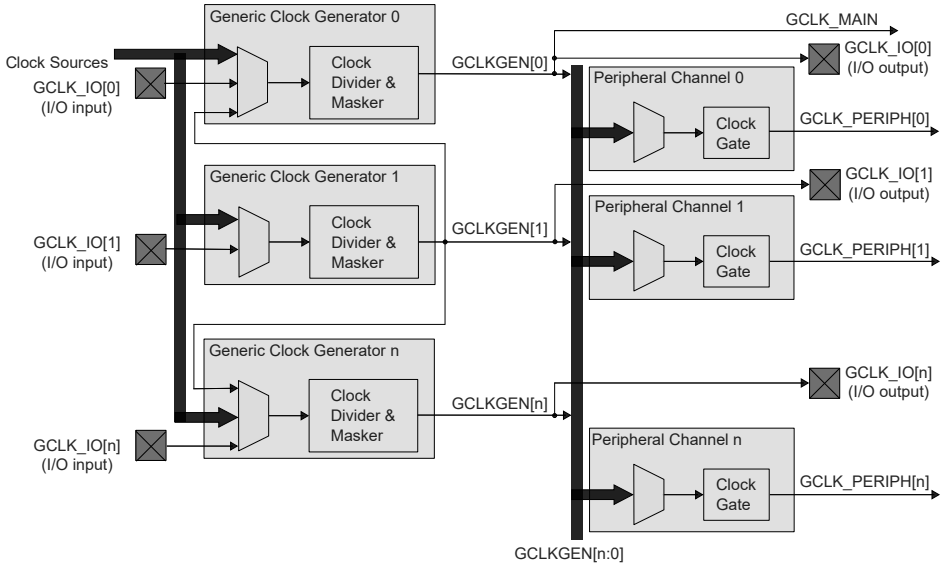
Figure 18-1. Device Clocking Diagram



The GCLK block diagram is shown below:

Downloaded from Arrow.com.

Figure 18-2. Generic Clock Controller Block Diagram



18.4 Signal Description

Table 18-1. GCLK Signal Description

Signal Name	Type	Description
GCLK_IO[4:0]	Digital I/O	Clock source for Generators when input Generic Clock signal when output

Note: One signal can be mapped on several pins.

18.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

18.5.1 I/O Lines

Using the GCLK I/O lines requires the I/O pins to be configured.

Related Links

32. PORT - I/O Pin Controller

18.5.2 Power Management

The GCLK can operate in sleep modes, if required. Refer to the sleep mode description in the Power Manager (PM) section.

Related Links

22. PM – Power Manager

18.5.3 Clocks

The GCLK bus clock (CLK_GCLK_APB) can be enabled and disabled in the Main Clock Controller.

Downloaded from Arrow.com.

Related Links

19.6.2.6 Peripheral Clock Masking

24. OSC32KCTRL – 32KHz Oscillators Controller

18.5.4 DMA

Not applicable.

18.5.5 Interrupts

Not applicable.

18.5.6 Events

Not applicable.

18.5.7 Debug Operation

When the CPU is halted in debug mode the GCLK continues normal operation. If the GCLK is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

18.5.8 Register Access Protection

All registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC).

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

Write-protection does not apply for accesses through an external debugger.

Related Links

15. PAC - Peripheral Access Controller

18.5.9 SAM L11 TrustZone Specific Register Access Protection

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

18.5.10 Analog Connections

Not applicable.

18.6 Functional Description

18.6.1 Principle of Operation

The GCLK module is comprised of five Generic Clock Generators (Generators) sourcing up to 64 Peripheral Channels and the Main Clock signal CLK_MAIN.

Downloaded from Arrow.com.

A clock source selected as input to a Generator can either be used directly, or it can be prescaled in the Generator. A generator output is used by one or more Peripheral Channels to provide a peripheral generic clock signal (GCLK_PERIPH) to the peripherals.

18.6.2 Basic Operation

18.6.2.1 Initialization

Before a Generator is enabled, the corresponding clock source should be enabled. The Peripheral clock must be configured as outlined by the following steps:

1. The Generator must be enabled (GENCTRLn.GENEN=1) and the division factor must be set (GENTRLn.DIVSEL and GENCTRLn.DIV) by performing a single 32-bit write to the Generator Control register (GENCTRLn).
2. The Generic Clock for a peripheral must be configured by writing to the respective Peripheral Channel Control register (PCHCTRLm). The Generator used as the source for the Peripheral Clock must be written to the GEN bit field in the Peripheral Channel Control register (PCHCTRLm.GEN).

Note: Each Generator n is configured by one dedicated register GENCTRLn.

Note: Each Peripheral Channel m is configured by one dedicated register PCHCTRLm.

18.6.2.2 Enabling, Disabling, and Resetting

The GCLK module has no enable/disable bit to enable or disable the whole module.

The GCLK is reset by setting the Software Reset bit in the Control A register (CTRLA.SWRST) to 1. All registers in the GCLK will be reset to their initial state, except for Peripheral Channels and associated Generators that have their Write Lock bit set to 1 (PCHCTRLm.WRTLOCK). For further details, refer to 18.6.3.4 Configuration Lock.

18.6.2.3 Generic Clock Generator

Each Generator (GCLK_GEN) can be set to run from one of eight different clock sources except GCLK_GEN[1], which can be set to run from one of seven sources. GCLK_GEN[1] is the only Generator that can be selected as source to others Generators.

Each generator GCLK_GEN[x] can be connected to one specific pin GCLK_IO[x]. A pin GCLK_IO[x] can be set either to act as source to GCLK_GEN[x] or to output the clock signal generated by GCLK_GEN[x].

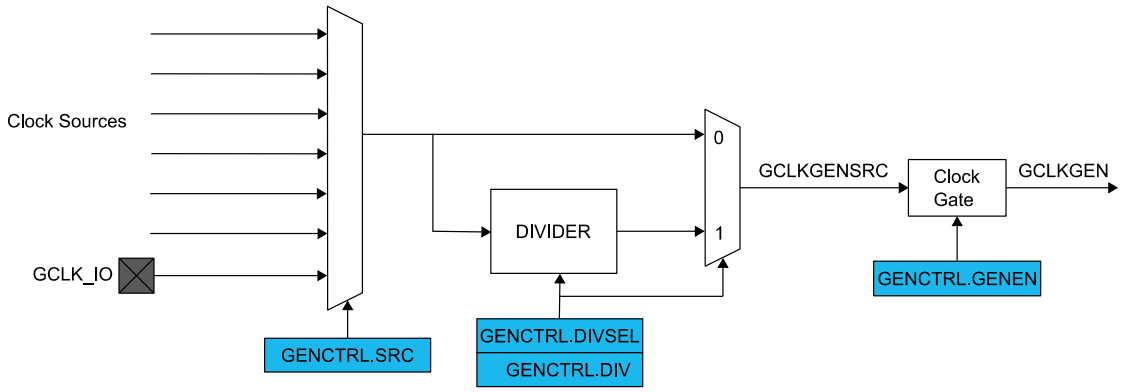
The selected source can be divided. Each Generator can be enabled or disabled independently.

Each GCLK_GEN clock signal can then be used as clock source for Peripheral Channels. Each Generator output is allocated to one or several Peripherals.

GCLK_GEN[0] is used as GCLK_MAIN for the synchronous clock controller inside the Main Clock Controller. Refer to the Main Clock Controller description for details on the synchronous clock generation.

Downloaded from Arrow.com.

Figure 18-3. Generic Clock Generator



Related Links

19. MCLK – Main Clock

18.6.2.4 Enabling a Generator

A Generator is enabled by writing a '1' to the Generator Enable bit in the Generator Control register (GENCTRLn.GENEN=1).

18.6.2.5 Disabling a Generator

A Generator is disabled by writing a '0' to GENCTRLn.GENEN. When GENCTRLn.GENEN=0, the GCLK_GEN[n] clock is disabled and gated.

18.6.2.6 Selecting a Clock Source for the Generator

Each Generator can individually select a clock source by setting the Source Select bit group in the Generator Control register (GENCTRLn.SRC).

Changing from one clock source, for example A, to another clock source, B, can be done on the fly: If clock source B is not ready, the Generator will continue using clock source A. As soon as source B is ready, the Generator will switch to it. During the switching operation, the Generator maintains clock requests to both clock sources A and B, and will release source A as soon as the switch is done. The according bit in SYNCBUSY register (SYNCBUSY.GENCTRLn) will remain '1' until the switch operation is completed.

The available clock sources are device dependent (usually the oscillators, RC oscillators, DPLL, and DFLLULP). Only Generator 1 can be used as a common source for all other generators.

18.6.2.7 Changing the Clock Frequency

The selected source for a Generator can be divided by writing a division value in the Division Factor bit field of the Generator Control register (GENCTRLn.DIV). How the actual division factor is calculated is depending on the Divide Selection bit (GENCTRLn.DIVSEL).

If GENCTRLn.DIVSEL=0 and GENCTRLn.DIV is either 0 or 1, the output clock will be undivided.

Note: The number of available DIV bits may vary from Generator to Generator.

18.6.2.8 Duty Cycle

When dividing a clock with an odd division factor, the duty-cycle will not be 50/50. Setting the Improve Duty Cycle bit of the Generator Control register (GENCTRLn.IDC) will result in a 50/50 duty cycle.

18.6.2.9 External Clock

The output clock (GCLK_GEN) of each Generator can be sent to I/O pins (GCLK_IO).

Downloaded from Arrow.com.

If the Output Enable bit in the Generator Control register is set (GENCTRLn.OE = 1) and the generator is enabled (GENCTRLn.GENEN=1), the Generator requests its clock source and the GCLK_GEN clock is output to an I/O pin.
Note: The I/O pin (GCLK/IO[n]) must first be configured as output by writing the corresponding PORT registers.

If GENCTRLn.OE is 0, the according I/O pin is set to an Output Off Value, which is selected by GENCTRLn.OOV: If GENCTRLn.OOV is '0', the output clock will be low. If this bit is '1', the output clock will be high.

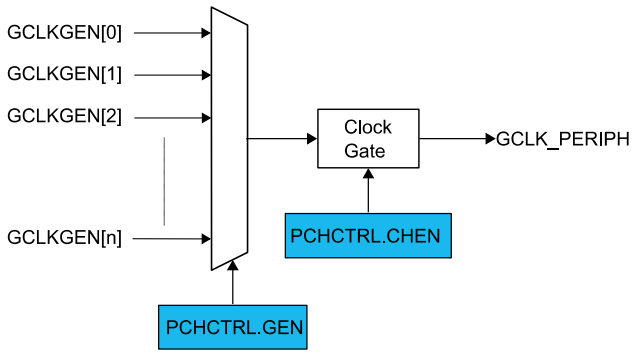
In Standby mode, if the clock is output (GENCTRLn.OE=1), the clock on the I/O pin is frozen to the OOV value if the Run In Standby bit of the Generic Control register (GENCTRLn.RUNSTDBY) is zero.
Note: With GENCTRLn.OE=1 and RUNSTDBY=0, entering the Standby mode can take longer due to a clock source dependent delay between turning off Power Domain PDSW. The maximum delay can be equal to the clock source period multiplied by the division factor.

If GENCTRLn.RUNSTDBY is '1', the GCLKGEN clock is kept running and output to the I/O pin.

Related Links
22.6.3.5 Power Domain Controller

18.6.3 Peripheral Clock

Figure 18-4. Peripheral Clock



18.6.3.1 Enabling a Peripheral Clock

Before a Peripheral Clock is enabled, one of the Generators must be enabled (GENCTRLn.GENEN) and selected as source for the Peripheral Channel by setting the Generator Selection bits in the Peripheral Channel Control register (PCHCTRL.GEN). Any available Generator can be selected as clock source for each Peripheral Channel.

When a Generator has been selected, the peripheral clock is enabled by setting the Channel Enable bit in the Peripheral Channel Control register, PCHCTRLm.CHEN = 1. The PCHCTRLm.CHEN bit must be synchronized to the generic clock domain. PCHCTRLm.CHEN will continue to read as its previous state until the synchronization is complete.

18.6.3.2 Disabling a Peripheral Clock

A Peripheral Clock is disabled by writing PCHCTRLm.CHEN=0. The PCHCTRLm.CHEN bit must be synchronized to the Generic Clock domain. PCHCTRLm.CHEN will stay in its previous state until the synchronization is complete. The Peripheral Clock is gated when disabled.

Related Links
18.8.4 PCHCTRLm

Downloaded from Arrow.com.

18.6.3.3 Selecting the Clock Source for a Peripheral

When changing a peripheral clock source by writing to PCHCTRLm.GEN, the peripheral clock must be disabled before re-enabling it with the new clock source setting. This prevents glitches during the transition:

1. Disable the Peripheral Channel by writing PCHCTRLm.CHEN=0
2. Assert that PCHCTRLm.CHEN reads '0'
3. Change the source of the Peripheral Channel by writing PCHCTRLm.GEN
4. Re-enable the Peripheral Channel by writing PCHCTRLm.CHEN=1

Related Links

18.8.4 PCHCTRLm

18.6.3.4 Configuration Lock

The peripheral clock configuration can be locked for further write accesses by setting the Write Lock bit in the Peripheral Channel Control register PCHCTRLm.WRTLOCK=1). All writing to the PCHCTRLm register will be ignored. It can only be unlocked by a Power Reset.

The Generator source of a locked Peripheral Channel will be locked, too: The corresponding GENCTRLn register is locked, and can be unlocked only by a Power Reset.

There is one exception concerning the Generator 0. As it is used as GCLK_MAIN, it cannot be locked. It is reset by any Reset and will start up in a known configuration. The software reset (CTRLA.SWRST) can not unlock the registers.

In case of an external Reset, the Generator source will be disabled. Even if the WRTLOCK bit is written to '1' the peripheral channels are disabled (PCHCTRLm.CHEN set to '0') until the Generator source is enabled again. Then, the PCHCTRLm.CHEN are set to '1' again.

Related Links

18.8.1 CTRLA

18.6.4 Additional Features

18.6.4.1 Peripheral Clock Enable after Reset

The Generic Clock Controller must be able to provide a generic clock to some specific peripherals after a Reset. That means that the configuration of the Generators and Peripheral Channels after Reset is device-dependent.

Refer to GENCTRLn.SRC for details on GENCTRLn reset.

Refer to PCHCTRLm.SRC for details on PCHCTRLm reset.

18.6.5 Sleep Mode Operation

18.6.5.1 SleepWalking

The GCLK module supports the SleepWalking feature.

If the system is in a sleep mode where the Generic Clocks are stopped, a peripheral that needs its clock in order to execute a process must request it from the Generic Clock Controller.

The Generic Clock Controller receives this request, determines which Generic Clock Generator is involved and which clock source needs to be awakened. It then wakes up the respective clock source, enables the Generator and Peripheral Channel stages successively, and delivers the clock to the peripheral.

Downloaded from Arrow.com.

The RUNSTDBY bit in the Generator Control register controls clock output to pin during standby sleep mode. If the bit is cleared, the Generator output is not available on pin. When set, the GCLK can continuously output the generator output to GCLK_IO. Refer to 18.6.2.9 External Clock for details.

Related Links

22. PM – Power Manager

18.6.5.2 Minimize Power Consumption in Standby

The following table identifies when a Clock Generator is off in Standby Mode, minimizing the power consumption:

Table 18-2. Clock Generator n Activity in Standby Mode

Request for Clock n present	GENCTRLn.RUNSTDBY	GENCTRLn.OE	Clock Generator n
yes	-	-	active
no	1	1	active
no	1	0	OFF
no	0	1	OFF
no	0	0	OFF

18.6.5.3 Entering Standby Mode

There may occur a delay when the device is put into Standby, until the power is turned off. This delay is caused by running Clock Generators: if the Run in Standby bit in the Generator Control register (GENCTRLn.RUNSTDBY) is '0', GCLK must verify that the clock is turned of properly. The duration of this verification is frequency-dependent.

Related Links

22. PM – Power Manager

18.6.6 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

An exception is the Channel Enable bit in the Peripheral Channel Control registers (PCHCTRLm.CHEN). When changing this bit, the bit value must be read-back to ensure the synchronization is complete and to assert glitch free internal operation. Note that changing the bit value under ongoing synchronization will *not* generate an error.

The following registers are synchronized when written:

- Generic Clock Generator Control register (GENCTRLn)
- Control A register (CTRLA)

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Related Links

18.8.1 CTRLA
18.8.4 PCHCTRLm

Downloaded from Arrow.com.

18.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0								SWRST
0x01 ... 0x03	Reserved									
0x04	SYNCBUSY	7:0		GENCTRL4	GENCTRL3	GENCTRL2	GENCTRL1	GENCTRL0		SWRST
		15:8								
		23:16								
		31:24								
0x08 ... 0x1F	Reserved									
0x20	GENCTRL0	7:0				SRC[4:0]				
		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
		23:16	DIV[7:0]							
		31:24	DIV[15:8]							
0x24	GENCTRL1	7:0				SRC[4:0]				
		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
		23:16	DIV[7:0]							
		31:24	DIV[15:8]							
0x28	GENCTRL2	7:0				SRC[4:0]				
		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
		23:16	DIV[7:0]							
		31:24	DIV[15:8]							
0x2C	GENCTRL3	7:0				SRC[4:0]				
		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
		23:16	DIV[7:0]							
		31:24	DIV[15:8]							
0x30	GENCTRL4	7:0				SRC[4:0]				
		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
		23:16	DIV[7:0]							
		31:24	DIV[15:8]							
0x34 ... 0x7F	Reserved									
0x80	PCHCTRL0	7:0	WRTLOCK	CHEN				GEN[2:0]		
		15:8								
		23:16								
		31:24								
0x84	PCHCTRL1	7:0	WRTLOCK	CHEN				GEN[2:0]		
		15:8								
		23:16								
		31:24								
0x88	PCHCTRL2	7:0	WRTLOCK	CHEN				GEN[2:0]		

Downloaded from Arrow.com.

Offset	Name	Bit Pos.								
		15:8								
		23:16								
		31:24								
0x8C	PCHCTRL3	7:0	WRTLOCK	CHEN				GEN[2:0]		
		15:8								
		23:16								
		31:24								
0x90	PCHCTRL4	7:0	WRTLOCK	CHEN				GEN[2:0]		
		15:8								
		23:16								
		31:24								
0x94	PCHCTRL5	7:0	WRTLOCK	CHEN				GEN[2:0]		
		15:8								
		23:16								
		31:24								
0x98	PCHCTRL6	7:0	WRTLOCK	CHEN				GEN[2:0]		
		15:8								
		23:16								
		31:24								
0x9C	PCHCTRL7	7:0	WRTLOCK	CHEN				GEN[2:0]		
		15:8								
		23:16								
		31:24								
0xA0	PCHCTRL8	7:0	WRTLOCK	CHEN				GEN[2:0]		
		15:8								
		23:16								
		31:24								
0xA4	PCHCTRL9	7:0	WRTLOCK	CHEN				GEN[2:0]		
		15:8								
		23:16								
		31:24								
0xA8	PCHCTRL10	7:0	WRTLOCK	CHEN				GEN[2:0]		
		15:8								
		23:16								
		31:24								
0xAC	PCHCTRL11	7:0	WRTLOCK	CHEN				GEN[2:0]		
		15:8								
		23:16								
		31:24								
0xB0	PCHCTRL12	7:0	WRTLOCK	CHEN				GEN[2:0]		
		15:8								
		23:16								
		31:24								
0xB4	PCHCTRL13	7:0	WRTLOCK	CHEN				GEN[2:0]		
		15:8								
		23:16								

Downloaded from Arrow.com.

Offset	Name	Bit Pos.								
		31:24								
0xB8	PCHCTRL14	7:0	WRTLOCK	CHEN				GEN[2:0]		
		15:8								
		23:16								
		31:24								
0xBC	PCHCTRL15	7:0	WRTLOCK	CHEN				GEN[2:0]		
		15:8								
		23:16								
		31:24								
0xC0	PCHCTRL16	7:0	WRTLOCK	CHEN				GEN[2:0]		
		15:8								
		23:16								
		31:24								
0xC4	PCHCTRL17	7:0	WRTLOCK	CHEN				GEN[2:0]		
		15:8								
		23:16								
		31:24								
0xC8	PCHCTRL18	7:0	WRTLOCK	CHEN				GEN[2:0]		
		15:8								
		23:16								
		31:24								
0xCC	PCHCTRL19	7:0	WRTLOCK	CHEN				GEN[2:0]		
		15:8								
		23:16								
		31:24								
0xD0	PCHCTRL20	7:0	WRTLOCK	CHEN				GEN[2:0]		
		15:8								
		23:16								
		31:24								

18.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to 18.5.8 Register Access Protection.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to 18.6.6 Synchronization.

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted

Downloaded from Arrow.com.

- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

Downloaded from Arrow.com.

18.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
								SWRST
Access								R/W
Reset								0

Bit 0 – SWRST Software Reset
Writing a zero to this bit has no effect.

Setting this bit to 1 will reset all registers in the GCLK to their initial state after a Power Reset, except for generic clocks and associated Generators that have their WRTLOCK bit in PCHCTRLm set to 1.

Refer to GENCTRL Reset Value for details on GENCTRL register reset.

Refer to PCHCTRL Reset Value for details on PCHCTRL register reset.

Due to synchronization, there is a waiting period between setting CTRLA.SWRST and a completed Reset. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

Value	Description
0	There is no Reset operation ongoing.
1	A Reset operation is ongoing.

Downloaded from Arrow.com.

18.8.2 Synchronization Busy

Name: SYNCBUSY
Offset: 0x04
Reset: 0x00000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		GENCTRL4	GENCTRL3	GENCTRL2	GENCTRL1	GENCTRL0		SWRST
Access		R	R	R	R	R		R
Reset		0	0	0	0	0		0

Bits 2, 3, 4, 5, 6 – GENCTRL Generator Control n Synchronization Busy
This bit is cleared when the synchronization of the Generator Control n register (GENCTRLn) between clock domains is complete, or when clock switching operation is complete.

This bit is set when the synchronization of the Generator Control n register (GENCTRLn) between clock domains is started.

Bit 0 – SWRST Software Reset Synchronization Busy
This bit is cleared when the synchronization of the CTRLA.SWRST register bit between clock domains is complete.

This bit is set when the synchronization of the CTRLA.SWRST register bit between clock domains is started.

Downloaded from Arrow.com.

18.8.3 Generator Control

Name: GENCTRLn
Offset: 0x20 + n*0x04 [n=0..4]
Reset: 0x00000105
Property: PAC Write-Protection, Write-Synchronized

GENCTRLn controls the settings of Generic Generator n (n=0..4). The reset value is 0x00000105 for Generator n=0, else 0x00000000

Bit	31	30	29	28	27	26	25	24
	DIV[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
Access								
Reset			0	0	0	0	0	1

Bit	7	6	5	4	3	2	1	0
				SRC[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 31:16 – DIV[15:0] Division Factor
These bits represent a division value for the corresponding Generator. The actual division factor is dependent on the state of DIVSEL. The number of relevant DIV bits for each Generator can be seen in this table. Written bits outside of the specified range will be ignored.

Table 18-3. Division Factor Bits

Generic Clock Generator	Division Factor Bits
Generator 0	8 division factor bits - DIV[7:0]
Generator 1	16 division factor bits - DIV[15:0]
Generator 2 - 4	8 division factor bits - DIV[7:0]

Bit 13 – RUNSTDBY Run in Standby
This bit is used to keep the Generator running in Standby as long as it is configured to output to a dedicated GCLK_IO pin. If GENCTRLn.OE is zero, this bit has no effect and the generator will only be running if a peripheral requires the clock.

Downloaded from Arrow.com.

Value	Description
0	The Generator is stopped in Standby and the GCLK_IO pin state (one or zero) will be dependent on the setting in GENCTRL.OOV.
1	The Generator is kept running and output to its dedicated GCLK_IO pin during Standby mode.

Bit 12 – DIVSEL Divide Selection
This bit determines how the division factor of the clock source of the Generator will be calculated from DIV. If the clock source should not be divided, DIVSEL must be 0 and the GENCTRLn.DIV value must be either 0 or 1.

Value	Description
0	The Generator clock frequency equals the clock source frequency divided by GENCTRLn.DIV.
1	The Generator clock frequency equals the clock source frequency divided by 2^(N+1), where N is the Division Factor Bits for the selected generator (refer to GENCTRLn.DIV).

Bit 11 – OE Output Enable
This bit is used to output the Generator clock output to the corresponding pin (GCLK_IO), as long as GCLK_IO is not defined as the Generator source in the GENCTRLn.SRC bit field.

Value	Description
0	No Generator clock signal on pin GCLK_IO.
1	The Generator clock signal is output on the corresponding GCLK_IO, unless GCLK_IO is selected as a generator source in the GENCTRLn.SRC bit field.

Bit 10 – OOV Output Off Value
This bit is used to control the clock output value on pin (GCLK_IO) when the Generator is turned off or the OE bit is zero, as long as GCLK_IO is not defined as the Generator source in the GENCTRLn.SRC bit field.

Value	Description
0	The GCLK_IO will be LOW when generator is turned off or when the OE bit is zero.
1	The GCLK_IO will be HIGH when generator is turned off or when the OE bit is zero.

Bit 9 – IDC Improve Duty Cycle
This bit is used to improve the duty cycle of the Generator output to 50/50 for odd division factors.

Value	Description
0	Generator output clock duty cycle is not balanced to 50/50 for odd division factors.
1	Generator output clock duty cycle is 50/50.

Bit 8 – GENEN Generator Enable
This bit is used to enable and disable the Generator.

Value	Description
0	Generator is disabled.
1	Generator is enabled.

Bits 4:0 – SRC[4:0] Generator Clock Source Selection
These bits select the Generator clock source, as shown in this table.

Downloaded from Arrow.com.

Table 18-4. Generator Clock Source Selection

Value	Name	Description
0x00	XOSC	XOSC oscillator output
0x01	GCLK_IN	Generator input pad (GCLK_IO)
0x02	GCLK_GEN1	Generic clock generator 1 output
0x03	OSCULP32K	OSCULP32K oscillator output
0x04	XOSC32K	XOSC32K oscillator output
0x05	OSC16M	OSC16M oscillator output
0x06	DFLLULP	DFLLULP ultra low power output
0x07	FDPLL96M	FDPLL96M output
0x08-0x1F	Reserved	Reserved for future use

A Power Reset will reset all GENCTRLn registers. the Reset values of the GENCTRLn registers are shown in table below.

Table 18-5. GENCTRLn Reset Value after a Power Reset

GCLK Generator	Reset Value after a Power Reset
0	0x00000105
others	0x00000000

A User Reset will reset the associated GENCTRL register unless the Generator is the source of a locked Peripheral Channel (PCHCTRLm.WRTLOCK=1). The reset values of the GENCTRL register are as shown in the table below.

Table 18-6. GENCTRLn Reset Value after a User Reset

GCLK Generator	Reset Value after a User Reset
0	0x00000105
others	No change if the generator is used by a Peripheral Channel m with PCHCTRLm.WRTLOCK=1 else 0x00000000

Related Links

18.8.4 PCHCTRLm

Downloaded from Arrow.com.

18.8.4 Peripheral Channel Control

Name: PCHCTRLm
Offset: 0x80 + m*0x04 [m=0..20]
Reset: 0x00000000
Property: PAC Write-Protection

PCHTRLm controls the settings of Peripheral Channel number m (m=0..20).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	WRTLOCK	CHEN				GEN[2:0]		
Access	R/W	R/W				R/W	R/W	R/W
Reset	0	0				0	0	0

Bit 7 – WRTLOCK Write Lock
After this bit is set to '1', further writes to the PCHCTRLm register will be discarded. The control register of the corresponding Generator n (GENCTRLn), as assigned in PCHCTRLm.GEN, will also be locked. It can only be unlocked by a Power Reset.

Note that Generator 0 cannot be locked.

Value	Description
0	The Peripheral Channel register and the associated Generator register are not locked
1	The Peripheral Channel register and the associated Generator register are locked

Bit 6 – CHEN Channel Enable
This bit is used to enable and disable a Peripheral Channel.

Value	Description
0	The Peripheral Channel is disabled
1	The Peripheral Channel is enabled

Bits 2:0 – GEN[2:0] Generator Selection
This bit field selects the Generator to be used as the source of a peripheral clock, as shown in the table below:

Downloaded from Arrow.com.

Table 18-7. Generator Selection

Value	Description
0x0	Generic Clock Generator 0
0x1	Generic Clock Generator 1
0x2	Generic Clock Generator 2
0x3	Generic Clock Generator 3
0x4	Generic Clock Generator 4
0x5	Generic Clock Generator 5
0x6	Generic Clock Generator 6
0x7	Generic Clock Generator 7
0xA	Generic Clock Generator 10
0xB	Generic Clock Generator 11

Table 18-8. Reset Value after a User Reset or a Power Reset

Reset	PCHCTRLm.GEN	PCHCTRLm.CHEN	PCHCTRLm.WRTLOCK
Power Reset	0x0	0x0	0x0
User Reset	If WRTLOCK = 0 : 0x0 If WRTLOCK = 1: no change	If WRTLOCK = 0 : 0x0 If WRTLOCK = 1: no change	No change

A Power Reset will reset all the PCHCTRLm registers.

A User Reset will reset a PCHCTRL if WRTLOCK=0, or else, the content of that PCHCTRL remains unchanged.

The PCHCTRL register Reset values are shown in the table below, PCHCTRLm Mapping.

Table 18-9. PCHCTRLm Mapping

index(m)	Name	Description
0	GCLK_DPLL	FDPLL96M input clock source for reference
1	GCLK_DPLL_32K	FDPLL96M 32 kHz clock for FDPLL96M internal clock timer
2	GCLK_DFLLULP	DFLLULP clock for DFLLULP
3	GCLK_EIC	EIC
4	GCLK_FREQM_MSR	FREQM Measure
5	GCLK_FREQM_REF	FREQM Reference
6	GCLK_EVSYS_CHANNEL_0	EVSYS_CHANNEL_0
7	GCLK_EVSYS_CHANNEL_1	EVSYS_CHANNEL_1
8	GCLK_EVSYS_CHANNEL_2	EVSYS_CHANNEL_2

Downloaded from Arrow.com.

index(m)	Name	Description
9	GCLK_EVSYS_CHANNEL_3	EVSYS_CHANNEL_3
10	GCLK_SERCOM[0,1,2]_SLOW	SERCOM[0,1,2]_SLOW
11	GCLK_SERCOM0_CORE	SERCOM0_CORE
12	GCLK_SERCOM1_CORE	SERCOM1_CORE
13	GCLK_SERCOM2_CORE	SERCOM2_CORE
14	GCLK_TC0, GCLK_TC1	TC0,TC1
15	GCLK_TC2	TC2
16	GCLK_ADC	ADC
17	GCLK_AC	AC
18	GCLK_DAC	DAC
19	GCLK_PTC	PTC
20	GCLK_CCL	CCL

Downloaded from Arrow.com.

19. MCLK – Main Clock

19.1 Overview

The Main Clock (MCLK) controls the synchronous clock generation of the device.

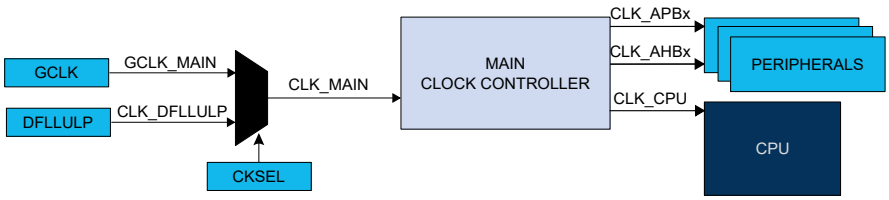
Using a clock provided by the Generic Clock Module (GCLK_MAIN) or the DFLLULP (CLK_DFLLULP), the Main Clock Controller provides synchronous system clocks to the CPU and the modules connected to the AHBx and the APBx bus. The synchronous system clocks are divided into a number of clock domains. Each clock domain can run at different frequencies, enabling the user to save power by running peripherals at a relatively low clock frequency, while maintaining high CPU performance or vice versa. In addition, the clock can be masked for individual modules, enabling the user to minimize power consumption.

19.2 Features

- Generates CPU, AHB, and APB system clocks
  - Clock source and division factor from GCLK
  - Clock prescaler with 1x to 128x division
- Safe run-time clock switching from GCLK
- Module-level clock gating through maskable peripheral clocks

19.3 Block Diagram

Figure 19-1. MCLK Block Diagram



19.4 Signal Description

Not applicable.

19.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

19.5.1 I/O Lines

Not applicable.

19.5.2 Power Management

The MCLK will operate in all sleep modes if a synchronous clock is required in these modes.

Related Links

Downloaded from Arrow.com.

22. PM – Power Manager

19.5.3 Clocks

The MCLK bus clock (CLK_MCLK_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_MCLK_APB can be found in the Peripheral Clock Masking section. If this clock is disabled, it can only be re-enabled by a reset.

The Generic Clock GCLK_MAIN or the DFLLULP Clock CLK_DFLLULP is required to generate the Main Clocks. GCLK_MAIN is configured in the Generic Clock Controller, and can be re-configured by the user if needed. CLK_DFLLULP is configured in the Oscillators Controller (OSCCTRL).

Related Links

18. GCLK - Generic Clock Controller

19.6.2.6 Peripheral Clock Masking

19.5.3.1 Main Clock

The main clock CLK_MAIN is the common source for the synchronous clocks. This is fed into the common 8-bit prescaler that is used to generate synchronous clocks to the CPU, AHBx, and APBx modules.

19.5.3.2 CPU Clock

The CPU clock (CLK_CPU) is routed to the CPU. Halting the CPU clock inhibits the CPU from executing instructions.

19.5.3.3 APBx and AHBx Clock

The APBx clocks (CLK_APBx) and the AHBx clocks (CLK_AHBx) are the root clock source used by modules requiring a clock on the APBx and the AHBx bus. These clocks are always synchronous to the CPU clock, and can run even when the CPU clock is turned off in sleep mode. A clock gater is inserted after the common APB clock to gate any APBx clock of a module on APBx bus, as well as the AHBx clock.

19.5.3.4 Clock Domains

The device has these synchronous clock domains:

- CPU synchronous clock domain (CPU Clock Domain). Frequency is f_{CPU}.

See also the related links for the clock domain partitioning.

Related Links

19.6.2.6 Peripheral Clock Masking

19.5.4 DMA

Not applicable.

19.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. Using the MCLK interrupt requires the Interrupt Controller to be configured first.

19.5.6 Events

Not applicable.

Downloaded from Arrow.com.

19.5.7 Debug Operation

When the CPU is halted in debug mode, the MCLK continues normal operation. In sleep mode, the clocks generated from the MCLK are kept running to allow the debugger accessing any module. As a consequence, power measurements are incorrect in debug mode.

19.5.8 Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Interrupt Flag register (INTFLAG)

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

Related Links

15. PAC - Peripheral Access Controller

19.5.9 SAM L11 TrustZone Specific Register Access Protection

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

19.5.10 Analog Connections

Not applicable.

19.6 Functional Description

19.6.1 Principle of Operation

The CLK_MAIN clock signal from the GCLK module or the DFLLULP is the source for the main clock, which in turn is the common root for the synchronous clocks for the CPU, APBx, and AHBx modules. The CLK_MAIN is divided by an 8-bit prescaler. Each of the derived clocks can run from any divided or undivided main clock, ensuring synchronous clock sources for each clock domain. The clock domain (CPU) can be changed on the fly to respond to variable load in the application. The clocks for each module in a clock domain can be masked individually to avoid power consumption in inactive modules. Depending on the sleep mode, some clock domains can be turned off.

19.6.2 Basic Operation

19.6.2.1 Initialization

After a Reset, the default clock source of the CLK_MAIN clock (GCLK_MAIN) is started and calibrated before the CPU starts running. The GCLK_MAIN clock is selected as the main clock without any prescaler division.

By default, only the necessary clocks are enabled.

Downloaded from Arrow.com.

Related Links

19.6.2.6 Peripheral Clock Masking

19.6.2.2 Enabling, Disabling, and Resetting

The MCLK module is always enabled and cannot be reset.

19.6.2.3 Selecting the Main Clock Source

Refer to the Generic Clock Controller description for details on how to configure the clock source of the GCLK_MAIN clock.

Refer to the Oscillators Controller (OSCCTRL) description for details on how to configure the clock source of the CLK_DFLLULP clock.

Related Links

18. GCLK - Generic Clock Controller

19.6.2.4 Selecting the Synchronous Clock Division Ratio

The main clock CLK_MAIN feeds an 8-bit prescaler, which can be used to generate the synchronous clocks. By default, the synchronous clocks run on the undivided main clock. The user can select a prescaler division for the CPU clock domain by writing the Division (DIV) bits in the CPU Clock Division register CPUDIV, resulting in a CPU clock domain frequency determined by this equation:

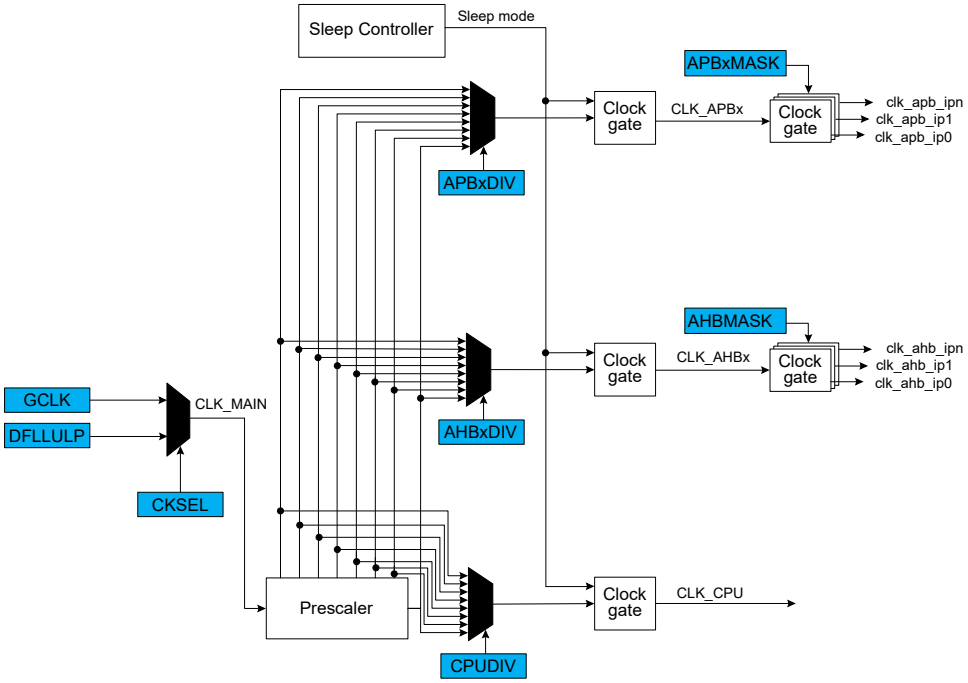
$$f_{CPU} = \frac{f_{main}}{CPUDIV}$$

If the application attempts to write forbidden values in CPUDIV register, registers are written but these bad values are not used and a violation is reported to the PAC module.

Division bits (DIV) can be written without halting or disabling peripheral modules. Writing DIV bits allows a new clock setting to be written to all synchronous clocks belonging to the corresponding clock domain at the same time.

Downloaded from Arrow.com.

Figure 19-2. Synchronous Clock Selection and Prescaler



Related Links

15. PAC - Peripheral Access Controller

19.6.2.5 Clock Ready Flag

There is a slight delay between writing to CPUDIV until the new clock settings become effective.

During this interval, the Clock Ready flag in the Interrupt Flag Status and Clear register (INTFLAG.CKRDY) will return zero when read. If CKRDY in the INTENSET register is set to '1', the Clock Ready interrupt will be triggered when the new clock setting is effective. The clock settings (CLKCFG) must not be re-written while INTFLAG. CKRDY reads '0'. The system may become unstable or hang, and a violation is reported to the PAC module.

Related Links

15. PAC - Peripheral Access Controller

19.6.2.6 Peripheral Clock Masking

It is possible to disable/enable the AHB or APB clock for a peripheral by writing the corresponding bit in the Clock Mask registers (APBxMASK) to '0'/'1'. The default state of the peripheral clocks is shown here.

Table 19-1. Peripheral Clock Default State

CPU Clock Domain	
Peripheral Clock	**Default State**
CLK_AC_APB	Enabled
CLK_ADC_APB	Enabled

Downloaded from Arrow.com.

CPU Clock Domain	
Peripheral Clock	**Default State**
CLK_BRIDGE_A_AHB	Enabled
CLK_BRIDGE_B_AHB	Enabled
CLK_BRIDGE_C_AHB	Enabled
CLK_CCL_APB	Enabled
CLK_DAC_APB	Enabled
CLK_DMAC_AHB	Enabled
CLK_DSU_AHB	Enabled
CLK_DSU_APB	Enabled
CLK_EIC_APB	Enabled
CLK_EVSYS_APB	Enabled
CLK_FREQM_APB	Enabled
CLK_GCLK_APB	Enabled
CLK_HMATRIXHS_AHB	Enabled
CLK_MCLK_APB	Enabled
CLK_NVMCTRL_AHB	Enabled
CLK_NVMCTRL_APB	Enabled
CLK_OPAMP_APB	Enabled
CLK_OSCCTRL_APB	Enabled
CLK_OSC32CTRL_APB	Enabled
CLK_PAC_AHB	Enabled
CLK_PAC_APB	Enabled
CLK_PORT_APB	Enabled
CLK_PM_APB	Enabled
CLK_PTC_APB	Enabled
CLK_RSTC_APB	Enabled
CLK_RTC_APB	Enabled
CLK_SERCOM0_APB	Enabled
CLK_SERCOM1_APB	Enabled
CLK_SERCOM2_APB(1)	Enabled
CLK_SUPC_APB	Enabled
CLK_TC0_APB	Enabled
CLK_TC1_APB	Enabled
CLK_TC2_APB	Enabled
CLK_TRAM_AHB	Enabled
CLK_WDT_APB	Enabled

When the APB clock is not provided to a module, its registers cannot be read or written. The module can be re-enabled later by writing the corresponding mask bit to '1'.

A module may be connected to several clock domains (for instance, AHB and APB), in which case it will have several mask bits.

Downloaded from Arrow.com.

Note that clocks should only be switched off if it is certain that the module will not be used: Switching off the clock for the NVM Controller (NVMCTRL) will cause a problem if the CPU needs to read from the Flash Memory. Switching off the clock to the MCLK module (which contains the mask registers) or the corresponding APBx bridge, will make it impossible to write the mask registers again. In this case, they can only be re-enabled by a system reset.

19.6.3 DMA Operation

Not applicable.

19.6.4 Interrupts

The peripheral has the following interrupt sources:

- Clock Ready (CKRDY): indicates that CPU clocks are ready. This interrupt is a synchronous wake-up source.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be enabled individually by writing a '1' to the corresponding enabling bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding clearing bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled or the peripheral is reset. An interrupt flag is cleared by writing a '1' to the corresponding bit in the INTFLAG register. Each peripheral can have one interrupt request line per interrupt source or one common interrupt request line for all the interrupt sources.If the peripheral has one common interrupt request line for all the interrupt sources, the user must read the INTFLAG register to determine which interrupt condition is present.

Related Links

22. PM – Power Manager

22.6.3.3 Sleep Mode Controller

19.6.5 Events

Not applicable.

19.6.6 Sleep Mode Operation

In all IDLE sleep modes, the MCLK is still running on the selected main clock.

In STANDBY sleep mode, the MCLK is frozen if no synchronous clock is required.

Downloaded from Arrow.com.

19.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0						CKSEL		
0x01	INTENCLR	7:0								CKRDY
0x02	INTENSET	7:0								CKRDY
0x03	INTFLAG	7:0								CKRDY
0x04	Reserved									
0x05	CPUDIV	7:0	CPUDIV[7:0]							
0x06 ... 0x0F	Reserved									
0x10	AHBMASK	7:0	NVMCTRL	PAC	Reserved	DSU	DMAC	APBC	APBB	APBA
		15:8				TRAM	Reserved	Reserved	Reserved	Reserved
		23:16								
		31:24								
0x14	APBAMASK	7:0	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM	PAC
		15:8		Reserved	AC	PORT	FREQM	EIC	RTC	WDT
		23:16								
		31:24								
0x18	APBBMASK	7:0				HMATRIXHS		NVMCTRL	DSU	IDAU
		15:8								
		23:16								
		31:24								
0x1C	APBCMASK	7:0	ADC	TC2	TC1	TC0	SERCOM2	SERCOM1	SERCOM0	EVSYS
		15:8				OPAMP	CCL	TRNG	PTC	DAC
		23:16								
		31:24								

19.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers can be write-protected optionally by the Peripheral Access Controller (PAC). This is denoted by the property "PAC Write-Protection" in each individual register description. Refer to the 19.5.8 Register Access Protection for details.

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Downloaded from Arrow.com.

Refer to *Peripherals Security Attribution* for more information.

Downloaded from Arrow.com.

19.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
						CKSEL		
Access						R/W		
Reset						0		

Bit 2 – CKSEL Main Clock Select

Value	Description
0	The GCLKMAIN clock is selected for the main clock.
1	The DFLLULP clock is selected for the main clock.

Downloaded from Arrow.com.

19.8.2 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x01
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Bit	7	6	5	4	3	2	1	0
								CKRDY
Access								R/W
Reset								0

Bit 0 – CKRDY Clock Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Clock Ready Interrupt Enable bit and the corresponding interrupt request.

Value	Description
0	The Clock Ready interrupt is enabled and will generate an interrupt request when the Clock Ready Interrupt Flag is set.
1	The Clock Ready interrupt is disabled.

Downloaded from Arrow.com.

19.8.3 Interrupt Enable Set

Name: INTENSET
Offset: 0x02
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Bit	7	6	5	4	3	2	1	0
								CKRDY
Access								R/W
Reset								0

Bit 0 – CKRDY Clock Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Clock Ready Interrupt Enable bit and enable the Clock Ready interrupt.

Value	Description
0	The Clock Ready interrupt is disabled.
1	The Clock Ready interrupt is enabled.

Downloaded from Arrow.com.

19.8.4 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x03
Reset: 0x01
Property: –

Bit	7	6	5	4	3	2	1	0
								CKRDY
Access								R/W
Reset								1

Bit 0 – CKRDY Clock Ready
This flag is cleared by writing a '1' to the flag.

This flag is set when the synchronous CPU, APBx, and AHBx clocks have frequencies as indicated in the CLKCFG registers and will generate an interrupt if INTENCLR/SET.CKRDY is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Clock Ready interrupt flag.

Downloaded from Arrow.com.

19.8.5 CPU Clock Division

Name: CPUDIV
Offset: 0x05
Reset: 0x01
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	CPUDIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bits 7:0 – CPUDIV[7:0] CPU Clock Division Factor
These bits define the division ratio of the main clock prescaler related to the CPU clock domain.

Frequencies must never exceed the specified maximum frequency for each clock domain.

Value	Name	Description
0x01	DIV1	Divide by 1
0x02	DIV2	Divide by 2
0x04	DIV4	Divide by 4
0x08	DIV8	Divide by 8
0x10	DIV16	Divide by 16
0x20	DIV32	Divide by 32
0x40	DIV64	Divide by 64
0x80	DIV128	Divide by 128
others	-	Reserved

Downloaded from Arrow.com.

19.8.6 AHB Mask

Name: AHBMASK
Offset: 0x10
Reset: 0x000001FFF
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				TRAM	Reserved	Reserved	Reserved	Reserved
Access				R/W	R/W	R/W	R/W	R/W
Reset				1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	NVMCTRL	PAC	Reserved	DSU	DMAC	APBC	APBB	APBA
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit 12 – TRAM TRAM AHB Clock Enable

Value	Description
0	The AHB clock for the TRAM is stopped
1	The AHB clock for the TRAM is enabled

Bit 11 – Reserved Must Be Set to 1
Bit 11 must always be set to ‘1’ when programming the AHBMASK register.

Bit 10 – Reserved Must Be Set to 1
Bit 10 must always be set to ‘1’ when programming the AHBMASK register.

Bit 9 – Reserved Must Be Set to 1
Bit 9 must always be set to ‘1’ when programming the AHBMASK register.

Bit 8 – Reserved Must Be Set to 1
Bit 8 must always be set to ‘1’ when programming the AHBMASK register.

Bit 7 – NVMCTRL NVMCTRL AHB Clock Enable

Downloaded from Arrow.com.

Value	Description
0	The AHB clock for the NVMCTRL is stopped
1	The AHB clock for the NVMCTRL is enabled

Bit 6 – PAC PAC AHB Clock Enable

Value	Description
0	The AHB clock for the PAC is stopped.
1	The AHB clock for the PAC is enabled.

Bit 5 – Reserved Must Be Set to 1
Bit 5 must always be set to '1' when programming the AHBMASK register.

Bit 4 – DSU DSU AHB Clock Enable

Value	Description
0	The AHB clock for the DSU is stopped.
1	The AHB clock for the DSU is enabled.

Bit 3 – DMAC DMAC AHB Clock Enable

Value	Description
0	The AHB clock for the DMAC is stopped.
1	The AHB clock for the DMAC is enabled.

Bit 2 – APBC APBC AHB Clock Enable

Value	Description
0	The AHB clock for the APBC is stopped.
1	The AHB clock for the APBC is enabled

Bit 1 – APBB APBB AHB Clock Enable

Value	Description
0	The AHB clock for the APBB is stopped.
1	The AHB clock for the APBB is enabled.

Bit 0 – APBA APBA AHB Clock Enable

Value	Description
0	The AHB clock for the APBA is stopped.
1	The AHB clock for the APBA is enabled.

Downloaded from Arrow.com.

19.8.7 APBA Mask

Name: APBAMASK
Offset: 0x14
Reset: 0x000007FFF
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		Reserved	AC	PORT	FREQM	EIC	RTC	WDT
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM	PAC
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit 14 – Reserved For future use
Reserved bits are unused and reserved for future use. For compatibility with future devices, always write reserved bits to their reset value. If no reset value is given, write 0.

Bit 13 – AC AC APBA Clock Enable

Value	Description
0	The APBA clock for the AC is stopped.
1	The APBA clock for the AC is enabled.

Bit 12 – PORT PORT APBA Clock Enable

Value	Description
0	The APBA clock for the PORT is stopped.
1	The APBA clock for the PORT is enabled.

Bit 11 – FREQM FREQM APBA Clock Enable

Value	Description
0	The APBA clock for the FREQM is stopped.
1	The APBA clock for the FREQM is enabled.

Downloaded from Arrow.com.

Bit 10 – EIC EIC APBA Clock Enable

Value	Description
0	The APBA clock for the EIC is stopped.
1	The APBA clock for the EIC is enabled.

Bit 9 – RTC RTC APBA Clock Enable

Value	Description
0	The APBA clock for the RTC is stopped.
1	The APBA clock for the RTC is enabled.

Bit 8 – WDT WDT APBA Clock Enable

Value	Description
0	The APBA clock for the WDT is stopped.
1	The APBA clock for the WDT is enabled.

Bit 7 – GCLK GCLK APBA Clock Enable

Value	Description
0	The APBA clock for the GCLK is stopped.
1	The APBA clock for the GCLK is enabled.

Bit 6 – SUPC SUPC APBA Clock Enable

Value	Description
0	The APBA clock for the SUPC is stopped.
1	The APBA clock for the SUPC is enabled.

Bit 5 – OSC32KCTRL OSC32KCTRL APBA Clock Enable

Value	Description
0	The APBA clock for the OSC32KCTRL is stopped.
1	The APBA clock for the OSC32KCTRL is enabled.

Bit 4 – OSCCTRL OSCCTRL APBA Clock Enable

Value	Description
0	The APBA clock for the OSCCTRL is stopped.
1	The APBA clock for the OSCCTRL is enabled.

Bit 3 – RSTC RSTC APBA Clock Enable

Value	Description
0	The APBA clock for the RSTC is stopped.
1	The APBA clock for the RSTC is enabled.

Bit 2 – MCLK MCLK APBA Clock Enable

Value	Description
0	The APBA clock for the MCLK is stopped.
1	The APBA clock for the MCLK is enabled.

Downloaded from Arrow.com.

Bit 1 – PM PM APBA Clock Enable

Value	Description
0	The APBA clock for the PM is stopped.
1	The APBA clock for the PM is enabled.

Bit 0 – PAC PAC APBA Clock Enable

Value	Description
0	The APBA clock for the PAC is stopped.
1	The APBA clock for the PAC is enabled.

Downloaded from Arrow.com.

19.8.8 APBB Mask

Name: APBBMASK
Offset: 0x18
Reset: 0x00000017
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				HMATRIXHS		NVMCTRL	DSU	IDAU
Access				R/W		R/W	R/W	R/W
Reset				1		1	1	1

Bit 4 – HMATRIXHS HMATRIXHS APBB Clock Enable

Value	Description
0	The APBB clock for the HMATRIXHS is stopped
1	The APBB clock for the HMATRIXHS is enabled

Bit 2 – NVMCTRL NVMCTRL APBB Clock Enable

Value	Description
0	The APBB clock for the NVMCTRL is stopped
1	The APBB clock for the NVMCTRL is enabled

Bit 1 – DSU DSU APBB Clock Enable

Value	Description
0	The APBB clock for the DSU is stopped
1	The APBB clock for the DSU is enabled

Bit 0 – IDAU IDAU APBB Clock Enable

Value	Description
0	The APBB clock for the IDAU is stopped
1	The APBB clock for the IDAU is enabled

Downloaded from Arrow.com.

19.8.9 APBC Mask

Name: APBCMASK
Offset: 0x1C
Reset: 0x00001FFF for 32-pin packages / 0x00001FF7 for 24-pin packages
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				OPAMP	CCL	TRNG	PTC	DAC
Access				R/W	R	R	R	R
Reset				1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	ADC	TC2	TC1	TC0	SERCOM2	SERCOM1	SERCOM0	EVSYS
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit 12 – OPAMP OPAMP APBC Clock Enable

Value	Description
0	The APBC clock for the OPAMP is stopped.
1	The APBC clock for the OPAMP is enabled.

Bit 11 – CCL CCL APBC Mask Clock Enable

Value	Description
0	The APBC clock for the CCL is stopped.
1	The APBC clock for the CCL is enabled.

Bit 10 – TRNG TRNG APBC Mask Clock Enable

Value	Description
0	The APBC clock for the TRNG is stopped.
1	The APBC clock for the TRNG is enabled.

Bit 9 – PTC PTC APBC Mask Clock Enable

Value	Description
0	The APBC clock for the PTC is stopped.
1	The APBC clock for the PTC is enabled.

Downloaded from Arrow.com.

Bit 8 – DAC DAC APBC Mask Clock Enable

Value	Description
0	The APBC clock for the DAC is stopped.
1	The APBC clock for the DAC is enabled.

Bit 7 – ADC ADC APBC Mask Clock Enable

Value	Description
0	The APBC clock for the ADC is stopped.
1	The APBC clock for the ADC is enabled.

Bit 6 – TC2 TC2 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the TC2 is stopped.
1	The APBC clock for the TC2 is enabled.

Bit 5 – TC1 TC1 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the TC1 is stopped.
1	The APBC clock for the TC1 is enabled.

Bit 4 – TC0 TC0 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the TC0 is stopped.
1	The APBC clock for the TC0 is enabled.

Bit 3 – SERCOM2 SERCOM2 APBC Mask Clock Enable
SERCOM2 Peripheral Clock is disabled for all 24-pin packages as SERCOM2 is not present.

Value	Description
0	The APBC clock for the SERCOM2 is stopped.
1	The APBC clock for the SERCOM2 is enabled.

Bit 2 – SERCOM1 SERCOM1 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the SERCOM1 is stopped.
1	The APBC clock for the SERCOM1 is enabled.

Bit 1 – SERCOM0 SERCOM0 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the SERCOM0 is stopped.
1	The APBC clock for the SERCOM0 is enabled.

Bit 0 – EVSYS EVSYS APBC Clock Enable

Downloaded from Arrow.com.

Value	Description
0	The APBC clock for the EVSYS is stopped.
1	The APBC clock for the EVSYS is enabled.

Downloaded from Arrow.com.

20. FREQM – Frequency Meter

20.1 Overview

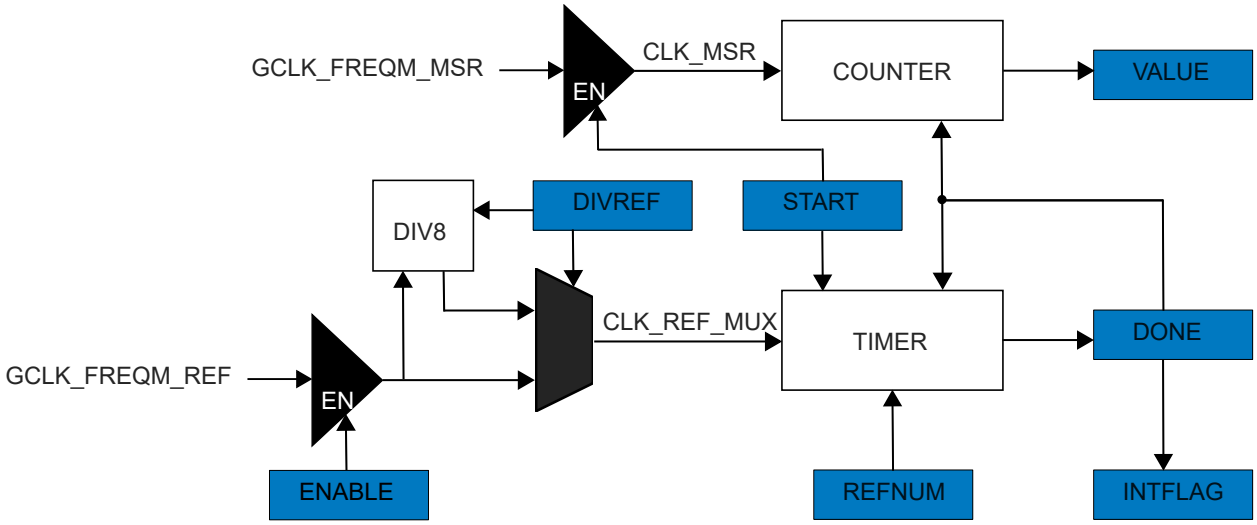
The Frequency Meter (FREQM) can be used to accurately measure the frequency of a clock by comparing it to a known reference clock.

20.2 Features

- Ratio can be measured with 24-bit accuracy
- Accurately measures the frequency of an input clock with respect to a reference clock
- Reference clock can be selected from the available GCLK_FREQM_REF sources
- Measured clock can be selected from the available GCLK_FREQM_MSR sources

20.3 Block Diagram

Figure 20-1. FREQM Block Diagram



20.4 Signal Description

Not applicable.

20.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

20.5.1 I/O Lines

The GCLK I/O lines (GCLK_IO[7:0]) can be used as measurement or reference clock sources. This requires the I/O pins to be configured.

Downloaded from Arrow.com.

20.5.2 Power Management

The FREQM will continue to operate in idle sleep mode where the selected source clock is running. The FREQM's interrupts can be used to wake up the device from idle sleep mode. Refer to the Power Manager chapter for details on the different sleep modes.

Related Links

22. PM – Power Manager

20.5.3 Clocks

The clock for the FREQM bus interface (CLK_APB_FREQM) is enabled and disabled by the Main Clock Controller, the default state of CLK_APB_FREQM can be found in Peripheral Clock Masking.

Two generic clocks are used by the FREQM: Reference Clock (GCLK_FREQM_REF) and Measurement Clock (GCLK_FREQM_MSR).

GCLK_FREQM_REF is required to clock the internal reference timer, which acts as the frequency reference.

GCLK_FREQM_MSR is required to clock a ripple counter for frequency measurement. These clocks must be configured and enabled in the generic clock controller before using the FREQM.

Related Links

19. MCLK – Main Clock

19.6.2.6 Peripheral Clock Masking

18. GCLK - Generic Clock Controller

20.5.4 DMA

Not applicable.

20.5.5 Interrupts

The interrupt request line is connected to the interrupt controller. Using FREQM interrupt requires the interrupt controller to be configured first.

20.5.6 Events

Not applicable

20.5.7 Debug Operation

When the CPU is halted in debug mode the FREQM continues its normal operation. The FREQM cannot be halted when the CPU is halted in debug mode. If the FREQM is configured in a way that requires it to be periodically serviced by the CPU, improper operation or data loss may result during debugging.

20.5.8 Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except the following registers:

- Control B register (CTRLB)
- Interrupt Flag Status and Clear register (INTFLAG)
- Status register (STATUS)

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Write-protection does not apply to accesses through an external debugger.

Downloaded from Arrow.com.

Related Links

15. PAC - Peripheral Access Controller

20.5.9 SAM L11 TrustZone Specific Register Access Protection

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

20.6 Functional Description

20.6.1 Principle of Operation

FREQM counts the number of periods of the measured clock (GCLK_FREQM_MSR) with respect to the reference clock (GCLK_FREQM_REF). The measurement is done for a period of REFNUM/f_{CLK_REF} and stored in the Value register (VALUE.VALUE). REFNUM is the number of Reference clock cycles selected in the Configuration A register (CFGA.REFNUM).

The frequency of the measured clock, f_{CLK_MSR}, is calculated by

$$f_{CLK_MSR} = \left(\frac{VALUE}{REFNUM}\right) f_{CLK_REF}$$

20.6.2 Basic Operation

20.6.2.1 Initialization

Before enabling FREQM, the device and peripheral must be configured:

- Each of the generic clocks (GCLK_FREQM_REF and GCLK_FREQM_MSR) must be configured and enabled.
-

Important: The reference clock must be slower than the measurement clock.

- Write the number of Reference clock cycles for which the measurement is to be done in the Configuration A register (CFGA.REFNUM). This must be a non-zero number.

The following register is enable-protected, meaning that it can only be written when the FREQM is disabled (CTRLA.ENABLE=0):

- Configuration A register (CFGA)

Enable-protection is denoted by the "Enable-Protected" property in the register description.

Related Links

18. GCLK - Generic Clock Controller

Downloaded from Arrow.com.

20.6.2.2 Enabling, Disabling and Resetting

The FREQM is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The peripheral is disabled by writing CTRLA.ENABLE=0.

The FREQM is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). On software reset, all registers in the FREQM will be reset to their initial state, and the FREQM will be disabled.

Then ENABLE and SWRST bits are write-synchronized.

Related Links

20.6.7 Synchronization

20.6.2.3 Measurement

In the Configuration A register, the Number of Reference Clock Cycles field (CFGA.REFNUM) selects the duration of the measurement. The measurement is given in number of GCLK_FREQM_REF periods.
Note: The REFNUM field must be written before the FREQM is enabled.

After the FREQM is enabled, writing a '1' to the START bit in the Control B register (CTRLB.START) starts the measurement. The BUSY bit in Status register (STATUS.BUSY) is set when the measurement starts, and cleared when the measurement is complete.

There is also an interrupt request for Measurement Done: When the Measurement Done bit in Interrupt Enable Set register (INTENSET.DONE) is '1' and a measurement is finished, the Measurement Done bit in the Interrupt Flag Status and Clear register (INTFLAG.DONE) will be set and an interrupt request is generated.

The result of the measurement can be read from the Value register (VALUE.VALUE). The frequency of the measured clock GCLK_FREQM_MSR is then:

$$f_{CLK_MSR} = \left(\frac{VALUE}{REFNUM}\right) f_{CLK_REF}$$

Note: In order to make sure the measurement result (VALUE.VALUE[23:0]) is valid, the overflow status (STATUS.OVF) should be checked.

In case an overflow condition occurred, indicated by the Overflow bit in the STATUS register (STATUS.OVF), either the number of reference clock cycles must be reduced (CFGA.REFNUM), or a faster reference clock must be configured. Once the configuration is adjusted, clear the overflow status by writing a '1' to STATUS.OVF. Then another measurement can be started by writing a '1' to CTRLB.START.

20.6.3 DMA Operation

Not applicable.

20.6.4 Interrupts

The FREQM has one interrupt source:

- DONE: A frequency measurement is done.

The interrupt flag in the Interrupt Flag Status and Clear (20.8.6 INTFLAG) register is set when the interrupt condition occurs. The interrupt can be enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (20.8.5 INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (20.8.4 INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the

Downloaded from Arrow.com.

FREQM is reset. See 20.8.6 INTFLAG for details on how to clear interrupt flags. All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. The user must read the 20.8.6 INTFLAG register to determine which interrupt condition is present.

This interrupt is a synchronous wake-up source.

Note that interrupts must be globally enabled for interrupt requests to be generated.

20.6.5 Events

Not applicable.

20.6.6 Sleep Mode Operation

The FREQM will continue to operate in idle sleep mode where the selected source clock is running. The FREQM's interrupts can be used to wake up the device from idle sleep mode.

For lowest chip power consumption in sleep modes, FREQM should be disabled before entering a sleep mode.

Related Links

22. PM – Power Manager

20.6.7 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits and registers are write-synchronized:

- Software Reset bit in Control A register (CTRLA.SWRST)
- Enable bit in Control A register (CTRLA.ENABLE)

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Downloaded from Arrow.com.

20.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0							ENABLE	SWRST
0x01	CTRLB	7:0								START
0x02	CFGA	7:0	REFNUM[7:0]							
		15:8	DIVREF							
0x04 ... 0x07	Reserved									
0x08	INTENCLR	7:0								DONE
0x09	INTENSET	7:0								DONE
0x0A	INTFLAG	7:0								DONE
0x0B	STATUS	7:0							OVF	BUSY
0x0C	SYNCBUSY	7:0							ENABLE	SWRST
		15:8								
		23:16								
		31:24								
0x10	VALUE	7:0	VALUE[7:0]							
		15:8	VALUE[15:8]							
		23:16	VALUE[23:16]							
		31:24								

20.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description.

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

Downloaded from Arrow.com.

20.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Bit	7	6	5	4	3	2	1	0
							ENABLE	SWRST
Access							R/W	R/W
Reset							0	0

Bit 1 – ENABLE Enable
Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the ENABLE bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

This bit is not enable-protected.

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled.

Bit 0 – SWRST Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the FREQM to their initial state, and the FREQM will be disabled. Writing a '1' to this bit will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Due to synchronization there is a delay from writing CTRLA.SWRST until the Reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the Reset is complete.

This bit is not enable-protected.

Value	Description
0	There is no ongoing Reset operation.
1	The Reset operation is ongoing.

Downloaded from Arrow.com.

20.8.2 Control B

Name: CTRLB
Offset: 0x01
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
								START
Access								W
Reset								0

Bit 0 – START Start Measurement

Value	Description
0	Writing a '0' has no effect.
1	Writing a '1' starts a measurement.

Downloaded from Arrow.com.

20.8.3 Configuration A

Name: CFGA
Offset: 0x02
Reset: 0x0000
Property: PAC Write-Protection, Enable-protected

Bit	15	14	13	12	11	10	9	8
	DIVREF							
Access	R/W							
Reset	0							

Bit	7	6	5	4	3	2	1	0
	REFNUM[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – DIVREF Divide Reference Clock
Divides the reference clock by 8

Value	Description
0	The reference clock is divided by 1.
1	The reference clock is divided by 8.

Bits 7:0 – REFNUM[7:0] Number of Reference Clock Cycles
Selects the duration of a measurement in number of CLK_FREQM_REF cycles. This must be a non-zero value, i.e. 0x01 (one cycle) to 0xFF (255 cycles).

Downloaded from Arrow.com.

20.8.4 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x08
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DONE
Access								R/W
Reset								0

Bit 0 – DONE Measurement Done Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Measurement Done Interrupt Enable bit, which disables the Measurement Done interrupt.

Value	Description
0	The Measurement Done interrupt is disabled.
1	The Measurement Done interrupt is enabled.

Downloaded from Arrow.com.

20.8.5 Interrupt Enable Set

Name: INTENSET
Offset: 0x09
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DONE
Access								R/W
Reset								0

Bit 0 – DONE Measurement Done Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Measurement Done Interrupt Enable bit, which enables the Measurement Done interrupt.

Value	Description
0	The Measurement Done interrupt is disabled.
1	The Measurement Done interrupt is enabled.

Downloaded from Arrow.com.

20.8.6 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x0A
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
								DONE
Access								R/W
Reset								0

Bit 0 – DONE Mesurement Done
This flag is cleared by writing a '1' to it.

This flag is set when the STATUS.BUSY bit has a one-to-zero transition.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the DONE interrupt flag.

Downloaded from Arrow.com.

20.8.7 Status

Name: STATUS
Offset: 0x0B
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
							OVF	BUSY
Access							R/W	R
Reset							0	0

Bit 1 – OVF Sticky Count Value Overflow
This bit is cleared by writing a '1' to it.

This bit is set when an overflow condition occurs to the value counter.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the OVF status.

Bit 0 – BUSY FREQM Status

Value	Description
0	No ongoing frequency measurement.
1	Frequency measurement is ongoing.

Downloaded from Arrow.com.

20.8.8 Synchronization Busy

Name: SYNCBUSY
Offset: 0x0C
Reset: 0x00000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							ENABLE	SWRST
Access							R	R
Reset							0	0

Bit 1 – ENABLE Enable
This bit is cleared when the synchronization of CTRLA.ENABLE is complete.

This bit is set when the synchronization of CTRLA.ENABLE is started.

Bit 0 – SWRST Synchronization Busy
This bit is cleared when the synchronization of CTRLA.SWRST is complete.

This bit is set when the synchronization of CTRLA.SWRST is started.

Downloaded from Arrow.com.

20.8.9 Value

Name: VALUE
Offset: 0x10
Reset: 0x00000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	VALUE[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	VALUE[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	VALUE[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – VALUE[23:0] Measurement Value
Result from measurement.

Downloaded from Arrow.com.

21. RSTC – Reset Controller

21.1 Overview

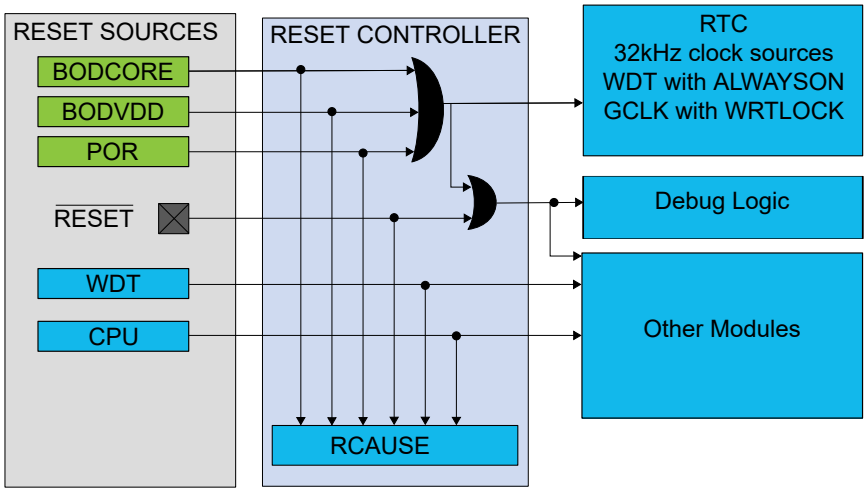
The Reset Controller (RSTC) manages the reset of the microcontroller. It issues a microcontroller reset, sets the device to its initial state and allows the reset source to be identified by software.

21.2 Features

- Reset the microcontroller and set it to an initial state according to the reset source
- Reset cause register for reading the reset source from the application code
- Multiple reset sources
  - Power supply reset sources: POR, BOD12, BOD33
  - User reset sources: External reset ($\overline{\text{RESET}}$), Watchdog reset, and System Reset Request

21.3 Block Diagram

Figure 21-1. Reset System



21.4 Signal Description

Signal Name	Type	Description
$\overline{\text{RESET}}$	Digital input	External reset

One signal can be mapped on several pins.

21.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

Downloaded from Arrow.com.

21.5.1 I/O Lines

Not applicable.

21.5.2 Power Management

The Reset Controller module is always on.

21.5.3 Clocks

The RSTC bus clock (CLK_RSTC_APB) can be enabled and disabled in the Main Clock Controller.

Related Links

19. MCLK – Main Clock

19.6.2.6 Peripheral Clock Masking

21.5.4 DMA

Not applicable.

21.5.5 Interrupts

Not applicable.

21.5.6 Events

Not applicable.

21.5.7 Debug Operation

When the CPU is halted in debug mode, the RSTC continues normal operation.

21.5.8 Register Access Protection

All registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC).

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

Write-protection does not apply for accesses through an external debugger.

21.5.9 SAM L11 TrustZone Specific Register Access Protection

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

21.5.10 Analog Connections

Not applicable.

Downloaded from Arrow.com.

21.6 Functional Description

21.6.1 Principle of Operation

The Reset Controller collects the various Reset sources and generates Reset for the device.

21.6.2 Basic Operation

21.6.2.1 Initialization

After a power-on Reset, the RSTC is enabled and the Reset Cause (RCAUSE) register indicates the POR source.

21.6.2.2 Enabling, Disabling, and Resetting

The RSTC module is always enabled.

21.6.2.3 Reset Causes and Effects

The latest Reset cause is available in RCAUSE register, and can be read during the application boot sequence in order to determine proper action.

These are the groups of Reset sources:

- Power supply Reset: Resets caused by an electrical issue. It covers POR and BODs Resets
- User Reset: Resets caused by the application. It covers external Resets, system Reset requests and watchdog Resets

The following table lists the parts of the device that are reset, depending on the Reset type.

Table 21-1. Effects of the Different Reset Causes

	Power Supply Reset	User Reset	
	POR, BOD33, BOD12	External Reset	WDT Reset, System Reset Request
RTC, OSC32KCTRL, RSTC	Y	N	N
GCLK with WRTLOCK	Y	N	N
Debug logic	Y	Y	N
Others	Y	Y	Y

The external Reset is generated when pulling the $\overline{\text{RESET}}$ pin low.

The POR, BOD12, and BOD33 Reset sources are generated by their corresponding module in the Supply Controller Interface (SUPC).

The WDT Reset is generated by the Watchdog Timer.

The System Reset Request is a Reset generated by the CPU when asserting the SYSRESETREQ bit located in the Reset Control register of the CPU (for details refer to the ARM® Cortex™ Technical Reference Manual on http://www.arm.com).

Note: Refer to the Timing Characteristics section of the Electrical Characteristics chapter.

Related Links

26. WDT – Watchdog Timer
25. SUPC – Supply Controller

Downloaded from Arrow.com.

46.14.1 External Reset Pin

21.6.3 Additional Features
Not applicable.

21.6.4 DMA Operation
Not applicable.

21.6.5 Interrupts
Not applicable.

21.6.6 Events
Not applicable.

21.6.7 Sleep Mode Operation
The RSTC module is active in all sleep modes.

Downloaded from Arrow.com.

21.7 Register Summary

Offset	Name	Bit Pos.								
0x00	RCAUSE	7:0		SYST	WDT	EXT		BOD33	BOD12	POR

21.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to 21.5.8 Register Access Protection.

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

Downloaded from Arrow.com.

21.8.1 Reset Cause

Name: RCAUSE
Offset: 0x00
Property: –

When a Reset occurs, the bit corresponding to the Reset source is set to '1' and all other bits are written to '0'.

Bit	7	6	5	4	3	2	1	0
		SYST	WDT	EXT		BOD33	BOD12	POR
Access		R	R	R		R	R	R
Reset		x	x	x		x	x	x

Bit 6 – SYST System Reset Request
This bit is set if a System Reset Request has occurred. Refer to the Cortex processor documentation for more details.

Bit 5 – WDT Watchdog Reset
This bit is set if a Watchdog Timer Reset has occurred.

Bit 4 – EXT External Reset
This bit is set if an external Reset has occurred.

Bit 2 – BOD33 Brown Out 33 Detector Reset
This bit is set if a BOD33 Reset has occurred.

Bit 1 – BOD12 Brown Out 12 Detector Reset
This bit is set if a BOD12 Reset has occurred.

Bit 0 – POR Power On Reset
This bit is set if a POR has occurred.

Downloaded from Arrow.com.

22. PM – Power Manager

22.1 Overview

The Power Manager (PM) controls the sleep modes and the power domain gating of the device.

Various sleep modes are provided in order to fit power consumption requirements. This enables the PM to stop unused modules in order to save power. In active mode, the CPU is executing application code. When the device enters a sleep mode, program execution is stopped and some modules and clock domains are automatically switched off by the PM according to the sleep mode. The application code decides which sleep mode to enter and when. Interrupts from enabled peripherals and all enabled reset sources can restore the device from a sleep mode to active mode.

Performance level technique consists of adjusting the regulator output voltage to reduce power consumption. The user can select on the fly the performance level configuration which best suits the application.

The power domain gating technique enables the PM to turn off unused power domain supplies individually, while keeping others powered up. Based on activity monitoring, power domain gating is managed automatically by hardware without software intervention. This technique is transparent for the application while minimizing the static consumption. The user can also manually control which power domains will be turned on and off in standby sleep mode.

The internal state of the logic is retained (retention state) allowing the application context to be kept in non-active states.

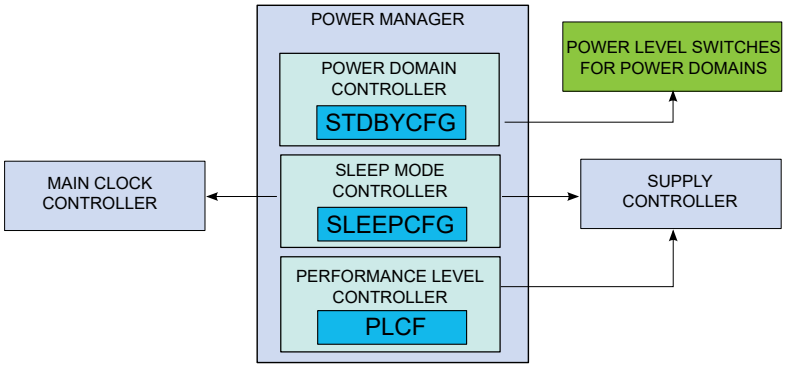
22.2 Features

- Power management control
  - Sleep modes: Idle, Standby, and Off
  - Performance levels: PL0 and PL2
  - SleepWalking available in Standby mode.
  - Full retention state in Standby mode
- Power Domain Control
  - Standby Sleep Mode with static power gating
  - SleepWalking extension to power gating
  - SRAM sub-blocks retention in Standby mode

Downloaded from Arrow.com.

22.3 Block Diagram

Figure 22-1. PM Block Diagram



22.4 Signal Description

Not applicable.

22.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

22.5.1 I/O Lines

Not applicable.

22.5.2 Clocks

The PM bus clock (CLK_PM_APB) can be enabled and disabled in the Main Clock module. If this clock is disabled, it can only be re-enabled by a system reset.

22.5.3 DMA

Not applicable.

22.5.4 Interrupts

The interrupt request line is connected to the interrupt controller. Using the PM interrupt requires the interrupt controller to be configured first.

22.5.5 Events

Not applicable.

22.5.6 Debug Operation

When the CPU is halted in debug mode, the PM continues normal operation. If standby sleep mode is requested by the system while in debug mode, the power domains are not turned off. As a consequence, power measurements while in debug mode are not relevant.

If OFF sleep mode is requested by the system while in debug mode, the core domains are kept on, and the debug modules are kept running to allow the debugger to access internal registers. When exiting the OFF mode upon a reset condition, the core domains are reset except the debug logic, allowing users to keep using their current debug session.

Downloaded from Arrow.com.

Hot plugging in standby mode is supported except if the power domain PDSW is in retention state.

Cold plugging in OFF mode is supported as long as the reset duration is superior to (Tmin).

22.5.7 Register Access Protection

Registers with write-access can be write-protected optionally by the peripheral access controller (PAC).

PAC Write-Protection is not available for the following registers:

- Interrupt Flag register (INTFLAG).

Optional PAC Write-Protection is denoted by the "PAC Write-Protection" property in each individual register description.

Write-protection does not apply to accesses through an external debugger.

Related Links

15. PAC - Peripheral Access Controller

22.5.8 SAM L11 TrustZone Specific Register Access Protection

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

22.5.9 Analog Connections

Not applicable.

22.6 Functional Description

22.6.1 Terminology

The following is a list of terms used to describe the Power Managemement features of this microcontroller.

22.6.1.1 Performance Levels

To help balance between performance and power consumption, the device has two performance levels. Each of the performance levels has a maximum operating frequency and a corresponding maximum consumption in µA/MHz.

It is the application's responsibility to configure the appropriate PL depending on the application activity level. When the application selects a new PL, the voltage applied on the full logic area moves from one value to another. This voltage scaling technique allows to reduce the active power consumption while decreasing the maximum frequency of the device.

22.6.1.1.1 PL0

Performance Level 0 (PL0) provides the maximum energy efficiency configuration.

Refer to the *Electrical Characteristics* chapters for details on energy consumption and maximum operating frequency.

Downloaded from Arrow.com.

22.6.1.1.2 PL2

Performance Level 2 (PL2) provides the maximum operating frequency.

Refer to the *Electrical Characteristics* chapters for details on energy consumption and maximum operating frequency.

22.6.1.2 Power Domains

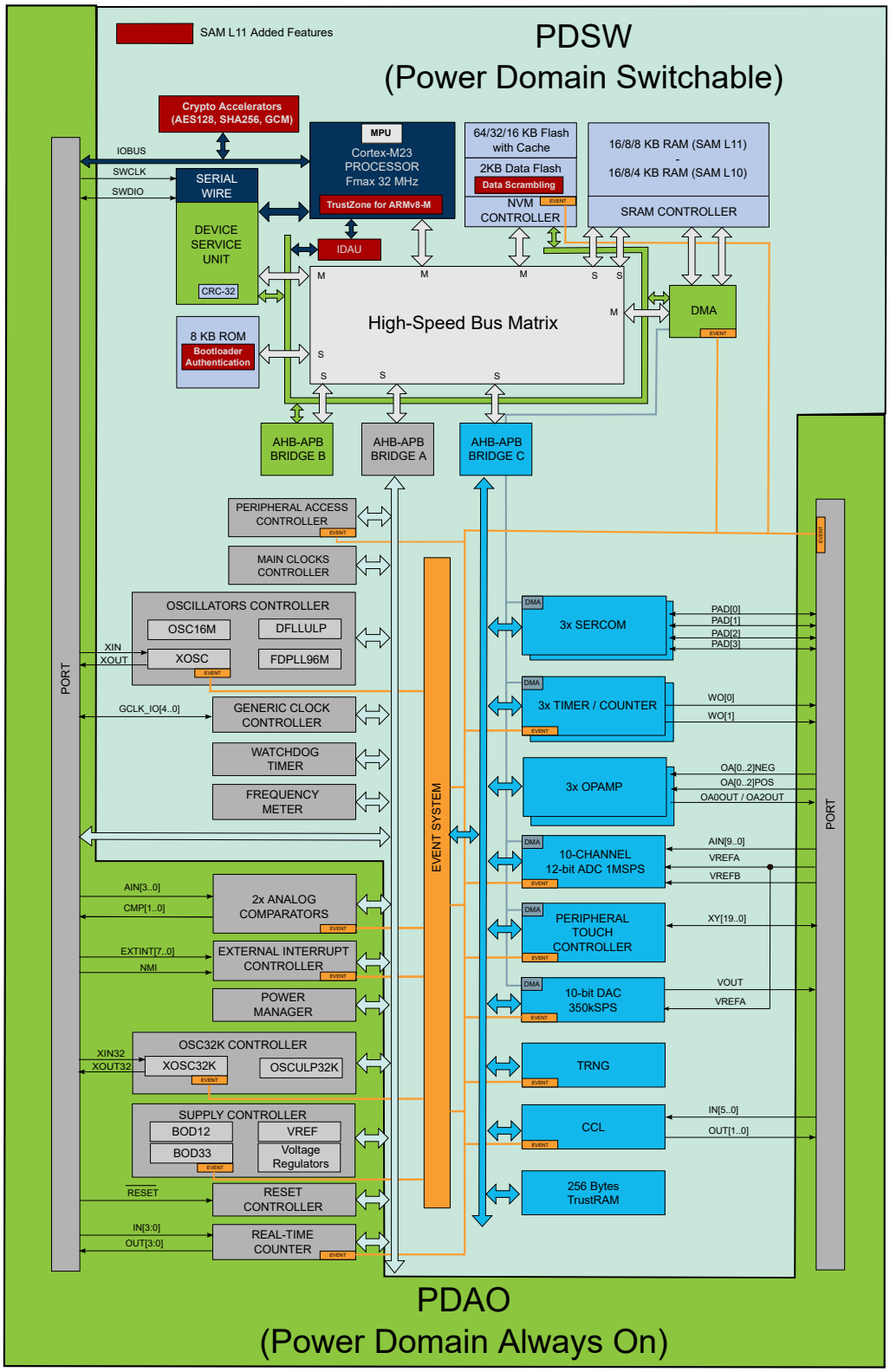
In addition to the supply domains, such as VDDIO and VDDANA, the device provides these power domains:

- PDSW
- PDAO

The PDSW is a "switchable power domain". In standby sleep mode, it can be turned off to save leakage consumption according to user configuration.

Downloaded from Arrow.com.

Figure 22-2. Power Domain Partitioning



Downloaded from Arrow.com.

22.6.1.2.1 PDSW - Power Domain Switchable

PDSW is the switchable power domain. It contains the Event System, Generic Clock Controller, Main Clock Controller, Oscillator Controller, Non-Volatile Memory Controller, DMA Controller, the Device Service Unit, and the ARM core. PDSW also contains a number of peripherals that allow the device to wake up from an interrupt: SERCOM, Timer, ADC, DAC, OPAMP, CCL, PTC.

22.6.1.2.2 PDAO - Power Domain Always On

PDAO contains all controllers located in the always-on domain. It is powered when in Active, Idle, or Standby mode.

22.6.1.3 Sleep Modes

The device can be set in a sleep mode. In sleep mode, the CPU is stopped and the peripherals are either active or idle, according to the sleep mode depth:

- Idle sleep mode: The CPU is stopped. Synchronous clocks are stopped except when requested. The logic is retained.
- Standby sleep mode: The CPU is stopped as well as the peripherals. The logic is retained, and power domain gating can be used to reduce power consumption further.
- Off sleep mode: The entire device is powered off.

22.6.1.4 Power Domain States and Gating

In Standby sleep mode, the Power Domain Gating technique allows for selecting the state of PDSW power domain automatically (e.g. for executing sleepwalking tasks) or manually:

Active State	The power domain is powered according to the performance level
Retention State	The main voltage supply for the power domain is switched off, while maintaining a secondary low-power supply for sequential cells. The logic context is restored when waking up.

22.6.2 Principle of Operation

In active mode, all clock domains and power domains are active, allowing software execution and peripheral operation. The PM Sleep Mode Controller allows to save power by choosing between different sleep modes depending on application requirements, see 22.6.3.3 Sleep Mode Controller.

The PM Performance Level Controller allows to optimize either for low power consumption or high performance.

The PM Power Domain Controller allows to reduce the power consumption in standby mode even further.

22.6.3 Basic Operation

22.6.3.1 Initialization

After a Power-on Reset (POR), the PM is enabled, the device is in Active mode, the performance level is PL0 (the lowest power consumption) and all the power domains are in active state.

22.6.3.2 Enabling, Disabling and Resetting

The PM is always enabled and can not be reset.

22.6.3.3 Sleep Mode Controller

Sleep mode is entered by executing the Wait For Interrupt instruction (WFI). The Sleep Mode bits in the Sleep Configuration register (SLEEPCFG.SLEEPMODE) select the level of the sleep mode.

Note: A small latency happens between the store instruction and actual writing of the SLEEPCFG register due to bridges. Software must ensure that the SLEEPCFG register reads the desired value before issuing a WFI instruction.

Downloaded from Arrow.com.

Note: After power-up, the MAINVREG low power mode takes some time to stabilize. Once stabilized, the SUPC->STATUS.ULPVREFRDY bit is set. Before entering Standby, software must ensure that the SUPC->STATUS.ULPVREFRDY bit is set.

Table 22-1. Sleep Mode Entry and Exit Table

Mode	Mode Entry	Wake-Up Sources
IDLE	SLEEPCFG.SLEEPMODE = IDLE _n	Synchronous (2) (APB, AHB), asynchronous (1)
STANDBY	SLEEPCFG.SLEEPMODE = STANDBY	Synchronous(3), Asynchronous
OFF	SLEEPCFG.SLEEPMODE = OFF	External Reset

Note:

1. Asynchronous: interrupt generated on generic clock, external clock, or external event.
2. Synchronous: interrupt generated on the APB clock.
3. Synchronous interrupt only for peripherals configured to run in standby.

Note: The type of wake-up sources (synchronous or asynchronous) is given in each module interrupt section.

The sleep modes (idle, standby, and off) and their effect on the clocks activity, the regulator and the NVM state are described in the table and the sections below. Refer to Power Domain Controller for the power domain gating effect.

Table 22-2. Sleep Mode Overview

Mode	Main clock	CPU	AHBx and APBx clock	GCLK clocks	Oscillators		Regulator	NVM
					ONDEMAND = 0	ONDEMAND = 1		
Active	Run	Run	Run	Run(3)	Run	Run if requested	MAINVREG	active
IDLE	Run	Stop	Stop(1)	Run(3)	Run	Run if requested	MAINVREG	active
STANDBY	Stop(1)	Stop	Stop(1)	Stop(1)	Run if requested or RUNSTDBY=1	Run if requested	MAINVREG in low power mode	Ultra Low- power
OFF	Stop	Stop	Stop	OFF	OFF	OFF	OFF	OFF

Note:

1. Running if requested by peripheral during SleepWalking.
2. Running during SleepWalking.
3. Following On-Demand Clock Request principle.

22.6.3.3.1 IDLE Mode

IDLE mode allows power optimization with the fastest wake-up time.

The CPU is stopped, and peripherals are still working. As in Active mode, the AHBx and APBx clocks for peripheral are still provided if requested. As the main clock source is still running, wake-up time is very fast.

- Entering Idle mode: The Idle mode is entered by executing the WFI instruction. Additionally, if the SLEEPONEXIT bit in the Cortex System Control register (SCR) is set, the Idle mode will be entered when the CPU exits the lowest priority ISR (Interrupt Service Routine, refer to the ARM Cortex

Downloaded from Arrow.com.

documentation for details). This mechanism can be useful for applications that only require the processor to run when an interrupt occurs. Before entering the Idle mode, the user must select the Idle Sleep mode in the Sleep Configuration register (SLEEPCFG.SLEEPMODE=IDLE).

- Exiting Idle mode: The processor wakes the system up when it detects any non-masked interrupt with sufficient priority to cause exception entry. The system goes back to the Active mode. The CPU and affected modules are restarted.

GCLK clocks, regulators and RAM are not affected by the Idle Sleep mode and operate in normal mode.

22.6.3.3.2 STANDBY Mode

The Standby mode is the lowest power configuration while keeping the state of the logic and the content of the RAM.

In this mode, all clocks are stopped except those configured to be running sleepwalking tasks. The clocks can also be active on request or at all times, depending on their on-demand and run-in-standby settings. Either synchronous (CLK_APBx or CLK_AHBx) or generic (GCLK_x) clocks or both can be involved in sleepwalking tasks. This is the case when for example the SERCOM RUNSTDBY bit is written to '1'.

- Entering Standby mode: This mode is entered by executing the WFI instruction after writing the Sleep Mode bit in the Sleep Configuration register (SLEEPCFG.SLEEPMODE=STANDBY). The SLEEPONEXIT feature is also available as in Idle mode.
- Exiting Standby mode: Any peripheral able to generate an asynchronous interrupt can wake up the system. For example, a peripheral running on a GCLK clock can trigger an interrupt. When the enabled asynchronous wake-up event occurs and the system is woken up, the device will either execute the interrupt service routine or continue the normal program execution according to the Priority Mask Register (PRIMASK) configuration of the CPU.

Refer to 22.6.3.5 Power Domain Controller for the RAM state.

The regulator operates in Low-Power mode by default and switches automatically to the normal mode in case of a sleepwalking task requiring more power. It returns automatically to low power mode when the sleepwalking task is completed.

22.6.3.3.3 OFF Mode

In Off mode, the device is entirely powered-off.

- Entering Off mode: This mode is entered by selecting the Off mode in the Sleep Configuration register by writing the Sleep Mode bits (SLEEPCFG.SLEEPMODE=OFF), and subsequent execution of the WFI instruction.
- Exiting Off mode: This mode is left by pulling the $\overline{\text{RESET}}$ pin low, or when a power Reset is done.

22.6.3.4 Performance Level

The application can change the performance level on the fly writing to the by Performance Level Select bit in the Performance Level Configuration register (PLCFG.PLSEL).

When changing to a lower performance level, the bus frequency must be reduced before writing PLCFG.PLSEL in order to avoid exceeding the limit of the target performance level.

When changing to a higher performance level, the bus frequency can be increased only after the Performance Level Ready flag in the Interrupt Flag Status and Clear (INTFLAG.PLRDY) bit set to '1', indicating that the performance level transition is complete.

After a reset, the device starts in the lowest PL (lowest power consumption and lowest max frequency). The application can then switch to another PL at anytime without any stop in the code execution. As shown in Figure 22-3, performance level transition is possible only when the device is in active mode.

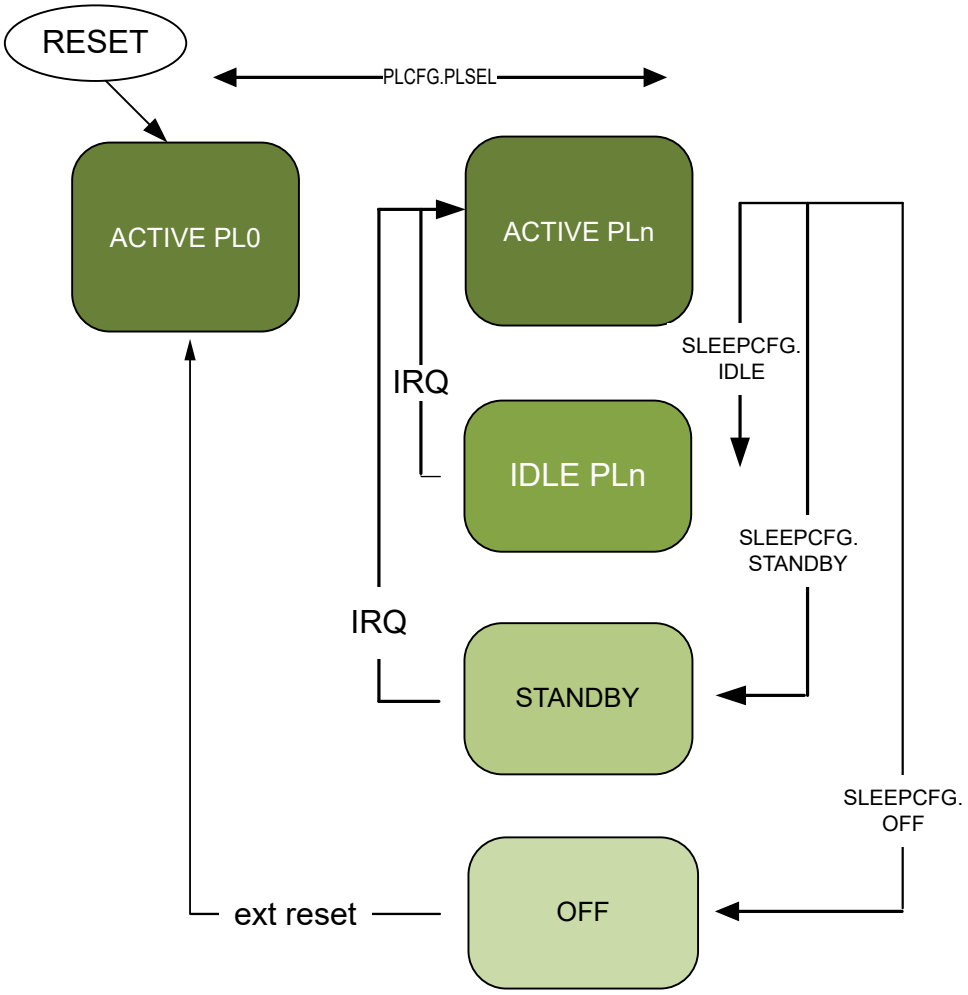
Downloaded from Arrow.com.

The Performance Level Disable bit in the Performance Level Configuration register (PLCFG.PLDIS) can be used to freeze the performance level to PL0. This disables the performance level hardware mechanism in order to reduce both the power consumption and the wake-up startup time from standby sleep mode.

Note: This bit PLCFG.PLDIS must be changed only when the current performance level is PL0.

Any attempt to modify this bit while the performance level is not PL0 is discarded and a violation is reported to the PAC module. Any attempt to change the performance level to PLn (with n>0) while PLCFG.PLDIS=1 is discarded and a violation is reported to the PAC module.

Figure 22-3. Sleep Modes and Performance Level Transitions



22.6.3.5 Power Domain Controller

The Power Domain Controller provides several ways of how power domains are handled while the device is in Standby mode or entering Standby mode:

- Default operation - all peripherals idle
  When entering Standby mode, the power domain PDSW is set in retention state. This allows for very low power consumption while retaining all the logic content of these power domains. When exiting Standby mode, all power domains are set back to active state.
- Default operation - Standby Sleep Mode with static power gating

Downloaded from Arrow.com.

Static Power Domain Gating is a technique that allows to automatically turn off the PDSW power domain supply when not used while keeping PDAO powered up.

- SleepWalking extension to power gating (SleepWalking with dynamic power gating)
  SleepWalking is the capability for a device in Standby Sleep mode, to temporarily wake-up clocks for a peripheral to perform a task without waking-up the CPU. The SleepWalking feature has been expanded to control power gating in addition to clock gating. The power domain PDSW can be automatically controlled (active or retention state) depending on peripheral requirements (PDCFG bit from the STDBYCFG register).

The static and dynamic power gating features are fully transparent for the user.

Table 22-3. Sleep Modes versus Power Domain States Overview

	Power Domain State	
Sleep Mode	**PDSW**	**PDAO**
Active	active	active
Idle	active	active
Standby - At least one peripheral from PDSW with RUNSTDBY = 1 OR PDCFG = 1	active (1)	active
Standby - No peripheral from PDSW with RUNSTDBY = 1	retention	active
Off	off	off

Note:

1. PDSW can be switched automatically in retention mode if the dynamic power gating feature is enabled.

22.6.3.6 Regulators, RAMs, and NVM State in Sleep Mode

By default, in Standby Sleep mode, the RAMs, NVM, and regulators are automatically set in Low-Power mode to reduce power consumption:

- The RAM is in Low-Power mode if its power domain is in retention or off state.
- Non-Volatile Memory - the NVM is located in the power domain PDSW. By default, the NVM is automatically set in low power mode in these conditions:
  - When the power domain PDSW is in retention or off state.
  - When the device is in Standby Sleep mode and the NVM is not accessed. This behavior can be changed by software by configuring the SLEEPPRM bit group of the CTRLB register in the NVMCTRL peripheral.
  - When the device is in Idle Sleep mode and the NVM is not accessed. This behavior can be changed by software by configuring the SLEEPPRM bit group of the CTRLB register in the NVMCTRL peripheral.
- Regulators: by default, in Standby Sleep mode, the PM analyzes the device activity to use either the main or the low-power voltage regulator to supply the VDDCORE.

GCLK clocks, regulators and RAM are not affected in Idle Sleep mode and will operate as normal.

Downloaded from Arrow.com.

Table 22-4. Regulators, RAMs, and NVM state in Sleep Mode

Sleep Mode	PDSW	SRAM Mode(1)	NVM	Regulators	
				VDDCORE	
				main	ULP
Active	active	normal	normal	on	on
Idle	active	auto(2)	on	on	on
Standby - PDSW in Active mode	active	normal(6)	auto(2)	auto(3)	on(5)
Standby - PDSW in Retention mode	retention	low power(6)	low power	auto(4)	on(5)
OFF	off	off	off	off	off

Note:

1. RAMs mode by default: STDBYCFG.BBIAS bits are set to their default value.
2. auto: by default, NVM is in low-power mode if not accessed.
3. auto: by default, the main voltage regulator is on if GCLK, APBx, or AHBx clock is running during SleepWalking.
4. auto: by default ULP regulator is selected in retention, but main regulator will be selected if VREG RUNSTDBY register bit in Supply Controller is set to 1.
5. on: low power voltage reference must be ready, and this is confirmed if STATUS.ULPVREFRDY register bit in SUPC equals to 1
6. SRAM can be partially retained in STANDBY using SRAM Power Switch

Related Links

22.6.4.4 Regulator Automatic Low Power Mode

22.6.4 Advanced Features

22.6.4.1 Power Domain Configuration

When entering Standby Sleep mode, a power domain is set automatically to retention state if no activity is required in it, refer to 22.6.3.5 Power Domain Controller for details. This behavior can be changed by writing the Power Domain Configuration bit group in the Standby Configuration register (STDBYCFG.PDCFG). For example, all power domains can be forced to remain in active state during Standby Sleep mode, this will accelerate wake-up time.

22.6.4.2 RAM Automatic Low Power Mode

The RAM is by default put in Low-Power mode (back-biased) if its power domain is in retention state and the device is in Standby Sleep mode.

This behavior can be changed by configuring BBIASxx bit groups in the Standby Configuration register (STDBYCFG.BBIASxx), refer to the table below for details.

Note: in Standby Sleep mode, the DMAC can access the SRAM in Standby Sleep mode only when the power domain PDSW is not in retention and PM.STDBYCFG.BBIASxx=0x0.

Downloaded from Arrow.com.

Table 22-5. RAM Back-Biasing Mode

STBYCDFG.BBIASxx config		RAM
0x0	Retention Back Biasing mode	RAM is back-biased if its power domain is in retention state
0x1	Standby Back Biasing mode	RAM is back-biased if the device is in Standby Sleep mode

22.6.4.3 SRAM Power Switch Configuration

The SRAM is divided in sub-blocks which can be retained or not in STANDBY low power mode to optimize power consumption.

By default, all sub-blocks are retained but it is possible to switch them off depending on SRAM memory size.

This behavior can be changed by configuring RAMPSWC bit groups in the Power Configuration register (PWCFG).

This configuration takes effect immediately. So, this is the responsibility of the user to ensure that no access is performed on OFF RAM.

When a SRAM sub-block is switched ON, the user has to wait 1 us before accessing it.

The first sub-block to be switched off is always at the top of the SRAM block memory and this is the same for the next ones.

22.6.4.4 Regulator Automatic Low Power Mode

In standby mode, the PM selects either the main or the low power voltage regulator to supply the VDDCORE. If switchable power domain is in retention state, the low power voltage regulator is used.

If a sleepwalking task is working on either asynchronous clocks (generic clocks) or synchronous clock (APB/AHB clocks), the main voltage regulator is used. This behavior can be changed by writing the Voltage Regulator Standby Mode bits in the Standby Configuration register (STDBYCFG.VREGSMOD). Refer to the following table for details.

Table 22-6. Regulator State in Sleep Mode

Sleep Modes	STDBYCFG. VREGSMOD	SleepWalking(1)	Regulator state for VDDCORE
Active	-	-	main voltage regulator
Idle	-	-	main voltage regulator
Standby (active)	0x0: AUTO	NO	low power regulator
		YES	main voltage regulator
	0x1: PERFORMANCE	-	main voltage regulator
	0x2: LP(2)	-(2)	low power regulator
Standby (retention)	-	-	low power regulator

Note:

1. SleepWalking is running on GCLK clock or synchronous clock. This is not related to XOSC32K or OSCULP32K clocks.
2. Must only be used when SleepWalking is running on GCLK with 32KHz source.

Downloaded from Arrow.com.

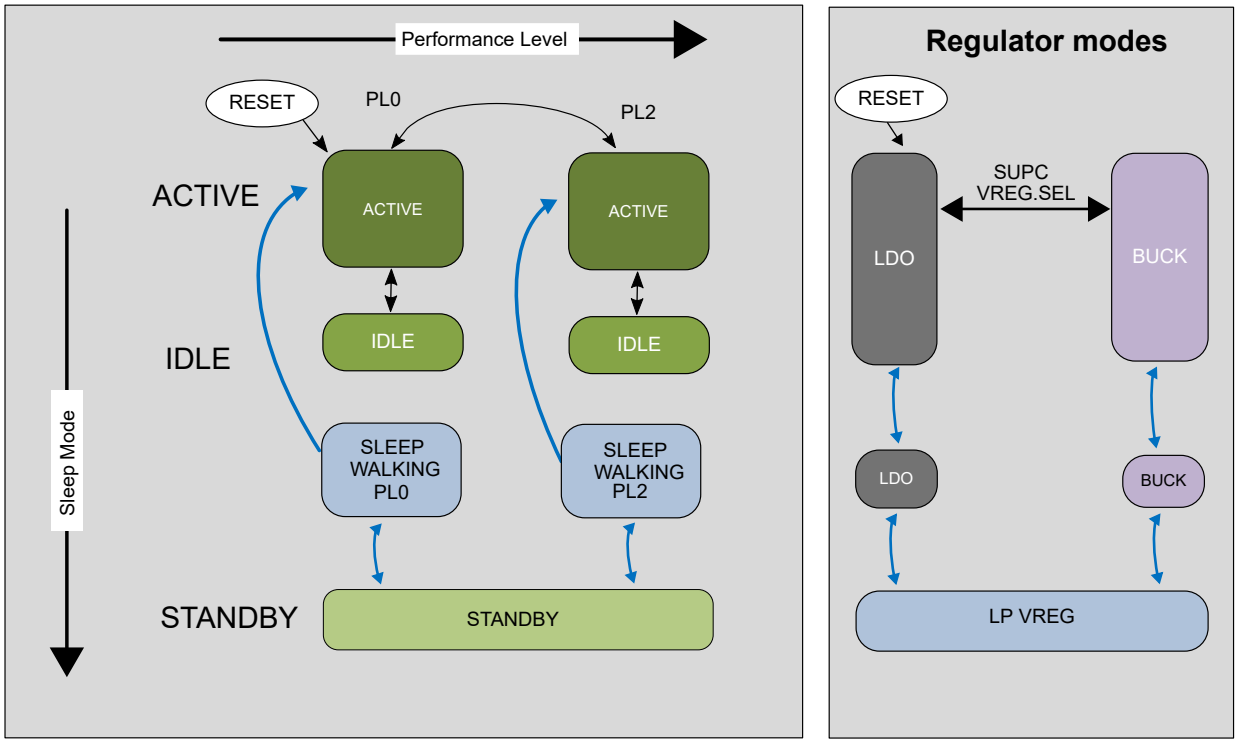
22.6.4.5 SleepWalking and Performance Level

SleepWalking is the capability for a device to temporarily wake up clocks for a peripheral to perform a task without waking up the CPU from STANDBY sleep mode. At the end of the sleepwalking task, the device can either be woken up by an interrupt (from a peripheral involved in SleepWalking) or enter again into STANDBY sleep mode. In this device, SleepWalking is supported only on GCLK clocks by using the on-demand clock principle of the clock sources.

In standby mode, when SleepWalking is ongoing, the performance level used to execute the sleepwalking task is the current configured performance level (used in active mode), and the main voltage regulator used to execute the SleepWalking task is the selected regulator used in active mode (LDO or Buck converter).

These are illustrated in the figure below.

Figure 22-4. Operating Conditions and SleepWalking



22.6.4.6 Wake-Up Time

The total wake-up time depends on the following:

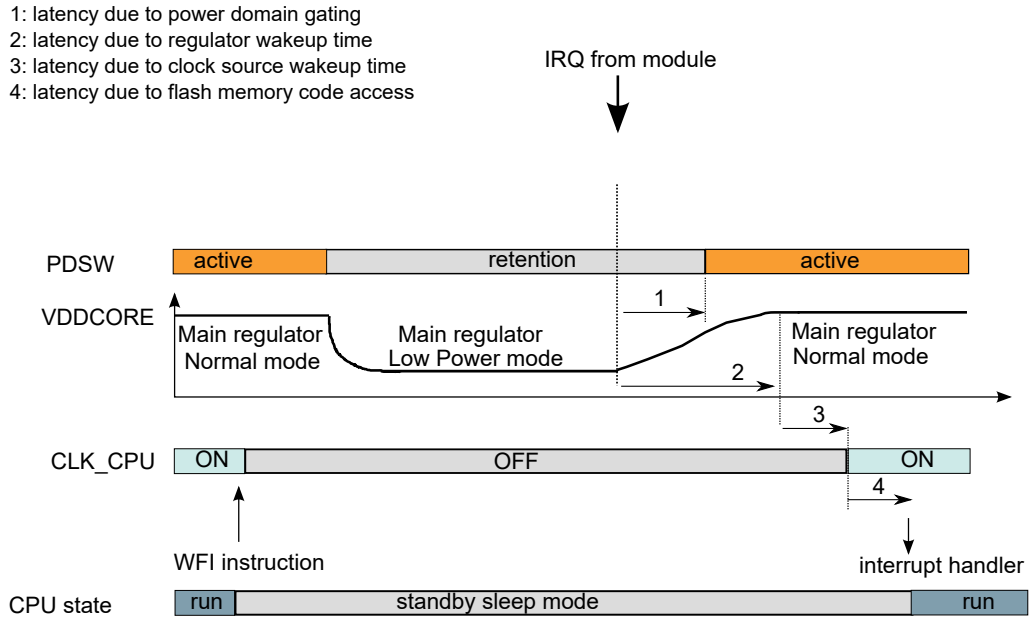
- Latency due to Power Domain Gating:
  Usually, wake-up time is measured with the assumption that the power domain is already in active state. When using Power Domain Gating, changing a power domain from retention to active state will take a certain time, refer to Electrical Characteristics. If power domain was already in active state in standby sleep mode, this latency is zero. If wake-up time is critical for the application, power domain can be forced to active state in Standby Sleep mode, refer to 22.6.4.1 Power Domain Configuration for details.
- Latency due to Performance Level and Regulator effect:
  Performance Level has to be taken into account for the global wake-up time. As example, if PL2 is selected and the device is in Standby Sleep mode, the voltage level supplied by the ULP voltage regulator is lower than the one used in Active mode. When the device wakes up, it takes a certain

Downloaded from Arrow.com.

amount of time for the main regulator to transition to the voltage level corresponding to PL2, causing additional wake-up time.

- Latency due to the CPU clock source wake-up time.
- Latency due to the NVM memory access.
- Latency due to Switchable Power Domain back-bias wake-up time:
  If back-bias is enabled, and the device wakes up from retention, it takes a certain amount of time for the regulator to settle.

Figure 22-5. Total Wake-up Time from Standby Sleep Mode



22.6.5 Standby with Static Power Domain Gating in Details

In Standby Sleep mode, the switchable power domain (PDSW) of a peripheral can remain in active state to perform the peripheral's tasks. This Static Power Domain Gating feature is supported by all peripherals. For some peripherals it must be enabled by writing a Run in Standby bit in the respective Control A register (CTRLA.RUNSTDBY) to '1'. Refer to each peripheral chapter for details.

The following examples illustrate Standby with static Power Domain Gating:

TC0 Standby with Static Power Domain Gating

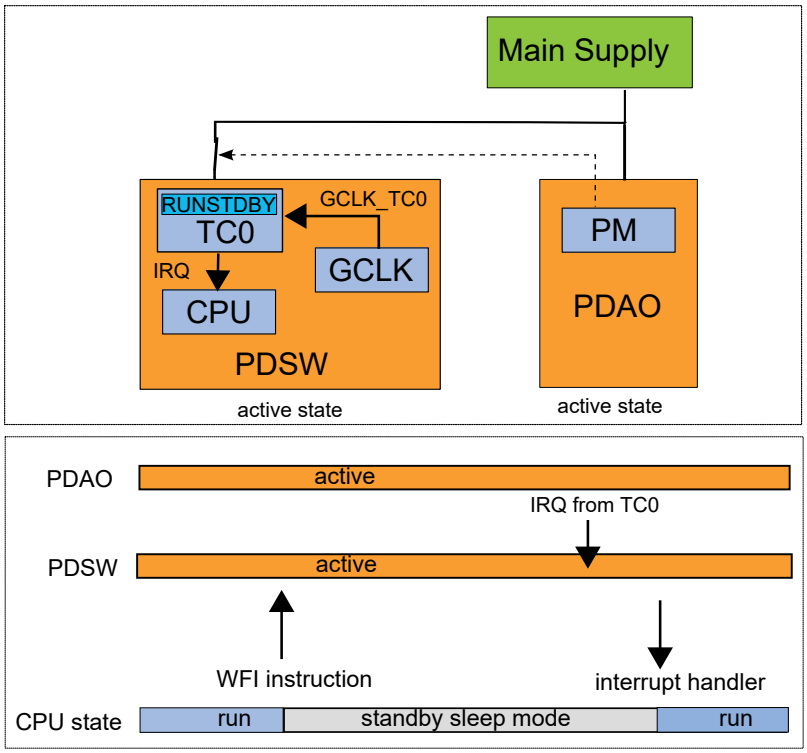
TC0 peripheral is used in counter operation mode. An interrupt is generated to wake-up the device based on the TC0 peripheral configuration. To make the TC0 peripheral continue to run in Standby Sleep mode, the RUNSTDBY bit is written to '1'.

- Entering Standby mode: As shown in Figure 22-6, PDSW remains active. Refer to 22.6.3.5 Power Domain Controller for details.
- Exiting Standby mode: When conditions are met, the TC0 peripheral generates an interrupt to wake-up the device, and the CPU is able to operate normally and execute the TC0 interrupt handler accordingly.
- Wake-up time:
  - The required time to set PDSW to active state has to be considered for the global wake-up time, refer to 22.6.4.6 Wake-Up Time for details.

Downloaded from Arrow.com.

– In this case, the VDDCORE voltage is still supplied by the main voltage regulator, refer to 22.6.4.4 Regulator Automatic Low Power Mode for details. Thus, global wake-up time is not affected by the regulator.

Figure 22-6. TC0 in Standby with Static Power Domain Gating



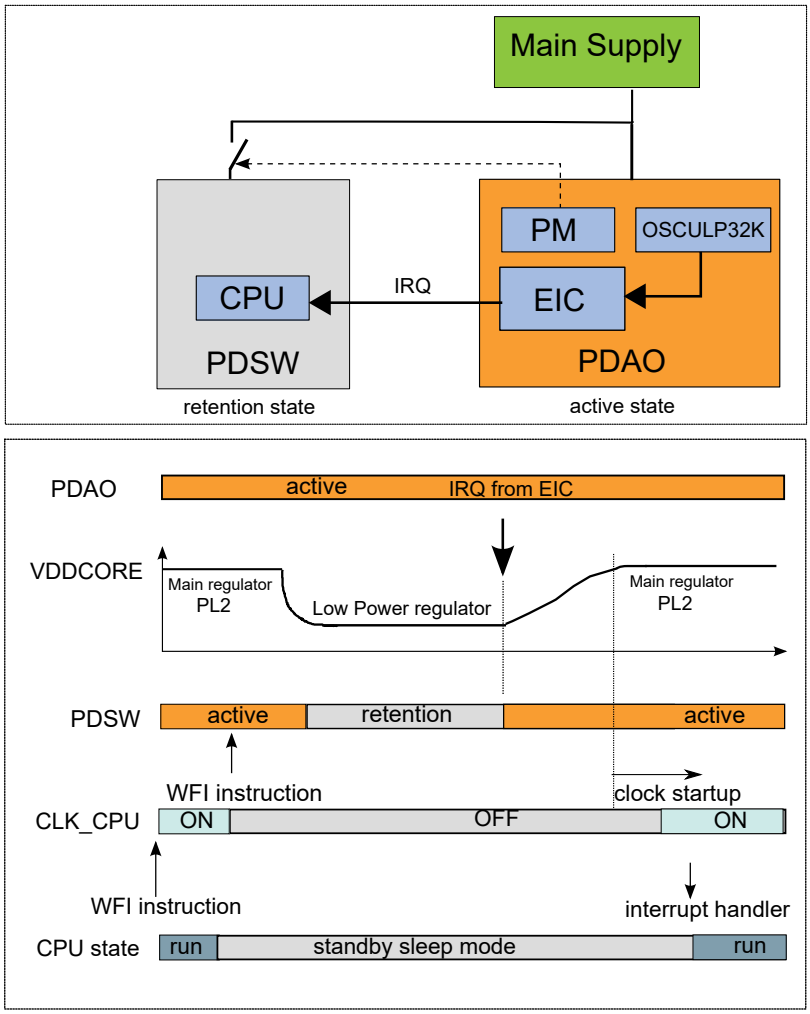
EIC in Standby with Static Power Domain Gating

In this example, EIC peripheral is used to detect an edge condition to generate interrupt to the CPU. An External interrupt pin is filtered by the CLK_ULP32K clock, GCLK peripheral is not used. Refer to Chapter 29. EIC – External Interrupt Controller for details. The EIC peripheral is located in the power domain PDAO (which is not switchable), and there is no RUNSTDBY bit in the EIC peripheral.

- Entering Standby mode: As shown in Figure 22-7, the switchable power domain is set in retention state by the Power Manager peripheral. The low power regulator supplies the VDDCORE voltage level.
- Exiting Standby mode: When conditions are met, the EIC peripheral generates an interrupt to wake the device up. Successively, the PM peripheral sets PDSW to active state, and the main voltage regulator restarts. Once PDSW is in active state and the main voltage regulator is ready, the CPU is able to operate normally and execute the EIC interrupt handler accordingly.
- Wake-up time:
  - The required time to set the switchable power domains to active state has to be considered for the global wake-up time, refer to 22.6.4.6 Wake-Up Time for details.
  - When in standby Sleep mode, the GCLK peripheral is not used, allowing the VDDCORE to be supplied by the low power regulator to reduce consumption, see 22.6.4.4 Regulator Automatic Low Power Mode. Consequently, main voltage regulator wake-up time has to be considered for the global wake-up time as shown in Figure 22-7.

Downloaded from Arrow.com.

Figure 22-7. EIC in Standby with Static Power Domain Gating



22.6.6 Sleepwalking with Dynamic Power Domain Gating in Details

To reduce power consumption even further, Sleepwalking with dynamic Power Domain Gating (also referred to as "Dynamic Sleepwalking") is used to turn power domain state from retention to active and vice-versa, based on event or DMA trigger.

22.6.6.1 Dynamic SleepWalking based on Event

To enable SleepWalking with dynamic power domain gating, the Dynamic Power Gating for Power Domain SW bit in the Standby Configuration register (STDBYCFG.DPGPDSW) has to be written to '1'.

When in retention state, a power domain can be automatically set to active state by the PM if an event is directed to this power domain. In this device, this concerns the event users located in power domain PDSW.

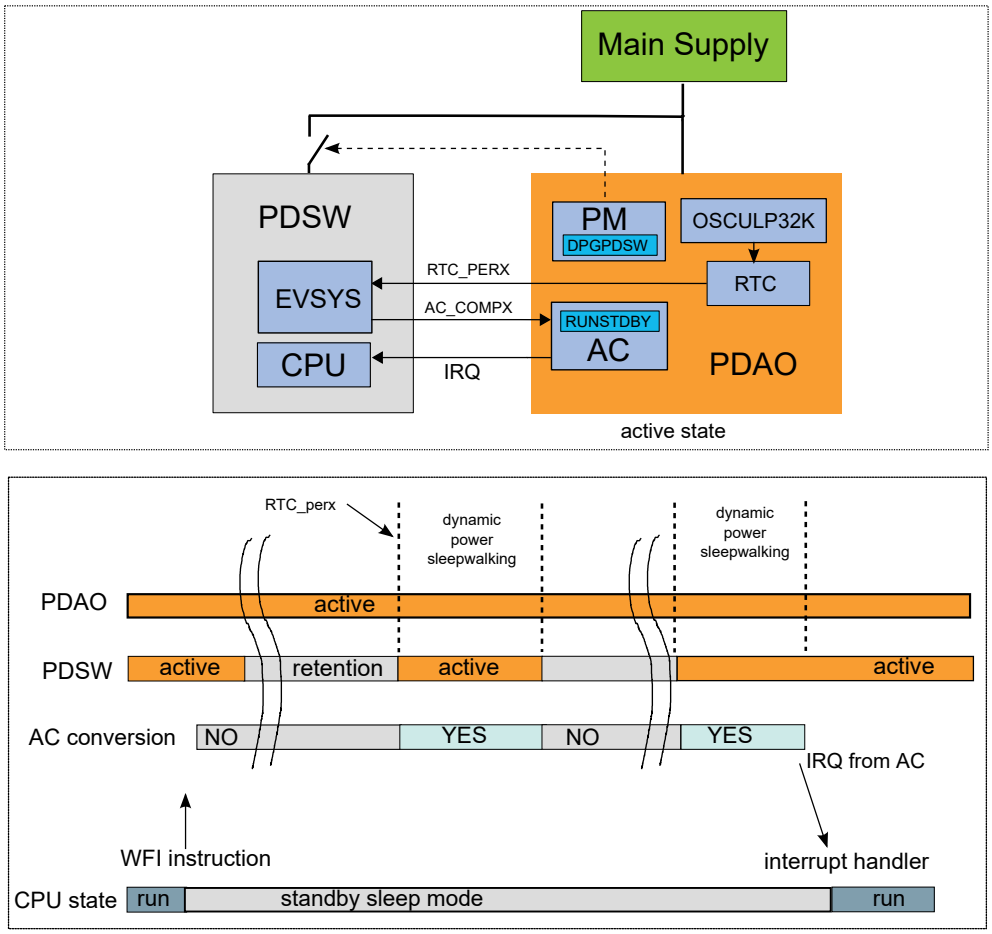
- When PDSW is in retention state, dynamic SleepWalking can be triggered by:
  - AC output event
  - RTC output event
  - EIC output event (if using the CLK_ULP32K clock and debouncing is enabled)

Refer also to 22.6.1.2 Power Domains.

Downloaded from Arrow.com.

Dynamic SleepWalking based on event is illustrated in the following example:

Figure 22-8. Dynamic SleepWalking based on Event: AC Periodic Comparison



The Analog Comparator (AC) peripheral is used in single shot mode to monitor voltage levels on input pins. A comparator interrupt, based on the AC peripheral configuration, is generated to wake up the device. In the GCLK module, the AC generic clock (GCLK_AC) source is routed a 32.768kHz oscillator (for low power applications, OSC32KULP is recommended). RTC and EVSYS modules are configured to generate periodic events to the AC. To make the comparator continue to run in standby sleep mode, the RUNSTDBY bit is written to '1'. To enable the dynamic SleepWalking for PDSW power domain, STDBYCFG.PDSW must be written to '1'.

Entering standby mode: The Power Manager sets the PDSW power domain in retention state. The AC comparators, COMPx, are OFF. The GCLK_AC clock is stopped. The VDDCORE is supplied by the low power regulator.

Dynamic SleepWalking: The RTC event (RTC_PERX) is routed by the Event System to the Analog Comparator to trigger a single-shot measurement. This event is detected by the Power Manager, which sets the PDSW power domain to active state and starts the main voltage regulator.

After enabling the AC comparator and starting the GCLK_AC, the single-shot measurement can be performed during Sleep mode (sleepwalking task), refer to 42.6.14.2 Single-Shot Measurement during Sleep for details. At the end of the conversion, if conditions to generate an interrupt are not met, the GCLK_AC clock is stopped again, as well as the AC comparator.

Downloaded from Arrow.com.

The low-power regulator starts again and the PDSW power domain is set back to retention state by the PM. During this dynamic SleepWalking period, the CPU is still sleeping.

Exiting standby mode: during the dynamic SleepWalking sequence, if conditions are met, the AC module generates an interrupt to wake up the device.

Related Links

27. RTC – Real-Time Counter
33. EVSYS – Event System

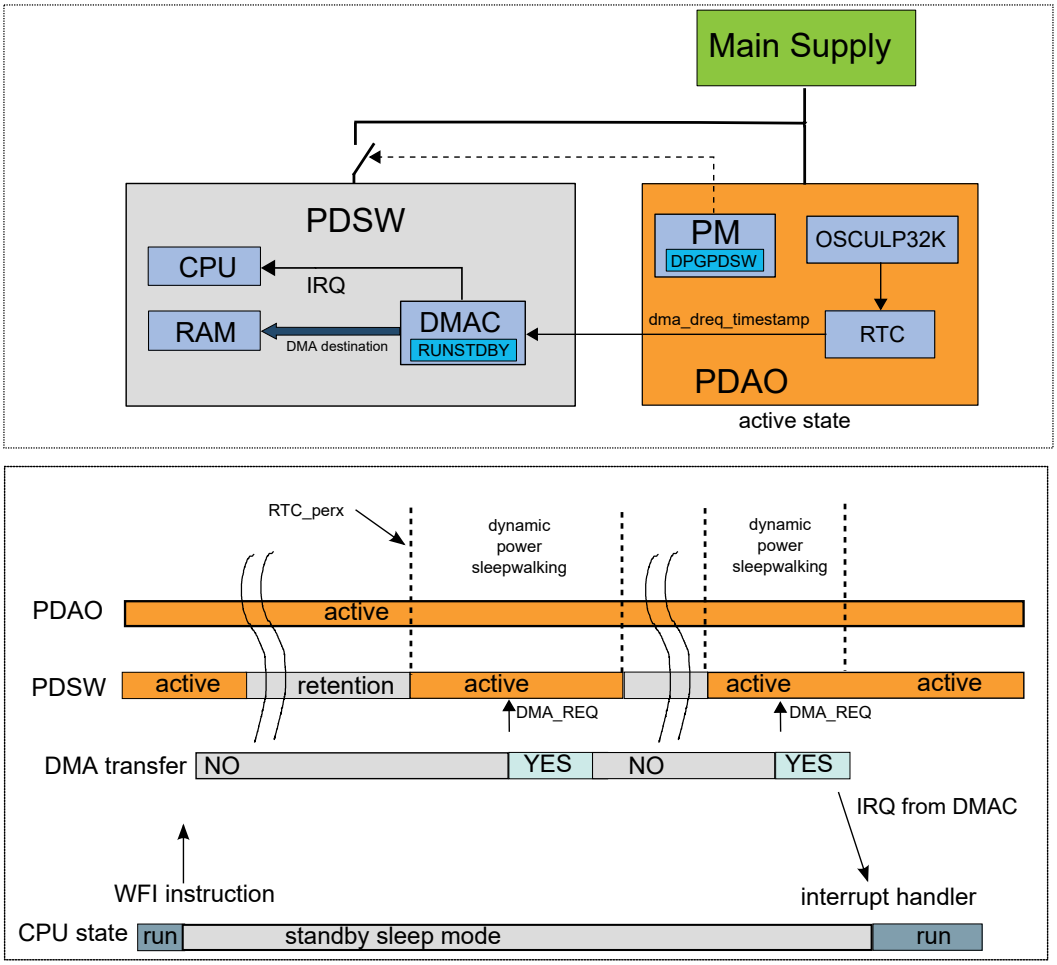
22.6.6.2 Dynamic SleepWalking Based on Peripheral DMA Trigger

To enable this advanced feature, the Dynamic Power Gating for Power Domain SW bit in the Standby Configuration register (STDBYCFG.DPGPDSW) have to be written to '1'.

When in retention state, the power domain PDSW (containing the DMAC) can be automatically set to active state if the PM detects a valid DMA trigger that is coming from a peripheral located in PDAO. A peripheral DMA trigger is valid if the corresponding DMA channel is enabled and its Run in Standby bit (RUNSTDBY) is written to '1'.

This is illustrated in the following example:

Figure 22-9. Dynamic Sleepwalking based on Peripheral DMA Trigger



Downloaded from Arrow.com.

The DMAC is configured to operate in standby sleep mode by using its respective RUNSTDBY bit. A DMAC channel is configured to set the DMA destination. The Run in Standby bit of this DMAC channel is written to '1' to allow it running in Standby Sleep mode.

Entering Standby mode: The Power Manager peripheral sets PDSW to retention state. The VDDCORE is supplied by the low-power regulator.

Dynamic SleepWalking: based on RTC conditions, an RTC output signal (DMA request for timestamp) triggers DMAC to put timestamp value at configured DMA destination.

This event is detected by the Power Manager which sets the PDSW power domain to active state and starts the main voltage regulator.

This DMA transfer request is detected by the PM, which sets PDSW (containing the DMAC) to active state. The DMAC requests the CLK_DMAC_AHB clock and transfer the timestamp value to the memory. When the DMA beat transfer is completed, the CLK_DMAC_AHB clock is stopped again.

The low-power regulator starts again and the PDSW power domain is set back to retention state by the PM. Note that during this dynamic SleepWalking period, the CPU is still sleeping.

Exiting Standby mode: during SleepWalking with Dynamic Power Gating sequence, if conditions are met, the DMAC generates an interrupt to wake up the device.

Related Links

27. RTC – Real-Time Counter
33. EVSYS – Event System

22.6.7 DMA Operation

Not applicable.

22.6.8 Interrupts

The peripheral has the following interrupt sources:

- Performance Level Ready (PLRDY)
  This interrupt is a synchronous wake-up source. See Table 22-1 for details.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled or the peripheral is reset.

An interrupt flag is cleared by writing a '1' to the corresponding bit in the INTFLAG register. Each peripheral can have one interrupt request line per interrupt source or one common interrupt request line for all the interrupt sources. Refer to the Nested Vector Interrupt Controller (NVIC) for details. If the peripheral has one common interrupt request line for all the interrupt sources, the user must read the INTFLAG register to determine which interrupt condition is present.

22.6.9 Events

Not applicable.

Downloaded from Arrow.com.

22.6.10 Sleep Mode Operation

The Power Manager is always active.

Downloaded from Arrow.com.

22.7 Register Summary

Offset	Name	Bit Pos.								
0x01	SLEEPCFG	7:0						SLEEPMODE[2:0]		
0x02	PLCFG	7:0	PLDIS						PLSEL[1:0]	
0x03	PWCFG	7:0							RAMPSWC[1:0]	
0x04	INTENCLR	7:0								PLRDY
0x05	INTENSET	7:0								PLRDY
0x06	INTFLAG	7:0								PLRDY
0x07	Reserved									
0x08	STDBYCFG	7:0	VREGSMOD[1:0]			DPGPDSW				PDCFG
		15:8				BBIASTR		BBIASHS		

22.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to 22.5.7 Register Access Protection.

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

Downloaded from Arrow.com.

22.8.1 Sleep Configuration

Name: SLEEPCFG
Offset: 0x01
Reset: 0x2
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
						SLEEPMODE[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:0 – SLEEPMODE[2:0] Sleep Mode
Note: A small latency happens between the store instruction and actual writing of the SLEEPCFG register due to bridges. Software has to make sure the SLEEPCFG register reads the wanted value before issuing WFI instruction.

Value	Name	Definition
0x0	Reserved	Reserved
0x1	Reserved	Reserved
0x2	IDLE	CPU, AHBx, and APBx clocks are OFF
0x3	Reserved	Reserved
0x4	STANDBY	ALL clocks are OFF, unless requested by sleepwalking peripheral
0x5	Reserved	Reserved
0x6	OFF	All power domains are powered OFF
0x7	Reserved	Reserved

Downloaded from Arrow.com.

22.8.2 Performance Level Configuration

Name: PLCFG
Offset: 0x02
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	PLDIS						PLSEL[1:0]	
Access	R/W						R/W	R/W
Reset	0						0	0

Bit 7 – PLDIS Performance Level Disable
Disabling the automatic PL selection forces the device to run in PL0 , reducing the power consumption and the wake-up time from standby sleep mode.

Changing this bit when the current performance level is not PL0 is discarded and a violation is reported to the PAC module.

Value	Description
0	The Performance Level mechanism is enabled.
1	The Performance Level mechanism is disabled.

Bits 1:0 – PLSEL[1:0] Performance Level Select

Value	Name	Definition
0x0	PL0	Performance Level 0
0x1	Reserved	Reserved
0x2	PL2	Performance Level 2
0x3	Reserved	Reserved

Downloaded from Arrow.com.

22.8.3 Power Configuration

Name: PWCFG
Offset: 0x03
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
							RAMPSWC[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 1:0 – RAMPSWC[1:0] RAM Power Switch Configuration

Value	Name	Definition
0x0	16KB	16KB Available
0x1	12KB	12KB Available
0x2	8KB	8KB Available
0x3	4KB	4KB Available

Downloaded from Arrow.com.

22.8.4 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Bit	7	6	5	4	3	2	1	0
								PLRDY
Access								R/W
Reset								0

Bit 0 – PLRDY Performance Level Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Performance Ready Interrupt Enable bit and the corresponding interrupt request.

Value	Description
0	The Performance Ready interrupt is disabled.
1	The Performance Ready interrupt is enabled and will generate an interrupt request when the Performance Ready Interrupt Flag is set.

Downloaded from Arrow.com.

22.8.5 Interrupt Enable Set

Name: INTENSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Bit	7	6	5	4	3	2	1	0
								PLRDY
Access								R/W
Reset								0

Bit 0 – PLRDY Performance Level Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Performance Ready Interrupt Enable bit and enable the Performance Ready interrupt.

Value	Description
0	The Performance Ready interrupt is disabled.
1	The Performance Ready interrupt is enabled.

Downloaded from Arrow.com.

22.8.6 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x06
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
								PLRDY
Access								R/W
Reset								0

Bit 0 – PLRDY Performance Level Ready
This flag is set when the performance level is ready and will generate an interrupt if INTENCLR/SET.PLRDY is '1'.

Writing a '1' to this bit has no effect.

Writing a '1' to this bit clears the Performance Ready interrupt flag.

Downloaded from Arrow.com.

22.8.7 Standby Configuration

Name: STDBYCFG
Offset: 0x08
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
				BBIASTR		BBIASHS		
Access				R/W		R/W		
Reset				0		0		

Bit	7	6	5	4	3	2	1	0
	VREGSMOD[1:0]			DPGPDSW				PDCFG
Access	R	R		R/W				R/W
Reset	0	0		0				0

Bit 12 – BBIASTR Back Bias for Trust RAM
Refer to 22.6.4.2 RAM Automatic Low Power Mode for details.

Value	Description
0	Retention Back Biasing mode
1	Standby Back Biasing mode

Bit 10 – BBIASHS Back Bias for HMCRAMCHS
Refer to 22.6.4.2 RAM Automatic Low Power Mode for details.

Value	Description
0	Retention Back Biasing mode
1	Standby Back Biasing mode

Bits 7:6 – VREGSMOD[1:0] VREG Switching Mode
Refer to 22.6.4.4 Regulator Automatic Low Power Mode for details.

Value	Name	Description
0x0	AUTO	Automatic Mode
0x1	PERFORMANCE	Performance oriented
0x2	LP	Low Power consumption oriented

Bit 4 – DPGPDSW Dynamic Power Gating for Switchable Power Domain

Value	Description
0	Dynamic SleepWalking for switchable power domain is disabled
1	Dynamic SleepWalking for switchable power domain PDSW is enabled

Bit 0 – PDCFG Power Domain Configuration

Value	Name	Description
0x0	DEFAULT	In standby mode, all power domain switching are handled by hardware.
0x1	PDSW	In standby mode, PDSW is forced ACTIVE.

Downloaded from Arrow.com.

23. OSCCTRL – Oscillators Controller

23.1 Overview

The Oscillators Controller (OSCCTRL) provides a user interface to the XOSC, OSC16M, DFLLULP and FDPLL96M.

Through the interface registers, it is possible to enable, disable, calibrate, and monitor the OSCCTRL oscillators.

All oscillators statuses are collected in the Status register (STATUS). They can additionally trigger interrupts upon status changes via the INTENSET, INTENCLR, and INTFLAG registers.

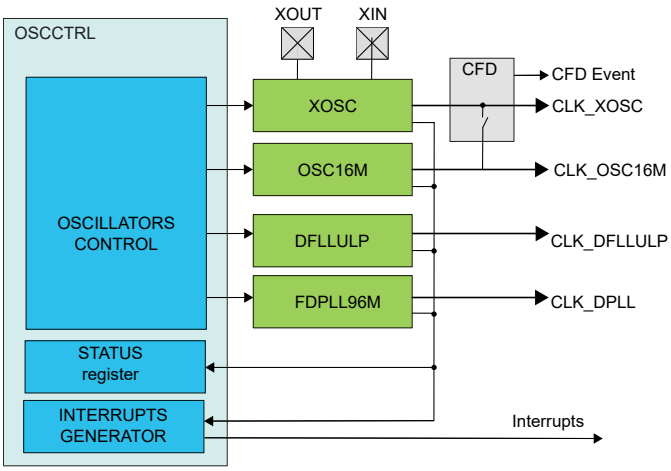
23.2 Features

- 0.4-32MHz Crystal Oscillator (XOSC)
  - Tunable gain control
  - Programmable start-up time
  - Crystal or external input clock on XIN I/O
  - Clock failure detection with safe clock switch
  - Clock failure event output
- 16MHz Internal Oscillator (OSC16M)
  - Fast startup
  - 4/8/12/16MHz output frequencies available
- Ultra Low-Power Digital Frequency Locked Loop (DFLLULP)
  - Operates as a frequency multiplier against a known frequency in closed loop mode
  - Optional frequency dithering
- Fractional Digital Phase Locked Loop (FDPLL96M)
  - 32 MHz to 96 MHz output frequency
  - 32 kHz to 2MHz reference clock
  - A selection of sources for the reference clock
  - Adjustable proportional integral controller
  - Fractional part used to achieve 1/16th of reference clock step

Downloaded from Arrow.com.

23.3 Block Diagram

Figure 23-1. OSCCTRL Block Diagram



23.4 Signal Description

Signal	Description	Type
XIN	Multipurpose Crystal Oscillator or external clock generator input	Analog input
XOUT	Multipurpose Crystal Oscillator output	Analog output

The I/O lines are automatically selected when XOSC is enabled.

23.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

23.5.1 I/O Lines

I/O lines are configured by OSCCTRL when XOSC is enabled, and need no user configuration.

23.5.2 Power Management

The OSCCTRL can continue to operate in any sleep mode where the selected source clock is running. The OSCCTRL interrupts can be used to wake up the device from sleep modes. The events can trigger other operations in the system without exiting sleep modes.

Related Links

22. PM – Power Manager

23.5.3 Clocks

The OSCCTRL gathers controls for all device oscillators and provides clock sources to the Generic Clock Controller (GCLK). The available clock sources are XOSC, OSC16M, DFLLULP and FDPLL96M.

The OSCCTRL bus clock (CLK_OSCCTRL_APB) can be enabled and disabled in the Main Clock module (MCLK).

Downloaded from Arrow.com.

The control logic uses the oscillator output, which is also asynchronous to the user interface clock (CLK_OSCCTRL_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to 23.6.11 Synchronization for further details.

A generic clock (GCLK_DFLLULP) is required to clock the DFLLULP tuner in closed-loop operation. This clock must be configured and enabled in the generic clock controller before using the DFLLULP tuner. Refer to the *Generic Clock Controller (GCLK)* chapter for details.

Related Links

19. MCLK – Main Clock

19.6.2.6 Peripheral Clock Masking

23.5.4 DMA

Not applicable.

23.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. Using the OSCCTRL interrupts requires the interrupt controller to be configured first.

23.5.6 Events

The events of this peripheral are connected to the Event System.

Related Links

33. EVSYS – Event System

23.5.7 Debug Operation

When the CPU is halted in debug mode the OSCCTRL continues normal operation. If the OSCCTRL is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

23.5.8 Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Interrupt Flag Status and Clear register (INTFLAG)

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

Write-protection does not apply for accesses through an external debugger.

23.5.9 SAM L11 TrustZone Specific Register Access Protection

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

Downloaded from Arrow.com.

23.5.10 Analog Connections

The 0.4-32MHz crystal must be connected between the XIN and XOUT pins, along with any required load capacitors.

23.6 Functional Description

23.6.1 Principle of Operation

XOSC, OSC16M, and FDPLL96M. are configured via OSCCTRL control registers. Through this interface, the oscillators are enabled, disabled, or have their calibration values updated.

The Status register gathers different status signals coming from the oscillators controlled by the OSCCTRL. The status signals can be used to generate system interrupts, and in some cases wake the system from Sleep mode, provided the corresponding interrupt is enabled.

23.6.2 External Multipurpose Crystal Oscillator (XOSC) Operation

The XOSC can operate in two different modes:

- External clock, with an external clock signal connected to the XIN pin
- Crystal oscillator, with an external 0.4-32MHz crystal

The XOSC can be used as a clock source for generic clock generators. This is configured by the Generic Clock Controller.

At reset, the XOSC is disabled, and the XIN/XOUT pins can be used as General Purpose I/O (GPIO) pins or by other peripherals in the system. When XOSC is enabled, the operating mode determines the GPIO usage. When in crystal oscillator mode, the XIN and XOUT pins are controlled by the OSCCTRL, and GPIO functions are overridden on both pins. When in external clock mode, only the XIN pin will be overridden and controlled by the OSCCTRL, while the XOUT pin can still be used as a GPIO pin.

The XOSC is enabled by writing a '1' to the Enable bit in the External Multipurpose Crystal Oscillator Control register (XOSCCTRL.ENABLE).

To enable XOSC as an external crystal oscillator, the XTAL Enable bit (XOSCCTRL.XTALEN) must be written to '1'. If XOSCCTRL.XTALEN is zero, the external clock input on XIN will be enabled.

When in crystal oscillator mode (XOSCCTRL.XTALEN=1), the External Multipurpose Crystal Oscillator Gain (XOSCCTRL.GAIN) must be set to match the external crystal oscillator frequency. If the External Multipurpose Crystal Oscillator Automatic Amplitude Gain Control (XOSCCTRL.AMPGC) is '1', the oscillator amplitude will be automatically adjusted, and in most cases result in a lower power consumption.

The XOSC will behave differently in different sleep modes, based on the settings of XOSCCTRL.RUNSTDBY, XOSCCTRL.ONDEMAND, and XOSCCTRL.ENABLE. If XOSCCTRL.ENABLE=0, the XOSC will be always stopped. For XOSCCTRL.ENABLE=1, this table is valid:

Table 23-1. XOSC Sleep Behavior

CPU Mode	XOSCCTRL.RUNSTDBY	XOSCCTRL.ONDEMAND	Sleep Behavior
Active or Idle	-	0	Always run
Active or Idle	-	1	Run if requested by peripheral

Downloaded from Arrow.com.

CPU Mode	XOSCCTRL.RUNSTDBY	XOSCCTRL.ONDEMAND	Sleep Behavior
Standby	1	0	Always run
Standby	1	1	Run if requested by peripheral
Standby	0	-	Run if requested by peripheral

After a hard reset, or when waking up from a sleep mode where the XOSC was disabled, the XOSC will need a certain amount of time to stabilize on the correct frequency. This start-up time can be configured by changing the Oscillator Start-Up Time bit group (XOSCCTRL.STARTUP) in the External Multipurpose Crystal Oscillator Control register. During the start-up time, the oscillator output is masked to ensure that no unstable clock propagates to the digital logic.

The External Multipurpose Crystal Oscillator Ready bit in the Status register (STATUS.XOSCRDY) is set once the external clock or crystal oscillator is stable and ready to be used as a clock source. An interrupt is generated on a zero-to-one transition on STATUS.XOSCRDY if the External Multipurpose Crystal Oscillator Ready bit in the Interrupt Enable Set register (INTENSET.XOSCRDY) is set.

Related Links

18. GCLK - Generic Clock Controller

23.6.3 Clock Failure Detection Operation

The Clock Failure Detector (CFD) enables the user to monitor the external clock or crystal oscillator signal provided by the external oscillator (XOSC). The CFD detects failing operation of the XOSC clock with reduced latency, and allows to switch to a safe clock source in case of clock failure. The user can also switch from the safe clock back to XOSC in case of recovery. The safe clock is derived from the OSC16M oscillator with a configurable prescaler. This allows to configure the safe clock in order to fulfill the operative conditions of the microcontroller.

In sleep modes, CFD operation is automatically disabled when the external oscillator is not requested to run by a peripheral. See the Sleep Behavior table above when this is the case.

The user interface registers allow to enable, disable, and configure the CFD. The Status register provides status flags on failure and clock switch conditions. The CFD can optionally trigger an interrupt or an event when a failure is detected.

Clock Failure Detection

The CFD is disabled at reset. The CFD does not monitor the XOSC clock when the oscillator is disabled (XOSCCTRL.ENABLE=0).

Before starting CFD operation, the user must start and enable the safe clock source (OSC16M oscillator).

CFD operation is started by writing a '1' to the CFD Enable bit in the External Oscillator Control register (XOCCTRL.CFDEN). After starting or restarting the XOSC, the CFD does not detect failure until the start-up time has elapsed. The start-up time is configured by the Oscillator Start-Up Time in the External Multipurpose Crystal Oscillator Control register (XOSCCTRL.STARTUP). Once the XOSC Start-Up Time is elapsed, the XOSC clock is constantly monitored.

During a period of 4 safe clocks (monitor period), the CFD watches for a clock activity from the XOSC. There must be at least one rising and one falling XOSC clock edge during 4 safe clock periods to meet non-failure conditions. If no or insufficient activity is detected, the failure status is asserted: The Clock Failure Detector status bit in the Status register (STATUS.CLKFAIL) and the Clock Failure Detector

Downloaded from Arrow.com.

interrupt flag bit in the Interrupt Flag register (INTFLAG.CLKFAIL) are set. If the CLKFAIL bit in the Interrupt Enable Set register (INTENSET.CLKFAIL) is set, an interrupt is generated as well. If the Event Output enable bit in the Event Control register (EVCTRL.CFDEO) is set, an output event is generated, too.

After a clock failure was issued the monitoring of the XOSC clock is continued, and the Clock Failure Detector status bit in the Status register (STATUS.CLKFAIL) reflects the current XOSC activity.

Clock Switch

When a clock failure is detected, the XOSC clock is replaced by the safe clock in order to maintain an active clock during the XOSC clock failure. The safe clock source is the OSC16M oscillator clock. The safe clock source can be scaled down by a configurable prescaler to ensure that the safe clock frequency does not exceed the operating conditions selected by the application. When the XOSC clock is switched to the safe clock, the Clock Switch bit in the Status register (STATUS.CLKSW) is set.

When the CFD has switched to the safe clock, the XOSC is not disabled. If desired, the application must take the necessary actions to disable the oscillator. The application must also take the necessary actions to configure the system clocks to continue normal operations.

If the application can recover the XOSC, the application can switch back to the XOSC clock by writing a '1' to Switch Back Enable bit in the Clock Failure Control register (XOSCCTRL.SWBACK). Once the XOSC clock is switched back, the Switch Back bit (XOSCCTRL.SWBACK) is cleared by hardware.

Prescaler

The CFD has an internal configurable prescaler to generate the safe clock from the OSC16M oscillator. The prescaler size allows to scale down the OSC16M oscillator so the safe clock frequency is not higher than the XOSC clock frequency monitored by the CFD. The division factor is 2^P, with P being the value of the CFD Prescaler bits in the CFD Prescaler Register (CFDPRESC.CFDPRESC).

Example 23-1.

For an external crystal oscillator at 0.4 MHz and the OSC16M frequency at 16 MHz, the CFDPRESC.CFDPRESC value should be set scale down by more than factor 16/0.4=80, for example 128, for a safe clock of adequate frequency.

Event

If the Event Output Enable bit in the Event Control register (EVCTRL.CFDEO) is set, the CFD clock failure will be output on the Event Output. When the CFD is switched to the safe clock, the CFD clock failure will not be output on the Event Output.

Sleep Mode

The CFD is halted depending on configuration of the XOSC and the peripheral clock request. For further details, refer to the Sleep Behavior table above. The CFD interrupt can be used to wake up the device from sleep modes.

23.6.4 16MHz Internal Oscillator (OSC16M) Operation

The OSC16M is an internal oscillator operating in open-loop mode and generating 4, 8, 12, or 16MHz frequency. The OSC16M frequency is selected by writing to the Frequency Select field in the OSC16M register (OSC16MCTRL.FSEL). OSC16M is enabled by writing '1' to the Oscillator Enable bit in the OSC16M Control register (OSC16MCTRL.ENABLE), and disabled by writing a '0' to this bit. Frequency selection must be done when OSC16M is disabled.

Downloaded from Arrow.com.

After enabling OSC16M, the OSC16M clock is output as soon as the oscillator is ready (STATUS.OSC16MRDY=1). User must ensure that the OSC16M is fully disabled before enabling it by reading STATUS.OSC16MRDY=0.

After reset, OSC16M is enabled and serves as the default clock source at 4MHz.

OSC16M will behave differently in different sleep modes based on the settings of OSC16MCTRL.RUNSTDBY, OSC16MCTRL.ONDEMAND, and OSC16MCTRL.ENABLE. If OSC16MCTRL.ENABLE=0, the OSC16M will be always stopped. For OSC16MCTRL.ENABLE=1, this table is valid:

Table 23-2. OSC16M Sleep Behavior

CPU Mode	OSC16MCTRL.RUNSTDBY	OSC16MCTRL.ONDEMAND	Sleep Behavior
Active or Idle	-	0	Always run
Active or Idle	-	1	Run if requested by peripheral
Standby	1	0	Always run
Standby	1	1	Run if requested by peripheral
Standby	0	-	Run if requested by peripheral

OSC16M is used as a clock source for the generic clock generators. This is configured by the Generic Clock Generator Controller.

Related Links

18. GCLK - Generic Clock Controller

23.6.5 Ultra Low-Power Digital Frequency Locked Loop (DFLLULP) Operation

The Ultra Low-Power Digital Frequency Locked Loop (DFLLULP) is an internal oscillator that can output a selectable frequency based on user inputs. The frequency is a multiplication ratio relative to a given reference clock using the tuning feature. The oscillator has to be enabled for the tuner to work.

Figure 23-2. Block Diagram

23.6.5.1 Basic Operation

23.6.5.1.1 Initialization

The following bits are enable-protected, meaning that they can only be written when the DFLLULP is disabled (DFLLULPCTRL.ENABLE is zero):

- Binary Search Enable bit in Control register (DFLLULPCTRL.BINSE)
- Safe Mode bit in Control register (DFLLULPCTRL.SAFE)
- Dither Mode bit in Control register (DFLLULPCTRL.DITHER)
- Division Factor bits in Control register (DFLLULPCTRL.DIV)

The following registers are enable-protected:

- Dither Control register (DFLLULPDITHER)
- Target Ratio register (DFLLULPRATIO)

Enable-protected bits in the DFLLULPCTRL register can be written at the same time as DFLLULPCTRL.ENABLE is written to one, but not at the same time as DFLLULPCTRL.ENABLE is written to zero.

Downloaded from Arrow.com.

Enable-protection is denoted by the Enable-Protected property in the register description.

23.6.5.1.2 Enabling and Disabling

The DFLLULP is enabled by writing a one to the Enable bit in the Control register (DFLLULPCTRL.ENABLE). The DFLLULP is disabled by writing a zero to DFLLULPCTRL.ENABLE.

23.6.5.1.3 Closed Loop Mode

In closed loop mode the frequency is controlled by a tuner which measures the ratio between the DFLLULP output frequency and reference clock frequency. The reference clock is provided by a generic clock. The target ratio is written to the RATIO field in the Target Ratio Register (DFLLULPRATIO). When the oscillator is enabled, the output frequency will be tuned to the target frequency.

When the tuning is finished, the Lock bit in the OSCCTRL Status Register will be set (STATUS.DFLLULPLOCK). The No Lock bit in the Status register (STATUS.DFLLULPNOLOCK) will be set to indicate whether a frequency lock is achieved or not. The No Lock bit should be checked after the Lock bit has been set. Lock status is cleared only if the tuner is disabled or a new value is written to DFLLULPDLY. The DFLLULP will attempt to track any variation in the internal oscillator or reference clock, and will not release the lock. No Lock may be set after lock is achieved if the tuner ever reaches the minimum or maximum delay value.

Tuning starts from the delay value in DFLLULPDLY.DELAY. A write to DFLLULPDLY.DELAY while tuning is in progress will restart tuning from the newly written value. The tuned delay value can be read back from DFLLULPDLY.DELAY after requesting a synchronization via the Read Request bit in the DFLLULPRREQ register.

The accuracy of the tuner frequency comparison is limited by the inverse of the target ratio (1/RATIO). Larger ratios, i.e. much slower reference clocks will give better results.

23.6.5.1.4 Binary Search

By default, the tuner starts from the current value of the DFLLULPDLY register and increments or decrements every reference clock period. This linear search can take up to a maximum of 256 reference clock cycles before lock. To speed up the time to lock, binary search can be enabled by writing one to the Binary Search Enable bit in the Control register (DFLLULPCTRL.BINSE). Binary search takes a maximum of 8 reference clock cycles to lock. After 8 reference clock cycles the tuner will operate in normal linear mode to track any changes in the frequency. Note that neither search algorithm is guaranteed to lock if the target ratio is outside of the oscillator tunable range. Binary search will induce large swings in the oscillator frequency. If this is not desirable, an optional safe mode can be used to mask the output clock until the search is complete. Safe mode is enabled by writing a one to the Safe Mode bit in the Control register (DFLLULPCTRL.SAFE). The binary search is re-triggered if there is a write to DFLLULPDLY, or if the tuner is disabled and re-enabled. Ondemand or sleep modes will not re-trigger the binary search. If binary search will be used with ondemand behavior, it is recommended to first enable the tuner with DFLLULPCTRL.ONDEMAND=0 and then set DFLLULPCTRL.ONDEMAND=1 after the tuner has locked. This will ensure that each request will start from the locked state.

23.6.5.1.5 Dithering

Dithering operation can improve the precision of the closed-loop tuner. Dithering works on two aspects: the delay step size and the comparator resolution. Normally the delay can only be changed in steps of one unit of the 8-bit delay field every reference clock period, for a total of 256 steps. Dithering allows for 8-bits of fractional delay value by automatically changing between DELAY and DELAY+1 values with a weight determined by the tuner. This also has the effect of smoothing the frequency over time. Dithering is therefore equivalent to 16-bits of delay value. If this full range is not needed, the step size can be increased by writing the Step Size field in the Dithering register (DFLLULPDITHER.STEP). By default the frequency comparator resolution is limited to 1/RATIO over a single reference clock period. In dithering operation, the comparator resolution can be made finer by comparing over multiple reference clock

Downloaded from Arrow.com.

periods. This behavior is controlled by the Period field in the Dithering register (DFLLULPDITHER.PER). The fine control offered by dithering means that the tuner will take longer to adjust to coarse changes in the frequency. When dithering mode is active, the tuner will attempt to get close to the final locked value before starting the dithering engine. Dithering mode can be restarted if there is a write to DFLLULPDLY, or if the tuner is disabled and re-enabled.

23.6.6 Event Triggered Tuning

The EVCTRL.TUNEEI and EVCTRL.TUNEINV control bits allow to start a tuning sequence on an incoming event or inverted event. On an incoming rising or falling edge of the event input, the DFLLULP close loop tuner unlock and start over a frequency tuning, depending on the configuration of the DFLLULP registers, until the tuner achieves a new lock.

23.6.7 Digital Phase Locked Loop (DPLL) Operation

The task of the DPLL is to maintain coherence between the input (reference) signal and the respective output frequency, CLK_DPLL, via phase comparison. The DPLL controller supports three independent sources of reference clocks:

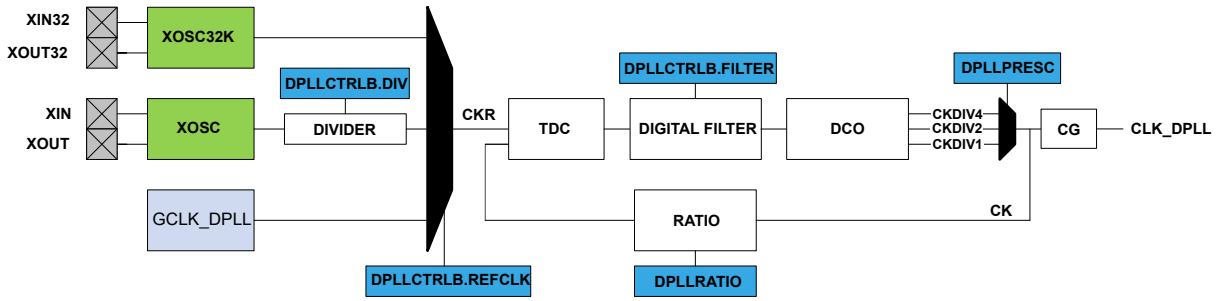
- XOSC32K: this clock is provided by the 32K External Crystal Oscillator (XOSC32K).
- XOSC: this clock is provided by the External Multipurpose Crystal Oscillator (XOSC).
- GCLK: this clock is provided by the Generic Clock Controller.

When the controller is enabled, the relationship between the reference clock frequency and the output clock frequency is:

$$f_{CK} = f_{CKR} \times \left(\mathrm{LDR} + 1 + \frac{\mathrm{LDRFRAC}}{16}\right) \times \frac{1}{2^{\mathrm{PRESC}}}$$

Where f_{CK} is the frequency of the DPLL output clock, LDR is the loop divider ratio integer part, LDRFRAC is the loop divider ratio fractional part, f_{CKR} is the frequency of the selected reference clock, and PRESC is the output prescaler value.

Figure 23-3. DPLL Block Diagram



When the controller is disabled, the output clock is low. If the Loop Divider Ratio Fractional part bit field in the DPLL Ratio register (DPLLRATIO.LDRFRAC) is zero, the DPLL works in integer mode. Otherwise, the fractional mode is activated. Note that the fractional part has a negative impact on the jitter of the DPLL.

> Example (integer mode only): assuming F_{CKR} = 32kHz and F_{CK} = 48MHz, the multiplication ratio is 1500. It means that LDR shall be set to 1499.

Downloaded from Arrow.com.

Example (fractional mode): assuming F_{CKR} = 32kHz and F_{CK} = 48.006MHz, the multiplication ratio is 1500.1875 (1500 + 3/16). Thus LDR is set to 1499 and LDRFRAC to 3.

Related Links

18. GCLK - Generic Clock Controller
24. OSC32KCTRL – 32KHz Oscillators Controller

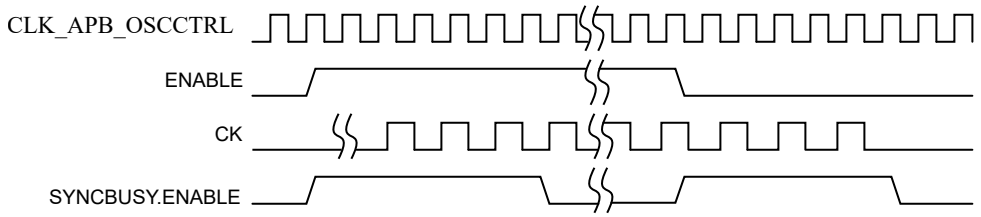
23.6.7.1 Basic Operation

23.6.7.1.1 Initialization, Enabling, Disabling, and Resetting

The DPLLC is enabled by writing a '1' to the Enable bit in the DPLL Control A register (DPLLCTRLA.ENABLE). The DPLLC is disabled by writing a zero to this bit.

The DPLLSYNCBUSY.ENABLE is set when the DPLLCTRLA.ENABLE bit is modified. It is cleared when the DPLL output clock CK has sampled the bit at the high level after enabling the DPLL. When disabling the DPLL, DPLLSYNCBUSY.ENABLE is cleared when the output clock is no longer running.

Figure 23-4. Enable Synchronization Busy Operation



The frequency of the DPLL output clock CK is stable when the module is enabled and when the Lock bit in the DPLL Status register is set (DPLLSTATUS.LOCK).

When the Lock Time bit field in the DPLL Control B register (DPLLCTRLB.LTIME) is non-zero, a user defined lock time is used to validate the lock operation. In this case the lock time is constant. If DPLLCTRLB.LTIME=0, the lock signal is linked with the status bit of the DPLL, and the lock time varies depending on the filter selection and the final target frequency.
Note: GCLK_DPLL_32K is responsible for counting the user defined lock time (LTIME different from 0x0), hence must be enabled.

When the Wake Up Fast bit (DPLLCTRLB.WUF) is set, the wake up fast mode is activated. In this mode the clock gating cell is enabled at the end of the startup time. At this time the final frequency is not stable, as it is still during the acquisition period, but it allows to save several milliseconds. After first acquisition, the Lock Bypass bit (DPLLCTRLB.LBYPASS) indicates if the lock signal is discarded from the control of the clock gater (CG) generating the output clock CLK_DPLL.

Table 23-3. CLK_DPLL Behavior from Startup to First Edge Detection

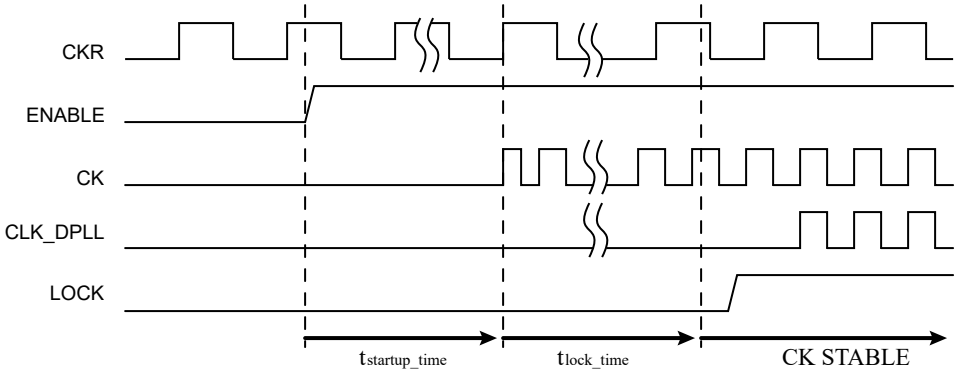
WUF	LTIME	CLK_DPLL Behavior
0	0	Normal Mode: First Edge when lock is asserted
0	Not Equal To Zero	Lock Timer Timeout mode: First Edge when the timer down-counts to 0.
1	X	Wake Up Fast Mode: First Edge when CK is active (startup time)

Downloaded from Arrow.com.

Table 23-4. CLK_DPLL Behavior after First Edge Detection

LBYPASS	CLK_DPLL Behavior
0	Normal Mode: the CLK_DPLL is turned off when lock signal is low.
1	Lock Bypass Mode: the CLK_DPLL is always running, lock is irrelevant.

Figure 23-5. CK and CLK_DPLL Output from DPLL Off Mode to Running Mode



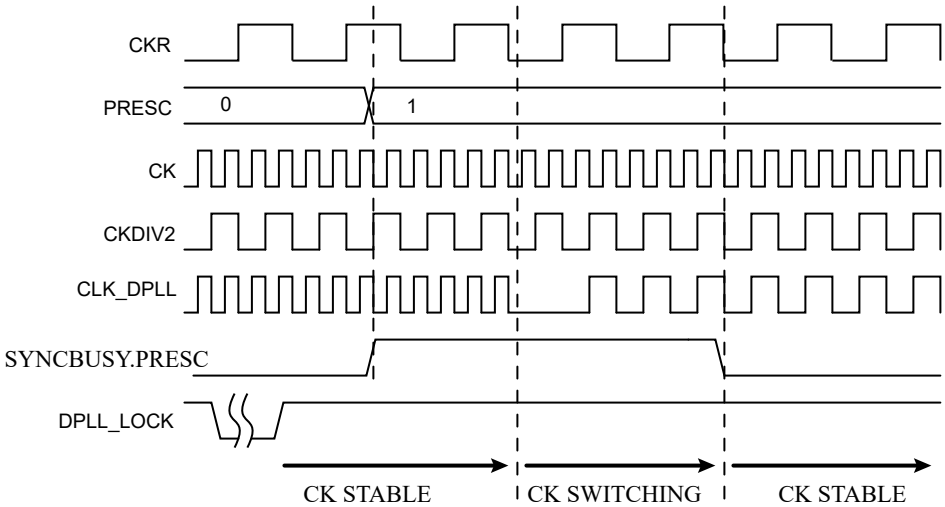
23.6.7.1.2 Reference Clock Switching

When a software operation requires reference clock switching, the recommended procedure is to turn the DPLL into the standby mode, modify the DPLLCTRLB.REFCLK to select the desired reference source, and activate the DPLL again.

23.6.7.1.3 Output Clock Prescaler

The DPLL controller includes an output prescaler. This prescaler provides three selectable output clocks CK, CKDIV2 and CKDIV4. The Prescaler bit field in the DPLL Prescaler register (DPLLPRESC.PRESC) is used to select a new output clock prescaler. When the prescaler field is modified, the DPLLSYNCBUSY.DPLLPRESC bit is set. It will be cleared by hardware when the synchronization is over.

Figure 23-6. Output Clock Switching Operation



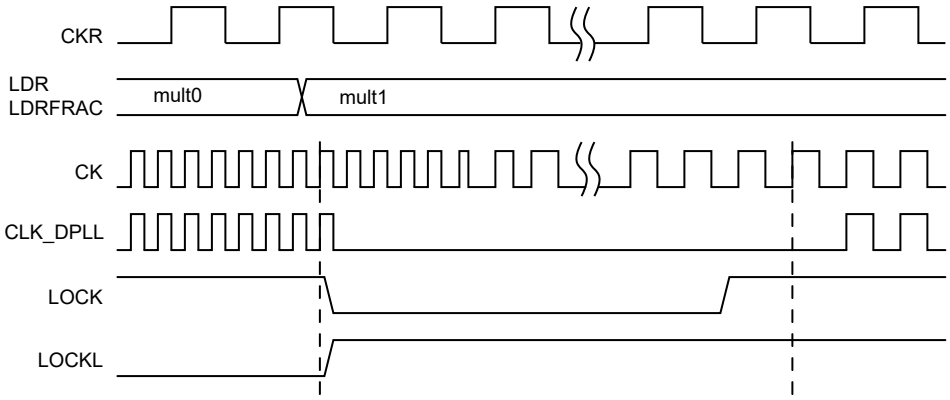
23.6.7.1.4 Loop Divider Ratio Updates

The DPLL Controller supports on-the-fly update of the DPLL Ratio Control (DPLLRATIO) register, allowing to modify the loop divider ratio and the loop divider ratio fractional part when the DPLL is enabled.

Downloaded from Arrow.com.

STATUS.DPLLLDRTO is set when the DPLLRATIO register has been modified and the DPLL analog cell has successfully sampled the updated value. At that time the DPLLSTATUS.LOCK bit is cleared and set again by hardware when the output frequency reached a stable state.

Figure 23-7. RATIOCTRL register update operation



23.6.7.1.5 Digital Filter Selection

The PLL digital filter (PI controller) is automatically adjusted in order to provide a good compromise between stability and jitter. Nevertheless a software operation can override the filter setting using the Filter bit field in the DPLL Control B register (DPLLCTRLB.FILTER). The Low Power Enable bit (DPLLCTRLB.LPEN) can be use to bypass the Time to Digital Converter (TDC) module.

23.6.8 DMA Operation

Not applicable.

23.6.9 Interrupts

The OSCCTRL has the following interrupt sources:

- XOSCRDY - Multipurpose Crystal Oscillator Ready: A 0-to-1 transition on the STATUS.XOSCRDY bit is detected
- CLKFAIL - Clock Failure. A 0-to-1 transition on the STATUS.CLKFAIL bit is detected
- OSC16MRDY - 16MHz Internal Oscillator Ready: A 0-to-1 transition on the STATUS.OSC16MRDY bit is detected
- DFLLULP-related:
  - DFLLULPRDY - DFLLULP Ready: A 0-to1 transition of the STATUS.DFLLULPRDY bit is detected.
  - DFLLULPLOCK - DFLLULP Lock: A 0-to-1 transition of the STATUS.DFLLULPLOCK bit is detected.
  - DFLLULPNOLOCK - DFLLULP No Lock: A 0-to-1 transition of the STATUS.DFLLULPNOLOCK bit is detected.
- DPLL-related:
  - DPLLLOCKR - DPLL Lock Rise: A 0-to-1 transition of the STATUS.DPLLLOCKR bit is detected
  - DPLLLOCKF - DPLL Lock Fall: A 0-to-1 transition of the STATUS.DPLLLOCKF bit is detected
  - DPLLLTTO - DPLL Lock Timer Time-out: A 0-to-1 transition of the STATUS.DPLLLTTO bit is detected
  - DPLLLDRTO - DPLL Loop Divider Ratio Update Complete. A 0-to-1 transition of the STATUS.DPLLLDRTO bit is detected

Downloaded from Arrow.com.

All these interrupts are synchronous wake-up source.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition occurs.

Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the OSCCTRL is reset. See the INTFLAG register for details on how to clear interrupt flags.

The OSCCTRL has one common interrupt request line for all the interrupt sources. The user must read the INTFLAG register to determine which interrupt condition is present. Refer to the INTFLAG register for details.

Note: The interrupts must be globally enabled for interrupt requests to be generated.

23.6.10 Events

The CFD can generate the following output event:

- Clock Failure (CLKFAIL): Generated when the Clock Failure status bit is set in the Status register (STATUS.CLKFAIL). The CFD event is not generated when the Clock Switch bit (STATUS.CLKSW) in the Status register is set.

Writing a '1' to an Event Output bit in the Event Control register (EVCTRL.CFDEO) enables the CFD output event. Writing a '0' to this bit disables the CFD output event. Refer to the *Event System* chapter for details on configuring the event system.

The DFLLULP can take the following actions on an input event:

- Unlock the DFLLULP close loop tuner and start over a frequency tuning, depending on the settings of the DFLLULP registers, until the tuner achieves a new lock.

Writing a '1' to the Event Input Enable bit in the Event Control register (EVCTRL.TUNEEI) enables the corresponding action on input event. Writing a '0' to this bit disables the corresponding action on input event. Refer to the *Event System* chapter for details on configuring the event system.

23.6.11 Synchronization

DFLLULP

Due to the asynchronicity between the main clock domain (CLK_OSCCTRL_APB) and the internal clock domain, some registers are synchronized when written. When a write-synchronized register is written, the corresponding bit in the Synchronization Busy register (DFLLULPSYNCBUSY) is set immediately. When the write-synchronization is complete, this bit is cleared. Reading a write-synchronized register while the synchronization is ongoing will return the value written, and not the current value in the peripheral clock domain. To read the current value in the peripheral clock domain after writing a register, the user must wait for the corresponding DFLLULPSYNCBUSY bit to be cleared before reading the value.

If a register is written while the corresponding bit in DFLLULPSYNCBUSY is one, the write is discarded and an error is generated.

The following bits and registers are write-synchronized:

- Delay Value register (DFLLULPDLY)

Write-synchronization is denoted by the Write-Synchronized property in the register description.

Downloaded from Arrow.com.

FDPLL96M

Due to the multiple clock domains, some registers in the FDPLL96M must be synchronized when accessed.

When executing an operation that requires synchronization, the relevant synchronization bit in the Synchronization Busy register (DPLLSYNCBUSY) will be set immediately, and cleared when synchronization is complete.

The following bits need synchronization when written:

- Enable bit in control register A (DPLLCTRLA.ENABLE)
- DPLL Ratio register (DPLLRATIO)
- DPLL Prescaler register (DPLLPRESC)

Downloaded from Arrow.com.

23.7 Register Summary

Offset	Name	Bit Pos.								
0x00	EVCTRL	7:0						TUNEINV	TUNEEI	CFDEO
0x01 ... 0x03	Reserved									
0x04	INTENCLR	7:0				OSC16MRDY			CLKFAIL	XOSCRDY
		15:8						DFLLULPNOLOCK	DFLLULPLOCK	DFLLULPRDY
		23:16					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
		31:24								
0x08	INTENSET	7:0				OSC16MRDY			CLKFAIL	XOSCRDY
		15:8						DFLLULPNOLOCK	DFLLULPLOCK	DFLLULPRDY
		23:16					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
		31:24								
0x0C	INTFLAG	7:0				OSC16MRDY			CLKFAIL	XOSCRDY
		15:8						DFLLULPNOLOCK	DFLLULPLOCK	DFLLULPRDY
		23:16					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
		31:24								
0x10	STATUS	7:0				OSC16MRDY		CLKSW	CLKFAIL	XOSCRDY
		15:8						DFLLULPNOLOCK	DFLLULPLOCK	DFLLULPRDY
		23:16					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
		31:24								
0x14	XOSCCTRL	7:0	ONDEMAND	RUNSTDBY		SWBACK	CFDEN	XTALEN	ENABLE	
		15:8	STARTUP[3:0]				AMPGC	GAIN[2:0]		
0x16	CFDPRESC	7:0						CFDPRESC[2:0]		
0x17	Reserved									
0x18	OSC16MCTRL	7:0	ONDEMAND	RUNSTDBY			FSEL[1:0]		ENABLE	
0x19 ... 0x1B	Reserved									
0x1C	DFLLULPCTRL	7:0	ONDEMAND	RUNSTDBY	DITHER	SAFE	BINSE		ENABLE	
		15:8						DIV[2:0]		
0x1E	DFLLULPDITHER	7:0		PER[2:0]				STEP[2:0]		
0x1F	DFLLULPRREQ	7:0	RREQ							
0x20	DFLLULPDLY	7:0	DELAY[7:0]							
		15:8								
		23:16								
		31:24								
0x24	DFLLULPRATIO	7:0	RATIO[7:0]							
		15:8						RATIO[10:8]		
		23:16								
		31:24								

Downloaded from Arrow.com.

Offset	Name	Bit Pos.								
0x28	DFLLULPSYNCBUSY	7:0					DELAY		ENABLE	
		15:8								
		23:16								
		31:24								
0x2C	DPLLCTRLA	7:0	ONDEMAND	RUNSTDBY					ENABLE	
0x2D ... 0x2F	Reserved									
0x30	DPLLRATIO	7:0	LDR[7:0]							
		15:8					LDR[11:8]			
		23:16					LDRFRAC[3:0]			
		31:24								
0x34	DPLLCTRLB	7:0			REFCLK[1:0]		WUF	LPEN	FILTER[1:0]	
		15:8				LBYPASS		LTIME[2:0]		
		23:16	DIV[7:0]							
		31:24						DIV[10:8]		
0x38	DPLLPRESC	7:0							PRESC[1:0]	
0x39 ... 0x3B	Reserved									
0x3C	DPLLSYNCBUSY	7:0					DPLLPRESC	DPLLRATIO	ENABLE	
0x3D ... 0x3F	Reserved									
0x40	DPLLSTATUS	7:0							CLKRDY	LOCK

23.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Write-protection is denoted by the "PAC Write-Protection" property in each individual register description. Refer to the 23.5.8 Register Access Protection section and the PAC - Peripheral Access Controller chapter for details.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" or "Write-Synchronized" property in each individual register description. Refer to the section on Synchronization for details.

Downloaded from Arrow.com.

23.8.1 Event Control

Name: EVCTRL
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
						TUNEINV	TUNEEI	CFDEO
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – TUNEINV Tune Event Input Invert
This bit is used to invert the input event of the DFLLULP tuner.

Value	Description
0	Tune event input source is not inverted.
1	Tune event input source is inverted.

Bit 1 – TUNEEI Tune Event Input Enable
This bit is used to enable the input event of the DFLLULP tuner.

Value	Description
0	A new closed loop tuning will not be triggered on any incoming event.
1	A new closed loop tuning will be triggered on any incoming event.

Bit 0 – CFDEO Clock Failure Detector Event Output Enable
This bit indicates whether the Clock Failure detector event output is enabled or not and an output event will be generated when the Clock Failure detector detects a clock failure

Value	Description
0	Clock Failure detector event output is disabled and no event will be generated.
1	Clock Failure detector event output is enabled and an event will be generated.

Downloaded from Arrow.com.

23.8.2 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
						DFLLULPNOLOCK	DFLLULPLOCK	DFLLULPRDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
				OSC16MRDY			CLKFAIL	XOSCRDY
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 19 – DPLLLDRTO DPLL Loop Divider Ratio Update Complete Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the DPLL Loop Divider Ratio Update Complete Interrupt Enable bit, which disables the DPLL Loop Divider Ratio Update Complete interrupt.

Value	Description
0	The DPLL Loop Divider Ratio Update Complete interrupt is disabled.
1	The DPLL Loop Divider Ratio Update Complete interrupt is enabled, and an interrupt request will be generated when the DPLL Loop Divider Ratio Update Complete Interrupt flag is set.

Bit 18 – DPLLLTO DPLL Lock Timeout Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the DPLL Lock Timeout Interrupt Enable bit, which disables the DPLL Lock Timeout interrupt.

Downloaded from Arrow.com.

Value	Description
0	The DPLL Lock Timeout interrupt is disabled.
1	The DPLL Lock Timeout interrupt is enabled, and an interrupt request will be generated when the DPLL Lock Timeout Interrupt flag is set.

Bit 17 – DPLLLCKF DPLL Lock Fall Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the DPLL Lock Fall Interrupt Enable bit, which disables the DPLL Lock Fall interrupt.

Value	Description
0	The DPLL Lock Fall interrupt is disabled.
1	The DPLL Lock Fall interrupt is enabled, and an interrupt request will be generated when the DPLL Lock Fall Interrupt flag is set.

Bit 16 – DPLLLCKR DPLL Lock Rise Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the DPLL Lock Rise Interrupt Enable bit, which disables the DPLL Lock Rise interrupt.

Value	Description
0	The DPLL Lock Rise interrupt is disabled.
1	The DPLL Lock Rise interrupt is enabled, and an interrupt request will be generated when the DPLL Lock Rise Interrupt flag is set.

Bit 10 – DFLLULPNOLOCK DFLLULP No Lock Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the DFLLULP No Lock interrupt Enable bit, which disables the DFLLULP No Lock interrupt.

Value	Description
0	The DFLLULP No Lock is disabled.
1	The DFLLULP No Lock interrupt is enabled, and an interrupt request will be generated when the DFLLULP No Lock Interrupt flag is set.

Bit 9 – DFLLULPLOCK DFLLULP Lock Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the DFLLULP Lock Interrupt Enable bit, which disables the DFLLULP Lock interrupt.

Value	Description
0	The DFLLULP Lock interrupt is disabled.
1	The DFLLULP Lock interrupt is enabled, and an interrupt request will be generated when the DFLLULP Lock Interrupt flag is set.

Bit 8 – DFLLULPRDY DFLLULP Ready interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the DFLLULP Ready Interrupt Enable bit, which disables the DFLLULP Ready interrupt.

Downloaded from Arrow.com.

Value	Description
0	The DFLLULP Ready interrupt is disabled.
1	The DFLLULP Ready interrupt is enabled, and an interrupt request will be generated when the DFLLULP Ready Interrupt flag is set.

Bit 4 – OSC16MRDY OSC16M Ready Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the OSC16M Ready Interrupt Enable bit, which disables the OSC16M Ready interrupt.

Value	Description
0	The OSC16M Ready interrupt is disabled.
1	The OSC16M Ready interrupt is enabled, and an interrupt request will be generated when the OSC16M Ready Interrupt flag is set.

Bit 1 – CLKFAIL Clock Failure Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the XOSC Clock Failure Interrupt Enable bit, which disables the XOSC Clock Failure interrupt.

Value	Description
0	The XOSC Clock Failure interrupt is disabled.
1	The XOSC Clock Failure interrupt is enabled, and an interrupt request will be generated when the XOSC Clock Failure Interrupt flag is set.

Bit 0 – XOSCRDY XOSC Ready Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the XOSC Ready Interrupt Enable bit, which disables the XOSC Ready interrupt.

Value	Description
0	The XOSC Ready interrupt is disabled.
1	The XOSC Ready interrupt is enabled, and an interrupt request will be generated when the XOSC Ready Interrupt flag is set.

Downloaded from Arrow.com.

23.8.3 Interrupt Enable Set

Name: INTENSET
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
						DFLLULPNOLOCK	DFLLULPLOCK	DFLLULPRDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
				OSC16MRDY			CLKFAIL	XOSCRDY
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 19 – DPLLLDRTO DPLL Loop Divider Ratio Update Complete Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the DPLL Loop Ratio Update Complete Interrupt Enable bit, which enables the DPLL Loop Ratio Update Complete interrupt.

Value	Description
0	The DPLL Loop Divider Ratio Update Complete interrupt is disabled.
1	The DPLL Loop Ratio Update Complete interrupt is enabled, and an interrupt request will be generated when the DPLL Loop Ratio Update Complete Interrupt flag is set.

Bit 18 – DPLLLTO DPLL Lock Timeout Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the DPLL Lock Timeout Interrupt Enable bit, which enables the DPLL Lock Timeout interrupt.

Downloaded from Arrow.com.

Value	Description
0	The DPLL Lock Timeout interrupt is disabled.
1	The DPLL Lock Timeout interrupt is enabled, and an interrupt request will be generated when the DPLL Lock Timeout Interrupt flag is set.

Bit 17 – DPLLLCKF DPLL Lock Fall Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the DPLL Lock Fall Interrupt Enable bit, which enables the DPLL Lock Fall interrupt.

Value	Description
0	The DPLL Lock Fall interrupt is disabled.
1	The DPLL Lock Fall interrupt is enabled, and an interrupt request will be generated when the DPLL Lock Fall Interrupt flag is set.

Bit 16 – DPLLLCKR DPLL Lock Rise Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the DPLL Lock Rise Interrupt Enable bit, which enables the DPLL Lock Rise interrupt.

Value	Description
0	The DPLL Lock Rise interrupt is disabled.
1	The DPLL Lock Rise interrupt is enabled, and an interrupt request will be generated when the DPLL Lock Rise Interrupt flag is set.

Bit 10 – DFLLULPNOLOCK DFLLULP No Lock Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the DFLLULP No Lock interrupt Enable bit, which enables the DFLLULP No Lock interrupt.

Value	Description
0	The DFLLULP No Lock is disabled.
1	The DFLL No Lock interrupt is enabled, and an interrupt request will be generated when the DFLL No Lock Interrupt flag is set.

Bit 9 – DFLLULPLOCK DFLLULP Lock Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the DFLLULP Lock Interrupt Enable bit, which enables the DFLLULP Lock interrupt.

Value	Description
0	The DFLLULP Lock interrupt is disabled.
1	The DFLLULP Lock interrupt is enabled, and an interrupt request will be generated when the DFLL Lock Interrupt flag is set.

Bit 8 – DFLLULPRDY DFLLULP Ready interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the DFLLULP Ready Interrupt Enable bit, which enables the DFLLULP Ready interrupt.

Downloaded from Arrow.com.

Value	Description
0	The DFLLULP Ready interrupt is disabled.
1	The DFLLULP Ready interrupt is enabled, and an interrupt request will be generated when the DFLLULP Ready Interrupt flag is set.

Bit 4 – OSC16MRDY OSC16M Ready Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the OSC16M Ready Interrupt Enable bit, which enables the OSC16M Ready interrupt.

Value	Description
0	The OSC16M Ready interrupt is disabled.
1	The OSC16M Ready interrupt is enabled, and an interrupt request will be generated when the OSC16M Ready Interrupt flag is set.

Bit 1 – CLKFAIL XOSC Clock Failure Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the XOSC Clock Failure Interrupt Enable bit, which enables the XOSC Clock Failure Interrupt.

Value	Description
0	The XOSC Clock Failure Interrupt is disabled.
1	The XOSC Clock Failure Interrupt is enabled, and an interrupt request will be generated when the XOSC Clock Failure Interrupt flag is set.

Bit 0 – XOSCRDY XOSC Ready Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the XOSC Ready Interrupt Enable bit, which enables the XOSC Ready interrupt.

Value	Description
0	The XOSC Ready interrupt is disabled.
1	The XOSC Ready interrupt is enabled, and an interrupt request will be generated when the XOSC Ready Interrupt flag is set.

Downloaded from Arrow.com.

23.8.4 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x0C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
						DFLLULPNOLOCK	DFLLULPLOCK	DFLLULPRDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
				OSC16MRDY			CLKFAIL	XOSCRDY
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 19 – DPLLLDRTO DPLL Loop Divider Ratio Update Complete
This flag is cleared by writing '1' to it.

This flag is set on a high to low transition of the DPLL Loop Divider Ratio Update Complete bit in the Status register (STATUS.DPLLLDRTO) and will generate an interrupt request if INTENSET.DPLLLDRTO is '1'.

Writing '0' to this bit has no effect.

Writing '1' to this bit clears the DPLL Loop Divider Ratio Update Complete interrupt flag.

Bit 18 – DPLLLTO DPLL Lock Timeout
This flag is cleared by writing '1' to it.

This flag is set on 0-to-1 transition of the DPLL Lock Timeout bit in the Status register (STATUS.DPLLLTO) and will generate an interrupt request if INTENSET.DPLLLTO is '1'.

Writing '0' to this bit has no effect.

Writing '1' to this bit clears the DPLL Lock Timeout interrupt flag.

Bit 17 – DPLLLCKF DPLL Lock Fall
This flag is cleared by writing '1' to it.

Downloaded from Arrow.com.

This flag is set on 0-to-1 transition of the DPLL Lock Fall bit in the Status register (STATUS.DPLLLCKF) and will generate an interrupt request if INTENSET.DPLLLCKF is '1'.

Writing '0' to this bit has no effect.

Writing '1' to this bit clears the DPLL Lock Fall interrupt flag.

Bit 16 – DPLLLCKR DPLL Lock Rise
This flag is cleared by writing '1' to it.

This flag is set on 0-to-1 transition of the DPLL Lock Rise bit in the Status register (STATUS.DPLLLCKR) and will generate an interrupt request if INTENSET.DPLLLCKR is '1'.

Writing '0' to this bit has no effect.

Writing '1' to this bit clears the DPLL Lock Rise interrupt flag.

Bit 10 – DFLLULPNOLOCK DFLLULP No Lock
This flag is cleared by writing '1' to it.

This flag is set on 0-to-1 transition of the DFLLULP No Lock bit in the Status register (STATUS.DFLLULPNOLOCK) and will generate an interrupt request if INTENSET.DFLLULPNOLOCK is '1'.

Writing '0' to this bit has no effect.

Writing '1' to this bit clears the DFLLULP No Lock interrupt flag.

Bit 9 – DFLLULPLOCK DFLLULP Lock
This flag is cleared by writing '1' to it.

This flag is set on 0-to-1 transition of the DFLLULP Lock bit in the Status register (STATUS.DFLLULPLOCK) and will generate an interrupt request if INTENSET.DFLLULPLOCK is '1'.

Writing '0' to this bit has no effect.

Writing '1' to this bit clears the DFLLULP Lock interrupt flag.

Bit 8 – DFLLULPRDY DFLLULP Ready
This flag is cleared by writing '1' to it.

This flag is set on 0-to-1 transition of the DFLLULP Ready bit in the Status register (STATUS.DFLLULPREADY) and will generate an interrupt request if INTENSET.DFLLULPREADY is '1'.

Writing '0' to this bit has no effect.

Writing '1' to this bit clears the DFLLULP Ready interrupt flag.

Bit 4 – OSC16MRDY OSC16M Ready
This flag is cleared by writing '1' to it.

This flag is set on 0-to-1 transition of the OSC16M Ready bit in the Status register (STATUS.OSC16MRDY) and will generate an interrupt request if INTENSET.OSC16MRDY is '1'.

Writing '0' to this bit has no effect.

Writing '1' to this bit clears the OSC16M Ready interrupt flag.

Downloaded from Arrow.com.

Bit 1 – CLKFAIL XOSC Failure Detection
This flag is cleared by writing '1' to it.

This flag is set on a 0-to-1 transition of the XOSC Clock Failure bit in the Status register (STATUS.CLKFAIL) and will generate an interrupt request if INTENSET.CLKFAIL is '1'.

Writing '0' to this bit has no effect.

Writing '1' to this bit clears the XOSC Clock Fail interrupt flag.

Bit 0 – XOSCRDY XOSC Ready
This flag is cleared by writing '1' to it.

This flag is set on a 0-to-1 transition of the XOSC Ready bit in the Status register (STATUS.XOSCRDY) and will generate an interrupt request if INTENSET.XOSCRDY is '1'.

Writing '0' to this bit has no effect.

Writing '1' to this bit clears the XOSC Ready interrupt flag.

Downloaded from Arrow.com.

23.8.5 Status

Name: STATUS
Offset: 0x10
Reset: 0x00000100
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
Access					R	R	R	R
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
						DFLLULPNOLOCK	DFLLULPLOCK	DFLLULPRDY
Access						R	R	R
Reset						0	0	1

Bit	7	6	5	4	3	2	1	0
				OSC16MRDY		CLKSW	CLKFAIL	XOSCRDY
Access				R		R	R	R
Reset				0		0	0	0

Bit 19 – DPLLLDRTO DPLL Loop Divider Ratio Update Complete

Value	Description
0	DPLL Loop Divider Ratio Update Complete not detected.
1	DPLL Loop Divider Ratio Update Complete detected.

Bit 18 – DPLLLTO DPLL Lock Timeout

Value	Description
0	DPLL Lock time-out not detected.
1	DPLL Lock time-out detected.

Bit 17 – DPLLLCKF DPLL Lock Fall

Value	Description
0	DPLL Lock fall edge not detected.
1	DPLL Lock fall edge detected.

Bit 16 – DPLLLCKR DPLL Lock Rise

Downloaded from Arrow.com.

Value	Description
0	DPLL Lock rise edge not detected.
1	DPLL Lock rise edge detected.

Bit 10 – DFLLULPNOLOCK DFLLULP No Lock

Value	Description
0	DFLLULP Tuner no lock state is not detected.
1	DFLLULP Tuner no lock state is detected.

Bit 9 – DFLLULPLOCK DFLLULP Lock

Value	Description
0	DFLLULP Tuner lock state is not detected.
1	DFLLULP Tuner lock state is detected.

Bit 8 – DFLLULPRDY DFLLULP Ready

Value	Description
0	DFLLULP is not ready.
1	DFLLULP is stable and ready to be used as a clock source.

Bit 4 – OSC16MRDY OSC16M Ready

Value	Description
0	OSC16M is not ready.
1	OSC16M is stable and ready to be used as a clock source.

Bit 2 – CLKSW XOSC Clock Switch

Value	Description
0	XOSC is not switched and provides the external clock or crystal oscillator clock.
1	XOSC is switched and provides the safe clock.

Bit 1 – CLKFAIL XOSC Clock Failure

Value	Description
0	No XOSC failure detected.
1	A XOSC failure was detected.

Bit 0 – XOSCRDY XOSC Ready

Value	Description
0	XOSC is not ready.
1	XOSC is stable and ready to be used as a clock source.

Downloaded from Arrow.com.

23.8.6 External Multipurpose Crystal Oscillator (XOSC) Control

Name: XOSCCTRL
Offset: 0x14
Reset: 0x0080
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
	STARTUP[3:0]				AMPGC	GAIN[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY		SWBACK	CFDEN	XTALEN	ENABLE	
Access	R/W	R/W		R/W	R/W	R/W	R/W	
Reset	1	0		0	0	0	0	

Bits 15:12 – STARTUP[3:0] Start-Up Time
These bits select start-up time for the oscillator.

The OSCULP32K oscillator is used to clock the start-up counter.

Table 23-5. Start-Up Time for External Multipurpose Crystal Oscillator

STARTUP[3:0]	Number of OSCULP32K Clock Cycles	Number of XOSC Clock Cycles	Approximate Equivalent Time [µs]
0x0	1	3	31
0x1	2	3	61
0x2	4	3	122
0x3	8	3	244
0x4	16	3	488
0x5	32	3	977
0x6	64	3	1953
0x7	128	3	3906
0x8	256	3	7813
0x9	512	3	15625
0xA	1024	3	31250
0xB	2048	3	62500µs
0xC	4096	3	125000
0xD	8192	3	250000

Downloaded from Arrow.com.

STARTUP[3:0]	Number of OSCULP32K Clock Cycles	Number of XOSC Clock Cycles	Approximate Equivalent Time [µs]
0xE	16384	3	500000
0xF	32768	3	1000000

Note:

1. Actual startup time is 1 OSCULP32K cycle + 3 XOSC cycles.
2. The given time neglects the three XOSC cycles before OSCULP32K cycle.

Bit 11 – AMPGC Automatic Amplitude Gain Control
Note: This bit must be set only after the XOSC has settled, indicated by the XOSC Ready flag in the Status register (STATUS.XOSCRDY).

Value	Description
0	The automatic amplitude gain control is disabled.
1	The automatic amplitude gain control is enabled. Amplitude gain will be automatically adjusted during Crystal Oscillator operation.

Bits 10:8 – GAIN[2:0] Oscillator Gain
These bits select the gain for the oscillator. The listed maximum frequencies are recommendations, and might vary based on capacitive load and crystal characteristics. Those bits must be properly configured even when the Automatic Amplitude Gain Control is active.

Value	Recommended Max Frequency [MHz]
0x0	2
0x1	4
0x2	8
0x3	16
0x4	30
0x5-0x7	Reserved

Bit 7 – ONDEMAND On Demand Control
The On Demand operation mode allows the oscillator to be enabled or disabled, depending on peripheral clock requests.

If the ONDEMAND bit has been previously written to '1', the oscillator will be running only when requested by a peripheral. If there is no peripheral requesting the oscillator's clock source, the oscillator will be in a disabled state.

If On Demand is disabled, the oscillator will always be running when enabled.

In standby sleep mode, the On Demand operation is still active.

Value	Description
0	The oscillator is always on, if enabled.
1	The oscillator is enabled when a peripheral is requesting the oscillator to be used as a clock source. The oscillator is disabled if no peripheral is requesting the clock source.

Downloaded from Arrow.com.

Bit 6 – RUNSTDBY Run in Standby
This bit controls how the XOSC behaves during Standby Sleep mode, together with the ONDEMAND bit:

Value	Description
0	The XOSC is not running in Standby sleep mode if no peripheral requests the clock.
1	The XOSC is running in Standby sleep mode. If ONDEMAND=1, the XOSC will be running when a peripheral is requesting the clock. If ONDEMAND=0, the clock source will always be running in Standby sleep mode.

Bit 4 – SWBACK Clock Switch Back
This bit controls the XOSC output switch back to the external clock or crystal oscillator in case of clock recovery:

Value	Description
0	The clock switch back is disabled.
1	The clock switch back is enabled. This bit is reset once the XOSC output clock is switched back to the external clock or crystal oscillator.

Bit 3 – CFDEN Clock Failure Detector Enable
This bit controls the clock failure detector:

Value	Description
0	The Clock Failure Detector is disabled.
1	the Clock Failure Detector is enabled.

Bit 2 – XTALEN Crystal Oscillator Enable
This bit controls the connections between the I/O pads and the external clock or crystal oscillator:

Value	Description
0	External clock connected on XIN. XOUT can be used as general-purpose I/O.
1	Crystal connected to XIN/XOUT.

Bit 1 – ENABLE Oscillator Enable

Value	Description
0	The oscillator is disabled.
1	The oscillator is enabled.

Downloaded from Arrow.com.

23.8.7 Clock Failure Detector Prescaler

Name: CFDPRESC
Offset: 0x16
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
						CFDPRESC[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:0 – CFDPRESC[2:0] Clock Failure Detector Prescaler
These bits select the prescaler for the clock failure detector.

The OSC16M oscillator is used to clock the CFD prescaler. The CFD safe clock frequency is the OSC16M frequency divided by 2^CFDPRESC.

Downloaded from Arrow.com.

23.8.8 16MHz Internal Oscillator (OSC16M) Control

Name: OSC16MCTRL
Offset: 0x18
Reset: 0x82
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY			FSEL[1:0]		ENABLE	
Access	R/W	R/W			R/W	R/W	R/W	
Reset	1	0			0	0	1	

Bit 7 – ONDEMAND On Demand Control
The On Demand operation mode allows the oscillator to be enabled or disabled depending on peripheral clock requests.

If the ONDEMAND bit has been previously written to '1', the oscillator will only be running when requested by a peripheral. If there is no peripheral requesting the oscillator's clock source, the oscillator will be in a disabled state.

If On Demand is disabled the oscillator will always be running when enabled.

In standby sleep mode, the On Demand operation is still active.

Value	Description
0	The oscillator is always on, if enabled.
1	The oscillator is enabled when a peripheral is requesting the oscillator to be used as a clock source. The oscillator is disabled if no peripheral is requesting the clock source.

Bit 6 – RUNSTDBY Run in Standby
This bit controls how the OSC16M behaves during standby sleep mode.

Value	Description
0	The OSC16M is disabled in standby sleep mode if no peripheral requests the clock.
1	The OSC16M is not stopped in standby sleep mode. If ONDEMAND=1, the OSC16M will be running when a peripheral is requesting the clock. If ONDEMAND=0, the clock source will always be running in standby sleep mode.

Bits 3:2 – FSEL[1:0] Oscillator Frequency Selection
These bits control the oscillator frequency range.

Value	Description
0x00	4MHz
0x01	8MHz
0x10	12MHz
0x11	16MHz

Bit 1 – ENABLE Oscillator Enable

Downloaded from Arrow.com.

Value	Description
0	The oscillator is disabled.
1	The oscillator is enabled.

Downloaded from Arrow.com.

23.8.9 DFLLULP Control

Name: DFLLULPCTRL
Offset: 0x1C
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected, Write-synchronized

Bit	15	14	13	12	11	10	9	8
						DIV[2:0]		
Access	R	R	R	R	R	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY	DITHER	SAFE	BINSE		ENABLE	
Access	R/W	R/W	R/W	R/W	R/W		R/W	R
Reset	0	0	0	0	0		0	0

Bits 10:8 – DIV[2:0] Division Factor
This field defines the division factor for the output frequency of the DFLLULP.

This value from production test, which depends on PL0 or PL2 mode, must be copied from the NVM software calibration row into the DFLLULPCTRL register by software.

The value must be changed before switching on a new Performance Level mode (PL0 or PL2).

These bits are not synchronized.

Value	Name	Description
0x0	DIV1	Frequency divided by 1
0x1	DIV2	Frequency divided by 2
0x2	DIV4	Frequency divided by 4
0x3	DIV8	Frequency divided by 8
0x4	DIV16	Frequency divided by 16
0x5	DIV32	Frequency divided by 32
0x6 – 0x7	-	Reserved

Bit 7 – ONDEMAND On Demand
The On Demand operation mode allows the oscillator to be enabled or disabled, depending on peripheral clock requests.

If the ONDEMAND bit has been previously written to '1', the oscillator will be running only when requested by a peripheral. If there is no peripheral requesting the oscillator's clock source, the oscillator will be in a disabled state.

If On Demand is disabled, the oscillator will always be running when enabled.

In standby sleep mode, the On Demand operation is still active.

This bit is not enabled-protected. This bit is not synchronized.

Downloaded from Arrow.com.

Bit 6 – RUNSTDBY Run in Standby
This bit controls how the DFLLULP behaves during standby sleep mode, together with the ONDEMAND bit.
This bit is not enabled-protected. This bit is not synchronized

Bit 5 – DITHER Tuner Dither Mode
This bit is not synchronized.

Value	Description
0	The dither mode is disabled.
1	The dither mode is enabled if tuning is enabled (DFLLULPCTRL.TUNE = 1).

Bit 4 – SAFE Tuner Safe Mode
This bit is not synchronized.

Value	Description
0	The clock output is not masked while binary search tuning is ongoing.
1	The clock output is masked while binary search tuning is ongoing (DFLLULPCTRL.BINSE = 1).

Bit 3 – BINSE Binary Search Enable
This bit is not synchronized.

Value	Description
0	Binary search tuning is disabled. Maximum number of reference clock cycles to acquire lock is 256.
1	Binary search tuning is enabled. Maximum number of reference clock cycles to acquire lock is 8.

Bit 1 – ENABLE Enable
This bit is not enable-protected.

Value	Description
0	The DFLLULP is disabled.
1	The DFLLULP is enabled.

Downloaded from Arrow.com.

23.8.10 DFLLULP Dither Control

Name: DFLLULPDITHER
Offset: 0x1E
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
		PER[2:0]				STEP[2:0]		
Access	R	R/W	R/W	R/W	R	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 6:4 – PER[2:0] Dither Period
These bits define the number of reference clock periods over which dithering is applied.

Value	Name	Description
0x0	PER1	Dither over 1 reference clock period
0x1	PER2	Dither over 2 reference clock periods
0x2	PER4	Dither over 4 reference clock periods
0x3	PER8	Dither over 8 reference clock periods
0x4	PER16	Dither over 16 reference clock periods
0x5	PER32	Dither over 32 reference clock periods
0x6 – 0x7	-	Reserved

Bits 2:0 – STEP[2:0] Dither Step
This field defines the dithering step size.

Value	Name	Description
0x0	STEP1	Dither step = 1
0x1	STEP2	Dither step = 2
0x2	STEP4	Dither step = 4
0x3	STEP8	Dither step = 8
0x4 – 0x7	-	Reserved

Downloaded from Arrow.com.

23.8.11 DFLLULP Read Request

Name: DFLLULPRREQ
Offset: 0x1F
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	RREQ							
Access	R/W	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 7 – RREQ Read Request
Writing a zero to this bit has no effect.

Writing a one to this bit requests synchronization of the DFLLULPDLY register with the current oscillator delay value and sets the Delay Busy bit in the Synchronization Busy register (DFLLULPSYNCBUSY.DELAY).

This bit is cleared automatically when synchronization is complete.

Downloaded from Arrow.com.

23.8.12 DFLLULP Delay Value

Name: DFLLULPDLY
Offset: 0x20
Reset: 0x00000080
Property: PAC Write-Protection, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DELAY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	0	0	0	0

Bits 7:0 – DELAY[7:0] Delay Value
Writing a value to this field sets the oscillator delay. A small value will produce a fast clock and a large value will produce a slow clock. If the tuner is enabled, writing to this field will cause the tuner to start tuning from the written value. Reading this value will return the last written delay or the oscillator delay when a synchronization was requested from the DFLLULPRREQ register. Writing a value to this register while a write synchronization or a read request synchronization is on-going will have no effect and produce a PAC error.

Downloaded from Arrow.com.

23.8.13 DFLLULP Target Ratio

Name: DFLLULPRATIO
Offset: 0x24
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
						RATIO[10:8]		
Access	R	R	R	R	R	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RATIO[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	0	0	0	0

Bits 10:0 – RATIO[10:0] Target Tuner Ratio
Writing a value to this field sets the target ratio between the DFLLULP output clock and the reference clock. The DFLLULPDLY.DELAY value will be updated in such a way that the target ratio and the actual ratio are as close as possible.

Downloaded from Arrow.com.

23.8.14 DFLLULP Synchronization Busy

Name: DFLLULPSYNCBUSY
Offset: 0x28
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					DELAY		ENABLE	
Access	R	R	R	R	R		R	R
Reset	0	0	0	0	0		0	0

Bit 3 – DELAY Delay Register Synchronization Busy
This bit is cleared when the synchronization of DFLLULPDLY is complete.

This bit is set when the synchronization of DFLLULPDLY is started.

Writing this bit has no effect.

Bit 1 – ENABLE Enable Bit Synchronization Busy
This bit is cleared when the synchronization of DFLLULPCTRL.ENABLE is complete.

This bit is set when the synchronization of DFLLULPCTRL.ENABLE is started.

Writing this bit has no effect.

Downloaded from Arrow.com.

23.8.15 DPLL Control A

Name: DPLLCTRLA
Offset: 0x2C
Reset: 0x80
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY					ENABLE	
Access	R/W	R/W					R/W	
Reset	1	0					0	

Bit 7 – ONDEMAND On Demand Clock Activation
The On Demand operation mode allows the DPLL to be enabled or disabled depending on peripheral clock requests.

If the ONDEMAND bit has been previously written to '1', the DPLL will only be running when requested by a peripheral. If there is no peripheral requesting the DPLL's clock source, the DPLL will be in a disabled state.

If On Demand is disabled the DPLL will always be running when enabled.

In standby sleep mode, the On Demand operation is still active.

Value	Description
0	The DPLL is always on, if enabled.
1	The DPLL is enabled when a peripheral is requesting the DPLL to be used as a clock source. The DPLL is disabled if no peripheral is requesting the clock source.

Bit 6 – RUNSTDBY Run in Standby
This bit controls how the DPLL behaves during standby sleep mode:

Value	Description
0	The DPLL is disabled in standby sleep mode if no peripheral requests the clock.
1	The DPLL is not stopped in standby sleep mode. If ONDEMAND=1, the DPLL will be running when a peripheral is requesting the clock. If ONDEMAND=0, the clock source will always be running in standby sleep mode.

Bit 1 – ENABLE DPLL Enable, Write-Synchronized (ENABLE)
The software operation of enabling or disabling the DPLL takes a few clock cycles, so the DPLLSYNCBUSY.ENABLE status bit indicates when the DPLL is successfully enabled or disabled.

Value	Description
0	The DPLL is disabled.
1	The DPLL is enabled.

Downloaded from Arrow.com.

23.8.16 DPLL Ratio Control

Name: DPLLRATIO
Offset: 0x30
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					LDRFRAC[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
					LDR[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 19:16 – LDRFRAC[3:0] Loop Divider Ratio Fractional Part
Writing these bits selects the fractional part of the frequency multiplier. Due to synchronization there is a delay between writing these bits and the effect on the DPLL output clock. The value written will read back immediately and the DPLLRATIO bit in the DPLL Synchronization Busy register (DPLLSYNCBUSY.DPLLRATIO) will be set. DPLLSYNCBUSY.DPLLRATIO will be cleared when the operation is completed.

Bits 11:0 – LDR[11:0] Loop Divider Ratio
Writing these bits selects the integer part of the frequency multiplier. The value written to these bits will read back immediately, and the DPLLRATIO bit in the DPLL Synchronization busy register (DPLLSYNCBUSY.DPLLRATIO), will be set. DPLLSYNCBUSY.DPLLRATIO will be cleared when the operation is completed.

Downloaded from Arrow.com.

23.8.17 DPLL Control B

Name: DPLLCTRLB
Offset: 0x34
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
						DIV[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	DIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
				LBYPASS		LTIME[2:0]		
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bit	7	6	5	4	3	2	1	0
			REFCLK[1:0]		WUF	LPEN	FILTER[1:0]	
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 26:16 – DIV[10:0] Clock Divider
These bits set the XOSC clock division factor and can be calculated with following formula:

$$f_{DIV} = \frac{f_{XOSC}}{2x(DIV + 1)}$$

Bit 12 – LBYPASS Lock Bypass

Value	Description
0	DPLL Lock signal drives the DPLL controller internal logic.
1	DPLL Lock signal is always asserted.

Bits 10:8 – LTIME[2:0] Lock Time
These bits select the lock time-out value:

Note: GCLK_DPLL_32K is responsible for counting the user defined lock time (LTIME different from 0x0), hence must be enabled.

Value	Name	Description
0x0	Default	No time-out. Automatic lock.
0x1	Reserved	
0x2	Reserved	
0x3	Reserved	

Downloaded from Arrow.com.

Value	Name	Description
0x4	8MS	Time-out if no lock within 8ms
0x5	9MS	Time-out if no lock within 9ms
0x6	10MS	Time-out if no lock within 10ms
0x7	11MS	Time-out if no lock within 11ms

Bits 5:4 – REFCLK[1:0] Reference Clock Selection
Write these bits to select the DPLL clock reference:

Value	Name	Description
0x0	XOSC32K	XOSC32K clock reference
0x1	XOSC	XOSC clock reference
0x2	GCLK	GCLK_DPLL clock reference
0x3	Reserved	-

Bit 3 – WUF Wake Up Fast

Value	Description
0	DPLL clock is output after startup and lock time.
1	DPLL clock is output after startup time.

Bit 2 – LPEN Low-Power Enable

Value	Description
0	The low-power mode is disabled. Time to Digital Converter is enabled.
1	The low-power mode is enabled. Time to Digital Converter is disabled. This will improve power consumption but increase the output jitter.

Bits 1:0 – FILTER[1:0] Proportional Integral Filter Selection
These bits select the DPLL filter type:

Value	Name	Description
0x0	DEFAULT	Default filter mode
0x1	LBFILT	Low bandwidth filter
0x2	HBFILT	High bandwidth filter
0x3	HDFILT	High damping filter

Downloaded from Arrow.com.

23.8.18 DPLL Prescaler

Name: DPLLPRESC
Offset: 0x38
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
							PRESC[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 1:0 – PRESC[1:0] Output Clock Prescaler
These bits define the output clock prescaler setting.

Value	Name	Description
0x0	DIV1	DPLL output is divided by 1
0x1	DIV2	DPLL output is divided by 2
0x2	DIV4	DPLL output is divided by 4
0x3	Reserved	

Downloaded from Arrow.com.

23.8.19 DPLL Synchronization Busy

Name: DPLLSYNCBUSY
Offset: 0x3C
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
					DPLLPRESC	DPLLRATIO	ENABLE	
Access					R	R	R	
Reset					0	0	0	

Bit 3 – DPLLPRESC DPLL Prescaler Synchronization Status

Value	Description
0	The DPLLRESC register has been synchronized.
1	The DPLLRESC register value has changed and its synchronization is in progress.

Bit 2 – DPLLRATIO DPLL Loop Divider Ratio Synchronization Status

Value	Description
0	The DPLLRATIO register has been synchronized.
1	The DPLLRATIO register value has changed and its synchronization is in progress.

Bit 1 – ENABLE DPLL Enable Synchronization Status

Value	Description
0	The DPLLCTRLA.ENABLE bit has been synchronized.
1	The DPLLCTRLA.ENABLE bit value has changed and its synchronization is in progress.

Downloaded from Arrow.com.

23.8.20 DPLL Status

Name: DPLLSTATUS
Offset: 0x40
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
							CLKRDY	LOCK
Access							R	R
Reset							0	0

Bit 1 – CLKRDY DPLL Clock Ready

Value	Description
0	The DPLL output clock is off.
1	The DPLL output clock in on.

Bit 0 – LOCK DPLL Lock

Value	Description
0	The DPLL Lock signal is cleared, when the DPLL is disabled or when the DPLL is trying to reach the target frequency.
1	The DPLL Lock signal is asserted when the desired frequency is reached.

Downloaded from Arrow.com.

24. OSC32KCTRL – 32KHz Oscillators Controller

24.1 Overview

The 32KHz Oscillators Controller (OSC32KCTRL) provides a user interface to the 32.768kHz oscillators: XOSC32K and OSCULP32K.

The OSC32KCTRL sub-peripherals can be enabled, disabled, calibrated, and monitored through interface registers.

All sub-peripheral statuses are collected in the Status register (STATUS). They can additionally trigger interrupts upon status changes via the INTENSET, INTENCLR, and INTFLAG registers.

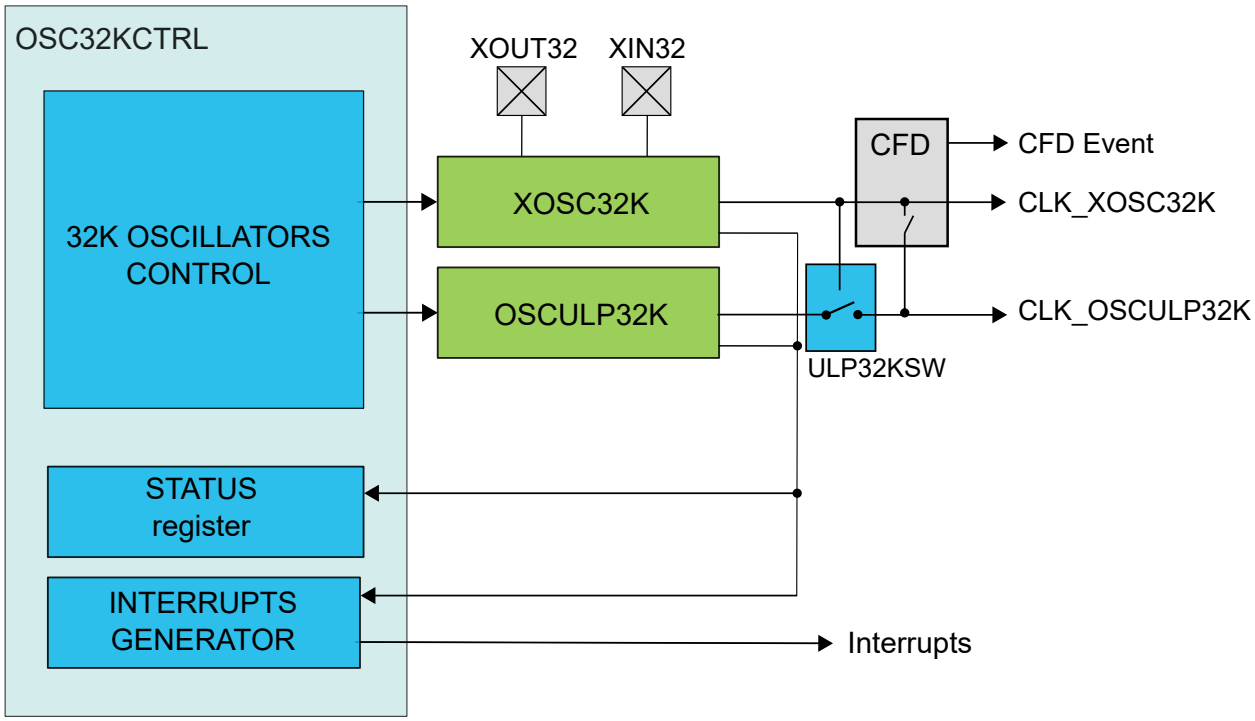
24.2 Features

- 32.768 kHz Crystal Oscillator (XOSC32K)
  - Programmable start-up time
  - Crystal or external input clock on XIN32 I/O
  - Clock failure detection with safe clock switch
  - Clock failure event output
- 32.768 kHz Ultra Low-Power Internal Oscillator (OSCULP32K)
  - Ultra low-power, always-on oscillator
  - Frequency fine tuning
- Calibration value loaded from Flash factory calibration at reset
- 1.024 kHz clock outputs available

Downloaded from Arrow.com.

24.3 Block Diagram

Figure 24-1. OSC32KCTRL Block Diagram



24.4 Signal Description

Signal	Description	Type
XIN32	Analog Input	32.768 kHz Crystal Oscillator or external clock input
XOUT32	Analog Output	32.768 kHz Crystal Oscillator output

The I/O lines are automatically selected when XOSC32K is enabled.

Note: The signal of the external crystal oscillator may affect the jitter of neighboring pads.

24.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

24.5.1 I/O Lines

I/O lines are configured by OSC32KCTRL when XOSC32K is enabled, and need no user configuration.

24.5.2 Power Management

The OSC32KCTRL will continue to operate in any sleep mode where a 32KHz oscillator is running as source clock. The OSC32KCTRL interrupts can be used to wake up the device from sleep modes.

Related Links

22. PM – Power Manager

Downloaded from Arrow.com.

24.5.3 Clocks

The OSC32KCTRL gathers controls for all 32KHz oscillators and provides clock sources to the Generic Clock Controller (GCLK), Real-Time Counter (RTC), and Watchdog Timer (WDT).

The available clock sources are: XOSC32K and OSCULP32K.

The OSC32KCTRL bus clock (CLK_OSC32KCTRL_APB) can be enabled and disabled in the Main Clock module (MCLK).

Related Links

19.6.2.6 Peripheral Clock Masking

24.5.4 Interrupts

The interrupt request lines are connected to the interrupt controller. Using the OSC32KCTRL interrupts requires the interrupt controller to be configured first.

24.5.5 Events

The events of this peripheral are connected to the Event System.

Related Links

33. EVSYS – Event System

24.5.6 Debug Operation

When the CPU is halted in debug mode, OSC32KCTRL will continue normal operation. If OSC32KCTRL is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

24.5.7 Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Interrupt Flag Status and Clear (INTFLAG) register

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

Related Links

15. PAC - Peripheral Access Controller

24.5.8 SAM L11 TrustZone Specific Register Access Protection

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

Downloaded from Arrow.com.

24.5.9 Analog Connections

The external 32.768kHz crystal must be connected between the XIN32 and XOUT32 pins, along with any required load capacitors. For details on recommended oscillator characteristics and capacitor load, refer to the related links.

24.6 Functional Description

24.6.1 Principle of Operation

XOSC32K and OSCULP32K are configured via OSC32KCTRL control registers. Through this interface, the sub-peripherals are enabled, disabled, or have their calibration values updated.

The STATUS register gathers different status signals coming from the sub-peripherals of OSC32KCTRL. The status signals can be used to generate system interrupts, and in some cases wake up the system from standby mode, provided the corresponding interrupt is enabled.

24.6.2 32KHz External Crystal Oscillator (XOSC32K) Operation

The XOSC32K can operate in two different modes:

- External clock, with an external clock signal connected to XIN32
- Crystal oscillator, with an external 32.768kHz crystal connected between XIN32 and XOUT32

At reset, the XOSC32K is disabled, and the XIN32/XOUT32 pins can either be used as General Purpose I/O (GPIO) pins or by other peripherals in the system.

When XOSC32K is enabled, the operating mode determines the GPIO usage. When in crystal oscillator mode, the XIN32 and XOUT32 pins are controlled by the OSC32KCTRL, and GPIO functions are overridden on both pins. When in external clock mode, the only XIN32 pin will be overridden and controlled by the OSC32KCTRL, while the XOUT32 pin can still be used as a GPIO pin.

The XOSC32K is enabled by writing a '1' to the Enable bit in the 32KHz External Crystal Oscillator Control register (XOSC32K.ENABLE=1). The XOSC32K is disabled by writing a '0' to the Enable bit in the 32KHz External Crystal Oscillator Control register (XOSC32K.ENABLE=0).

To enable the XOSC32K as a crystal oscillator, the XTALEN bit in the 32KHz External Crystal Oscillator Control register must be set (XOSC32K.XTALEN=1). If XOSC32K.XTALEN is '0', the external clock input will be enabled.

The XOSC32K 32.768kHz output is enabled by setting the 32KHz Output Enable bit in the 32KHz External Crystal Oscillator Control register (XOSC32K.EN32K=1). The XOSC32K also has a 1.024kHz clock output, which can only be used by the RTC. This clock output is enabled by setting the 1KHz Output Enable bit in the 32KHz External Crystal Oscillator Control register (XOSC32K.EN1K=1).

It is also possible to lock the XOSC32K configuration by setting the Write Lock bit in the 32KHz External Crystal Oscillator Control register (XOSC32K.WRTLOCK=1). If set, the XOSC32K configuration is locked until a Power-On Reset (POR) is detected.

The XOSC32K will behave differently in different sleep modes based on the settings of XOSC32K.RUNSTDBY, XOSC32K.ONDEMAND, and XOSC32K.ENABLE. If XOSC32KCTRL.ENABLE=0, the XOSC32K will be always stopped. For XOS32KCTRL.ENABLE=1, this table is valid:

Downloaded from Arrow.com.

Table 24-1. XOSC32K Sleep Behavior

CPU Mode	XOSC32K.RUNSTDBY	XOSC32K.ONDEMAND	Sleep Behavior of XOSC32K and CFD
Active or Idle	-	0	Always run
Active or Idle	-	1	Run if requested by peripheral
Standby	1	0	Always run
Standby	1	1	Run if requested by peripheral
Standby	0	-	Run if requested by peripheral

As a crystal oscillator usually requires a very long start-up time, the 32KHz External Crystal Oscillator will keep running across resets when XOSC32K.ONDEMAND=0, except for power-on reset (POR). After a reset or when waking up from a sleep mode where the XOSC32K was disabled, the XOSC32K will need a certain amount of time to stabilize on the correct frequency. This start-up time can be configured by changing the Oscillator Start-Up Time bit group (XOSC32K.STARTUP) in the 32KHz External Crystal Oscillator Control register. During the start-up time, the oscillator output is masked to ensure that no unstable clock propagates to the digital logic.

Once the external clock or crystal oscillator is stable and ready to be used as a clock source, the XOSC32K Ready bit in the Status register is set (STATUS.XOSC32KRDY=1). The transition of STATUS.XOSC32KRDY from '0' to '1' generates an interrupt if the XOSC32K Ready bit in the Interrupt Enable Set register is set (INTENSET.XOSC32KRDY=1).

The XOSC32K can be used as a source for Generic Clock Generators (GCLK) or for the Real-Time Counter (RTC). Before enabling the GCLK or the RTC module, the corresponding oscillator output must be enabled (XOSC32K.EN32K or XOSC32K.EN1K) in order to ensure proper operation. In the same way, the GCLK or RTC modules must be disabled before the clock selection is changed. For details on RTC clock configuration, refer also to 24.6.6 Real-Time Counter Clock Selection.

Related Links

18. GCLK - Generic Clock Controller
27. RTC – Real-Time Counter

24.6.3 Clock Failure Detection Operation

The Clock Failure Detector (CFD) allows the user to monitor the external clock or crystal oscillator signal provided by the external oscillator (XOSC32K). The CFD detects failing operation of the XOSC32K clock with reduced latency, and allows to switch to a safe clock source in case of clock failure. The user can also switch from the safe clock back to XOSC32K in case of recovery. The safe clock is derived from the OSCULP32K oscillator with a configurable prescaler. This allows to configure the safe clock in order to fulfill the operative conditions of the microcontroller.

In sleep modes, CFD operation is automatically disabled when the external oscillator is not requested to run by a peripheral. See the Sleep Behavior table above when this is the case.

The user interface registers allow to enable, disable, and configure the CFD. The Status register provides status flags on failure and clock switch conditions. The CFD can optionally trigger an interrupt or an event when a failure is detected.

Downloaded from Arrow.com.

Clock Failure Detection

The CFD is reset only at power-on (POR). The CFD does not monitor the XOSC32K clock when the oscillator is disabled (XOSC32K.ENABLE=0).

Before starting CFD operation, the user must start and enable the safe clock source (OSCULP32K oscillator).

CFD operation is started by writing a '1' to the CFD Enable bit in the External Oscillator Control register (CFDCTRL.CFDEN). After starting or restarting the XOSC32K, the CFD does not detect failure until the start-up time has elapsed. The start-up time is configured by the Oscillator Start-Up Time in the External Multipurpose Crystal Oscillator Control register (XOSC32K.STARTUP). Once the XOSC32K Start-Up Time is elapsed, the XOSC32K clock is constantly monitored.

During a period of 4 safe clocks (monitor period), the CFD watches for a clock activity from the XOSC32K. There must be at least one rising and one falling XOSC32K clock edge during 4 safe clock periods to meet non-failure conditions. If no or insufficient activity is detected, the failure status is asserted: The Clock Failure Detector status bit in the Status register (STATUS.CLKFAIL) and the Clock Failure Detector interrupt flag bit in the Interrupt Flag register (INTFLAG.CLKFAIL) are set. If the CLKFAIL bit in the Interrupt Enable Set register (INTENSET.CLKFAIL) is set, an interrupt is generated as well. If the Event Output enable bit in the Event Control register (EVCTRL.CFDEO) is set, an output event is generated, too.

After a clock failure was issued the monitoring of the XOSC32K clock is continued, and the Clock Failure Detector status bit in the Status register (STATUS.CLKFAIL) reflects the current XOSC32K activity.

Clock Switch

When a clock failure is detected, the XOSC32K clock is replaced by the safe clock in order to maintain an active clock during the XOSC32K clock failure. The safe clock source is the OSCULP32K oscillator clock. Both 32KHz and 1KHz outputs of the XOSC32K are replaced by the respective OSCULP32K 32KHz and 1KHz outputs. The safe clock source can be scaled down by a configurable prescaler to ensure that the safe clock frequency does not exceed the operating conditions selected by the application. When the XOSC32K clock is switched to the safe clock, the Clock Switch bit in the Status register (STATUS.CLKSW) is set.

When the CFD has switched to the safe clock, the XOSC32K is not disabled. If desired, the application must take the necessary actions to disable the oscillator. The application must also take the necessary actions to configure the system clocks to continue normal operations. In the case the application can recover the XOSC32K, the application can switch back to the XOSC32K clock by writing a '1' to Switch Back Enable bit in the Clock Failure Control register (CFDCTRL.SWBACK). Once the XOSC32K clock is switched back, the Switch Back bit (CFDCTRL.SWBACK) is cleared by hardware.

Prescaler

The CFD has an internal configurable prescaler to generate the safe clock from the OSCULP32K oscillator. The prescaler size allows to scale down the OSCULP32K oscillator so the safe clock frequency is not higher than the XOSC32K clock frequency monitored by the CFD. The maximum division factor is 2.

The prescaler is applied on both outputs (32KHz and 1KHz) of the safe clock.

Example 24-1.

For an external crystal oscillator at 32KHz and the OSCULP32K frequency is 32KHz, the XOSC32K.CFDPRESC should be set to 0 for a safe clock of equal frequency.

Downloaded from Arrow.com.

Event

If the Event Output Enable bit in the Event Control register (EVCTRL.CFDEO) is set, the CFD clock failure will be output on the Event Output. When the CFD is switched to the safe clock, the CFD clock failure will not be output on the Event Output.

Sleep Mode

The CFD is halted depending on configuration of the XOSC32K and the peripheral clock request. For further details, refer to the Sleep Behavior table above. The CFD interrupt can be used to wake up the device from sleep modes.

24.6.4 32 kHz Ultra Low-Power Internal Oscillator (OSCULP32K) Operation

The OSCULP32K provides a tunable, low-speed, and ultra low-power clock source. The OSCULP32K is factory-calibrated under typical voltage and temperature conditions.

The OSCULP32K is enabled by default after a Power-on Reset (POR), and will always run except during POR. The frequency of the OSCULP32K Oscillator is controlled by the value in the Calibration bits in the 32 kHz Ultra Low-Power Internal Oscillator Control register (OSCULP32K.CALIB). This data is used to compensate for process variations.

OSCULP32K.CALIB is automatically loaded from Flash Factory Calibration during start-up. The calibration value can be overridden by the user by writing to OSCULP32K.CALIB.

Users can lock the OSCULP32K configuration by setting the Write Lock bit in the 32 kHz Ultra Low-Power Internal Oscillator Control register (OSCULP32K.WRTLOCK = 1). If set, the OSCULP32K configuration is locked until POR is detected.

The OSCULP32K can be used as a source for Generic Clock Generators (GCLK) or for the Real-Time Counter (RTC). To ensure proper operation, the GCLK or RTC modules must be disabled before the clock selection is changed.

OSCULP32K Clock Switch

The Clock switch operation requires the XOSC32K to be enabled (XOSC32K.ENABLE=1 and STATUS.XOSC32KRDY = 1). When the OSCULP32K Clock Switch Enable bit (OSCULP32K.ULP32KSW) is set, the CLK_OSCULP32K clock is switched to the XOSC32K Clock Oscillator. When the clock switch process is complete, the OSCULP32K Clock Switch bit in Status register (STATUS.ULP32KSW) is set. The OSCULP32K oscillator is shut off, and the XOSC32K oscillator becomes always running. The CFD feature is also disabled by hardware. When set, the OSCULP32K.ULP32KSW can be reset only by POR operation.

Related Links

27. RTC – Real-Time Counter

24.6.6 Real-Time Counter Clock Selection

18. GCLK - Generic Clock Controller

24.6.5 Watchdog Timer Clock Selection

The Watchdog Timer (WDT) uses the internal 1.024kHz OSCULP32K output clock. This clock is running all the time and internally enabled when requested by the WDT module.

Related Links

26. WDT – Watchdog Timer

Downloaded from Arrow.com.

24.6.6 Real-Time Counter Clock Selection

Before enabling the RTC module, the RTC clock must be selected first. All oscillator outputs are valid as RTC clock. The selection is done in the RTC Control register (RTCCTRL). To ensure a proper operation, it is highly recommended to disable the RTC module first, before the RTC clock source selection is changed.

Related Links

27. RTC – Real-Time Counter

24.6.7 Interrupts

The OSC32KCTRL has the following interrupt sources:

- XOSC32KRDY - 32KHz Crystal Oscillator Ready: A 0-to-1 transition on the STATUS.XOSC32KRDY bit is detected
- CLKFAIL - Clock Failure Detector: A 0-to-1 transition on the STATUS.CLKFAIL bit is detected

All these interrupts are synchronous wake-up source.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition occurs. Each interrupt can be enabled individually by setting the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by setting the corresponding bit in the Interrupt Enable Clear register (INTENCLR). An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled or the OSC32KCTRL is reset. See the INTFLAG register for details on how to clear interrupt flags.

The OSC32KCTRL has one common interrupt request line for all the interrupt sources. The user must read the INTFLAG register to determine which interrupt condition is present. Refer to the INTFLAG register for details.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

Related Links

22. PM – Power Manager

24.6.8 Events

The CFD can generate the following output event:

- Clock Failure Detector (CLKFAIL): Generated when the Clock Failure Detector status bit is set in the Status register (STATUS.CLKFAIL). The CFD event is not generated when the Clock Switch bit (STATUS.SWBACK) in the Status register is set.

Writing a '1' to an Event Output bit in the Event Control register (EVCTRL.CFDEO) enables the CFD output event. Writing a '0' to this bit disables the CFD output event. Refer to the Event System chapter for details on configuring the event system.

Downloaded from Arrow.com.

24.7 Register Summary

Offset	Name	Bit Pos.								
0x00	INTENCLR	7:0						CLKFAIL		XOSC32KRDY
		15:8								
		23:16								
		31:24								
0x04	INTENSET	7:0						CLKFAIL		XOSC32KRDY
		15:8								
		23:16								
		31:24								
0x08	INTFLAG	7:0						CLKFAIL		XOSC32KRDY
		15:8								
		23:16								
		31:24								
0x0C	STATUS	7:0				ULP32KSW	CLKSW	CLKFAIL		XOSC32KRDY
		15:8								
		23:16								
		31:24								
0x10	RTCCTRL	7:0						RTCSEL[2:0]		
0x11 ... 0x13	Reserved									
0x14	XOSC32K	7:0	ONDEMAND	RUNSTDBY		EN1K	EN32K	XTALEN	ENABLE	
		15:8				WRTLOCK		STARTUP[2:0]		
0x16	CFDCTRL	7:0						CFDPRESC	SWBACK	CFDEN
0x17	EVCTRL	7:0								CFDEO
0x18 ... 0x1B	Reserved									
0x1C	OSCULP32K	7:0			ULP32KSW					
		15:8	WRTLOCK			CALIB[4:0]				
		23:16								
		31:24								

24.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register and the 8-bit halves of a 16-bit register can be accessed directly.

All registers with write-access can be write-protected optionally by the peripheral access controller (PAC). Optional Write-Protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-

Downloaded from Arrow.com.

Protection" property in the register description. Write-protection does not apply to accesses through an external debugger.

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

Related Links

15. PAC - Peripheral Access Controller

Downloaded from Arrow.com.

24.8.1 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						CLKFAIL		XOSC32KRDY
Access						R/W		R/W
Reset						0		0

Bit 2 – CLKFAIL XOSC32K Clock Failure Detection Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the XOSC32K Clock Failure Interrupt Enable bit, which disables the XOSC32K Clock Failure interrupt.

Value	Description
0	The XOSC32K Clock Failure Detection is disabled.
1	The XOSC32K Clock Failure Detection is enabled. An interrupt request will be generated when the XOSC32K Clock Failure Detection interrupt flag is set.

Bit 0 – XOSC32KRDY XOSC32K Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the XOSC32K Ready Interrupt Enable bit, which disables the XOSC32K Ready interrupt.

Value	Description
0	The XOSC32K Ready interrupt is disabled.
1	The XOSC32K Ready interrupt is enabled.

Downloaded from Arrow.com.

24.8.2 Interrupt Enable Set

Name: INTENSET
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						CLKFAIL		XOSC32KRDY
Access						R/W		R/W
Reset						0		0

Bit 2 – CLKFAIL XOSC32K Clock Failure Detection Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the XOSC32K Clock Failure Interrupt Enable bit, which enables the XOSC32K Clock Failure interrupt.

Value	Description
0	The XOSC32K Clock Failure Detection is disabled.
1	The XOSC32K Clock Failure Detection is enabled. An interrupt request will be generated when the XOSC32K Clock Failure Detection interrupt flag is set.

Bit 0 – XOSC32KRDY XOSC32K Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the XOSC32K Ready Interrupt Enable bit, which enables the XOSC32K Ready interrupt.

Value	Description
0	The XOSC32K Ready interrupt is disabled.
1	The XOSC32K Ready interrupt is enabled.

Downloaded from Arrow.com.

24.8.3 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x08
Reset: 0x00000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						CLKFAIL		XOSC32KRDY
Access						R/W		R/W
Reset						0		0

Bit 2 – CLKFAIL XOSC32K Clock Failure Detection
This flag is cleared by writing a '1' to it.

This flag is set on a zero-to-one transition of the XOSC32K Clock Failure Detection bit in the Status register (STATUS.CLKFAIL) and will generate an interrupt request if INTENSET.CLKFAIL is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the XOSC32K Clock Failure Detection flag.

Bit 0 – XOSC32KRDY XOSC32K Ready
This flag is cleared by writing a '1' to it.

This flag is set by a zero-to-one transition of the XOSC32K Ready bit in the Status register (STATUS.XOSC32KRDY), and will generate an interrupt request if INTENSET.XOSC32KRDY=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the XOSC32K Ready interrupt flag.

Downloaded from Arrow.com.

24.8.4 Status

Name: STATUS
Offset: 0x0C
Reset: 0x00000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				ULP32KSW	CLKSW	CLKFAIL		XOSC32KRDY
Access				R	R	R		R
Reset				0	0	0		0

Bit 4 – ULP32KSW OSCULP32K Clock Switch

Value	Description
0	OSCULP32K is not switched and provided by the ULP32K oscillator.
1	OSCULP32K is switched to be provided by the XOSC32K clock.

Bit 3 – CLKSW XOSC32K Clock Switch

Value	Description
0	XOSC32K is not switched and provided the crystal oscillator.
1	XOSC32K is switched to be provided by the safe clock.

Bit 2 – CLKFAIL XOSC32K Clock Failure Detector

Value	Description
0	XOSC32K is passing failure detection.
1	XOSC32K is not passing failure detection.

Bit 0 – XOSC32KRDY XOSC32K Ready

Value	Description
0	XOSC32K is not ready.
1	XOSC32K is stable and ready to be used as a clock source.

Downloaded from Arrow.com.

24.8.5 RTC Clock Selection Control

Name: RTCCTRL
Offset: 0x10
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
						RTCSEL[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:0 – RTCSEL[2:0] RTC Clock Selection
These bits select the source for the RTC.

Value	Name	Description
0x0	ULP1K	1.024kHz from 32KHz internal ULP oscillator
0x1	ULP32K	32.768kHz from 32KHz internal ULP oscillator
0x2, 0x3	Reserved	-
0x4	XOSC1K	1.024kHz from 32KHz external oscillator
0x5	XOSC32K	32.768kHz from 32KHz external crystal oscillator
0x6	Reserved	
0x7	Reserved	

Downloaded from Arrow.com.

24.8.6 32KHz External Crystal Oscillator (XOSC32K) Control

Name: XOSC32K
Offset: 0x14
Reset: 0x00000080
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
				WRTLOCK		STARTUP[2:0]		
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY		EN1K	EN32K	XTALEN	ENABLE	
Access	R/W	R/W		R/W	R/W	R/W	R/W	
Reset	1	0		0	0	0	0	

Bit 12 – WRTLOCK Write Lock
This bit locks the XOSC32K register for future writes, effectively freezing the XOSC32K configuration.

Value	Description
0	The XOSC32K configuration is not locked.
1	The XOSC32K configuration is locked.

Bits 10:8 – STARTUP[2:0] Oscillator Start-Up Time
These bits select the start-up time for the oscillator.

The OSCULP32K oscillator is used to clock the start-up counter.

Table 24-2. Start-Up Time for 32KHz External Crystal Oscillator

STARTUP[2:0]	Number of OSCULP32K Clock Cycles	Number of XOSC32K Clock Cycles	Approximate Equivalent Time [s]
0x0	2048	3	0.06
0x1	4096	3	0.13
0x2	16384	3	0.5
0x3	32768	3	1
0x4	65536	3	2
0x5	131072	3	4
0x6	262144	3	8
0x7	-	-	Reserved

Note:
1. Actual Start-Up time is 1 OSCULP32K cycle + 3 XOSC32K cycles.
2. The given time assumes an XTAL frequency of 32.768kHz.

Downloaded from Arrow.com.

Bit 7 – ONDEMAND On Demand Control
This bit controls how the XOSC32K behaves when a peripheral clock request is detected. For details, refer to XOSC32K Sleep Behavior.

Bit 6 – RUNSTDBY Run in Standby
This bit controls how the XOSC32K behaves during standby sleep mode. For details, refer to XOSC32K Sleep Behavior.

Bit 4 – EN1K 1KHz Output Enable

Value	Description
0	The 1KHz output is disabled.
1	The 1KHz output is enabled.

Bit 3 – EN32K 32KHz Output Enable

Value	Description
0	The 32KHz output is disabled.
1	The 32KHz output is enabled.

Bit 2 – XTALEN Crystal Oscillator Enable
This bit controls the connections between the I/O pads and the external clock or crystal oscillator.

Value	Description
0	External clock connected on XIN32. XOUT32 can be used as general-purpose I/O.
1	Crystal connected to XIN32/XOUT32.

Bit 1 – ENABLE Oscillator Enable

Value	Description
0	The oscillator is disabled.
1	The oscillator is enabled.

Downloaded from Arrow.com.

24.8.7 Clock Failure Detector Control

Name: CFDCTRL
Offset: 0x16
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
						CFDPRESC	SWBACK	CFDEN
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – CFDPRESC Clock Failure Detector Prescaler
This bit selects the prescaler for the Clock Failure Detector.

Value	Description
0	The CFD safe clock frequency is the OSCULP32K frequency
1	The CFD safe clock frequency is the OSCULP32K frequency divided by 2

Bit 1 – SWBACK Clock Switch Back
This bit clontrols the XOSC32K output switch back to the external clock or crystal scillator in case of clock recovery.

Value	Description
0	The clock switch is disabled.
1	The clock switch is enabled. This bit is reset when the XOSC32K output is switched back to the external clock or crystal oscillator.

Bit 0 – CFDEN Clock Failure Detector Enable
This bit selects the Clock Failure Detector state.

Value	Description
0	The CFD is disabled.
1	The CFD is enabled.

Downloaded from Arrow.com.

24.8.8 Event Control

Name: EVCTRL
Offset: 0x17
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								CFDEO
Access								R/W
Reset								0

Bit 0 – CFDEO Clock Failure Detector Event Out Enable
This bit controls whether the Clock Failure Detector event output is enabled and an event will be generated when the CFD detects a clock failure.

Value	Description
0	Clock Failure Detector Event output is disabled, no event will be generated.
1	Clock Failure Detector Event output is enabled, an event will be generated.

Downloaded from Arrow.com.

24.8.9 32KHz Ultra Low-Power Internal Oscillator (OSCULP32K) Control

Name: OSCULP32K
Offset: 0x1C
Reset: 0x0000XX06
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	WRTLOCK			CALIB[4:0]				
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			x	x	x	x	x

Bit	7	6	5	4	3	2	1	0
			ULP32KSW					
Access			R/W					
Reset			0					

Bit 15 – WRTLOCK Write Lock
This bit locks the OSCULP32K register for future writes to fix the OSCULP32K configuration.

Value	Description
0	The OSCULP32K configuration is not locked.
1	The OSCULP32K configuration is locked.

Bits 12:8 – CALIB[4:0] Oscillator Calibration
These bits control the oscillator calibration.

These bits are loaded from Flash Calibration at startup.

Bit 5 – ULP32KSW OSCULP32K Clock Switch Enable

Value	Description
0	OSCULP32K is not switched and provided by the ULP32K oscillator.
1	OSCULP32K is switched to be provided by the XOSC32K oscillator.

Downloaded from Arrow.com.

25. SUPC – Supply Controller

25.1 Overview

The Supply Controller (SUPC) manages the voltage reference and power supply of the device.

The SUPC controls the voltage regulators for the core (VDDCORE) domain. It sets the voltage regulators according to the sleep modes, or the user configuration. In active mode, the voltage regulators can be selected on the fly between LDO (low-dropout) type regulator or Buck converter.

The SUPC embeds two Brown-Out Detectors. BOD33 monitors the voltage applied to the device (VDD) and BOD12 monitors the internal voltage to the core (VDDCORE). The BOD can monitor the supply voltage continuously (continuous mode) or periodically (sampling mode).

The SUPC generates also a selectable reference voltage and a voltage dependent on the temperature which can be used by analog modules like the ADC or DAC.

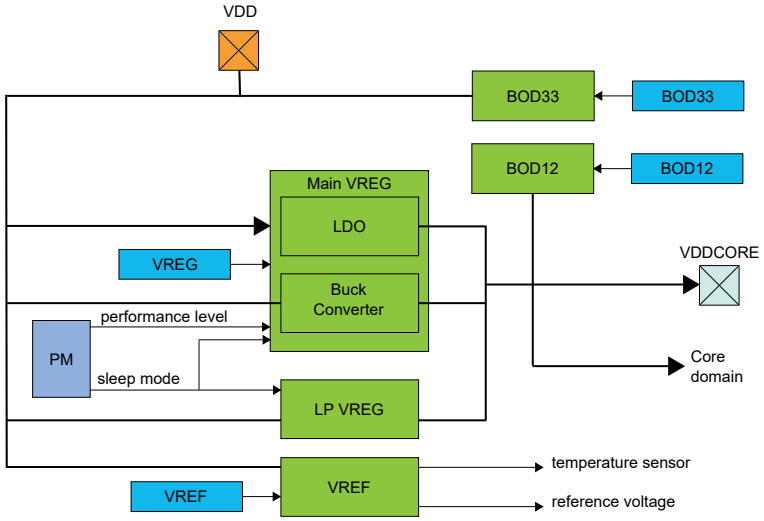
25.2 Features

- Voltage Regulator System
  - Main voltage regulator: LDO or Buck Converter in active mode (MAINVREG)
  - Low-Power voltage regulator in Standby mode (LPVREG)
  - Adjustable VDDCORE to the Sleep mode or the performance level
  - Controlled VDDCORE voltage slope when changing VDDCORE
- Voltage Reference System
  - Reference voltage for ADC and DAC
  - Temperature sensor
- 3.3V Brown-Out Detector (BOD33)
  - Programmable threshold
  - Threshold value loaded from NVM User Row at startup
  - Triggers resets or interrupts or event. Action loaded from NVM User Row
  - Operating modes:
    - Continuous mode
    - Sampled mode for low power applications with programmable sample frequency
  - Hysteresis value from Flash User Calibration
- 1.2V Brown-Out Detector (BOD12)
  - Internal non-configurable Brown-Out Detector

Downloaded from Arrow.com.

25.3 Block Diagram

Figure 25-1. SUPC Block Diagram



25.4 Signal Description

Not appclicable.

25.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

25.5.1 I/O Lines

Not applicable.

25.5.2 Power Management

The SUPC can operate in all sleep modes.

Related Links

22. PM – Power Manager

25.5.3 Clocks

The SUPC bus clock (CLK_SUPC_APB) can be enabled and disabled in the Main Clock module.

A 32KHz clock, asynchronous to the user interface clock (CLK_SUPC_APB), is required to run BOD33 and BOD12 in sampled mode. Due to this asynchronicity, writing to certain registers will require synchronization between the clock domains. Refer to 25.6.6 Synchronization for further details.

Related Links

24. OSC32KCTRL – 32KHz Oscillators Controller

19.6.2.6 Peripheral Clock Masking

25.5.4 DMA

Not applicable.

Downloaded from Arrow.com.

25.5.5 Interrupts

The interrupt request lines are connected to the interrupt controller. Using the SUPC interrupts requires the interrupt controller to be configured first.

25.5.6 Events

The events are connected to the Event System. Refer to the Event System section for details on how to configure the Event System.

25.5.7 Debug Operation

When the CPU is halted in debug mode, the SUPC continues normal operation. If the SUPC is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

If debugger cold-plugging is detected by the system, BOD33 and BOD12 resets will be masked. The BOD resets keep running under hot-plugging. This allows to correct a BOD33 user level too high for the available supply.

25.5.8 Register Access Protection

Registers with write-access can be write-protected optionally by the peripheral access controller (PAC).

Note: Not all registers with write-access can be write-protected.

PAC Write-Protection is not available for the following registers:

- Interrupt Flag Status and Clear register (INTFLAG)

Optional PAC Write-Protection is denoted by the "PAC Write-Protection" property in each individual register description.

Related Links

15. PAC - Peripheral Access Controller

25.5.9 SAM L11 TrustZone Specific Register Access Protection

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

25.5.10 Analog Connections

Not applicable.

25.6 Functional Description

25.6.1 Voltage Regulator System Operation

25.6.1.1 Enabling, Disabling, and Resetting

The LDO main voltage regulator is enabled after any Reset. The main voltage regulator (MAINVREG) can be disabled by writing the Enable bit in the VREG register (VREG.ENABLE) to zero. The main

Downloaded from Arrow.com.

voltage regulator output supply level is automatically defined by the performance level or the sleep mode selected in the Power Manager module.

Related Links

22. PM – Power Manager

25.6.1.2 Initialization

After a Reset, the LDO voltage regulator supplying VDDCORE is enabled.

25.6.1.3 Selecting a Voltage Regulator

In Active mode, the type of the main voltage regulator supplying VDDCORE can be switched on the fly. The two alternatives are a LDO regulator and a Buck converter.

The main voltage regulator switching sequences are as follows:

- The user changes the value of the Voltage Regulator Selection bit in the Voltage Regulator System Control register (VREG.SEL)
- The start of the switching sequence is indicated by clearing the Voltage Regulator Ready bit in the STATUS register (STATUS.VREGRDY=0)
- Once the switching sequence is completed, STATUS.VREGRDY will read '1'

The Voltage Regulator Ready (VREGRDY) interrupt can also be used to detect a zero-to-one transition of the STATUS.VREGRDY bit.

25.6.1.4 Voltage Scaling Control

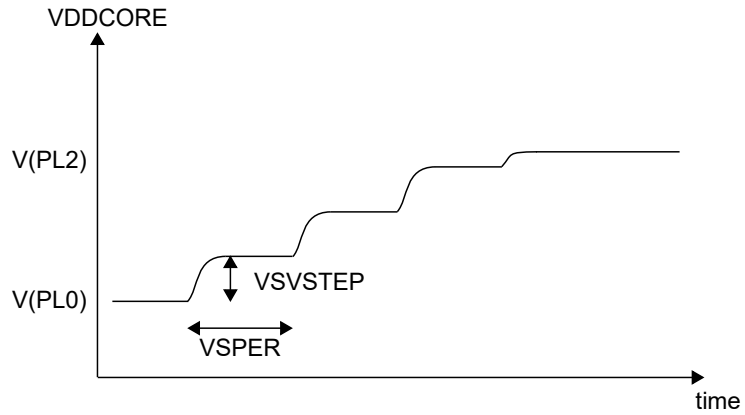
The VDDCORE supply will change under certain circumstances:

- When a new performance level (PL) is set
- When the Standby Sleep mode is entered or left
- When a sleepwalking task is requested in Standby Sleep mode

To prevent high peak current on the main power supply and to have a smooth transition of VDDCORE, both the voltage scaling step size and the voltage scaling frequency can be controlled: VDDCORE is changed by the selected step size of the selected period until the target voltage is reached.

The Voltage Scaling Voltage Step field is in the VREG register, VREG.VSVSTEP. The Voltage Scaling Period field is VREG.VSPER.

The following waveform shows an example of changing performance level from PL0 to PL2.



Setting VREG.VSVSTEP to the maximum value allows to transition in one voltage step.

Downloaded from Arrow.com.

The STATUS.VCORERDY bit is set to '1' as soon as the VDDCORE voltage has reached the target voltage. During voltage transition, STATUS.VCORERDY will read '0'. The Voltage Ready interrupt (VCORERDY) can be used to detect a 0-to-1 transition of STATUS.VCORERDY, see also 25.6.4 Interrupts.

When entering the Standby Sleep mode and when no sleepwalking task is requested, the VDDCORE Voltage scaling control is not used.

25.6.1.5 Sleep Mode Operation

In Standby mode, the low-power voltage regulator (LPVREG) is used to supply VDDCORE.

When the Run in Standby bit in the VREG register (VREG.RUNSTDBY) is written to '1', VDDCORE is supplied by the main voltage regulator. Depending on the Standby in PL0 bit in the Voltage Regulator register (VREG.STDBYPL0), the VDDCORE level is either set to the PL0 voltage level, or remains in the current performance level.

Table 25-1. VDDCORE Level in Standby Mode

VREG.RUNSTDBY	VREG.STDBYPL0	VDDCORE Supply in Standby Mode
0	-	LPVREG
1	0	MAINVREG in current performance level(1)
1	1	MAINVREG in PL0

Note:

1. When the device is in PL0 but VREG.STDBYPL0=0, the MAINVREG is operating in normal power mode. To minimize power consumption, operate MAINVREG in PL0 mode by selecting VREG.STDBYPL0=1.

By writing the Low-Power mode Efficiency bit in the VREG register (VREG.LPEFF) to '1', the efficiency of the regulator in LPVREG can be improved when the application uses a limited VDD range (2.5 to 3.63V). It is also possible to use the BOD33 in order to monitor the VDD and change this LPEFF value on the fly according to VDD level.

Related Links

22.6.3.3 Sleep Mode Controller

25.6.2 Voltage Reference System Operation

The reference voltages are generated by a functional block DETREF inside of the SUPC. DETREF is providing a fixed-voltage source, BANDGAP=1.1V, and a variable voltage, INTREF.

25.6.2.1 Initialization

The voltage reference output and the temperature sensor are disabled after any Reset.

25.6.2.2 Enabling, Disabling, and Resetting

The voltage reference output is enabled/disabled by setting/clearing the Voltage Reference Output Enable bit in the Voltage Reference register (VREF.VREFOE).

The temperature sensor is enabled/disabled by setting/clearing the Temperature Sensor Enable bit in the Voltage Reference register (VREF.TSEN).

Note: When VREF.ONDEMAND=0, it is not recommended to enable both voltage reference output and temperature sensor at the same time - only the voltage reference output will be present at both ADC inputs.

Downloaded from Arrow.com.

25.6.2.3 Selecting a Voltage Reference

The Voltage Reference Selection bit field in the VREF register (VREF.SEL) selects the voltage of INTREF to be applied to analog modules, e.g. the ADC.

25.6.2.4 Sleep Mode Operation

The Voltage Reference output and the Temperature Sensor output behavior during sleep mode can be configured using the Run in Standby bit and the On Demand bit in the Voltage Reference register (VREF.RUNSTDBY, VREF.ONDEMAND), see the following table:

Table 25-2. VREF Sleep Mode Operation

VREF.ONDEMAND	VREF.RUNSTDBY	Voltage Reference Sleep behavior
-	-	Disable
0	0	Always run in all sleep modes *except* standby sleep mode
0	1	Always run in all sleep modes *including* standby sleep mode
1	0	Only run if requested by the ADC, in all sleep modes *except* standby sleep mode
1	1	Only run if requested by the ADC, in all sleep modes *including* standby sleep mode

25.6.3 Brown-Out Detectors

25.6.3.1 Initialization

Before a Brown-Out Detector (BOD33) is enabled, it must be configured, as outlined by the following:

- Set the BOD threshold level (BOD33.LEVEL)
- Set the configuration in Active, Standby (BOD33.ACTION, BOD33.STDBYCFG)
- Set the prescaling value if the BOD will run in sampling mode (BOD33.PSEL)
- Set the action and hysteresis (BOD33.ACTION and BOD33.HYST)

The BOD33 register is Enable-Protected, meaning that they can only be written when the BOD is disabled (BOD33.ENABLE=0 and STATUS.B33SRDY=0). As long as the Enable bit is '1', any writes to Enable-Protected registers will be discarded, and an APB error will be generated. The Enable bits are not Enable-Protected.

25.6.3.2 Enabling, Disabling, and Resetting

After power or user reset, the BOD33 and BOD12 register values are loaded from the NVM User Page.

The BOD33 is enabled by writing a '1' to the Enable bit in the BOD control register (BOD33.ENABLE). The BOD33 is disabled by writing a '0' to the BOD33.ENABLE.

25.6.3.3 3.3V Brown-Out Detector (BOD33)

The 3.3V Brown-Out Detector (BOD33) is able to monitor the VDD supply and compares the voltage with the brown-out threshold level set in the BOD33 Level field (BOD33.LEVEL) in the BOD33 register.

When VDD crosses below the brown-out threshold level, the BOD33 can generate either an interrupt or a Reset, depending on the BOD33 Action bit field (BOD33.ACTION).

The BOD33 detection status can be read from the BOD33 Detection bit in the Status register (STATUS.BOD33DET).

At start-up or at Power-On Reset (POR), the BOD33 register values are loaded from the NVM User Row.

Downloaded from Arrow.com.

25.6.3.4 1.2V Brown-Out Detector (BOD12)

The BOD12 is calibrated in production and its calibration configuration is stored in the NVM User Row. This configuration must not be changed to assure the correct behavior of the BOD12. The BOD12 generates a reset when 1.2V crosses below the preset brown-out level. The BOD12 is always disabled in Standby Sleep mode.

25.6.3.5 Continuous Mode

Continuous mode is the default mode for BOD33.

The BOD33 is continuously monitoring the VDD supply voltage if it is enabled (BOD33.ENABLE=1) and if the BOD33 Configuration bit in the BOD33 register is cleared (BOD33.ACTCFG=0 for active mode, BOD33.STDBYCFG=0 for standby mode).

25.6.3.6 Sampling Mode

The Sampling Mode is a low-power mode where the BOD33 is being repeatedly enabled on a sampling clock's ticks. The BOD33 will monitor the supply voltage for a short period of time and then go to a low-power disabled state until the next sampling clock tick.

Sampling mode is enabled in Active mode for BOD33 by writing the ACTCFG bit (BOD33.ACTCFG=1). Sampling mode is enabled in Standby mode by writing to the STDBYCFG bit (BOD33.STBYCFG=1). The frequency of the clock ticks ($F_{clksampling}$) is controlled by the Prescaler Select bit groups in the BOD33 register (BOD33.PSEL).

$$F_{clksampling} = \frac{F_{clkprescaler}}{2^{(\mathrm{PSEL}+1)}}$$

The prescaler signal ($F_{clkprescaler}$) is a 1KHz clock, output by the 32KHz Ultra Low Power Oscillator OSCULP32K.

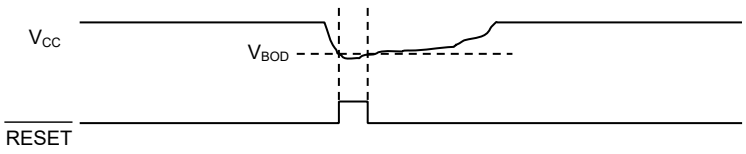
As the sampling clock is different from the APB clock domain, synchronization among the clocks is necessary. See also 25.6.6 Synchronization.

25.6.3.7 Hysteresis

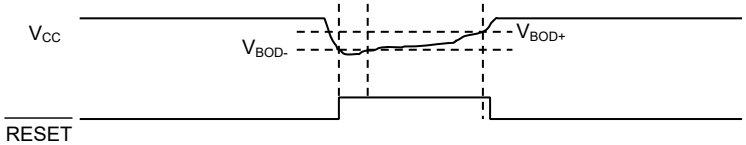
A hysteresis on the trigger threshold of a BOD will reduce the sensitivity to ripples on the monitored voltage: instead of switching $\overline{\text{RESET}}$ at each crossing of V_{BOD}, the thresholds for switching $\overline{\text{RESET}}$ on and off are separated (V_{BOD-} and V_{BOD+}, respectively).

Figure 25-2. BOD Hysteresis Principle

Hysteresis OFF:



Hysteresis ON:



Enabling the BOD33 hysteresis by writing the Hysteresis bit in the BOD33 register (BOD33.HYST) to '1' will add hysteresis to the BOD33 threshold level.

Downloaded from Arrow.com.

The hysteresis functionality can be used in both Continuous and Sampling Mode.

25.6.3.8 Sleep Mode Operation

25.6.3.8.1 Standby Mode

The BOD33 can be used in standby mode if the BOD is enabled and the corresponding Run in Standby bit is written to '1' (BOD33.RUNSTDBY).

The BOD33 can be configured to work in either Continuous or Sampling Mode by writing a '1' to the Configuration in Standby Sleep Mode bit (BOD33.STDBYCFG).

25.6.4 Interrupts

The SUPC has the following interrupt sources, which are either synchronous or asynchronous wake-up sources:

- VDDCORE Voltage Ready (VCORERDY), asynchronous
- Voltage Regulator Ready (VREGRDY) asynchronous
- BOD33 Ready (BOD33RDY), synchronous
- BOD33 Detection (BOD33DET), asynchronous
- BOD33 Synchronization Ready (B33SRDY), synchronous

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition occurs.

Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is disabled, or the SUPC is reset. See the INTFLAG register for details on how to clear interrupt flags. The SUPC has one common interrupt request line for all the interrupt sources. The user must read the INTFLAG register to determine which interrupt condition is present.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

Related Links

22.6.3.3 Sleep Mode Controller

25.6.5 Events

The SUPC can gemerate the following output event:

- BOD12 Detection (BOD12DET): Generated when the VDDCORE crosses below the brown-out threshold level.
- BOD33 Detection (BOD33DET): Generated when the VDD crosses below the brown-out threshold level.

Writing a one to an Event Output bit in the Event Control Register (EVCTRL.xxEO) enables the corresponding output event. Writing a zero to this bit disables the corresponding output event. Refer to the Event System chapter for details on configuring the event system.

25.6.6 Synchronization

The prescaler counters that are used to trigger brown-out detections operate asynchronously from the peripheral bus. As a consequence, the BOD33 Enable bit (BOD33.ENABLE) need synchronization when written.

Downloaded from Arrow.com.

The Write-Synchronization of the Enable bit is triggered by writing a '1' to the Enable bit of the BOD33 Control register. The Synchronization Ready bit (STATUS.B33SRDY) in the STATUS register will be cleared when the Write-Synchronization starts, and set again when the Write-Synchronization is complete. Writing to the same register while the Write-Synchronization is ongoing (STATUS.B33SRDY is '0') will generate a PAC error without stalling the APB bus.

25.6.7 Low Power VREF in Active Mode

During active functional mode, the brownout detector BOD33 and the main voltage regulator (VREG) can reduce their power consumption by using the low power voltage reference (ULPVREF).

The low power voltage reference is ready and can be selected when ULPVREFRDY bit in STATUS register is high. The ULPVREF Ready (ULPVREFRDY) interrupt can also be used to detect a zero-to-one transition of the STATUS.ULPVREFRDY bit.

Writing the VREF bit in the BOD33 register to '1' selects ULPVREF as voltage reference for the BOD33.

If the chip operated in PL0 ((PM->PLCFG.PLSEL=0) or Performance Level is disabled (PM->PLCFG.PLDIS=1), writing the VREFSEL bit in the VREG register to '1' selects ULPVREF as voltage reference for the main voltage regulator.

Downloaded from Arrow.com.

25.7 Register Summary

Offset	Name	Bit Pos.								
0x00	INTENCLR	7:0						B33SRDY	BOD33DET	BOD33RDY
		15:8					ULPVREFRDY	VCORERDY		VREGRDY
		23:16								
		31:24								
0x04	INTENSET	7:0						B33SRDY	BOD33DET	BOD33RDY
		15:8					ULPVREFRDY	VCORERDY		VREGRDY
		23:16								
		31:24								
0x08	INTFLAG	7:0						B33SRDY	BOD33DET	BOD33RDY
		15:8					ULPVREFRDY	VCORERDY		VREGRDY
		23:16								
		31:24								
0x0C	STATUS	7:0						B33SRDY	BOD33DET	BOD33RDY
		15:8				ULPVREFRDY		VCORERDY		VREGRDY
		23:16								
		31:24								
0x10	BOD33	7:0		RUNSTDBY	STDBYCFG	ACTION[1:0]		HYST	ENABLE	
		15:8	PSEL[3:0]				VREFSEL			ACTCFG
		23:16			LEVEL[5:0]					
		31:24								
0x14 ... 0x17	Reserved									
0x18	VREG	7:0		RUNSTDBY	STDBYPL0			SEL	ENABLE	
		15:8							VREFSEL	LPEFF
		23:16					VSVSTEP[3:0]			
		31:24	VSPER[7:0]							
0x1C	VREF	7:0	ONDEMAND	RUNSTDBY				VREFOE	TSEN	
		15:8								
		23:16					SEL[3:0]			
		31:24								
0x20 ... 0x2B	Reserved									
0x2C	EVCTRL	7:0							BOD33DETEO	
		15:8								
		23:16								
		31:24								

Downloaded from Arrow.com.

25.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). PAC Write-protection is denoted by the "PAC Write-Protection" property in each individual register description. Refer to 25.5.8 Register Access Protection for details.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. Refer to 25.6.6 Synchronization for details.

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

Downloaded from Arrow.com.

25.8.1 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					ULPVREFRDY	VCORERDY		VREGRDY
Access					R/W	R/W		R/W
Reset					0	0		0

Bit	7	6	5	4	3	2	1	0
						B33SRDY	BOD33DET	BOD33RDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 11 – ULPVREFRDY Low Power Voltage Reference Ready Interrupt Enable
Writing a '0' to this bit has no effect.

The ULPVREFRDY bit will clear on a zero-to-one transition of the Low Power Voltage Reference Ready bit in the Status register (STATUS.ULPVREFRDY).

Value	Description
0	The Low Power Ready interrupt is disabled.
1	The Low Power Ready interrupt is enabled and an interrupt request will be generated when the ULPVREFRDY Interrupt Flag is set.

Bit 10 – VCORERDY VDDCORE Voltage Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the VDDCORE Ready Interrupt Enable bit, which disables the VDDCORE Ready interrupt.

Value	Description
0	The VDDCORE Ready interrupt is disabled.
1	The VDDCORE Ready interrupt is enabled and an interrupt request will be generated when the VCORERDY Interrupt Flag is set.

Downloaded from Arrow.com.

Bit 8 – VREGRDY Voltage Regulator Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Voltage Regulator Ready Interrupt Enable bit, which disables the Voltage Regulator Ready interrupt.

Value	Description
0	The Voltage Regulator Ready interrupt is disabled.
1	The Voltage Regulator Ready interrupt is enabled and an interrupt request will be generated when the Voltage Regulator Ready Interrupt Flag is set.

Bit 2 – B33SRDY BOD33 Synchronization Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the BOD33 Synchronization Ready Interrupt Enable bit, which disables the BOD33 Synchronization Ready interrupt.

Value	Description
0	The BOD33 Synchronization Ready interrupt is disabled.
1	The BOD33 Synchronization Ready interrupt is enabled, and an interrupt request will be generated when the BOD33 Synchronization Ready Interrupt flag is set.

Bit 1 – BOD33DET BOD33 Detection Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the BOD33 Detection Interrupt Enable bit, which disables the BOD33 Detection interrupt.

Value	Description
0	The BOD33 Detection interrupt is disabled.
1	The BOD33 Detection interrupt is enabled, and an interrupt request will be generated when the BOD33 Detection Interrupt flag is set.

Bit 0 – BOD33RDY BOD33 Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the BOD33 Ready Interrupt Enable bit, which disables the BOD33 Ready interrupt.

Value	Description
0	The BOD33 Ready interrupt is disabled.
1	The BOD33 Ready interrupt is enabled, and an interrupt request will be generated when the BOD33 Ready Interrupt flag is set.

Downloaded from Arrow.com.

25.8.2 Interrupt Enable Set

Name: INTENSET
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					ULPVREFRDY	VCORERDY		VREGRDY
Access					R/W	R/W		R/W
Reset					0	0		0

Bit	7	6	5	4	3	2	1	0
						B33SRDY	BOD33DET	BOD33RDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 11 – ULPVREFRDY Low Power Voltage Reference Ready Interrupt Enable
Writing a '0' to this bit has no effect.

The ULPVREFRDY bit is set on a zero-to-one transition of the Low Power Voltage Reference Ready bit in the Status register (STATUS.ULPVREFRDY).

Value	Description
0	The Low Power Ready interrupt is disabled.
1	The Low Power Ready interrupt is enabled and an interrupt request will be generated when the ULPVREFRDY Interrupt Flag is set.

Bit 10 – VCORERDY VDDCORE Voltage Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the VDDCORE Ready Interrupt Enable bit, which enables the VDDCORE Ready interrupt.

Value	Description
0	The VDDCORE Ready interrupt is disabled.
1	The VDDCORE Ready interrupt is enabled and an interrupt request will be generated when the VCORERDY Interrupt Flag is set.

Downloaded from Arrow.com.

Bit 8 – VREGRDY Voltage Regulator Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Voltage Regulator Ready Interrupt Enable bit, which enables the Voltage Regulator Ready interrupt.

Value	Description
0	The Voltage Regulator Ready interrupt is disabled.
1	The Voltage Regulator Ready interrupt is enabled and an interrupt request will be generated when the Voltage Regulator Ready Interrupt Flag is set.

Bit 2 – B33SRDY BOD33 Synchronization Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the BOD33 Synchronization Ready Interrupt Enable bit, which enables the BOD33 Synchronization Ready interrupt.

Value	Description
0	The BOD33 Synchronization Ready interrupt is disabled.
1	The BOD33 Synchronization Ready interrupt is enabled, and an interrupt request will be generated when the BOD33 Synchronization Ready Interrupt flag is set.

Bit 1 – BOD33DET BOD33 Detection Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the BOD33 Detection Interrupt Enable bit, which enables the BOD33 Detection interrupt.

Value	Description
0	The BOD33 Detection interrupt is disabled.
1	The BOD33 Detection interrupt is enabled, and an interrupt request will be generated when the BOD33 Detection Interrupt flag is set.

Bit 0 – BOD33RDY BOD33 Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the BOD33 Ready Interrupt Enable bit, which enables the BOD33 Ready interrupt.

Value	Description
0	The BOD33 Ready interrupt is disabled.
1	The BOD33 Ready interrupt is enabled, and an interrupt request will be generated when the BOD33 Ready Interrupt flag is set.

Downloaded from Arrow.com.

25.8.3 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x08
Reset: x initially determined from NVM User Row after reset
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					ULPVREFRDY	VCORERDY		VREGRDY
Access					R/W	R/W		R/W
Reset					0	0		1

Bit	7	6	5	4	3	2	1	0
						B33SRDY	BOD33DET	BOD33RDY
Access						R/W	R/W	R/W
Reset						0	0	x

Bit 11 – ULPVREFRDY Low Power Voltage Reference Ready Interrupt Enable
Writing a '0' to this bit has no effect.

The ULPVREFRDY bit will clear on a zero-to-one transition of the Low Power Voltage Reference Ready bit in the Status register (STATUS.ULPVREFRDY) and will generate an interrupt request if INTENSET.ULPVREFRDY=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the ULPVREFRDY interrupt flag.

Bit 10 – VCORERDY VDDCORE Voltage Ready
This flag is cleared by writing a '1 to it.

This flag is set on a zero-to-one transition of the VDDCORE Ready bit in the Status register (STATUS.VCORERDY) and will generate an interrupt request if INTENSET.VCORERDY=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the VCORERDY interrupt flag.

Bit 8 – VREGRDY Voltage Regulator Ready
This flag is cleared by writing a '1' to it.

Downloaded from Arrow.com.

This flag is set on a zero-to-one transition of the Voltage Regulator Ready bit in the Status register (STATUS.VREGRDY) and will generate an interrupt request if INTENSET.VREGRDY=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the VREGRDY interrupt flag.

Bit 2 – B33SRDY BOD33 Synchronization Ready
This flag is cleared by writing a '1' to it.

This flag is set on a zero-to-one transition of the BOD33 Synchronization Ready bit in the Status register (STATUS.B33SRDY) and will generate an interrupt request if INTENSET.B33SRDY=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the BOD33 Synchronization Ready interrupt flag.

Bit 1 – BOD33DET BOD33 Detection
This flag is cleared by writing a '1' to it.

This flag is set on a zero-to-one transition of the BOD33 Detection bit in the Status register (STATUS.BOD33DET) and will generate an interrupt request if INTENSET.BOD33DET=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the BOD33 Detection interrupt flag.

Bit 0 – BOD33RDY BOD33 Ready
This flag is cleared by writing a '1' to it.

This flag is set on a zero-to-one transition of the BOD33 Ready bit in the Status register (STATUS.BOD33RDY) and will generate an interrupt request if INTENSET.BOD33RDY=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the BOD33 Ready interrupt flag.

The BOD33 can be enabled.

Downloaded from Arrow.com.

25.8.4 Status

Name: STATUS
Offset: 0x0C
Reset: x,y initially determined from NVM User Row after reset
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				ULPVREFRDY		VCORERDY		VREGRDY
Access				R		R		R
Reset				x		1		1

Bit	7	6	5	4	3	2	1	0
						B33SRDY	BOD33DET	BOD33RDY
Access						R	R	R
Reset						0	0	y

Bit 12 – ULPVREFRDY Low Power Voltage Reference Ready

Value	Description
0	The ULPVREF voltage is not as expected.
1	The ULPVREF voltage is the target voltage.

Bit 10 – VCORERDY VDDCORE Voltage Ready

Value	Description
0	The VDDCORE voltage is not as expected.
1	The VDDCORE voltage is the target voltage.

Bit 8 – VREGRDY Voltage Regulator Ready

Value	Description
0	The selected voltage regulator in VREG.SEL is not ready.
1	The voltage regulator selected in VREG.SEL is ready and the core domain is supplied by this voltage regulator.

Bit 2 – B33SRDY BOD33 Synchronization Ready

Downloaded from Arrow.com.

Value	Description
0	BOD33 synchronization is ongoing.
1	BOD33 synchronization is complete.

Bit 1 – BOD33DET BOD33 Detection

Value	Description
0	No BOD33 detection.
1	BOD33 has detected that the I/O power supply is going below the BOD33 reference value.

Bit 0 – BOD33RDY BOD33 Ready
The BOD33 can be enabled at start-up from NVM User Row.

Value	Description
0	BOD33 is not ready.
1	BOD33 is ready.

Downloaded from Arrow.com.

25.8.5 3.3V Brown-Out Detector (BOD33) Control

Name: BOD33
Offset: 0x10
Reset: x initially determined from NVM User Row after reset
Property: Write-Synchronized, Enable-Protected, PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			LEVEL[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			x	x	x	x	x	x

Bit	15	14	13	12	11	10	9	8
	PSEL[3:0]				VREFSEL			ACTCFG
Access	R/W	R/W	R/W	R/W	R/W			R/W
Reset	0	0	0	0	0			0

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY	STDBYCFG	ACTION[1:0]		HYST	ENABLE	
Access		R/W	R/W	R/W	R/W	R/W	R/W	
Reset		0	0	x	x	x	x	

Bits 21:16 – LEVEL[5:0] BOD33 Threshold Level on VDD
These bits set the triggering voltage threshold for the BOD33 when the BOD33 monitors the VDD.

These bits are loaded from NVM User Row at start-up.

This bit field is not synchronized.

Bits 15:12 – PSEL[3:0] Prescaler Select
Selects the prescaler divide-by output for the BOD33 sampling mode. The input clock comes from the OSCULP32K 1KHz output.

Value	Name	Description
0x0	DIV2	Divide clock by 2
0x1	DIV4	Divide clock by 4
0x2	DIV8	Divide clock by 8
0x3	DIV16	Divide clock by 16
0x4	DIV32	Divide clock by 32
0x5	DIV64	Divide clock by 64
0x6	DIV128	Divide clock by 128
0x7	DIV256	Divide clock by 256
0x8	DIV512	Divide clock by 512
0x9	DIV1024	Divide clock by 1024

Downloaded from Arrow.com.

Value	Name	Description
0xA	DIV2048	Divide clock by 2048
0xB	DIV4096	Divide clock by 4096
0xC	DIV8192	Divide clock by 8192
0xD	DIV16384	Divide clock by 16384
0xE	DIV32768	Divide clock by 32768
0xF	DIV65536	Divide clock by 65536

Bit 11 – VREFSEL BOD33 Voltage Reference Selection
This bit is not synchronized.

Value	Description
0	Selects VREF for the BOD33.
1	Selects ULPVREF for the BOD33.

Bit 8 – ACTCFG BOD33 Configuration in Active Sleep Mode
This bit is not synchronized.

Value	Description
0	In active mode, the BOD33 operates in continuous mode.
1	In active mode, the BOD33 operates in sampling mode.

Bit 6 – RUNSTDBY Run in Standby
This bit is not synchronized.

Value	Description
0	In standby sleep mode, the BOD33 is disabled.
1	In standby sleep mode, the BOD33 is enabled.

Bit 5 – STDBYCFG BOD33 Configuration in Standby Sleep Mode
If the RUNSTDBY bit is set to '1', the STDBYCFG bit sets the BOD33 configuration in standby sleep mode.

This bit is not synchronized.

Value	Description
0	In standby sleep mode, the BOD33 is enabled and configured in continuous mode.
1	In standby sleep mode, the BOD33 is enabled and configured in sampling mode.

Bits 4:3 – ACTION[1:0] BOD33 Action
These bits are used to select the BOD33 action when the supply voltage crosses below the BOD33 threshold.

These bits are loaded from NVM User Row at start-up.

This bit field is not synchronized.

Value	Name	Description
0x0	NONE	No action
0x1	RESET	The BOD33 generates a reset

Downloaded from Arrow.com.

Value	Name	Description
0x2	INT	The BOD33 generates an interrupt
0x3	-	Reserved

Bit 2 – HYST Hysteresis
This bit indicates whether hysteresis is enabled for the BOD33 threshold voltage.

This bit is loaded from NVM User Row at start-up.

This bit is not synchronized.

Value	Description
0	No hysteresis.
1	Hysteresis enabled.

Bit 1 – ENABLE Enable
This bit is loaded from NVM User Row at start-up.

This bit is not enable-protected.

Value	Description
0	BOD33 is disabled.
1	BOD33 is enabled.

Downloaded from Arrow.com.

25.8.6 Voltage Regulator System (VREG) Control

Name: VREG
Offset: 0x18
Reset: 0x00000002
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	VSPER[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
					VSVSTEP[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
							VREFSEL	LPEFF
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY	STDBYPL0			SEL	ENABLE	
Access		R/W	R/W			R/W	R/W	
Reset		0	1			0	1	

Bits 31:24 – VSPER[7:0] Voltage Scaling Period
This bitfield sets the period between the voltage steps when the VDDCORE voltage is changing in µs.

If VSPER=0, the period between two voltage steps is 1µs.

Bits 19:16 – VSVSTEP[3:0] Voltage Scaling Voltage Step
This field sets the voltage step height when the VDDCORE voltage is changing to reach the target VDDCORE voltage.

The voltage step is equal to $2^{VSVSTEP*}$ min_step.

See the Electrical Characteristics chapters for the min_step voltage level.

Bit 9 – VREFSEL Voltage Regulator Voltage Reference Selection
This bit provides support of using ULPVREF during active function mode.

Value	Description
0	Selects VREF for the voltage regulator.
1	Selects ULPVREF for the voltage regulator.

Bit 8 – LPEFF Low power Mode Efficiency

Downloaded from Arrow.com.

Value	Description
0	The voltage regulator in Low power mode has the default efficiency and supports the whole VDD range (1.62V to 3.63V).
1	The voltage regulator in Low power mode has the highest efficiency and supports a limited VDD range (2.5V to 3.63V).

Bit 6 – RUNSTDBY Run in Standby

Value	Description
0	The voltage regulator is in low power mode in Standby sleep mode.
1	The voltage regulator is in normal mode in Standby sleep mode.

Bit 5 – STDBYPL0 Standby in PL0
This bit selects the performance level (PL) of the main voltage regulator for the Standby sleep mode. This bit is only considered when RUNSTDBY=1.

Value	Description
0	In Standby sleep mode, the voltage regulator remains in the current performance level.
1	In Standby sleep mode, the voltage regulator is used in PL0.

Bit 2 – SEL Voltage Regulator Selection

Value	Description
0	The voltage regulator in active mode is a LDO voltage regulator.
1	The voltage regulator in active mode is a buck converter.

Bit 1 – ENABLE Enable

Value	Description
0	The voltage regulator is disabled.
1	The voltage regulator is enabled.

Downloaded from Arrow.com.

25.8.7 Voltage References System (VREF) Control

Name: VREF
Offset: 0x1C
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					SEL[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY				VREFOE	TSEN	
Access	R/W	R/W				R/W	R/W	
Reset	0	0				0	0	

Bits 19:16 – SEL[3:0] Voltage Reference Selection
These bits select the Voltage Reference for the ADC/DAC.

Value	Name	Description
0x0	1V0	1.0V voltage reference typical value
0x1	1V1	1.1V voltage reference typical valueThe 1.1V voltage reference typical value must be selected for DAC use. Other values are not permitted.
0x2	1V2	1.2V voltage reference typical value
0x3	1V25	1.25V voltage reference typical value
0x4	2V0	2.0V voltage reference typical value
0x5	2V2	2.2V voltage reference typical value
0x6	2V4	2.4V voltage reference typical value
0x7	2V5	2.5V voltage reference typical value
Others		Reserved

Bit 7 – ONDEMAND On Demand Control
The On Demand operation mode allows to enable or disable the voltage reference depending on peripheral requests.

Downloaded from Arrow.com.

Value	Description
0	The voltage reference is always on, if enabled.
1	The voltage reference is enabled when a peripheral is requesting it. The voltage reference is disabled if no peripheral is requesting it.

Bit 6 – RUNSTDBY Run In Standby
The bit controls how the voltage reference behaves during standby sleep mode.

Value	Description
0	The voltage reference is halted during standby sleep mode.
1	The voltage reference is not stopped in standby sleep mode. If VREF.ONDEMAND=1, the voltage reference will be running when a peripheral is requesting it. If VREF.ONDEMAND=0, the voltage reference will always be running in standby sleep mode.

Bit 2 – VREFOE Voltage Reference Output Enable

Value	Description
0	The Voltage Reference output (INTREF) is not available as an ADC input channel.
1	The Voltage Reference output (INTREF) is routed to an ADC input channel.

Bit 1 – TSEN Temperature Sensor Enable

Value	Description
0	Temperature Sensor is disabled.
1	Temperature Sensor is enabled and routed to an ADC input channel.

Downloaded from Arrow.com.

25.8.8 Event Control

Name: EVCTRL
Offset: 0x2C
Reset: 0x0000000
Property: Enable-Protected, PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							BOD33DETEO	
Access							R/W	
Reset							0	

Bit 1 – BOD33DETEO BOD33 Detection Event Output Enable

Value	Description
0	BOD33 detection event output is disabled and event will not be generated
1	BOD33 detection event output is enabled and event will be generated

Downloaded from Arrow.com.

26. WDT – Watchdog Timer

26.1 Overview

The Watchdog Timer (WDT) is a system function for monitoring correct program operation. It makes it possible to recover from error situations such as runaway or deadlocked code. The WDT is configured to a predefined time-out period, and is constantly running when enabled. If the WDT is not cleared within the time-out period, it will issue a system reset. An early-warning interrupt is available to indicate an upcoming watchdog time-out condition.

The window mode makes it possible to define a time slot (or window) inside the total time-out period during which the WDT must be cleared. If the WDT is cleared outside this window, either too early or too late, a system reset will be issued. Compared to the normal mode, this can also catch situations where a code error causes the WDT to be cleared frequently.

When enabled, the WDT will run in active mode and all sleep modes. It is asynchronous and runs from a CPU-independent clock source. The WDT will continue operation and issue a system reset or interrupt even if the main clocks fail.

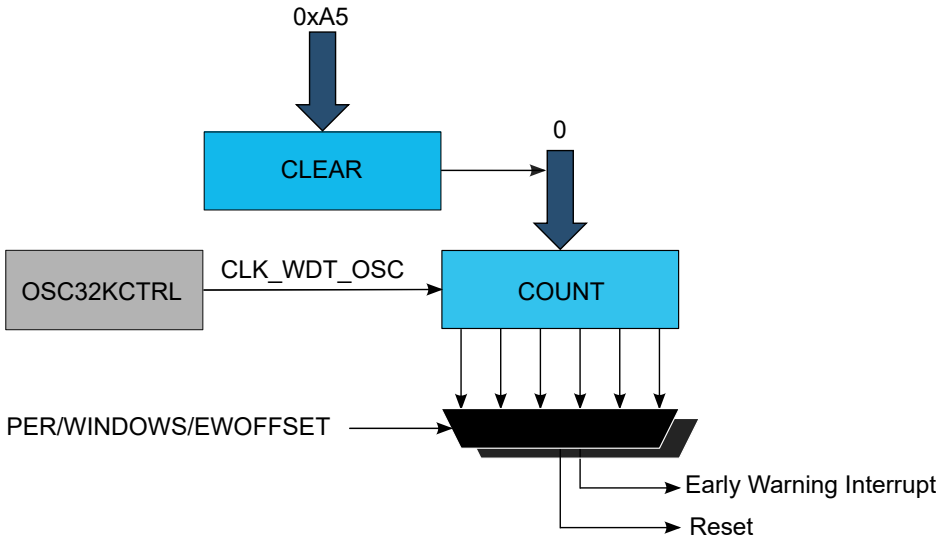
26.2 Features

- Issues a system reset if the Watchdog Timer is not cleared before its time-out period
- Early Warning interrupt generation
- Asynchronous operation from dedicated oscillator
- Two types of operation
  - Normal
  - Window mode
- Selectable time-out periods
  - From 8 cycles to 16,384 cycles in Normal mode
  - From 16 cycles to 32,768 cycles in Window mode
- Always-On capability

Downloaded from Arrow.com.

26.3 Block Diagram

Figure 26-1. WDT Block Diagram



26.4 Signal Description

Not applicable.

26.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

26.5.1 I/O Lines

Not applicable.

26.5.2 Power Management

The WDT can continue to operate in any sleep modes where the selected source clock is running. The WDT interrupts can be used to wake up the device from sleep modes. The events can trigger other operations in the system without exiting sleep modes.

Related Links

22. PM – Power Manager

26.5.3 Clocks

The WDT bus clock (CLK_WDT_APB) can be enabled and disabled (masked) in the Main Clock module (MCLK).

A 1.024 kHz oscillator clock (CLK_WDT_OSC) is required to clock the WDT internal counter.

The CLK_WDT_OSC CLOCK is sourced from the clock of the internal Ultra Low-Power Oscillator (OSCULP32K). Due to ultra low-power design, the oscillator is not accurate, hence the exact time-out period may vary from device-to-device. This variation must be considered when designing software that uses the WDT to ensure that the time-out periods used are valid for all devices.

Downloaded from Arrow.com.

The counter clock CLK_WDT_OSC is asynchronous to the bus clock (CLK_WDT_APB). Due to this asynchronicity, writing to certain registers will require synchronization between the clock domains. Refer to 26.6.7 Synchronization for further details.

Related Links

19.6.2.6 Peripheral Clock Masking

24. OSC32KCTRL – 32KHz Oscillators Controller

26.5.4 DMA

Not applicable.

26.5.5 Interrupts

The interrupt request line is connected to the interrupt controller. Using the WDT interrupt(s) requires the interrupt controller to be configured first.

26.5.6 Events

Not applicable.

26.5.7 Debug Operation

When the CPU is halted in debug mode the WDT will halt normal operation.

26.5.8 Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Interrupt Flag Status and Clear (INTFLAG) register

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

26.5.9 SAM L11 TrustZone Specific Register Access Protection

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

26.5.10 Analog Connections

Not applicable.

26.6 Functional Description

26.6.1 Principle of Operation

The Watchdog Timer (WDT) is a system for monitoring correct program operation, making it possible to recover from error situations such as runaway code, by issuing a Reset. When enabled, the WDT is a

Downloaded from Arrow.com.

constantly running timer that is configured to a predefined time-out period. Before the end of the time-out period, the WDT should be set back, or else, a system Reset is issued.

The WDT has two modes of operation, Normal mode and Window mode. Both modes offer the option of Early Warning interrupt generation. The description for each of the basic modes is given below. The settings in the Control A register (CTRLA) and the Interrupt Enable register (handled by INTENCLR/INTENSET) determine the mode of operation:

Table 26-1. WDT Operating Modes

CTRLA.ENABLE	CTRLA.WEN	Interrupt Enable	Mode
0	x	x	Stopped
1	0	0	Normal mode
1	0	1	Normal mode with Early Warning interrupt
1	1	0	Window mode
1	1	1	Window mode with Early Warning interrupt

26.6.2 Basic Operation

26.6.2.1 Initialization

The following bits are enable-protected, meaning that they can only be written when the WDT is disabled (CTRLA.ENABLE=0):

- Control A register (CTRLA), except the Enable bit (CTRLA.ENABLE)
- Configuration register (CONFIG)
- Early Warning Interrupt Control register (EWCTRL)

Enable-protected bits in the CTRLA register can be written at the same time as CTRLA.ENABLE is written to '1', but not at the same time as CTRLA.ENABLE is written to '0'.

The WDT can be configured only while the WDT is disabled. The WDT is configured by defining the required Time-Out Period bits in the Configuration register (CONFIG.PER). If Window mode operation is desired, the Window Enable bit in the Control A register must be set (CTRLA.WEN=1) and the Window Period bits in the Configuration register (CONFIG.WINDOW) must be defined.

Enable-protection is denoted by the "Enable-Protected" property in the register description.

26.6.2.2 Configurable Reset Values

After a Power-on Reset, some registers will be loaded with initial values from the NVM User Row.

This includes the following bits and bit groups:

- Enable bit in the Control A register, CTRLA.ENABLE
- Always-On bit in the Control A register, CTRLA.ALWAYSON
- Run In Standby Enable bit in the Control A register (CTRLA.RUNSTDBY)
- Watchdog Timer Windows Mode Enable bit in the Control A register, CTRLA.WEN
- Watchdog Timer Windows Mode Time-Out Period bits in the Configuration register, CONFIG.WINDOW
- Time-Out Period bits in the Configuration register, CONFIG.PER
- Early Warning Interrupt Time Offset bits in the Early Warning Interrupt Control register, EWCTRL.EWOFFSET

Downloaded from Arrow.com.

26.6.2.3 Enabling, Disabling, and Resetting

The WDT is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The WDT is disabled by writing a '0' to CTRLA.ENABLE.

The WDT can be disabled only if the Always-On bit in the Control A register (CTRLA.ALWAYSON) is '0'.

26.6.2.4 Normal Mode

In Normal mode operation, the length of a time-out period is configured in CONFIG.PER. The WDT is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). Once enabled, the WDT will issue a system reset if a time-out occurs. This can be prevented by clearing the WDT at any time during the time-out period.

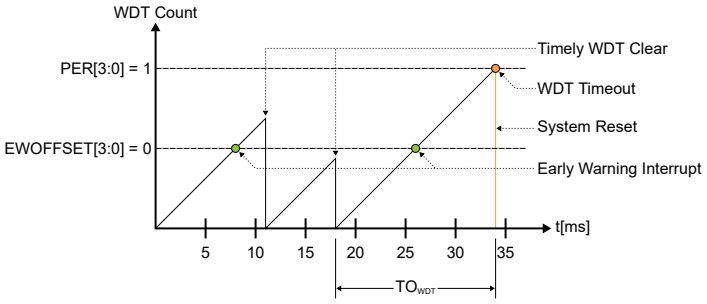
The WDT is cleared and a new WDT time-out period is started by writing 0xA5 to the Clear register (CLEAR). Writing any other value than 0xA5 to CLEAR will issue an immediate system reset.

There are 12 possible WDT time-out (TO_{WDT}) periods, selectable from 8ms to 16s.

By default, the early warning interrupt is disabled. If it is desired, the Early Warning Interrupt Enable bit in the Interrupt Enable register (INTENSET.EW) must be written to '1'. The Early Warning Interrupt is disabled again by writing a '1' to the Early Warning Interrupt bit in the Interrupt Enable Clear register (INTENCLR.EW).

If the Early Warning Interrupt is enabled, an interrupt is generated prior to a WDT time-out condition. In Normal mode, the Early Warning Offset bits in the Early Warning Interrupt Control register, EWCTRL.EWOFFSET, define the time when the early warning interrupt occurs. The Normal mode operation is illustrated in the figure Normal-Mode Operation.

Figure 26-2. Normal-Mode Operation



26.6.2.5 Window Mode

In Window mode operation, the WDT uses two different time specifications: the WDT can only be cleared by writing 0xA5 to the CLEAR register *after* the closed window time-out period (TO_{WDTW}), during the subsequent Normal time-out period (TO_{WDT}). If the WDT is cleared before the time window opens (before TO_{WDTW} is over), the WDT will issue a system reset.

Both parameters TO_{WDTW} and TO_{WDT} are periods in a range from 8ms to 16s, so the total duration of the WDT time-out period is the sum of the two parameters.

The closed window period is defined by the Window Period bits in the Configuration register (CONFIG.WINDOW), and the open window period is defined by the Period bits in the Configuration register (CONFIG.PER).

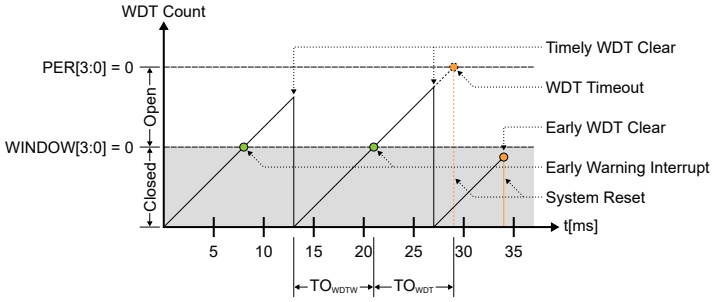
By default, the Early Warning interrupt is disabled. If it is desired, the Early Warning Interrupt Enable bit in the Interrupt Enable register (INTENSET.EW) must be written to '1'. The Early Warning Interrupt is

Downloaded from Arrow.com.

disabled again by writing a '1' to the Early Warning Interrupt bit in the Interrupt Enable Clear (INTENCLR.EW) register.

If the Early Warning interrupt is enabled in Window mode, the interrupt is generated at the start of the open window period, i.e. after TO_{WDTW}. The Window mode operation is illustrated in figure Window-Mode Operation.

Figure 26-3. Window-Mode Operation



26.6.3 DMA Operation

Not applicable.

26.6.4 Interrupts

The WDT has the following interrupt source:

- Early Warning (EW): Indicates that the counter is approaching the time-out condition.
  - This interrupt is an asynchronous wake-up source.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs.

Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the WDT is reset. See the 26.8.6 INTFLAG register description for details on how to clear interrupt flags. All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. The user must read the INTFLAG register to determine which interrupt condition is present.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

Related Links

22. PM – Power Manager

22.6.3.3 Sleep Mode Controller

26.6.5 Events

Not applicable.

Downloaded from Arrow.com.

26.6.6 Sleep Mode Operation

The Run-In-Standby bit in Control A (CTRLA.RUNSTDBY) control the behavior of the WDT during standby sleep mode. When the bit is zero, the watchdog is disabled during sleep, but maintains its current configuration. When CTRLA.RUNSTDBY is '1', the WDT continues to operate during sleep.

Related Links

26.8.1 CTRLA

26.6.7 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following registers are synchronized when written:

- Enable bit in Control A register (CTRLA.ENABLE)
- Window Enable bit in Control A register (CTRLA.WEN)
- Run-In-Standby bit in Control A register (CTRLA.RUNSTDBY)
- Always-On bit in control Control A (CTRLA.ALWAYSON)
- Watchdog Clear register (CLEAR)

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Required read-synchronization is denoted by the "Read-Synchronized" property in the register description.

26.6.8 Additional Features

26.6.8.1 Always-On Mode

The Always-On mode is enabled by setting the Always-On bit in the Control A register (CTRLA.ALWAYSON=1). When the Always-On mode is enabled, the WDT runs continuously, regardless of the state of CTRLA.ENABLE. Once written, the Always-On bit can only be cleared by a power-on reset. The Configuration (CONFIG) and Early Warning Control (EWCTRL) registers are read-only registers while the CTRLA.ALWAYSON bit is set. Thus, the time period configuration bits (CONFIG.PER, CONFIG.WINDOW, EWCTRL.EWOFFSET) of the WDT cannot be changed.

Enabling or disabling Window mode operation by writing the Window Enable bit (CTRLA.WEN) is allowed while in Always-On mode, but note that CONFIG.PER cannot be changed.

The Interrupt Clear and Interrupt Set registers are accessible in the Always-On mode. The Early Warning interrupt can still be enabled or disabled while in the Always-On mode, but note that EWCTRL.EWOFFSET cannot be changed.

Table WDT Operating Modes With Always-On shows the operation of the WDT for CTRLA.ALWAYSON=1.

Table 26-2. WDT Operating Modes With Always-On

WEN	Interrupt Enable	Mode
0	0	Always-on and normal mode
0	1	Always-on and normal mode with Early Warning interrupt

Downloaded from Arrow.com.

WEN	Interrupt Enable	Mode
1	0	Always-on and window mode
1	1	Always-on and window mode with Early Warning interrupt

26.6.8.2 Early Warning

The Early Warning interrupt notifies that the WDT is approaching its time-out condition. The Early Warning interrupt behaves differently in Normal mode and in Window mode.

In Normal mode, the Early Warning interrupt generation is defined by the Early Warning Offset in the Early Warning Control register (EWCTRL.EWOFFSET). The Early Warning Offset bits define the number of CLK_WDT_OSC clocks before the interrupt is generated, relative to the start of the watchdog time-out period.

The user must take caution when programming the Early Warning Offset bits. If these bits define an Early Warning interrupt generation time greater than the watchdog time-out period, the watchdog time-out system reset is generated prior to the Early Warning interrupt. Consequently, the Early Warning interrupt will never be generated.

In window mode, the Early Warning interrupt is generated at the start of the open window period. In a typical application where the system is in sleep mode, the Early Warning interrupt can be used to wake up and clear the Watchdog Timer, after which the system can perform other tasks or return to sleep mode.

> If the WDT is operating in Normal mode with CONFIG.PER = 0x2 and EWCTRL.EWOFFSET = 0x1, the Early Warning interrupt is generated 16 CLK_WDT_OSC clock cycles after the start of the time-out period. The time-out system reset is generated 32 CLK_WDT_OSC clock cycles after the start of the watchdog time-out period.

Downloaded from Arrow.com.

26.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	ALWAYSON	RUNSTDBY				WEN	ENABLE	
0x01	CONFIG	7:0	WINDOW[3:0]				PER[3:0]			
0x02	EWCTRL	7:0					EWOFFSET[3:0]			
0x03	Reserved									
0x04	INTENCLR	7:0								EW
0x05	INTENSET	7:0								EW
0x06	INTFLAG	7:0								EW
0x07	Reserved									
0x08	SYNCBUSY	7:0			CLEAR	ALWAYSON	RUNSTDBY	WEN	ENABLE	
		15:8								
		23:16								
		31:24								
0x0C	CLEAR	7:0	CLEAR[7:0]							

26.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to 26.5.8 Register Access Protection.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to 26.6.7 Synchronization.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

Downloaded from Arrow.com.

26.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: x initially determined from NVM User Row after reset
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	ALWAYSON	RUNSTDBY				WEN	ENABLE	
Access	R/W	R/W				R/W	R/W	
Reset	x	x				x	x	

Bit 7 – ALWAYSON Always-On
This bit allows the WDT to run continuously. After being set, this bit cannot be written to '0', and the WDT will remain enabled until a power-on Reset is received. When this bit is '1', the Control A register (CTRLA), the Configuration register (CONFIG) and the Early Warning Control register (EWCTRL) will be read-only, and any writes to these registers are not allowed.

Writing a '0' to this bit has no effect.

This bit is not Enable-Protected.

This bit is loaded from NVM User Row at start-up.

Value	Description
0	The WDT is enabled and disabled through the ENABLE bit.
1	The WDT is enabled and can only be disabled by a power-on reset (POR).

Bit 6 – RUNSTDBY Run in Standby
This bit controls the behavior of the watchdog during standby sleep mode. This bit can only be written when CTRLA.ENABLE is zero or CTRLA.ALWAYSON is one:

- When CTRLA.ALWAYSON=0, this bit is enable-protected by CTRLA.ENABLE.
- When CTRLA.ALWAYSON=1, this bit is not enable-protected by CTRLA.ENABLE.

These bits are loaded from NVM User Row at startup.

Value	Description
0	The WDT is disabled during standby sleep.
1	The WDT is enabled continues to operate during standby sleep.

Bit 2 – WEN Watchdog Timer Window Mode Enable
This bit enables Window mode. It can only be written if the peripheral is disabled unless CTRLA.ALWAYSON=1. The initial value of this bit is loaded from Flash Calibration.

This bit is loaded from NVM User Row at startup.

Value	Description
0	Window mode is disabled (normal operation).
1	Window mode is enabled.

Downloaded from Arrow.com.

Bit 1 – ENABLE Enable
This bit enables or disables the WDT. It can only be written if CTRLA.ALWAYSON=0.

Due to synchronization, there is delay between writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately, and the Enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

This bit is not Enable-Protected.

This bit is loaded from NVM User Row at startup.

Value	Description
0	The WDT is disabled.
1	The WDT is enabled.

Downloaded from Arrow.com.

26.8.2 Configuration

Name: CONFIG
Offset: 0x01
Reset: x initially determined from NVM User Row after reset
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	WINDOW[3:0]				PER[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bits 7:4 – WINDOW[3:0] Window Mode Time-Out Period
In Window mode, these bits determine the watchdog closed window period as a number of cycles of the 1.024kHz CLK_WDT_OSC clock.

These bits are loaded from NVM User Row at start-up.

Value	Name	Description
0x0	CYC8	8 clock cycles
0x1	CYC16	16 clock cycles
0x2	CYC32	32 clock cycles
0x3	CYC64	64 clock cycles
0x4	CYC128	128 clock cycles
0x5	CYC256	256 clock cycles
0x6	CYC512	512 clock cycles
0x7	CYC1024	1024 clock cycles
0x8	CYC2048	2048 clock cycles
0x9	CYC4096	4096 clock cycles
0xA	CYC8192	8192 clock cycles
0xB	CYC16384	16384 clock cycles
0xC-0xF	Reserved	Reserved

Bits 3:0 – PER[3:0] Time-Out Period
These bits determine the watchdog time-out period as a number of 1.024kHz CLK_WDTOSC clock cycles. In Window mode operation, these bits define the open window period.

These bits are loaded from NVM User Row at startup.

Value	Name	Description
0x0	CYC8	8 clock cycles
0x1	CYC16	16 clock cycles
0x2	CYC32	32 clock cycles
0x3	CYC64	64 clock cycles
0x4	CYC128	128 clock cycles
0x5	CYC256	256 clock cycles
0x6	CYC512	512 clock cycles
0x7	CYC1024	1024 clock cycles
0x8	CYC2048	2048 clock cycles

Downloaded from Arrow.com.

Value	Name	Description
0x9	CYC4096	4096 clock cycles
0xA	CYC8192	8192 clock cycles
0xB	CYC16384	16384 clock cycles
0xC – 0xF	-	Reserved

Downloaded from Arrow.com.

26.8.3 Early Warning Control

Name: EWCTRL
Offset: 0x02
Reset: x initially determined from NVM User Row after reset
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
					EWOFFSET[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					x	x	x	x

Bits 3:0 – EWOFFSET[3:0] Early Warning Interrupt Time Offset
These bits determine the number of GCLK_WDT clock cycles between the start of the watchdog time-out period and the generation of the Early Warning interrupt. These bits are loaded from NVM User Row at start-up.

Value	Name	Description
0x0	CYC8	8 clock cycles
0x1	CYC16	16 clock cycles
0x2	CYC32	32 clock cycles
0x3	CYC64	64 clock cycles
0x4	CYC128	128 clock cycles
0x5	CYC256	256 clock cycles
0x6	CYC512	512 clock cycles
0x7	CYC1024	1024 clock cycles
0x8	CYC2048	2048 clock cycles
0x9	CYC4096	4096 clock cycles
0xA	CYC8192	8192 clock cycles
0xB – 0xF	Reserved	Reserved

Downloaded from Arrow.com.

26.8.4 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Bit	7	6	5	4	3	2	1	0
								EW
Access								R/W
Reset								0

Bit 0 – EW Early Warning Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Early Warning Interrupt Enable bit, which disables the Early Warning interrupt.

Value	Description
0	The Early Warning interrupt is disabled.
1	The Early Warning interrupt is enabled.

Downloaded from Arrow.com.

26.8.5 Interrupt Enable Set

Name: INTENSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Bit	7	6	5	4	3	2	1	0
								EW
Access								R/W
Reset								0

Bit 0 – EW Early Warning Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the Early Warning Interrupt Enable bit, which enables the Early Warning interrupt.

Value	Description
0	The Early Warning interrupt is disabled.
1	The Early Warning interrupt is enabled.

Downloaded from Arrow.com.

26.8.6 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x06
Reset: 0x00
Property: N/A

Bit	7	6	5	4	3	2	1	0
								EW
Access								R/W
Reset								0

Bit 0 – EW Early Warning
This flag is cleared by writing a '1' to it.

This flag is set when an Early Warning interrupt occurs, as defined by the EWOFFSET bit group in EWCTRL.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Early Warning interrupt flag.

Downloaded from Arrow.com.

26.8.7 Synchronization Busy

Name: SYNCBUSY
Offset: 0x08
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			CLEAR	ALWAYSON	RUNSTDBY	WEN	ENABLE	
Access			R	R	R	R	R	
Reset			0	0	0	0	0	

Bit 5 – CLEAR Clear Synchronization Busy

Value	Description
0	Write synchronization of the CLEAR register is complete.
1	Write synchronization of the CLEAR register is ongoing.

Bit 4 – ALWAYSON Always-On Synchronization Busy

Value	Description
0	Write synchronization of the CTRLA.ALWAYSON bit is complete.
1	Write synchronization of the CTRLA.ALWAYSON bit is ongoing.

Bit 3 – RUNSTDBY Run-In-Standby Synchronization Busy

Value	Description
0	Write synchronization of the CTRLA.RUNSTDBY bit is complete.
1	Write synchronization of the CTRLA.RUNSTDBY bit is ongoing.

Bit 2 – WEN Window Enable Synchronization Busy

Value	Description
0	Write synchronization of the CTRLA.WEN bit is complete.
1	Write synchronization of the CTRLA.WEN bit is ongoing.

Downloaded from Arrow.com.

Bit 1 – ENABLE Enable Synchronization Busy

Value	Description
0	Write synchronization of the CTRLA.ENABLE bit is complete.
1	Write synchronization of the CTRLA.ENABLE bit is ongoing.

Downloaded from Arrow.com.

26.8.8 Clear

Name: CLEAR
Offset: 0x0C
Reset: 0x00
Property: Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	CLEAR[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – CLEAR[7:0] Watchdog Clear
In Normal mode, writing 0xA5 to this register during the watchdog time-out period will clear the Watchdog Timer and the watchdog time-out period is restarted.

In Window mode, any writing attempt to this register before the time-out period started (i.e., during TO_{WDTW}) will issue an immediate system Reset. Writing 0xA5 during the time-out period TO_{WDT} will clear the Watchdog Timer and the complete time-out sequence (first TO_{WDTW} then TO_{WDT}) is restarted.

In both modes, writing any other value than 0xA5 will issue an immediate system Reset.

Downloaded from Arrow.com.

27. RTC – Real-Time Counter

27.1 Overview

The Real-Time Counter (RTC) is a 32-bit counter with a 10-bit programmable prescaler that typically runs continuously to keep track of time. The RTC can wake up the device from sleep modes using the alarm/compare wake up, periodic wake up, or overflow wake up mechanisms, or from the wake inputs.

The RTC can generate periodic peripheral events from outputs of the prescaler, as well as alarm/compare interrupts and peripheral events, which can trigger at any counter value. Additionally, the timer can trigger an overflow interrupt and peripheral event, and can be reset on the occurrence of an alarm/compare match. This allows periodic interrupts and peripheral events at very long and accurate intervals.

The 10-bit programmable prescaler can scale down the clock source. By this, a wide range of resolutions and time-out periods can be configured. With a 32.768kHz clock source, the minimum counter tick interval is 30.5µs, and time-out periods can range up to 36 hours. For a counter tick interval of 1s, the maximum time-out period is more than 136 years.

27.2 Features

- 32-bit counter with 10-bit prescaler
- Multiple clock sources
- 32-bit or 16-bit counter mode
- One 32-bit or two 16-bit compare values
- Clock/Calendar mode
  - Time in seconds, minutes, and hours (12/24)
  - Date in day of month, month, and year
  - Leap year correction
- Digital prescaler correction/tuning for increased accuracy
- Overflow, alarm/compare match and prescaler interrupts and events
  - Optional clear on alarm/compare match
- 2 general purpose registers
- Tamper Detection
  - Timestamp on event or up to 5 inputs with debouncing
  - Active layer protection

Downloaded from Arrow.com.

27.3 Block Diagram

Figure 27-1. RTC Block Diagram (Mode 0 — 32-Bit Counter)

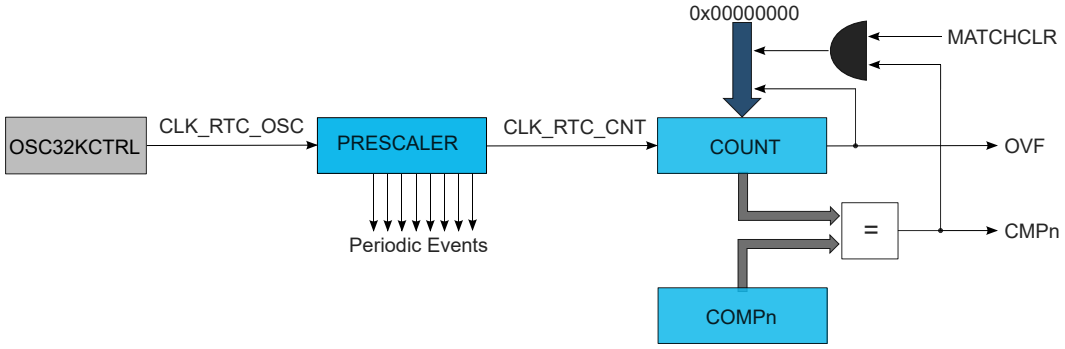


Figure 27-2. RTC Block Diagram (Mode 1 — 16-Bit Counter)

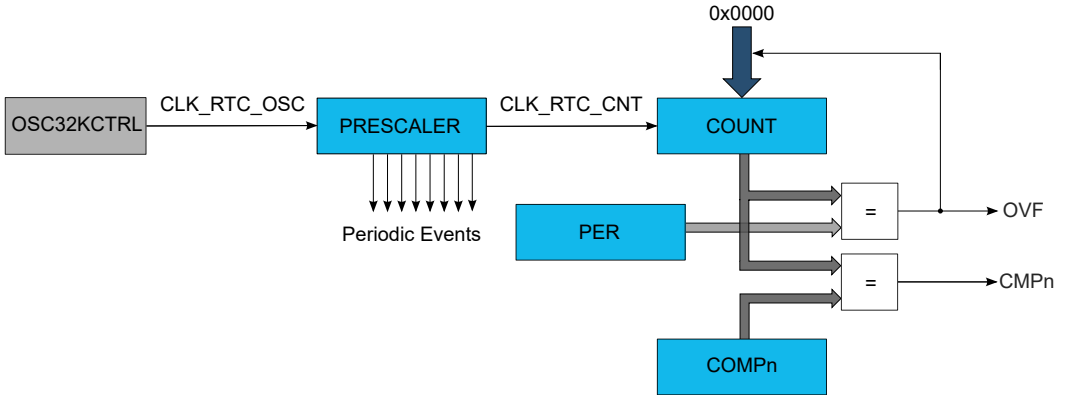
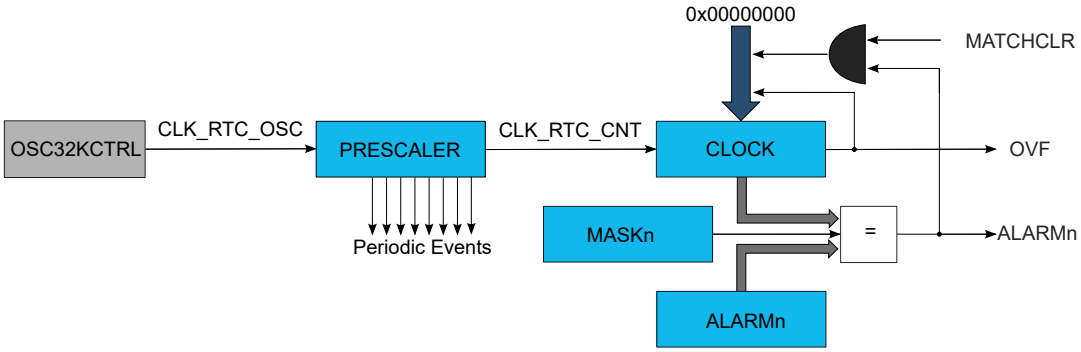
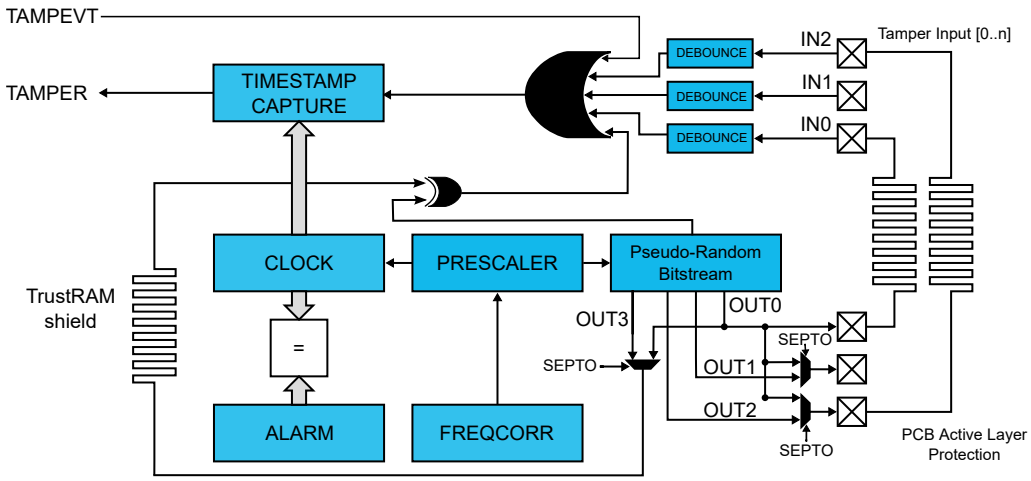


Figure 27-3. RTC Block Diagram (Mode 2 — Clock/Calendar)



Downloaded from Arrow.com.

Figure 27-4. RTC Block Diagram (Tamper Detection Use Case)



Related Links

27.6.2.3 32-Bit Counter (Mode 0)
27.6.2.4 16-Bit Counter (Mode 1)
27.6.2.5 Clock/Calendar (Mode 2)
27.6.8.4 Tamper Detection

27.4 Signal Description

Table 27-1. Signal Description

Signal	Description	Type
INn [n=0..4]	Tamper Detection Input	Digital input
OUT	Tamper Detection Output	Digital output

One signal can be mapped to one of several pins.

Related Links

4.1 Multiplexed Signals

27.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

27.5.1 I/O Lines

For more information on I/O configurations, refer to the "RTC Pinout" section.

Related Links: I/O Multiplexing and Considerations

27.5.2 Power Management

The RTC will continue to operate in any sleep mode where the selected source clock is running. The RTC interrupts can be used to wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes. Refer to the *Power Manager* for details on the different sleep modes.

Downloaded from Arrow.com.

The RTC will be reset only at power-on (POR) or by setting the Software Reset bit in the Control A register (CTRLA.SWRST=1).

Related Links

22. PM – Power Manager

27.5.3 Clocks

The RTC bus clock (CLK_RTC_APB) can be enabled and disabled in the Main Clock module MCLK, and the default state of CLK_RTC_APB can be found in Peripheral Clock Masking section.

A 32KHz or 1KHz oscillator clock (CLK_RTC_OSC) is required to clock the RTC. This clock must be configured and enabled in the 32KHz oscillator controller (OSC32KCTRL) before using the RTC.

This oscillator clock is asynchronous to the bus clock (CLK_RTC_APB). Due to this asynchronicity, writing to certain registers will require synchronization between the clock domains. Refer to 27.6.7 Synchronization for further details.

Related Links

24. OSC32KCTRL – 32KHz Oscillators Controller

19.6.2.6 Peripheral Clock Masking

27.5.4 DMA

The DMA request lines (or line if only one request) are connected to the DMA Controller (DMAC). Using the RTC DMA requests requires the DMA Controller to be configured first.

Related Links

28. DMAC – Direct Memory Access Controller

27.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. Using the RTC interrupt requires the Interrupt Controller to be configured first.

27.5.6 Events

The events are connected to the *Event System*.

Related Links

33. EVSYS – Event System

27.5.7 Debug Operation

When the CPU is halted in debug mode the RTC will halt normal operation. The RTC can be forced to continue operation during debugging. Refer to 27.8.7 DBGCTRL for details.

27.5.8 Register Access Protection

All registers with write-access are optionally write-protected by the peripheral access controller (PAC), except the following registers:

- Interrupt Flag Status and Clear (INTFLAG) register

Write-protection is denoted by the "PAC Write-Protection" property in the register description.

Write-protection does not apply to accesses through an external debugger. Refer to the PAC - Peripheral Access Controller for details.

Related Links

Downloaded from Arrow.com.

15. PAC - Peripheral Access Controller

27.5.9 SAM L11 TrustZone Specific Register Access Protection

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

27.5.10 Analog Connections

A 32.768kHz crystal can be connected to the XIN32 and XOUT32 pins, along with any required load capacitors. See the Electrical Characteristics Chapters for details on recommended crystal characteristics and load capacitors.

27.6 Functional Description

27.6.1 Principle of Operation

The RTC keeps track of time in the system and enables periodic events, as well as interrupts and events at a specified time. The RTC consists of a 10-bit prescaler that feeds a 32-bit counter. The actual format of the 32-bit counter depends on the RTC operating mode.

The RTC can function in one of these modes:

- Mode 0 - COUNT32: RTC serves as 32-bit counter
- Mode 1 - COUNT16: RTC serves as 16-bit counter
- Mode 2 - CLOCK: RTC serves as clock/calendar with alarm functionality

27.6.2 Basic Operation

27.6.2.1 Initialization

The following bits are enable-protected, meaning that they can only be written when the RTC is disabled (CTRLA.ENABLE=0):

- Operating Mode bits in the Control A register (CTRLA.MODE)
- Prescaler bits in the Control A register (CTRLA.PRESCALER)
- Clear on Match bit in the Control A register (CTRLA.MATCHCLR)
- Clock Representation bit in the Control A register (CTRLA.CLKREP)

The following registers are enable-protected:

- Control B register (CTRLB)
- Event Control register (EVCTRL)
- Tamper Control register (TAMPCTRL)
- Tamper Control B register (TAMPCTRLB)

Enable-protected bits and registers can be changed only when the RTC is disabled (CTRLA.ENABLE=0). If the RTC is enabled (CTRLA.ENABLE=1), these operations are necessary: first write

Downloaded from Arrow.com.

CTRLA.ENABLE=0 and check whether the write synchronization has finished, then change the desired bit field value. Enable-protected bits in CTRLA register can be written at the same time as CTRLA.ENABLE is written to '1', but not at the same time as CTRLA.ENABLE is written to '0'.

Enable-protection is denoted by the "Enable-Protected" property in the register description.

The RTC prescaler divides the source clock for the RTC counter.

Note: In Clock/Calendar mode, the prescaler must be configured to provide a 1Hz clock to the counter for correct operation.

The frequency of the RTC clock (CLK_RTC_CNT) is given by the following formula:

$$f_{\mathrm{CLK_RTC_CNT}} = \frac{f_{\mathrm{CLK_RTC_OSC}}}{2^{\mathrm{PRESCALER}}}$$

The frequency of the oscillator clock, CLK_RTC_OSC, is given by $f_{CLK_RTC_OSC}$, and $f_{CLK_RTC_CNT}$ is the frequency of the internal prescaled RTC clock, CLK_RTC_CNT.

27.6.2.2 Enabling, Disabling, and Resetting

The RTC is enabled by setting the Enable bit in the Control A register (CTRLA.ENABLE=1). The RTC is disabled by writing CTRLA.ENABLE=0.

The RTC is reset by setting the Software Reset bit in the Control A register (CTRLA.SWRST=1). All registers in the RTC, except DEBUG, will be reset to their initial state, and the RTC will be disabled. The RTC must be disabled before resetting it.

27.6.2.3 32-Bit Counter (Mode 0)

When the RTC Operating Mode bits in the Control A register (CTRLA.MODE) are written to 0x0, the counter operates in 32-bit Counter mode. The block diagram of this mode is shown in Figure 27-1. When the RTC is enabled, the counter will increment on every 0-to-1 transition of CLK_RTC_CNT. The counter will increment until it reaches the top value of 0xFFFFFFFF, and then wrap to 0x00000000. This sets the Overflow Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.OVF).

The RTC counter value can be read from or written to the Counter Value register (COUNT) in 32-bit format.

The counter value is continuously compared with the 32-bit Compare register (COMP0). When a compare match occurs, the Compare 0 Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMP0) is set on the next 0-to-1 transition of CLK_RTC_CNT.

If the Clear on Match bit in the Control A register (CTRLA.MATCHCLR) is '1', the counter is cleared on the next counter cycle when a compare match with COMP0 occurs. This allows the RTC to generate periodic interrupts or events with longer periods than the prescaler events. Note that when CTRLA.MATCHCLR is '1', INTFLAG.CMP0 and INTFLAG.OVF will both be set simultaneously on a compare match with COMP0.

27.6.2.4 16-Bit Counter (Mode 1)

When the RTC Operating Mode bits in the Control A register (CTRLA.MODE) are written to 0x1, the counter operates in 16-bit Counter mode as shown in Figure 27-2. When the RTC is enabled, the counter will increment on every 0-to-1 transition of CLK_RTC_CNT. In 16-bit Counter mode, the 16-bit Period register (PER) holds the maximum value of the counter. The counter will increment until it reaches the PER value, and then wrap to 0x0000. This sets the Overflow Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.OVF).

The RTC counter value can be read from or written to the Counter Value register (COUNT) in 16-bit format.

Downloaded from Arrow.com.

The counter value is continuously compared with the 16-bit Compare registers (COMPn, n=0..1). When a compare match occurs, the Compare n Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMPn, n=0..1) is set on the next 0-to-1 transition of CLK_RTC_CNT.

27.6.2.5 Clock/Calendar (Mode 2)

When the RTC Operating Mode bits in the Control A register (CTRLA.MODE) are written to 0x2, the counter operates in Clock/Calendar mode, as shown in Figure 27-3. When the RTC is enabled, the counter will increment on every 0-to-1 transition of CLK_RTC_CNT. The selected clock source and RTC prescaler must be configured to provide a 1Hz clock to the counter for correct operation in this mode.

The time and date can be read from or written to the Clock Value register (CLOCK) in a 32-bit time/date format. Time is represented as:

- Seconds
- Minutes
- Hours

Hours can be represented in either 12- or 24-hour format, selected by the Clock Representation bit in the Control A register (CTRLA.CLKREP). This bit can be changed only while the RTC is disabled.

The date is represented in this form:

- Day as the numeric day of the month (starting at 1)
- Month as the numeric month of the year (1 = January, 2 = February, etc.)
- Year as a value from 0x00 to 0x3F. This value must be added to a user-defined reference year. The reference year must be a leap year (2016, 2020 etc). Example: the year value 0x2D, added to a reference year 2016, represents the year 2061.

The RTC will increment until it reaches the top value of 23:59:59 December 31 of year value 0x3F, and then wrap to 00:00:00 January 1 of year value 0x00. This will set the Overflow Interrupt flag in the Interrupt Flag Status and Clear registers (INTFLAG.OVF).

The clock value is continuously compared with the 32-bit Alarm register (ALARM0). When an alarm match occurs, the Alarm 0 Interrupt flag in the Interrupt Flag Status and Clear registers (INTFLAG.ALARM0) is set on the next 0-to-1 transition of CLK_RTC_CNT. E.g. For a 1Hz clock counter, it means the Alarm 0 Interrupt flag is set with a delay of 1s after the occurrence of alarm match.

A valid alarm match depends on the setting of the Alarm Mask Selection bits in the Alarm 0 Mask register (MASK0.SEL). These bits determine which time/date fields of the clock and alarm values are valid for comparison and which are ignored.

If the Clear on Match bit in the Control A register (CTRLA.MATCHCLR) is set, the counter is cleared on the next counter cycle when an alarm match with ALARM0 occurs. This allows the RTC to generate periodic interrupts or events with longer periods than it would be possible with the prescaler events only (see 27.6.8.1 Periodic Intervals).
Note: When CTRLA.MATCHCLR is 1, INTFLAG.ALARM0 and INTFLAG.OVF will both be set simultaneously on an alarm match with ALARM0.

27.6.3 DMA Operation

The RTC generates the following DMA request:

- Tamper (TAMPER): The request is set on capture of the timestamp. The request is cleared when the Timestamp register is read.

Downloaded from Arrow.com.

If the CPU accesses the registers which are source for DMA request set/clear condition, the DMA request can be lost or the DMA transfer can be corrupted, if enabled.

27.6.4 Interrupts

The RTC has the following interrupt sources:

- Overflow (OVF): Indicates that the counter has reached its top value and wrapped to zero.
- Tamper (TAMPER): Indicates detection of valid signal on a tamper input pin or tamper event input.
- Compare (CMPn): Indicates a match between the counter value and the compare register.
- Alarm (ALARMn): Indicates a match between the clock value and the alarm register.
- Period n (PERn): The corresponding bit in the prescaler has toggled. Refer to 27.6.8.1 Periodic Intervals for details.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by setting the corresponding bit in the Interrupt Enable Set register (INTENSET=1), and disabled by setting the corresponding bit in the Interrupt Enable Clear register (INTENCLR=1).

An interrupt request is generated when the interrupt flag is raised and the corresponding interrupt is enabled. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is disabled or the RTC is reset. See the description of the INTFLAG registers for details on how to clear interrupt flags.

All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. Refer to the Nested Vector Interrupt Controller for details. The user must read the INTFLAG register to determine which interrupt condition is present.

Note: Interrupts must be globally enabled for interrupt requests to be generated. Refer to the Nested Vector Interrupt Controller for details.

27.6.5 Events

The RTC can generate the following output events:

- Overflow (OVF): Generated when the counter has reached its top value and wrapped to zero.
- Tamper (TAMPER): Generated on detection of valid signal on a tamper input pin or tamper event input.
- Compare (CMPn): Indicates a match between the counter value and the compare register.
- Alarm (ALARM): Indicates a match between the clock value and the alarm register.
- Period n (PERn): The corresponding bit in the prescaler has toggled. Refer to 27.6.8.1 Periodic Intervals for details.
- Periodic Daily (PERD): Generated when the COUNT/CLOCK has incremented at a fixed period of time.

Setting the Event Output bit in the Event Control Register (EVCTRL.xxxEO=1) enables the corresponding output event. Writing a zero to this bit disables the corresponding output event. Refer to the EVSYS - Event System for details on configuring the event system.

The RTC can take the following actions on an input event:

- Tamper (TAMPEVT): Capture the RTC counter to the timestamp register. See *Tamper Detection*.

Writing a one to an Event Input bit into the Event Control register (EVCTRL.xxxEI) enables the corresponding action on input event. Writing a zero to this bit disables the corresponding action on input event.

Downloaded from Arrow.com.

Related Links

33. EVSYS – Event System

27.6.6 Sleep Mode Operation

The RTC will continue to operate in any sleep mode where the source clock is active. The RTC *interrupts* can be used to wake up the device from a sleep mode. RTC *events* can trigger other operations in the system without exiting the sleep mode.

An interrupt request will be generated after the wake-up if the Interrupt Controller is configured accordingly. Otherwise the CPU will wake up directly, without triggering any interrupt. In this case, the CPU will continue executing right from the first instruction that followed the entry into sleep.

The periodic events can also wake up the CPU through the interrupt function of the Event System. In this case, the event must be enabled and connected to an event channel with its interrupt enabled. See *Event System* for more information.

27.6.7 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in Control A register, CTRLA.SWRST
- Enable bit in Control A register, CTRLA.ENABLE
- Count Read Synchronization bit in Control A register (CTRLA.COUNTSYNC)
- Clock Read Synchronization bit in Control A register (CTRLA.COUNTSYNC)

The following registers are synchronized when written:

- Counter Value register, COUNT
- Clock Value register, CLOCK
- Counter Period register, PER
- Compare n Value registers, COMPn
- Alarm n Value registers, ALARMn
- Frequency Correction register, FREQCORR
- Alarm n Mask register, MASKn
- The General Purpose n registers (GPn)

The following registers are synchronized when read:

- The Counter Value register, COUNT, if the Counter Read Sync Enable bit in CTRLA (CTRLA.COUNTSYNC) is '1'
- The Clock Value register, CLOCK, if the Clock Read Sync Enable bit in CTRLA (CTRLA.CLOCKSYNC) is '1'
- The Timestamp Value register (TIMESTAMP)

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Required read-synchronization is denoted by the "Read-Synchronized" property in the register description.

Downloaded from Arrow.com.

27.6.8 Additional Features

27.6.8.1 Periodic Intervals

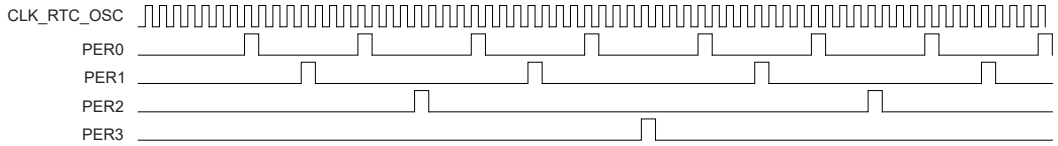
The RTC prescaler can generate interrupts and events at periodic intervals, allowing flexible system tick creation. Any of the upper eight bits of the prescaler (bits 2 to 9) can be the source of an interrupt/event. When one of the eight Periodic Event Output bits in the Event Control register (EVCTRL.PEREO[n=0..7]) is '1', an event is generated on the 0-to-1 transition of the related bit in the prescaler, resulting in a periodic event frequency of:

$$f_{\mathrm{PERIODIC(n)}} = \frac{f_{\mathrm{CLK_RTC_OSC}}}{2^{n+3}}$$

$f_{CLK_RTC_OSC}$ is the frequency of the internal prescaler clock CLK_RTC_OSC, and n is the position of the EVCTRL.PEREOn bit. For example, PER0 will generate an event every eight CLK_RTC_OSC cycles, PER1 every 16 cycles, etc. This is shown in the figure below.

Periodic events are independent of the prescaler setting used by the RTC counter, except if CTRLA.PRESCALER is zero. Then, no periodic events will be generated.

Figure 27-5. Example Periodic Events



27.6.8.2 Frequency Correction

The RTC Frequency Correction module employs periodic counter corrections to compensate for a too-slow or too-fast oscillator. Frequency correction requires that CTRLA.PRESCALER is greater than 1.

The digital correction circuit adds or subtracts cycles from the RTC prescaler to adjust the frequency in approximately 1ppm steps. Digital correction is achieved by adding or skipping a single count in the prescaler once every 8192 CLK_RTC_OSC cycles. The Value bit group in the Frequency Correction register (FREQCORR.VALUE) determines the number of times the adjustment is applied over 128 of these periods. The resulting correction is as follows:

$$\text{Correction in ppm} = \frac{\text{FREQCORR.VALUE}}{8192 \cdot 128} \cdot 10^6 \text{ppm}$$

This results in a resolution of 0.95367ppm.

The Sign bit in the Frequency Correction register (FREQCORR.SIGN) determines the direction of the correction. A positive value will add counts and increase the period (reducing the frequency), and a negative value will reduce counts per period (speeding up the frequency).

Digital correction also affects the generation of the periodic events from the prescaler. When the correction is applied at the end of the correction cycle period, the interval between the previous periodic event and the next occurrence may also be shortened or lengthened depending on the correction value.

27.6.8.3 General Purpose Registers

The RTC includes four General Purpose registers (GPn). These registers are reset only when the RTC is reset or when tamper detection occurs while CTRLA.GPTRST=1, and remain powered while the RTC is powered. They can be used to store user-defined values while other parts of the system are powered off.

The general purpose registers 2*n and 2*n+1 are enabled by writing a '1' to the General Purpose Enable bit n in the Control B register (CTRLB.GPnEN).

Downloaded from Arrow.com.

The GP registers share internal resources with the COMPARE/ALARM features. Each COMPARE/ALARM register have a separate read buffer and write buffer. When the general purpose feature is enabled the even GP uses the read buffer while the odd GP uses the write buffer.

When the COMPARE/ALARM register is written, the write buffer hold temporarily the COMPARE/ALARM value until the synchronisation is complete (bit SYNCBUSY.COMPn going to 0). After the write is completed the write buffer can be used as a odd general purpose register whithout affecting the COMPARE/ALARM function.

If the COMPARE/ALARM function is not used, the read buffer can be used as an even general purpose register. In this case writing the even GP will temporarirely use the write buffer until the synchronisation is complete (bit SYNCBUSY.GPn going to 0). Thus an even GP must be written before writing the odd GP. Changing or writing an even GP needs to temporarily save the value of the odd GP.

Before using an even GP, the associated COMPARE/ALARM feature must be disabled by writing a '1' to the General Purpose Enable bit in the Control B register (CTRLB.GPnEN). To re-enable the compare/alarm, CTRLB.GPnEN must be written to zero and the associated COMPn/ALARMn must be written with the correct value.

An example procedure to write the general purpose registers GP0 and GP1 is:

1. Wait for any ongoing write to COMP0 to complete (SYNCBUSY.COMP0 = 0). If the RTC is operating in Mode 1, wait for any ongoing write to COMP1 to complete as well (SYNCBUSY.COMP1 = 0).
2. Write CTRLB.GP0EN = 1 if GP0 is needed.
3. Write GP0 if needed.
4. Wait for any ongoing write to GP0 to complete (SYNCBUSY.GP0 = 0). Note that GP1 will also show as busy when GP0 is busy.
5. Write GP1 if needed.

The following table provides the correspondence of General Purpose Registers and the COMPARE/ALARM read or write buffer in all RTC modes.

Table 27-2. General Purpose Registers Versus Compare/Alarm Registers: n in 0, 2, 4, 6...

Register	Mode 0	Mode 1	Mode 2	Write Before
GPn	COMPn/2 write buffer	(COMPn , COMPn +1) write buffer	ALARMn/2 write buffer	GPn+1
GPn+1	COMPn/2 read buffer	(COMPn , COMPn +1) read buffer	ALARMn/2 read buffer	-

27.6.8.4 Tamper Detection

The RTC provides four tamper channels that can be used for tamper detection.

The action of each tamper channel is configured using the Input n Action bits in the Tamper Control register (TAMPCTRL.INnACT):

- Off: Detection for tamper channel n is disabled.
- Wake: A transition on INn input (tamper channel n) matching TAMPCTRL.TAMPLVLn will be detected and the tamper interrupt flag (INTFLAG.TAMPER) will be set. The RTC value will not be captured in the TIMESTAMP register.
- Capture: A transition on INn input (tamper channel n) matching TAMPCTRL.TAMPLVLn will be detected and the tamper interrupt flag (INTFLAG.TAMPER) will be set. The RTC value will be captured in the TIMESTAMP register.

Downloaded from Arrow.com.

- Active Layer Protection: A mismatch of an internal RTC signal routed between INn and OUTn pins will be detected and the tamper interrupt flag (INTFLAG.TAMPER) will be set. The RTC value will be captured in the TIMESTAMP register.

In order to determine which tamper source caused a tamper event, the Tamper ID register (TAMPID) provides the detection status of each tamper channel. These bits remain active until cleared by software.

A single interrupt request (TAMPER) is available for all tamper channels.

The RTC also supports an input event (TAMPEVT) for generating a tamper condition within the Event System. The tamper input event is enabled by the Tamper Input Event Enable bit in the Event Control register (EVCTRL.TAMPEVEI).

Up to four polarity external inputs (INn) can be used for tamper detection. The polarity for each input is selected with the Tamper Level bits in the Tamper Control register (TAMPCTRL.TAMPLVLn).

Separate debouncers are embedded for each external input. The debouncer for each input is enabled/disabled with the Debounce Enable bits in the Tamper Control register (TAMPCTRL.DEBNCn). The debouncer configuration is fixed for all inputs as set by the Control B register (CTRLB). The debouncing period duration is configurable using the Debounce Frequency field in the Control B register (CTRLB.DEBF). The period is set for all debouncers (i.e., the duration cannot be adjusted separately for each debouncer).

When TAMPCTRL.DEBNCn = 0, INn is detected asynchronously. See Figure 27-6 for an example.

When TAMPCTRL.DEBNCn = 1, the detection time depends on whether the debouncer operates synchronously or asynchronously, and whether majority detection is enabled or not. Refer to the table below for more details. Synchronous versus asynchronous stability debouncing is configured by the Debounce Asynchronous Enable bit in the Control B register (CTRLB.DEBASYNC):

- Synchronous (CTRLB.DEBASYNC = 0): INn is synchronized in two CLK_RTC periods and then must remain stable for four CLK_RTC_DEB periods before a valid detection occurs. See Figure 27-7 for an example.
- Asynchronous (CTRLB.DEBASYNC = 1): The first edge on INn is detected. Further detection is blanked until INn remains stable for four CLK_RTC_DEB periods. See Figure 27-8 for an example.

Majority debouncing is configured by the Debounce Majority Enable bit in the Control B register (CTRLB.DEBMAJ). INn must be valid for two out of three CLK_RTC_DEB periods. See Figure 27-9 for an example.

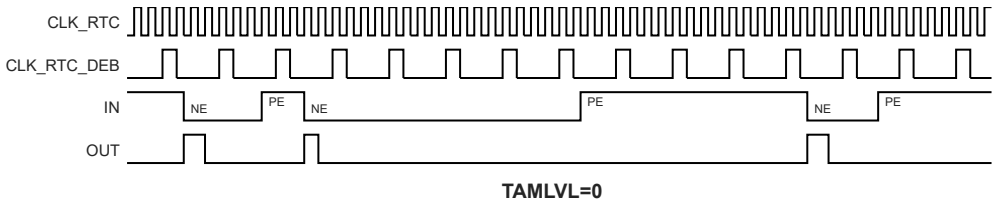
Table 27-3. Debouncer Configuration

TAMPCTRL.DEBNCn	CTRLB.DEBMAJ	CTRLB.DEBASYNC	Description
0	X	X	Detect edge on INn with no debouncing. Every edge detected is immediately triggered.
1	0	0	Detect edge on INn with synchronous stability debouncing. Edge detected is only triggered when INn is stable for 4 consecutive CLK_RTC_DEB periods.
1	0	1	Detect edge on INn with asynchronous stability debouncing. First detected edge is triggered immediately. All subsequent detected edges are

Downloaded from Arrow.com.

TAMPCTRL.DEBNCn	CTRLB.DEBMAJ	CTRLB.DEBASYNC	Description
			ignored until INn is stable for 4 consecutive CLK_RTC_DEB periods.
1	1	X	Detect edge on INn with majority debouncing. Pin INn is sampled for 3 consecutive CLK_RTC_DEB periods. Signal level is determined by majority-rule (LLL, LLH, LHL, HLL = '0' and LHH, HLH, HHL, HHH = '1').

Figure 27-6. Edge Detection with Debouncer Disabled



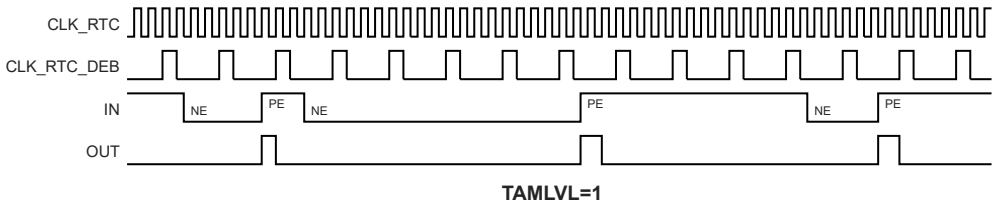
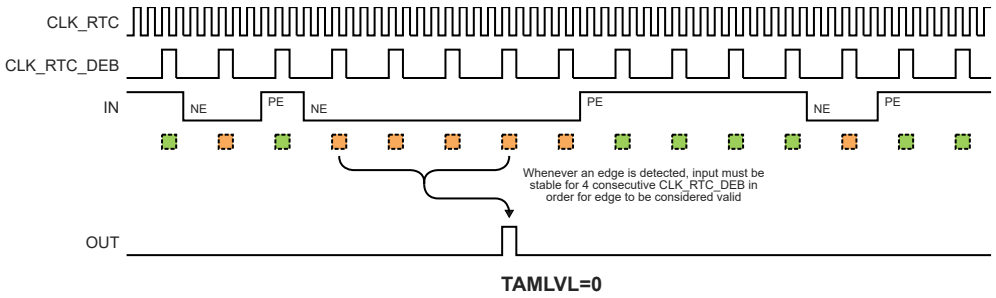
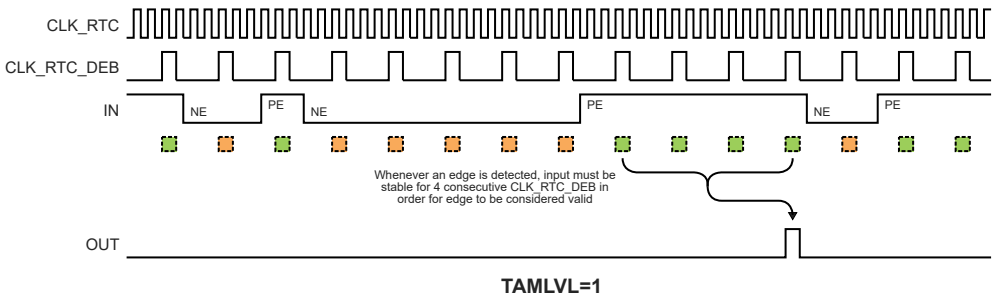


Figure 27-7. Edge Detection with Synchronous Stability Debouncing





Downloaded from Arrow.com.

Figure 27-8. Edge Detection with Asynchronous Stability Debouncing

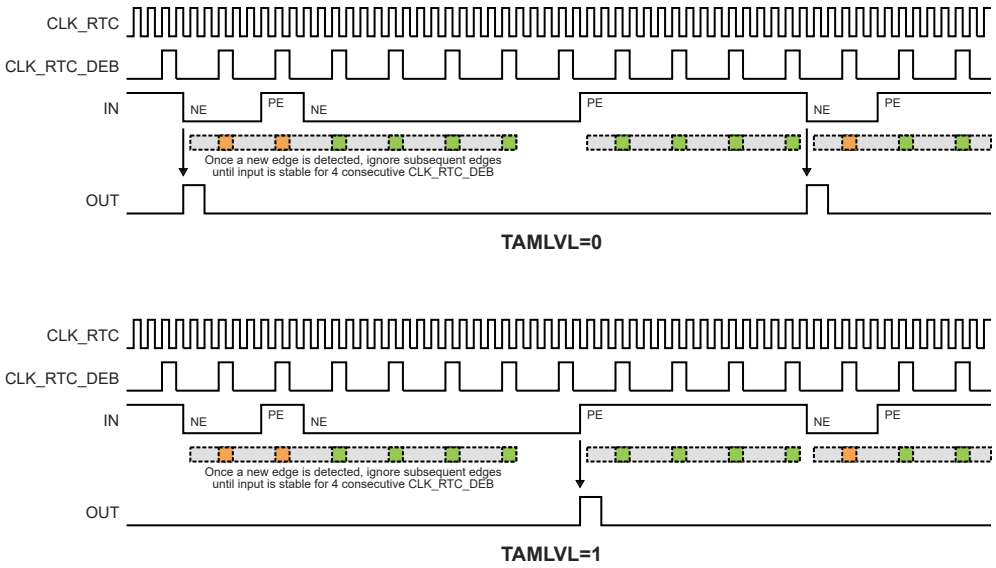
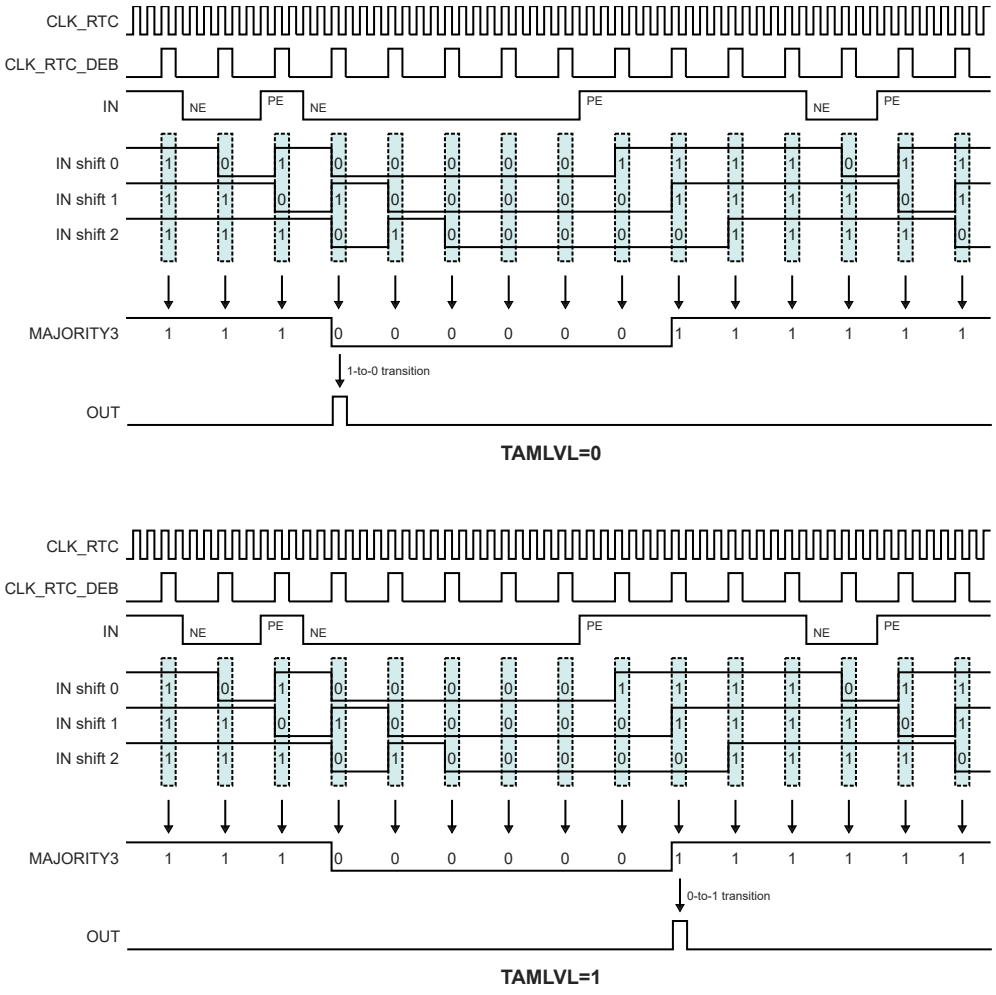


Figure 27-9. Edge Detection with Majority Debouncing



Downloaded from Arrow.com.

Related Links

27.3 Block Diagram

27.6.8.4.1 Timestamp

27.6.8.4.2 Active Layer Protection

27.6.8.4.1 Timestamp

As part of tamper detection the RTC can capture the counter value (COUNT/CLOCK) into the TIMESTAMP register. Three CLK_RTC periods are required to detect the tampering condition and capture the value. The TIMESTAMP value can be read once the Tamper flag in the Interrupt Flag register (INTFLAG.TAMPER) is set. If the DMA Enable bit in the Control B register (CTRLB.DMAEN) is '1', a DMA request will be triggered by the timestamp. In order to determine which tamper source caused a capture, the Tamper ID register (TAMPID) provides the detection status of each tamper channel and the tamper input event. A DMA transfer can then read both TIMESTAMP and TAMPID in succession.

A new timestamp value cannot be captured until the Tamper flag is cleared, either by reading the timestamp or by writing a '1' to INTFLAG.TAMPER. If several tamper conditions occur in a short window before the flag is cleared, only the first timestamp may be logged. However, the detection of each tamper will still be recorded in TAMPID.

The Tamper Input Event (TAMPEVT) will always perform a timestamp capture. To capture on the external inputs (INn), the corresponding Input Action field in the Tamper Control register (TAMPCTRL.INnACT) must be written to '1'. If an input is set for wake functionality it does not capture the timestamp; however the Tamper flag and TAMPID will still be updated.

Related Links

27.6.8.4 Tamper Detection

27.6.8.4.2 Active Layer Protection

The RTC provides a mean of detecting broken traces on the PCB , also known as Active layer Protection. In this mode, a generated internal RTC signal can be directly routed over critical components on the board using RTC OUT output pin to one RTC INn input pin. A tamper condition is detected if there is a mismatch on the generated RTC signal.

The Active Layer Protection mode and the generation of the RTC signal is enabled by setting the RTCOUT bit in the Control B register (CTRLB.RTCOUT).

Enabling active layer protection requires the following steps:

- Enable the RTC prescaler output by writing a one to the RTC Out bit in the Control B register (CTRLB.RTCOUT). The I/O pins must also be configured to correctly route the signal to the external pins.
- Select the frequency of the output signal by configuring the RTC Active Layer Frequency field in the Control B register (CTRLB.ACTF).

  $$\mathrm{GCLK_RTC_OUT} = \frac{\mathrm{CLK_RTC}}{2^{\mathrm{CTRLB.ACTF}+1}}$$
- Enable the tamper input n (INn) in active layer mode by writing 3 to the corresponding Input Action field in the Tamper Control register (TAMPCTRL.INnACT). When active layer protection is enabled and INn and OUTn pin are used, the value of INn is sampled on the falling edge of CLK_RTC and compared to the expected value of OUTn. Therefore up to one half of a CLK_RTC period is available for propagation delay through the trace.
- Enable Acitive Layer Protection by setting CTRLB.RTCOUT bit.

Related Links

Downloaded from Arrow.com.

27.6.8.4 Tamper Detection

Downloaded from Arrow.com.

27.7 Register Summary - Mode 0 - 32-Bit Counter

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	MATCHCLR				MODE[1:0]		ENABLE	SWRST
		15:8	COUNTSYNC	GPTRST			PRESCALER[3:0]			
0x02	CTRLB	7:0	DMAEN	RTCOUT	DEBASYNC	DEBMAJ				GP0EN
		15:8	SEPTO	ACTF[2:0]				DEBF[2:0]		
0x04	EVCTRL	7:0	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
		15:8	OVFEO	TAMPEREO						CMPEO0
		23:16								TAMPEVEI
		31:24								PERDEO
0x08	INTENCLR	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
		15:8	OVF	TAMPER						CMP0
0x0A	INTENSET	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
		15:8	OVF	TAMPER						CMP0
0x0C	INTFLAG	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
		15:8	OVF	TAMPER						CMP0
0x0E	DBGCTRL	7:0								DBGRUN
0x0F	Reserved									
0x10	SYNCBUSY	7:0			COMP0		COUNT	FREQCORR	ENABLE	SWRST
		15:8	COUNTSYNC							
		23:16							GPn[1:0]	
		31:24								
0x14	FREQCORR	7:0	SIGN	VALUE[6:0]						
0x15 ... 0x17	Reserved									
0x18	COUNT	7:0	COUNT[7:0]							
		15:8	COUNT[15:8]							
		23:16	COUNT[23:16]							
		31:24	COUNT[31:24]							
0x1C ... 0x1F	Reserved									
0x20	COMP	7:0	COMP[7:0]							
		15:8	COMP[15:8]							
		23:16	COMP[23:16]							
		31:24	COMP[31:24]							
0x24 ... 0x3F	Reserved									
0x40	GP0	7:0	GP[7:0]							
		15:8	GP[15:8]							
		23:16	GP[23:16]							
		31:24	GP[31:24]							
0x44	GP1	7:0	GP[7:0]							

Downloaded from Arrow.com.

Offset	Name	Bit Pos.								
		15:8	GP[15:8]							
		23:16	GP[23:16]							
		31:24	GP[31:24]							
0x48 ... 0x5F	Reserved									
0x60	TAMPCTRL	7:0	IN3ACT[1:0]		IN2ACT[1:0]		IN1ACT[1:0]		IN0ACT[1:0]	
		15:8								
		23:16					TAMLVL3	TAMLVL2	TAMLVL1	TAMLVL0
		31:24					DEBNC3	DEBNC2	DEBNC1	DEBNC0
0x64	TIMESTAMP	7:0	COUNT[7:0]							
		15:8	COUNT[15:8]							
		23:16	COUNT[23:16]							
		31:24	COUNT[31:24]							
0x68	TAMPID	7:0					TAMPID3	TAMPID2	TAMPID1	TAMPID0
		15:8								
		23:16								
		31:24	TAMPEVT							
0x6C	TAMPCTRLB	7:0					ALSI3	ALSI2	ALSI1	ALSI0
		15:8								
		23:16								
		31:24								

27.8 Register Description - Mode 0 - 32-Bit Counter

This Register Description section is valid if the RTC is in COUNT32 mode (CTRLA.MODE=0).

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

Downloaded from Arrow.com.

27.8.1 Control A in COUNT32 mode (CTRLA.MODE=0)

Name: CTRLA
Offset: 0x00
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	COUNTSYNC	GPTRST			PRESCALER[3:0]			
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MATCHCLR				MODE[1:0]		ENABLE	SWRST
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit 15 – COUNTSYNC COUNT Read Synchronization Enable
The COUNT register requires synchronization when reading. Disabling the synchronization will prevent reading valid values from the COUNT register.
This bit is not enable-protected.

Value	Description
0	COUNT read synchronization is disabled
1	COUNT read synchronization is enabled

Bit 14 – GPTRST GP Registers Reset On Tamper Enable
Only GP registers enabled by the CTRLB.GPnEN bits are affected. This bit can be written only when the peripheral is disabled.
This bit is not synchronized.

Bits 11:8 – PRESCALER[3:0] Prescaler
These bits define the prescaling factor for the RTC clock source (GCLK_RTC) to generate the counter clock (CLK_RTC_CNT). Periodic events and interrupts are not available when the prescaler is off. These bits are not synchronized.

Value	Name	Description
0x0	OFF	CLK_RTC_CNT = GCLK_RTC/1
0x1	DIV1	CLK_RTC_CNT = GCLK_RTC/1
0x2	DIV2	CLK_RTC_CNT = GCLK_RTC/2
0x3	DIV4	CLK_RTC_CNT = GCLK_RTC/4
0x4	DIV8	CLK_RTC_CNT = GCLK_RTC/8
0x5	DIV16	CLK_RTC_CNT = GCLK_RTC/16
0x6	DIV32	CLK_RTC_CNT = GCLK_RTC/32
0x7	DIV64	CLK_RTC_CNT = GCLK_RTC/64
0x8	DIV128	CLK_RTC_CNT = GCLK_RTC/128
0x9	DIV256	CLK_RTC_CNT = GCLK_RTC/256
0xA	DIV512	CLK_RTC_CNT = GCLK_RTC/512

Downloaded from Arrow.com.

Value	Name	Description
0xB	DIV1024	CLK_RTC_CNT = GCLK_RTC/1024
0xC-0xF	-	Reserved

Bit 7 – MATCHCLR Clear on Match
This bit defines if the counter is cleared or not on a match.

This bit is not synchronized.

Value	Description
0	The counter is not cleared on a Compare/Alarm 0 match
1	The counter is cleared on a Compare/Alarm 0 match

Bits 3:2 – MODE[1:0] Operating Mode
This bit group defines the operating mode of the RTC.

This bit is not synchronized.

Value	Name	Description
0x0	COUNT32	Mode 0: 32-bit counter
0x1	COUNT16	Mode 1: 16-bit counter
0x2	CLOCK	Mode 2: Clock/calendar
0x3	-	Reserved

Bit 1 – ENABLE Enable
Due to synchronization there is a delay between writing CTRLA.ENABLE and until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

Value	Description
0	The peripheral is disabled
1	The peripheral is enabled

Bit 0 – SWRST Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the RTC (except DBGCTRL) to their initial state, and the RTC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Due to synchronization there is a delay between writing CTRLA.SWRST and until the reset is complete. CTRLA.SWRST will be cleared when the reset is complete.

Value	Description
0	There is not reset operation ongoing
1	The reset operation is ongoing

Downloaded from Arrow.com.

27.8.2 Control B in COUNT32 mode (CTRLA.MODE=0)

Name: CTRLB
Offset: 0x02
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
	SEPTO	ACTF[2:0]				DEBF[2:0]		
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
	DMAEN	RTCOUT	DEBASYNC	DEBMAJ				GP0EN
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bit 15 – SEPTO Separate Tamper Outputs

Value	Description
0	IN[n] is compared to OUT[0].
1	IN[n] is compared to OUT[n].

Bits 14:12 – ACTF[2:0] Active Layer Frequency
These bits define the prescaling factor for the RTC clock output (OUT) used during active layer protection in terms of the CLK_RTC.

Value	Name	Description
0x0	DIV2	CLK_RTC_OUT = CLK_RTC / 2
0x1	DIV4	CLK_RTC_OUT = CLK_RTC / 4
0x2	DIV8	CLK_RTC_OUT = CLK_RTC / 8
0x3	DIV16	CLK_RTC_OUT = CLK_RTC / 16
0x4	DIV32	CLK_RTC_OUT = CLK_RTC / 32
0x5	DIV64	CLK_RTC_OUT = CLK_RTC / 64
0x6	DIV128	CLK_RTC_OUT = CLK_RTC / 128
0x7	DIV256	CLK_RTC_OUT = CLK_RTC / 256

Bits 10:8 – DEBF[2:0] Debounce Frequency
These bits define the prescaling factor for the input debouncers in terms of the CLK_RTC.

Value	Name	Description
0x0	DIV2	CLK_RTC_DEB = CLK_RTC / 2
0x1	DIV4	CLK_RTC_DEB = CLK_RTC / 4
0x2	DIV8	CLK_RTC_DEB = CLK_RTC / 8
0x3	DIV16	CLK_RTC_DEB = CLK_RTC / 16
0x4	DIV32	CLK_RTC_DEB = CLK_RTC / 32
0x5	DIV64	CLK_RTC_DEB = CLK_RTC / 64
0x6	DIV128	CLK_RTC_DEB = CLK_RTC / 128
0x7	DIV256	CLK_RTC_DEB = CLK_RTC / 256

Downloaded from Arrow.com.

Bit 7 – DMAEN DMA Enable
The RTC can trigger a DMA request when the timestamp is ready in the TIMESTAMP register.

Value	Description
0	Tamper DMA request is disabled. Reading TIMESTAMP has no effect on INTFLAG.TAMPER.
1	Tamper DMA request is enabled. Reading TIMESTAMP will clear INTFLAG.TAMPER.

Bit 6 – RTCOUT RTC Output Enable

Value	Description
0	The RTC active layer output is disabled.
1	The RTC active layer output is enabled.

Bit 5 – DEBASYNC Debouncer Asynchronous Enable

Value	Description
0	The tamper input debouncers operate synchronously.
1	The tamper input debouncers operate asynchronously.

Bit 4 – DEBMAJ Debouncer Majority Enable

Value	Description
0	The tamper input debouncers match three equal values.
1	The tamper input debouncers match majority two of three values.

Bit 0 – GP0EN General Purpose 0 Enable

Value	Description
0	COMP0 compare function enabled. GP0/GP1 disabled.
1	COMP0 compare function disabled. GP0/GP1 enabled.

Downloaded from Arrow.com.

27.8.3 Event Control in COUNT32 mode (CTRLA.MODE=0)

Name: EVCTRL
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
								PERDEO
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
								TAMPEVEI
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
	OVFEO	TAMPEREO						CMPEO0
Access	R/W	R/W						R/W
Reset	0	0						0

Bit	7	6	5	4	3	2	1	0
	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 24 – PERDEO Periodic Interval Daily Event Output Enable

Value	Description
0	Periodic Daily event is disabled and will not be generated.
1	Periodic Daily event is enabled and will be generated. The event occurs at the overflow of the RTC counter (i.e., when the RTC counter goes from 0xFFFF to 0x0000).

Bit 16 – TAMPEVEI Tamper Event Input Enable

Value	Description
0	Tamper event input is disabled and incoming events will be ignored.
1	Tamper event input is enabled and incoming events will capture the COUNT value.

Bit 15 – OVFEO Overflow Event Output Enable

Value	Description
0	Overflow event is disabled and will not be generated.
1	Overflow event is enabled and will be generated for every overflow.

Bit 14 – TAMPEREO Tamper Event Output Enable

Downloaded from Arrow.com.

Value	Description
0	Tamper event output is disabled and will not be generated.
1	Tamper event output is enabled and will be generated for every tamper input.

Bit 8 – CMPEO0 Compare 0 Event Output Enable

Value	Description
0	Compare 0 event is disabled and will not be generated.
1	Compare 0 event is enabled and will be generated for every compare match.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PEREOn Periodic Interval n Event Output Enable [n = 7..0]

Value	Description
0	Periodic Interval n event is disabled and will not be generated.
1	Periodic Interval n event is enabled and will be generated.

Downloaded from Arrow.com.

27.8.4 Interrupt Enable Clear in COUNT32 mode (CTRLA.MODE=0)

Name: INTENCLR
Offset: 0x08
Reset: 0x0000
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER						CMP0
Access	R/W	R/W						R/W
Reset	0	0						0

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bit 14 – TAMPER Tamper Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this but will clear the Tamper Interrupt Enable bit, which disables the Tamper interrupt.

Value	Description
0	The Tamper interrupt is disabled.
1	The Tamper interrupt is enabled.

Bit 8 – CMP0 Compare 0 Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Compare 0 Interrupt Enable bit, which disables the Compare 0 interrupt.

Value	Description
0	The Compare 0 interrupt is disabled.
1	The Compare 0 interrupt is enabled.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect.

Downloaded from Arrow.com.

Writing a '1' to this bit will clear the Periodic Interval n Interrupt Enable bit, which disables the Periodic Interval n interrupt.

Value	Description
0	Periodic Interval n interrupt is disabled.
1	Periodic Interval n interrupt is enabled.

Downloaded from Arrow.com.

27.8.5 Interrupt Enable Set in COUNT32 mode (CTRLA.MODE=0)

Name: INTENSET
Offset: 0x0A
Reset: 0x0000
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER						CMP0
Access	R/W	R/W						R/W
Reset	0	0						0

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bit 14 – TAMPER Tamper Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Tamper Interrupt Enable bit, which enables the Tamper interrupt.

Value	Description
0	The Tamper interrupt is disabled.
1	The Tamper interrupt is enabled.

Bit 8 – CMP0 Compare 0 Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Compare 0 Interrupt Enable bit, which enables the Compare 0 interrupt.

Value	Description
0	The Compare 0 interrupt is disabled.
1	The Compare 0 interrupt is enabled.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Periodic Interval n Interrupt Enable bit, which enables the Periodic Interval n interrupt.

Downloaded from Arrow.com.

Value	Description
0	Periodic Interval n interrupt is disabled.
1	Periodic Interval n interrupt is enabled.

Downloaded from Arrow.com.

27.8.6 Interrupt Flag Status and Clear in COUNT32 mode (CTRLA.MODE=0)

Name: INTFLAG
Offset: 0x0C
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER						CMP0
Access	R/W	R/W						R/W
Reset	0	0						0

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow
This flag is cleared by writing a '1' to the flag.

This flag is set on the next CLK_RTC_CNT cycle after an overflow condition occurs, and an interrupt request will be generated if INTENCLR/SET.OVF is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Overflow interrupt flag.

Bit 14 – TAMPER Tamper event
This flag is set after a damper condition occurs, and an interrupt request will be generated if INTENCLR.TAMPER/INTENSET.TAMPER is '1'. Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the Tamper interrupt flag.

Bit 8 – CMP0 Compare 0
This flag is cleared by writing a '1' to the flag.

This flag is set on the next CLK_RTC_CNT cycle after a match with the compare condition, and an interrupt request will be generated if INTENCLR/SET.COMP0 is one.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Compare 0 interrupt flag.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n [n = 7..0]
This flag is cleared by writing a '1' to the flag.

This flag is set on the 0-to-1 transition of prescaler bit [n+2], and an interrupt request will be generated if INTENCLR/SET.PERn is one.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Periodic Interval n interrupt flag.

Downloaded from Arrow.com.

27.8.7 Debug Control

Name: DBGCTRL
Offset: 0x0E
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN Debug Run
This bit is not reset by a software reset.

This bit controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The RTC is halted when the CPU is halted by an external debugger.
1	The RTC continues normal operation when the CPU is halted by an external debugger.

Downloaded from Arrow.com.

27.8.8 Synchronization Busy in COUNT32 mode (CTRLA.MODE=0)

Name: SYNCBUSY
Offset: 0x10
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							GPn[1:0]	
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	COUNTSYNC							
Access	R							
Reset	0							

Bit	7	6	5	4	3	2	1	0
			COMP0		COUNT	FREQCORR	ENABLE	SWRST
Access			R		R	R	R	R
Reset			0		0	0	0	0

Bits 17:16 – GPn[1:0] General Purpose n Synchronization Busy Status

Value	Description
0	Write synchronization for GPn register is complete.
1	Write synchronization for GPn register is ongoing.

Bit 15 – COUNTSYNC Count Read Sync Enable Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.COUNTSYNC bit is complete.
1	Write synchronization for CTRLA.COUNTSYNC bit is ongoing.

Bit 5 – COMP0 Compare 0 Synchronization Busy Status

Value	Description
0	Write synchronization for COMP0 register is complete.
1	Write synchronization for COMP0 register is ongoing.

Bit 3 – COUNT Count Value Synchronization Busy Status

Value	Description
0	Read/write synchronization for COUNT register is complete.
1	Read/write synchronization for COUNT register is ongoing.

Downloaded from Arrow.com.

Bit 2 – FREQCORR Frequency Correction Synchronization Busy Status

Value	Description
0	Write synchronization for FREQCORR register is complete.
1	Write synchronization for FREQCORR register is ongoing.

Bit 1 – ENABLE Enable Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.ENABLE bit is complete.
1	Write synchronization for CTRLA.ENABLE bit is ongoing.

Bit 0 – SWRST Software Reset Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.SWRST bit is complete.
1	Write synchronization for CTRLA.SWRST bit is ongoing.

Downloaded from Arrow.com.

27.8.9 Frequency Correction

Name: FREQCORR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	SIGN	VALUE[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7 – SIGN Correction Sign

Value	Description
0	The correction value is positive, i.e., frequency will be decreased.
1	The correction value is negative, i.e., frequency will be increased.

Bits 6:0 – VALUE[6:0] Correction Value
These bits define the amount of correction applied to the RTC prescaler.

Value	Description
0	Correction is disabled and the RTC frequency is unchanged.
1 - 127	The RTC frequency is adjusted according to the value.

Downloaded from Arrow.com.

27.8.10 Counter Value in COUNT32 mode (CTRLA.MODE=0)

Name: COUNT
Offset: 0x18
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	COUNT[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	COUNT[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – COUNT[31:0] Counter Value
These bits define the value of the 32-bit RTC counter in mode 0.

Downloaded from Arrow.com.

27.8.11 Compare 0 Value in COUNT32 mode (CTRLA.MODE=0)

Name: COMP
Offset: 0x20
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
	COMP[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	COMP[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COMP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COMP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – COMP[31:0] Compare Value
The 32-bit value of COMP0 is continuously compared with the 32-bit COUNT value. When a match occurs, the Compare 0 interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMP0) is set on the next counter cycle, and the counter value is cleared if CTRLA.MATCHCLR is '1'.

Downloaded from Arrow.com.

27.8.12 General Purpose n

Name: GP
Offset: 0x40 + n*0x04 [n=0..1]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	GP[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	GP[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – GP[31:0] General Purpose
These bits are for user-defined general purpose use, see 27.6.8.3 General Purpose Registers.

Downloaded from Arrow.com.

27.8.13 Tamper Control

Name: TAMPCTRL
Offset: 0x60
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
					DEBNC3	DEBNC2	DEBNC1	DEBNC0
Access								
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
					TAMLVL3	TAMLVL2	TAMLVL1	TAMLVL0
Access								
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	IN3ACT[1:0]		IN2ACT[1:0]		IN1ACT[1:0]		IN0ACT[1:0]	
Access								
Reset	0	0	0	0	0	0	0	0

Bits 24, 25, 26, 27 – DEBNC Debounce Enable of Tamper Input INn

Value	Description
0	Debouncing is disabled for Tamper input INn
1	Debouncing is enabled for Tamper input INn

Bits 16, 17, 18, 19 – TAMLVL Tamper Level Select of Tamper Input INn

Value	Description
0	A falling edge condition will be detected on Tamper input INn.
1	A rising edge condition will be detected on Tamper input INn.

Bits 0:1, 2:3, 4:5, 6:7 – INACT Tamper Channel n Action
These bits determine the action taken by Tamper Channel n.

Value	Name	Description
0x0	OFF	Off (Disabled)
0x1	WAKE	Wake and set Tamper flag
0x2	CAPTURE	Capture timestamp and set Tamper flag
0x3	ACTL	Compare RTC signal routed between INn and OUT pins . When a mismatch occurs, capture timestamp and set Tamper flag

Downloaded from Arrow.com.

27.8.14 Timestamp

Name: TIMESTAMP
Offset: 0x64
Reset: 0x0
Property: Read-Only

Bit	31	30	29	28	27	26	25	24
	COUNT[31:24]							
Access	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	COUNT[23:16]							
Access	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – COUNT[31:0] Count Timestamp Value
The 32-bit value of COUNT is captured by the TIMESTAMP when a tamper condition occurs

Downloaded from Arrow.com.

27.8.15 Tamper ID

Name: TAMPID
Offset: 0x68
Reset: 0x00000000

Bit	31	30	29	28	27	26	25	24
	TAMPEVT							
Access	R/W							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					TAMPID3	TAMPID2	TAMPID1	TAMPID0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 31 – TAMPEVT Tamper Event Detected
Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the tamper detection bit.

Value	Description
0	A tamper input event has not been detected
1	A tamper input event has been detected

Bits 0, 1, 2, 3 – TAMPID Tamper on Channel n Detected
Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the tamper detection bit.

Value	Description
0	A tamper condition has not been detected on Channel n
1	A tamper condition has been detected on Channel n

Downloaded from Arrow.com.

27.8.16 Tamper Control B

Name: TAMPCTRLB
Offset: 0x6C
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					ALSI3	ALSI2	ALSI1	ALSI0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 0, 1, 2, 3 – ALSI Active Layer Internal Select n

Value	Description
0	Active layer Protection is monitoring the RTC signal using INn and OUTn tamper pins
1	Active layer Protection is monitoring the RTC signal on the TrustRAM shield

Downloaded from Arrow.com.

27.9 Register Summary - Mode 1 - 16-Bit Counter

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0					MODE[1:0]		ENABLE	SWRST
		15:8	COUNTSYNC	GPTRST			PRESCALER[3:0]			
0x02	CTRLB	7:0	DMAEN	RTCOUT	DEBASYNC	DEBMAJ				GP0EN
		15:8	SEPTO	ACTF[2:0]				DEBF[2:0]		
0x04	EVCTRL	7:0	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
		15:8	OVFEO	TAMPEREO					CMPEO1	CMPEO0
		23:16								TAMPEVEI
		31:24								PERDEO
0x08	INTENCLR	7:0	PER7	PER6	PER5	PER4	PER3	PER2	CMP1	CMP0
		15:8	OVF	TAMPER						
0x0A	INTENSET	7:0	PER7	PER6	PER5	PER4	PER3	PER2	CMP1	CMP0
		15:8	OVF	TAMPER						
0x0C	INTFLAG	7:0	PER7	PER6	PER5	PER4	PER3	PER2	CMP1	CMP0
		15:8	OVF	TAMPER						
0x0E	DBGCTRL	7:0								DBGRUN
0x0F	Reserved									
0x10	SYNCBUSY	7:0		COMP1	COMP0	PER	COUNT	FREQCORR	ENABLE	SWRST
		15:8	COUNTSYNC							
		23:16							GPn[1:0]	
		31:24								
0x14	FREQCORR	7:0	SIGN	VALUE[6:0]						
0x15 ... 0x17	Reserved									
0x18	COUNT	7:0	COUNT[7:0]							
		15:8	COUNT[15:8]							
0x1A ... 0x1B	Reserved									
0x1C	PER	7:0	PER[7:0]							
		15:8	PER[15:8]							
0x1E ... 0x1F	Reserved									
0x20	COMP0	7:0	COMP[7:0]							
		15:8	COMP[15:8]							
0x22	COMP1	7:0	COMP[7:0]							
		15:8	COMP[15:8]							
0x24 ... 0x3F	Reserved									
0x40	GP0	7:0	GP[7:0]							
		15:8	GP[15:8]							
		23:16	GP[23:16]							

Downloaded from Arrow.com.

Offset	Name	Bit Pos.								
		31:24	GP[31:24]							
0x44	GP1	7:0	GP[7:0]							
		15:8	GP[15:8]							
		23:16	GP[23:16]							
		31:24	GP[31:24]							
0x48 ... 0x5F	Reserved									
0x60	TAMPCTRL	7:0	IN3ACT[1:0]		IN2ACT[1:0]		IN1ACT[1:0]		IN0ACT[1:0]	
		15:8								
		23:16					TAMLVL3	TAMLVL2	TAMLVL1	TAMLVL0
		31:24					DEBNC3	DEBNC2	DEBNC1	DEBNC0
0x64	TIMESTAMP	7:0	COUNT[7:0]							
		15:8	COUNT[15:8]							
		23:16								
		31:24								
0x68	TAMPID	7:0					TAMPID3	TAMPID2	TAMPID1	TAMPID0
		15:8								
		23:16								
		31:24	TAMPEVT							
0x6C	TAMPCTRLB	7:0					ALSI3	ALSI2	ALSI1	ALSI0
		15:8								
		23:16								
		31:24								

27.10 Register Description - Mode 1 - 16-Bit Counter

This Register Description section is valid if the RTC is in COUNT16 mode (CTRLA.MODE=1).

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Downloaded from Arrow.com.

Refer to *Peripherals Security Attribution* for more information.

Downloaded from Arrow.com.

27.10.1 Control A in COUNT16 mode (CTRLA.MODE=1)

Name: CTRLA
Offset: 0x00
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	COUNTSYNC	GPTRST			PRESCALER[3:0]			
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit	7	6	5	4	3	2	1	0
					MODE[1:0]		ENABLE	SWRST
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 15 – COUNTSYNC COUNT Read Synchronization Enable
The COUNT register requires synchronization when reading. Disabling the synchronization will prevent reading valid values from the COUNT register.
This bit is not enable-protected.

Value	Description
0	COUNT read synchronization is disabled
1	COUNT read synchronization is enabled

Bit 14 – GPTRST GP Registers Reset On Tamper Enable
Only GP registers enabled by the CTRLB.GPnEN bits are affected. This bit can be written only when the peripheral is disabled.
This bit is not synchronized.

Value	Description
0	GPn registers will not reset when a tamper condition occurs.
1	GPn registers will reset when a tamper condition occurs.

Bits 11:8 – PRESCALER[3:0] Prescaler
These bits define the prescaling factor for the RTC clock source (GCLK_RTC) to generate the counter clock (CLK_RTC_CNT). Periodic events and interrupts are not available when the prescaler is off. These bits are not synchronized.

Value	Name	Description
0x0	OFF	CLK_RTC_CNT = GCLK_RTC/1
0x1	DIV1	CLK_RTC_CNT = GCLK_RTC/1
0x2	DIV2	CLK_RTC_CNT = GCLK_RTC/2
0x3	DIV4	CLK_RTC_CNT = GCLK_RTC/4
0x4	DIV8	CLK_RTC_CNT = GCLK_RTC/8
0x5	DIV16	CLK_RTC_CNT = GCLK_RTC/16
0x6	DIV32	CLK_RTC_CNT = GCLK_RTC/32
0x7	DIV64	CLK_RTC_CNT = GCLK_RTC/64

Downloaded from Arrow.com.

Value	Name	Description
0x8	DIV128	CLK_RTC_CNT = GCLK_RTC/128
0x9	DIV256	CLK_RTC_CNT = GCLK_RTC/256
0xA	DIV512	CLK_RTC_CNT = GCLK_RTC/512
0xB	DIV1024	CLK_RTC_CNT = GCLK_RTC/1024
0xC-0xF	-	Reserved

Bits 3:2 – MODE[1:0] Operating Mode
This field defines the operating mode of the RTC. This bit is not synchronized.

Value	Name	Description
0x0	COUNT32	Mode 0: 32-bit counter
0x1	COUNT16	Mode 1: 16-bit counter
0x2	CLOCK	Mode 2: Clock/calendar
0x3	-	Reserved

Bit 1 – ENABLE Enable
Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

Value	Description
0	The peripheral is disabled
1	The peripheral is enabled

Bit 0 – SWRST Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the RTC (except DBGCTRL) to their initial state, and the RTC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST will be cleared when the reset is complete.

Value	Description
0	There is not reset operation ongoing
1	The reset operation is ongoing

Downloaded from Arrow.com.

27.10.2 Control B in COUNT16 mode (CTRLA.MODE=1)

Name: CTRLB
Offset: 0x02
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
	SEPTO	ACTF[2:0]				DEBF[2:0]		
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
	DMAEN	RTCOUT	DEBASYNC	DEBMAJ				GP0EN
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bit 15 – SEPTO Separate Tamper Outputs

Value	Description
0	IN[n] is compared to OUT[0] (backward-compatible).
1	IN[n] is compared to OUT[n].

Bits 14:12 – ACTF[2:0] Active Layer Frequency
These bits define the prescaling factor for the RTC clock output (OUT) used during active layer protection in terms of the CLK_RTC.

Value	Name	Description
0x0	DIV2	CLK_RTC_OUT = CLK_RTC / 2
0x1	DIV4	CLK_RTC_OUT = CLK_RTC / 4
0x2	DIV8	CLK_RTC_OUT = CLK_RTC / 8
0x3	DIV16	CLK_RTC_OUT = CLK_RTC / 16
0x4	DIV32	CLK_RTC_OUT = CLK_RTC / 32
0x5	DIV64	CLK_RTC_OUT = CLK_RTC / 64
0x6	DIV128	CLK_RTC_OUT = CLK_RTC / 128
0x7	DIV256	CLK_RTC_OUT = CLK_RTC / 256

Bits 10:8 – DEBF[2:0] Debounce Frequency
These bits define the prescaling factor for the input debouncers in terms of the CLK_RTC.

Value	Name	Description
0x0	DIV2	CLK_RTC_DEB = CLK_RTC / 2
0x1	DIV4	CLK_RTC_DEB = CLK_RTC / 4
0x2	DIV8	CLK_RTC_DEB = CLK_RTC / 8
0x3	DIV16	CLK_RTC_DEB = CLK_RTC / 16
0x4	DIV32	CLK_RTC_DEB = CLK_RTC / 32
0x5	DIV64	CLK_RTC_DEB = CLK_RTC / 64
0x6	DIV128	CLK_RTC_DEB = CLK_RTC / 128
0x7	DIV256	CLK_RTC_DEB = CLK_RTC / 256

Downloaded from Arrow.com.

Bit 7 – DMAEN DMA Enable
The RTC can trigger a DMA request when the timestamp is ready in the TIMESTAMP register.

Value	Description
0	Tamper DMA request is disabled. Reading TIMESTAMP has no effect on INTFLAG.TAMPER.
1	Tamper DMA request is enabled. Reading TIMESTAMP will clear INTFLAG.TAMPER.

Bit 6 – RTCOUT RTC Output Enable

Value	Description
0	The RTC active layer output is disabled.
1	The RTC active layer output is enabled.

Bit 5 – DEBASYNC Debouncer Asynchronous Enable

Value	Description
0	The tamper input debouncers operate synchronously.
1	The tamper input debouncers operate asynchronously.

Bit 4 – DEBMAJ Debouncer Majority Enable

Value	Description
0	The tamper input debouncers match three equal values.
1	The tamper input debouncers match majority two of three values.

Bit 0 – GP0EN General Purpose 0 Enable

Value	Description
0	COMP0 compare function enabled. GP0/GP1 disabled.
1	COMP0 compare function disabled. GP0/GP1 enabled.

Downloaded from Arrow.com.

27.10.3 Event Control in COUNT16 mode (CTRLA.MODE=1)

Name: EVCTRL
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
								PERDEO
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
								TAMPEVEI
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
	OVFEO	TAMPEREO					CMPEO1	CMPEO0
Access	R/W	R/W					R/W	R/W
Reset	0	0					0	0

Bit	7	6	5	4	3	2	1	0
	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 24 – PERDEO Periodic Interval Daily Event Output Enable

Value	Description
0	Periodic Daily event is disabled and will not be generated.
1	Periodic Daily event is enabled and will be generated. The event occurs at the overflow of the RTC counter (i.e., when the RTC counter goes from 0xFFFF to 0x0000).

Bit 16 – TAMPEVEI Tamper Event Input Enable

Value	Description
0	Tamper event input is disabled, and incoming events will be ignored
1	Tamper event input is enabled, and incoming events will capture the COUNT value

Bit 15 – OVFEO Overflow Event Output Enable

Value	Description
0	Overflow event is disabled and will not be generated.
1	Overflow event is enabled and will be generated for every overflow.

Bit 14 – TAMPEREO Tamper Event Output Enable

Downloaded from Arrow.com.

Value	Description
0	Tamper event output is disabled, and will not be generated.
1	Tamper event output is enabled, and will be generated for every tamper input.

Bits 8, 9 – CMPEOn Compare n Event Output Enable [n = 1..0]

Value	Description
0	Compare n event is disabled and will not be generated.
1	Compare n event is enabled and will be generated for every compare match.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PEREOn Periodic Interval n Event Output Enable [n = 7..0]

Value	Description
0	Periodic Interval n event is disabled and will not be generated.
1	Periodic Interval n event is enabled and will be generated.

Downloaded from Arrow.com.

27.10.4 Interrupt Enable Clear in COUNT16 mode (CTRLA.MODE=1)

Name: INTENCLR
Offset: 0x08
Reset: 0x0000
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER						
Access	R/W	R/W						
Reset	0	0						

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	CMP1	CMP0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bit 14 – TAMPER Tamper Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Tamper Interrupt Enable bit, which disables the Tamper interrupt.

Value	Description
0	The Tamper interrupt is disabled.
1	The Tamper interrupt is enabled.

Bits 0, 1 – CMPn Compare n Interrupt Enable [n = 1..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Compare n Interrupt Enable bit, which disables the Compare n interrupt.

Value	Description
0	The Compare n interrupt is disabled.
1	The Compare n interrupt is enabled.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Periodic Interval n Interrupt Enable bit, which disables the Periodic Interval n interrupt.

Value	Description
0	Periodic Interval n interrupt is disabled.
1	Periodic Interval n interrupt is enabled.

Downloaded from Arrow.com.

27.10.5 Interrupt Enable Set in COUNT16 mode (CTRLA.MODE=1)

Name: INTENSET
Offset: 0x0A
Reset: 0x0000
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER						
Access	R/W	R/W						
Reset	0	0						

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	CMP1	CMP0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bit 14 – TAMPER Tamper Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Tamper Interrupt Enable bit, which enables the Tamper interrupt.

Value	Description
0	The Tamper interrupt is disabled.
1	The Tamper interrupt is enabled.

Bits 0, 1 – CMPn Compare n Interrupt Enable [n = 1..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Compare n Interrupt Enable bit, which and enables the Compare n interrupt.

Value	Description
0	The Compare n interrupt is disabled.
1	The Compare n interrupt is enabled.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Periodic Interval n Interrupt Enable bit, which enables the Periodic Interval n interrupt.

Value	Description
0	Periodic Interval n interrupt is disabled.
1	Periodic Interval n interrupt is enabled.

Downloaded from Arrow.com.

27.10.6 Interrupt Flag Status and Clear in COUNT16 mode (CTRLA.MODE=1)

Name: INTFLAG
Offset: 0x0C
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER						
Access	R/W	R/W						
Reset	0	0						

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	CMP1	CMP0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow
This flag is cleared by writing a '1' to the flag.

This flag is set on the next CLK_RTC_CNT cycle after an overflow condition occurs, and an interrupt request will be generated if INTENCLR/SET.OVF is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Overflow interrupt flag.

Bit 14 – TAMPER Tamper
This flag is set after a tamper condition occurs, and an interrupt request will be generated if INTENCLR.TAMPER/ INTENSET.TAMPER is one.

Writing a '0' to this bit has no effect.

Writing a one to this bit clears the Tamper interrupt flag.

Bits 0, 1 – CMPn Compare n [n = 1..0]
This flag is cleared by writing a '1' to the flag.

This flag is set on the next CLK_RTC_CNT cycle after a match with the compare condition, and an interrupt request will be generated if INTENCLR/SET.COMPn is one.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Compare n interrupt flag.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n [n = 7..0]
This flag is cleared by writing a '1' to the flag.

This flag is set on the 0-to-1 transition of prescaler bit [n+2], and an interrupt request will be generated if INTENCLR/SET.PERx is one.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Periodic Interval n interrupt flag.

Downloaded from Arrow.com.

27.10.7 Debug Control

Name: DBGCTRL
Offset: 0x0E
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN Debug Run
This bit is not reset by a software reset.

This bit controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The RTC is halted when the CPU is halted by an external debugger.
1	The RTC continues normal operation when the CPU is halted by an external debugger.

Downloaded from Arrow.com.

27.10.8 Synchronization Busy in COUNT16 mode (CTRLA.MODE=1)

Name: SYNCBUSY
Offset: 0x10
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							GPn[1:0]	
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	COUNTSYNC							
Access	R							
Reset	0							

Bit	7	6	5	4	3	2	1	0
		COMP1	COMP0	PER	COUNT	FREQCORR	ENABLE	SWRST
Access		R/W	R/W	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bits 17:16 – GPn[1:0] General Purpose n Synchronization Busy Status

Value	Description
0	Write synchronization for GPn register is complete.
1	Write synchronization for GPn register is ongoing.

Bit 15 – COUNTSYNC Count Read Sync Enable Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.COUNTSYNC bit is complete.
1	Write synchronization for CTRLA.COUNTSYNC bit is ongoing.

Bits 5, 6 – COMPn Compare n Synchronization Busy Status [n = 1..0]

Value	Description
0	Write synchronization for COMPn register is complete.
1	Write synchronization for COMPn register is ongoing.

Bit 4 – PER Period Synchronization Busy Status

Value	Description
0	Write synchronization for PER register is complete.
1	Write synchronization for PER register is ongoing.

Downloaded from Arrow.com.

Bit 3 – COUNT Count Value Synchronization Busy Status

Value	Description
0	Read/write synchronization for COUNT register is complete.
1	Read/write synchronization for COUNT register is ongoing.

Bit 2 – FREQCORR Frequency Correction Synchronization Busy Status

Value	Description
0	Write synchronization for FREQCORR register is complete.
1	Write synchronization for FREQCORR register is ongoing.

Bit 1 – ENABLE Enable Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.ENABLE bit is complete.
1	Write synchronization for CTRLA.ENABLE bit is ongoing.

Bit 0 – SWRST Software Reset Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.SWRST bit is complete.
1	Write synchronization for CTRLA.SWRST bit is ongoing.

Downloaded from Arrow.com.

27.10.9 Frequency Correction

Name: FREQCORR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	SIGN	VALUE[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7 – SIGN Correction Sign

Value	Description
0	The correction value is positive, i.e., frequency will be decreased.
1	The correction value is negative, i.e., frequency will be increased.

Bits 6:0 – VALUE[6:0] Correction Value
These bits define the amount of correction applied to the RTC prescaler.

Value	Description
0	Correction is disabled and the RTC frequency is unchanged.
1 - 127	The RTC frequency is adjusted according to the value.

Downloaded from Arrow.com.

27.10.10 Counter Value in COUNT16 mode (CTRLA.MODE=1)

Name: COUNT
Offset: 0x18
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – COUNT[15:0] Counter Value
These bits define the value of the 16-bit RTC counter in COUNT16 mode (CTRLA.MODE=1).

Downloaded from Arrow.com.

27.10.11 Counter Period in COUNT16 mode (CTRLA.MODE=1)

Name: PER
Offset: 0x1C
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	PER[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PER[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – PER[15:0] Counter Period
These bits define the value of the 16-bit RTC period in COUNT16 mode (CTRLA.MODE=1).

Downloaded from Arrow.com.

27.10.12 Compare n Value in COUNT16 mode (CTRLA.MODE=1)

Name: COMP
Offset: 0x20 + n*0x02 [n=0..1]
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	COMP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COMP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – COMP[15:0] Compare Value
The 16-bit value of COMPn is continuously compared with the 16-bit COUNT value. When a match occurs, the Compare n interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMPn) is set on the next counter cycle.

Downloaded from Arrow.com.

27.10.13 General Purpose n

Name: GP
Offset: 0x40 + n*0x04 [n=0..1]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	GP[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	GP[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – GP[31:0] General Purpose
These bits are for user-defined general purpose use, see 27.6.8.3 General Purpose Registers.

Downloaded from Arrow.com.

27.10.14 Tamper Control

Name: TAMPCTRL
Offset: 0x60
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
					DEBNC3	DEBNC2	DEBNC1	DEBNC0
Access								
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
					TAMLVL3	TAMLVL2	TAMLVL1	TAMLVL0
Access								
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	IN3ACT[1:0]		IN2ACT[1:0]		IN1ACT[1:0]		IN0ACT[1:0]	
Access								
Reset	0	0	0	0	0	0	0	0

Bits 24, 25, 26, 27 – DEBNC Debounce Enable of Tamper Input INn

Value	Description
0	Debouncing is disabled for Tamper input INn
1	Debouncing is enabled for Tamper input INn

Bits 16, 17, 18, 19 – TAMLVL Tamper Level Select of Tamper Input INn

Value	Description
0	A falling edge condition will be detected on Tamper input INn.
1	A rising edge condition will be detected on Tamper input INn.

Bits 0:1, 2:3, 4:5, 6:7 – INACT Tamper Channel n Action
These bits determine the action taken by Tamper Channel n.

Value	Name	Description
0x0	OFF	Off (Disabled)
0x1	WAKE	Wake and set Tamper flag
0x2	CAPTURE	Capture timestamp and set Tamper flag
0x3	ACTL	Compare RTC signal routed between INn and OUT pins . When a mismatch occurs, capture timestamp and set Tamper flag

Downloaded from Arrow.com.

27.10.15 Timestamp

Name: TIMESTAMP
Offset: 0x64
Reset: 0x0000
Property: Read-Only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – COUNT[15:0] Count Timestamp Value
The 16-bit value of COUNT is captured by the TIMESTAMP when a tamper condition occurs.

Downloaded from Arrow.com.

27.10.16 Tamper ID

Name: TAMPID
Offset: 0x68
Reset: 0x00000000

Bit	31	30	29	28	27	26	25	24
	TAMPEVT							
Access	R/W							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					TAMPID3	TAMPID2	TAMPID1	TAMPID0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 31 – TAMPEVT Tamper Event Detected
Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the tamper detection bit.

Value	Description
0	A tamper input event has not been detected
1	A tamper input event has been detected

Bits 0, 1, 2, 3 – TAMPID Tamper on Channel n Detected
Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the tamper detection bit.

Value	Description
0	A tamper condition has not been detected on Channel n
1	A tamper condition has been detected on Channel n

Downloaded from Arrow.com.

27.10.17 Tamper Control B

Name: TAMPCTRLB
Offset: 0x6C
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					ALSI3	ALSI2	ALSI1	ALSI0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 0, 1, 2, 3 – ALSI Active Layer Internal Select n

Value	Description
0	Active layer Protection is monitoring the RTC signal using INn and OUTn tamper pins
1	Active layer Protection is monitoring the RTC signal on the TrustRAM shield

Downloaded from Arrow.com.

27.11 Register Summary - Mode 2 - Clock/Calendar

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	MATCHCLR	CLKREP			MODE[1:0]		ENABLE	SWRST
		15:8	CLOCKSYNC	GPTRST			PRESCALER[3:0]			
0x02	CTRLB	7:0	DMAEN	RTCOUT	DEBASYNC	DEBMAJ				GP0EN
		15:8	SEPTO	ACTF[2:0]				DEBF[2:0]		
0x04	EVCTRL	7:0	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
		15:8	OVFEO	TAMPEREO						ALARMEO
		23:16								TAMPEVEI
		31:24								PERDEO
0x08	INTENCLR	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
		15:8	OVF	TAMPER						ALARM0
0x0A	INTENSET	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
		15:8	OVF	TAMPER						ALARM0
0x0C	INTFLAG	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
		15:8	OVF	TAMPER						ALARM0
0x0E	DBGCTRL	7:0								DBGRUN
0x0F	Reserved									
0x10	SYNCBUSY	7:0			ALARM0		CLOCK	FREQCORR	ENABLE	SWRST
		15:8	CLOCKSYNC				MASK0			
		23:16							GPn[1:0]	
		31:24								
0x14	FREQCORR	7:0	SIGN	VALUE[6:0]						
0x15 ... 0x17	Reserved									
0x18	CLOCK	7:0	MINUTE[1:0]		SECOND[5:0]					
		15:8	HOUR[3:0]				MINUTE[5:2]			
		23:16	MONTH[1:0]		DAY[4:0]					HOUR[4:4]
		31:24	YEAR[5:0]						MONTH[3:2]	
0x1C ... 0x1F	Reserved									
0x20	ALARM	7:0	MINUTE[1:0]		SECOND[5:0]					
		15:8	HOUR[3:0]				MINUTE[5:2]			
		23:16	MONTH[1:0]		DAY[4:0]					HOUR[4:4]
		31:24	YEAR[5:0]						MONTH[3:2]	
0x24	MASK	7:0						SEL[2:0]		
0x25 ... 0x3F	Reserved									
0x40	GP0	7:0	GP[7:0]							
		15:8	GP[15:8]							
		23:16	GP[23:16]							
		31:24	GP[31:24]							

Downloaded from Arrow.com.

Offset	Name	Bit Pos.								
0x44	GP1	7:0	GP[7:0]							
		15:8	GP[15:8]							
		23:16	GP[23:16]							
		31:24	GP[31:24]							
0x48 ... 0x5F	Reserved									
0x60	TAMPCTRL	7:0	IN3ACT[1:0]		IN2ACT[1:0]		IN1ACT[1:0]		IN0ACT[1:0]	
		15:8								
		23:16					TAMLVL3	TAMLVL2	TAMLVL1	TAMLVL0
		31:24					DEBNC3	DEBNC2	DEBNC1	DEBNC0
0x64	TIMESTAMP	7:0	MINUTE[1:0]		SECOND[5:0]					
		15:8	HOUR[3:0]				MINUTE[5:2]			
		23:16	MONTH[1:0]		DAY[4:0]					HOUR[4:4]
		31:24	YEAR[5:0]						MONTH[3:2]	
0x68	TAMPID	7:0					TAMPID3	TAMPID2	TAMPID1	TAMPID0
		15:8								
		23:16								
		31:24	TAMPEVT							
0x6C	TAMPCTRLB	7:0					ALSI3	ALSI2	ALSI1	ALSI0
		15:8								
		23:16								
		31:24								

27.12 Register Description - Mode 2 - Clock/Calendar

This Register Description section is valid if the RTC is in Clock/Calendar mode (CTRLA.MODE=2).

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Downloaded from Arrow.com.

Refer to *Peripherals Security Attribution* for more information.

Downloaded from Arrow.com.

27.12.1 Control A in Clock/Calendar mode (CTRLA.MODE=2)

Name: CTRLA
Offset: 0x00
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	CLOCKSYNC	GPTRST			PRESCALER[3:0]			
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MATCHCLR	CLKREP			MODE[1:0]		ENABLE	SWRST
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit 15 – CLOCKSYNC CLOCK Read Synchronization Enable
The CLOCK register requires synchronization when reading. Disabling the synchronization will prevent reading valid values from the CLOCK register.
This bit is not enable-protected.

Value	Description
0	CLOCK read synchronization is disabled
1	CLOCK read synchronization is enabled

Bit 14 – GPTRST GP Registers Reset On Tamper Enable
Only GP registers enabled by the CTRLB.GPnEN bits are affected. This bit can be written only when the peripheral is disabled.
This bit is not synchronized.

Bits 11:8 – PRESCALER[3:0] Prescaler
These bits define the prescaling factor for the RTC clock source (GCLK_RTC) to generate the counter clock (CLK_RTC_CNT). Periodic events and interrupts are not available when the prescaler is off. These bits are not synchronized.

Value	Name	Description
0x0	OFF	CLK_RTC_CNT = GCLK_RTC/1
0x1	DIV1	CLK_RTC_CNT = GCLK_RTC/1
0x2	DIV2	CLK_RTC_CNT = GCLK_RTC/2
0x3	DIV4	CLK_RTC_CNT = GCLK_RTC/4
0x4	DIV8	CLK_RTC_CNT = GCLK_RTC/8
0x5	DIV16	CLK_RTC_CNT = GCLK_RTC/16
0x6	DIV32	CLK_RTC_CNT = GCLK_RTC/32
0x7	DIV64	CLK_RTC_CNT = GCLK_RTC/64
0x8	DIV128	CLK_RTC_CNT = GCLK_RTC/128
0x9	DIV256	CLK_RTC_CNT = GCLK_RTC/256
0xA	DIV512	CLK_RTC_CNT = GCLK_RTC/512

Downloaded from Arrow.com.

Value	Name	Description
0xB	DIV1024	CLK_RTC_CNT = GCLK_RTC/1024
0xC-0xF	-	Reserved

Bit 7 – MATCHCLR Clear on Match
This bit is valid only in Mode 0 (COUNT32) and Mode 2 (CLOCK). This bit can be written only when the peripheral is disabled. This bit is not synchronized.

Value	Description
0	The counter is not cleared on a Compare/Alarm 0 match
1	The counter is cleared on a Compare/Alarm 0 match

Bit 6 – CLKREP Clock Representation
This bit is valid only in Mode 2 and determines how the hours are represented in the Clock Value (CLOCK) register. This bit can be written only when the peripheral is disabled. This bit is not synchronized.

Value	Description
0	24 Hour
1	12 Hour (AM/PM)

Bits 3:2 – MODE[1:0] Operating Mode
This field defines the operating mode of the RTC. This bit is not synchronized.

Value	Name	Description
0x0	COUNT32	Mode 0: 32-bit counter
0x1	COUNT16	Mode 1: 16-bit counter
0x2	CLOCK	Mode 2: Clock/calendar
0x3	-	Reserved

Bit 1 – ENABLE Enable
Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

Value	Description
0	The peripheral is disabled
1	The peripheral is enabled

Bit 0 – SWRST Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the RTC, except DBGCTRL, to their initial state, and the RTC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST will be cleared when the reset is complete.

Downloaded from Arrow.com.

Value	Description
0	There is not reset operation ongoing
1	The reset operation is ongoing

Downloaded from Arrow.com.

27.12.2 Control B in Clock/Calendar mode (CTRLA.MODE=2)

Name: CTRLB
Offset: 0x2
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
	SEPTO	ACTF[2:0]				DEBF[2:0]		
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
	DMAEN	RTCOUT	DEBASYNC	DEBMAJ				GP0EN
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bit 15 – SEPTO Separate Tamper Outputs

Value	Description
0	IN[n] is compared tp OUT[0] (backward-compatible).
1	IN[n] is compared tp OUT[n].

Bits 14:12 – ACTF[2:0] Active Layer Frequency
These bits define the prescaling factor for the RTC clock output (OUT) used during active layer protection in terms of the CLK_RTC.

Value	Name	Description
0x0	DIV2	CLK_RTC_OUT = CLK_RTC / 2
0x1	DIV4	CLK_RTC_OUT = CLK_RTC / 4
0x2	DIV8	CLK_RTC_OUT = CLK_RTC / 8
0x3	DIV16	CLK_RTC_OUT = CLK_RTC / 16
0x4	DIV32	CLK_RTC_OUT = CLK_RTC / 32
0x5	DIV64	CLK_RTC_OUT = CLK_RTC / 64
0x6	DIV128	CLK_RTC_OUT = CLK_RTC / 128
0x7	DIV256	CLK_RTC_OUT = CLK_RTC / 256

Bits 10:8 – DEBF[2:0] Debounce Frequency
These bits define the prescaling factor for the input debouncers in terms of the CLK_RTC.

Value	Name	Description
0x0	DIV2	CLK_RTC_DEB = CLK_RTC / 2
0x1	DIV4	CLK_RTC_DEB = CLK_RTC / 4
0x2	DIV8	CLK_RTC_DEB = CLK_RTC / 8
0x3	DIV16	CLK_RTC_DEB = CLK_RTC / 16
0x4	DIV32	CLK_RTC_DEB = CLK_RTC / 32
0x5	DIV64	CLK_RTC_DEB = CLK_RTC / 64
0x6	DIV128	CLK_RTC_DEB = CLK_RTC / 128
0x7	DIV256	CLK_RTC_DEB = CLK_RTC / 256

Downloaded from Arrow.com.

Bit 7 – DMAEN DMA Enable
The RTC can trigger a DMA request when the timestamp is ready in the TIMESTAMP register.

Value	Description
0	Tamper DMA request is disabled. Reading TIMESTAMP has no effect on INTFLAG.TAMPER.
1	Tamper DMA request is enabled. Reading TIMESTAMP will clear INTFLAG.TAMPER.

Bit 6 – RTCOUT RTC Out Enable

Value	Description
0	The RTC active layer output is disabled.
1	The RTC active layer output is enabled.

Bit 5 – DEBASYNC Debouncer Asynchronous Enable

Value	Description
0	The tamper input debouncers operate synchronously.
1	The tamper input debouncers operate asynchronously.

Bit 4 – DEBMAJ Debouncer Majority Enable

Value	Description
0	The tamper input debouncers match three equal values.
1	The tamper input debouncers match majority two of three values.

Bit 0 – GP0EN General Purpose 0 Enable

Value	Description
0	COMP0 compare function enabled. GP0 disabled.
1	COMP0 compare function disabled. GP0 enabled.

Downloaded from Arrow.com.

27.12.3 Event Control in Clock/Calendar mode (CTRLA.MODE=2)

Name: EVCTRL
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
								PERDEO
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
								TAMPEVEI
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
	OVFEO	TAMPEREO						ALARMEO
Access	R/W	R/W						R/W
Reset	0	0						0

Bit	7	6	5	4	3	2	1	0
	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 24 – PERDEO Periodic Interval Daily Event Output Enable

Value	Description
0	Periodic Daily event is disabled and will not be generated.
1	Periodic Daily event is enabled and will be generated. The event occurs at the last second of each day depending on the CTRLA.CLKREP bit: • If CLKREP = 0, the event will occur at 23:59:59 • If CLKREP = 1, the event will occur at 11:59:59, PM = 1

Bit 16 – TAMPEVEI Tamper Event Input Enable

Value	Description
0	Tamper event input is disabled, and incoming events will be ignored.
1	Tamper event input is enabled, and all incoming events will capture the CLOCK value.

Bit 15 – OVFEO Overflow Event Output Enable

Value	Description
0	Overflow event is disabled and will not be generated.
1	Overflow event is enabled and will be generated for every overflow.

Downloaded from Arrow.com.

Bit 14 – TAMPEREO Tamper Event Output Enable

Value	Description
0	Tamper event output is disabled, and will not be generated
1	Tamper event output is enabled, and will be generated for every tamper input.

Bit 8 – ALARMEO Alarm 0 Event Output Enable

Value	Description
0	Alarm 0 event is disabled and will not be generated.
1	Alarm 0 event is enabled and will be generated for every compare match.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PEREOn Periodic Interval n Event Output Enable [n = 7..0]

Value	Description
0	Periodic Interval n event is disabled and will not be generated.
1	Periodic Interval n event is enabled and will be generated.

Downloaded from Arrow.com.

27.12.4 Interrupt Enable Clear in Clock/Calendar mode (CTRLA.MODE=2)

Name: INTENCLR
Offset: 0x08
Reset: 0x0000
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER						ALARM0
Access	R/W	R/W						R/W
Reset	0	0						0

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bit 14 – TAMPER Tamper Interrupt Enable

Bit 8 – ALARM0 Alarm 0 Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Alarm 0 Interrupt Enable bit, which disables the Alarm interrupt.

Value	Description
0	The Alarm 0 interrupt is disabled.
1	The Alarm 0 interrupt is enabled.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Periodic Interval n Interrupt Enable bit, which disables the Periodic Interval n interrupt.

Value	Description
0	Periodic Interval n interrupt is disabled.
1	Periodic Interval n interrupt is enabled.

Downloaded from Arrow.com.

27.12.5 Interrupt Enable Set in Clock/Calendar mode (CTRLA.MODE=2)

Name: INTENSET
Offset: 0x0A
Reset: 0x0000
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER						ALARM0
Access	R/W	R/W						R/W
Reset	0	0						0

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bit 14 – TAMPER Tamper Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Tamper Interrupt Enable bit, which enables the Tamper interrupt.

Value	Description
0	The Tamper interrupt it disabled.
1	The Tamper interrupt is enabled.

Bit 8 – ALARM0 Alarm 0 Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Alarm 0 Interrupt Enable bit, which enables the Alarm 0 interrupt.

Value	Description
0	The Alarm 0 interrupt is disabled.
1	The Alarm 0 interrupt is enabled.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Periodic Interval n Interrupt Enable bit, which enables the Periodic Interval n interrupt.

Value	Description
0	Periodic Interval n interrupt is disabled.
1	Periodic Interval n interrupt is enabled.

Downloaded from Arrow.com.

27.12.6 Interrupt Flag Status and Clear in Clock/Calendar mode (CTRLA.MODE=2)

Name: INTFLAG
Offset: 0x0C
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER						ALARM0
Access	R/W	R/W						R/W
Reset	0	0						0

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow
This flag is cleared by writing a '1' to the flag.

This flag is set on the next CLK_RTC_CNT cycle after an overflow condition occurs, and an interrupt request will be generated if INTENCLR/SET.OVF is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Overflow interrupt flag.

Bit 14 – TAMPER Tamper
This flag is set after a tamper condition occurs, and an interrupt request will be generated if INTENCLR.TAMPER/INTENSET.TAMPER is '1'. Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the Tamper interrupt flag.

Bit 8 – ALARM0 Alarm 0
This flag is cleared by writing a '1' to the flag.

This flag is set on the next CLK_RTC_CNT cycle after a match with the compare condition, and an interrupt request will be generated if INTENCLR/SET.ALARM0 is one.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Alarm 0 interrupt flag.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n [n = 7..0]
This flag is cleared by writing a '1' to the flag.

This flag is set on the 0-to-1 transition of prescaler bit [n+2], and an interrupt request will be generated if INTENCLR/SET.PERx is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Periodic Interval n interrupt flag.

Downloaded from Arrow.com.

27.12.7 Debug Control

Name: DBGCTRL
Offset: 0x0E
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN Debug Run
This bit is not reset by a software reset.

This bit controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The RTC is halted when the CPU is halted by an external debugger.
1	The RTC continues normal operation when the CPU is halted by an external debugger.

Downloaded from Arrow.com.

27.12.8 Synchronization Busy in Clock/Calendar mode (CTRLA.MODE=2)

Name: SYNCBUSY
Offset: 0x10
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							GPn[1:0]	
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	CLOCKSYNC				MASK0			
Access	R				R			
Reset	0				0			

Bit	7	6	5	4	3	2	1	0
			ALARM0		CLOCK	FREQCORR	ENABLE	SWRST
Access			R		R	R	R	R
Reset			0		0	0	0	0

Bits 17:16 – GPn[1:0] General Purpose n Synchronization Busy Status

Value	Description
0	Write synchronization for GPn register is complete.
1	Write synchronization for GPn register is ongoing.

Bit 15 – CLOCKSYNC Clock Read Sync Enable Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.CLOCKSYNC bit is complete.
1	Write synchronization for CTRLA.CLOCKSYNC bit is ongoing.

Bit 11 – MASK0 Mask 0 Synchronization Busy Status

Value	Description
0	Write synchronization for MASK0 register is complete.
1	Write synchronization for MASK0 register is ongoing.

Bit 5 – ALARM0 Alarm 0 Synchronization Busy Status

Value	Description
0	Write synchronization for ALARM0 register is complete.
1	Write synchronization for ALARM0 register is ongoing.

Downloaded from Arrow.com.

Bit 3 – CLOCK Clock Register Synchronization Busy Status

Value	Description
0	Read/write synchronization for CLOCK register is complete.
1	Read/write synchronization for CLOCK register is ongoing.

Bit 2 – FREQCORR Frequency Correction Synchronization Busy Status

Value	Description
0	Write synchronization for FREQCORR register is complete.
1	Write synchronization for FREQCORR register is ongoing.

Bit 1 – ENABLE Enable Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.ENABLE bit is complete.
1	Write synchronization for CTRLA.ENABLE bit is ongoing.

Bit 0 – SWRST Software Reset Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.SWRST bit is complete.
1	Write synchronization for CTRLA.SWRST bit is ongoing.

Downloaded from Arrow.com.

27.12.9 Frequency Correction

Name: FREQCORR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	SIGN	VALUE[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7 – SIGN Correction Sign

Value	Description
0	The correction value is positive, i.e., frequency will be decreased.
1	The correction value is negative, i.e., frequency will be increased.

Bits 6:0 – VALUE[6:0] Correction Value
These bits define the amount of correction applied to the RTC prescaler.

Value	Description
0	Correction is disabled and the RTC frequency is unchanged.
1 - 127	The RTC frequency is adjusted according to the value.

Downloaded from Arrow.com.

27.12.10 Clock Value in Clock/Calendar mode (CTRLA.MODE=2)

Name: CLOCK
Offset: 0x18
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	YEAR[5:0]						MONTH[3:2]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MONTH[1:0]		DAY[4:0]					HOUR[4:4]
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	HOUR[3:0]				MINUTE[5:2]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MINUTE[1:0]		SECOND[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:26 – YEAR[5:0] Year
The year offset with respect to the reference year (defined in software).

The year is considered a leap year if YEAR[1:0] is zero.

Bits 25:22 – MONTH[3:0] Month
1 – January

2 – February

...

12 – December

Bits 21:17 – DAY[4:0] Day
Day starts at 1 and ends at 28, 29, 30, or 31, depending on the month and year.

Bits 16:12 – HOUR[4:0] Hour
When CTRLA.CLKREP=0, the Hour bit group is in 24-hour format, with values 0-23. When CTRLA.CLKREP=1, HOUR[3:0] has values 1-12, and HOUR[4] represents AM (0) or PM (1).

Bits 11:6 – MINUTE[5:0] Minute
0 – 59

Downloaded from Arrow.com.

Bits 5:0 – SECOND[5:0] Second
0 – 59

Downloaded from Arrow.com.

27.12.11 Alarm Value in Clock/Calendar mode (CTRLA.MODE=2)

Name: ALARM
Offset: 0x20
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

The 32-bit value of ALARM is continuously compared with the 32-bit CLOCK value, based on the masking set by MASK.SEL. When a match occurs, the Alarm n interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.ALARM) is set on the next counter cycle, and the counter is cleared if CTRLA.MATCHCLR is '1'.

Bit	31	30	29	28	27	26	25	24
	YEAR[5:0]						MONTH[3:2]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MONTH[1:0]		DAY[4:0]					HOUR[4:4]
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	HOUR[3:0]				MINUTE[5:2]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MINUTE[1:0]		SECOND[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:26 – YEAR[5:0] Year
The alarm year. Years are only matched if MASK.SEL is 6

Bits 25:22 – MONTH[3:0] Month
The alarm month. Months are matched only if MASK.SEL is greater than 4.

Bits 21:17 – DAY[4:0] Day
The alarm day. Days are matched only if MASK.SEL is greater than 3.

Bits 16:12 – HOUR[4:0] Hour
The alarm hour. Hours are matched only if MASK.SEL is greater than 2.

Bits 11:6 – MINUTE[5:0] Minute
The alarm minute. Minutes are matched only if MASK.SEL is greater than 1.

Bits 5:0 – SECOND[5:0] Second
The alarm second. Seconds are matched only if MASK.SEL is greater than 0.

Downloaded from Arrow.com.

27.12.12 Alarm Mask in Clock/Calendar mode (CTRLA.MODE=2)

Name: MASK
Offset: 0x24
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
						SEL[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:0 – SEL[2:0] Alarm Mask Selection
These bits define which bit groups of ALARM are valid.

Value	Name	Description
0x0	OFF	Alarm Disabled
0x1	SS	Match seconds only
0x2	MMSS	Match seconds and minutes only
0x3	HHMMSS	Match seconds, minutes, and hours only
0x4	DDHHMMSS	Match seconds, minutes, hours, and days only
0x5	MMDDHHMMSS	Match seconds, minutes, hours, days, and months only
0x6	YYMMDDHHMMSS	Match seconds, minutes, hours, days, months, and years
0x7	-	Reserved

Downloaded from Arrow.com.

27.12.13 General Purpose n

Name: GP
Offset: 0x40 + n*0x04 [n=0..1]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	GP[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	GP[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – GP[31:0] General Purpose
These bits are for user-defined general purpose use, see 27.6.8.3 General Purpose Registers.

Downloaded from Arrow.com.

27.12.14 Tamper Control

Name: TAMPCTRL
Offset: 0x60
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
					DEBNC3	DEBNC2	DEBNC1	DEBNC0
Access								
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
					TAMLVL3	TAMLVL2	TAMLVL1	TAMLVL0
Access								
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	IN3ACT[1:0]		IN2ACT[1:0]		IN1ACT[1:0]		IN0ACT[1:0]	
Access								
Reset	0	0	0	0	0	0	0	0

Bits 24, 25, 26, 27 – DEBNC Debounce Enable of Tamper Input INn

Value	Description
0	Debouncing is disabled for Tamper input INn
1	Debouncing is enabled for Tamper input INn

Bits 16, 17, 18, 19 – TAMLVL Tamper Level Select of Tamper Input INn

Value	Description
0	A falling edge condition will be detected on Tamper input INn.
1	A rising edge condition will be detected on Tamper input INn.

Bits 0:1, 2:3, 4:5, 6:7 – INACT Tamper Channel n Action
These bits determine the action taken by Tamper Channel n.

Value	Name	Description
0x0	OFF	Off (Disabled)
0x1	WAKE	Wake and set Tamper flag
0x2	CAPTURE	Capture timestamp and set Tamper flag
0x3	ACTL	Compare RTC signal routed between INn and OUT pins . When a mismatch occurs, capture timestamp and set Tamper flag

Downloaded from Arrow.com.

27.12.15 Timestamp Value

Name: TIMESTAMP
Offset: 0x64
Reset: 0
Property: R

Bit	31	30	29	28	27	26	25	24
	YEAR[5:0]						MONTH[3:2]	
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MONTH[1:0]		DAY[4:0]					HOUR[4:4]
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	HOUR[3:0]				MINUTE[5:2]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MINUTE[1:0]		SECOND[5:0]					
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:26 – YEAR[5:0] Year
The year value is captured by the TIMESTAMP when a tamper condition occurs.

Bits 25:22 – MONTH[3:0] Month
The month value is captured by the TIMESTAMP when a tamper condition occurs.

Bits 21:17 – DAY[4:0] Day
The day value is captured by the TIMESTAMP when a tamper condition occurs.

Bits 16:12 – HOUR[4:0] Hour
The hour value is captured by the TIMESTAMP when a tamper condition occurs.

Bits 11:6 – MINUTE[5:0] Minute
The minute value is captured by the TIMESTAMP when a tamper condition occurs.

Bits 5:0 – SECOND[5:0] Second
The second value is captured by the TIMESTAMP when a tamper condition occurs.

Downloaded from Arrow.com.

27.12.16 Tamper ID

Name: TAMPID
Offset: 0x68
Reset: 0x00000000

Bit	31	30	29	28	27	26	25	24
	TAMPEVT							
Access	R/W							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					TAMPID3	TAMPID2	TAMPID1	TAMPID0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 31 – TAMPEVT Tamper Event Detected
Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the tamper detection bit.

Value	Description
0	A tamper input event has not been detected
1	A tamper input event has been detected

Bits 0, 1, 2, 3 – TAMPID Tamper on Channel n Detected
Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the tamper detection bit.

Value	Description
0	A tamper condition has not been detected on Channel n
1	A tamper condition has been detected on Channel n

Downloaded from Arrow.com.

27.12.17 Tamper Control B

Name: TAMPCTRLB
Offset: 0x6C
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					ALSI3	ALSI2	ALSI1	ALSI0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 0, 1, 2, 3 – ALSI Active Layer Internal Select n

Value	Description
0	Active layer Protection is monitoring the RTC signal using INn and OUTn tamper pins
1	Active layer Protection is monitoring the RTC signal on the TrustRAM shield

Downloaded from Arrow.com.

28. DMAC – Direct Memory Access Controller

28.1 Overview

The Direct Memory Access Controller (DMAC) contains both a Direct Memory Access engine and a Cyclic Redundancy Check (CRC) engine. The DMAC can transfer data between memories and peripherals, and thus off-load these tasks from the CPU. It enables high data transfer rates with minimum CPU intervention, and frees up CPU time. With access to all peripherals, the DMAC can handle automatic transfer of data between communication modules.

The DMA part of the DMAC has several DMA channels which all can receive different types of transfer triggers to generate transfer requests from the DMA channels to the arbiter, see also the Block Diagram. The arbiter will grant one DMA channel at a time to act as the active channel. When an active channel has been granted, the fetch engine of the DMAC will fetch a transfer descriptor from the SRAM and store it in the internal memory of the active channel, which will execute the data transmission.

An ongoing data transfer of an active channel can be interrupted by a higher prioritized DMA channel. The DMAC will write back the updated transfer descriptor from the internal memory of the active channel to SRAM, and grant the higher prioritized channel to start transfer as the new active channel. Once a DMA channel is done with its transfer, interrupts and events can be generated optionally.

The DMAC has four bus interfaces:

- The *data transfer bus* is used for performing the actual DMA transfer.
- The *AHB/APB Bridge bus* is used when writing and reading the I/O registers of the DMAC.
- The *descriptor fetch bus* is used by the fetch engine to fetch transfer descriptors before data transfer can be started or continued.
- The *write-back bus* is used to write the transfer descriptor back to SRAM.

All buses are AHB master interfaces but the AHB/APB Bridge bus, which is an APB slave interface.

The CRC engine can be used by software to detect an accidental error in the transferred data and to take corrective action, such as requesting the data to be sent again or simply not using the incorrect data.

28.2 Features

- Data transfer from:
  - Peripheral to peripheral
  - Peripheral to memory
  - Memory to peripheral
  - Memory to memory
- Transfer trigger sources
  - Software
  - Events from Event System
  - Dedicated requests from peripherals
- SRAM based transfer descriptors
  - Single transfer using one descriptor
  - Multi-buffer or circular buffer modes by linking multiple descriptors

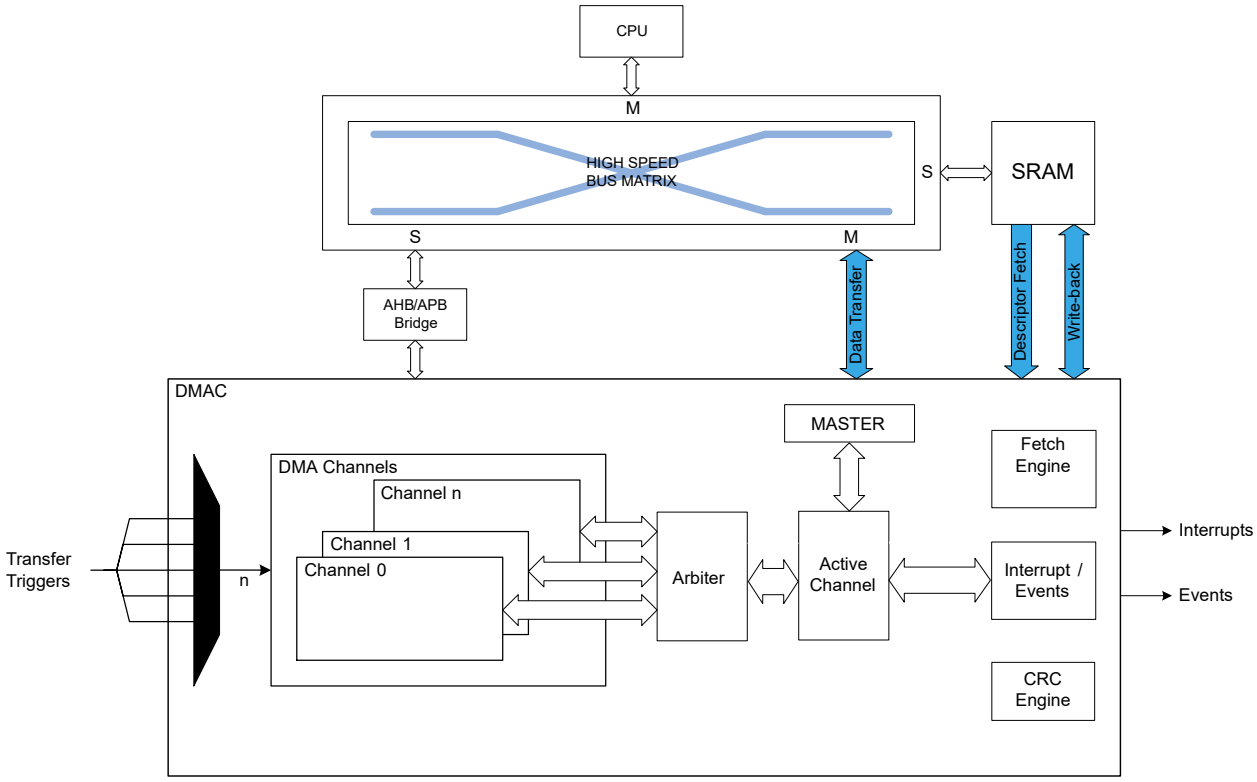
Downloaded from Arrow.com.

- Up to 8 channels
  - Enable 8 independent transfers
  - Automatic descriptor fetch for each channel
  - Suspend/resume operation support for each channel
- Flexible arbitration scheme
  - 4 configurable priority levels for each channel
  - Fixed or round-robin priority scheme within each priority level
- From 1 to 256KB data transfer in a single block transfer
- Multiple addressing modes
  - Static
  - Configurable increment scheme
- Optional interrupt generation
  - On block transfer complete
  - On error detection
  - On channel suspend
- 4 event inputs
  - One event input for each of the 4 least significant DMA channels
  - Can be selected to trigger normal transfers, periodic transfers or conditional transfers
  - Can be selected to suspend or resume channel operation
- 4 event outputs
  - One output event for each of the 4 least significant DMA channels
  - Selectable generation on AHB, block, or transaction transfer complete
- Error management supported by write-back function
  - Dedicated Write-Back memory section for each channel to store ongoing descriptor transfer
- CRC polynomial software selectable to
  - CRC-16 (CRC-CCITT)
  - CRC-32 (IEEE® 802.3)

Downloaded from Arrow.com.

28.3 Block Diagram

Figure 28-1. DMAC Block Diagram



28.4 Signal Description

Not applicable.

28.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

28.5.1 I/O Lines

Not applicable.

28.5.2 Power Management

The DMAC will continue to operate in any sleep mode where the selected source clock is running. The DMAC's interrupts can be used to wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes. On hardware or software reset, all registers are set to their reset value.

Related Links

22. PM – Power Manager

28.5.3 Clocks

The DMAC bus clock (CLK_DMAC_APB) must be configured and enabled in the Main Clock module before using the DMAC.

Downloaded from Arrow.com.

This bus clock (CLK_DMAC_APB) is always synchronous to the module clock (CLK_DMAC_AHB), but can be divided by a prescaler and may run even when the module clock is turned off.

Related Links

19.6.2.6 Peripheral Clock Masking

28.5.4 DMA

Not applicable.

28.5.5 Interrupts

The interrupt request line is connected to the interrupt controller. Using the DMAC interrupt requires the interrupt controller to be configured first.

28.5.6 Events

The events are connected to the event system.

Related Links

33. EVSYS – Event System

28.5.7 Debug Operation

When the CPU is halted in debug mode the DMAC will halt normal operation. The DMAC can be forced to continue operation during debugging. Refer to 28.8.6 DBGCTRL for details.

28.5.8 Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Interrupt Pending register (INTPEND)
- Channel ID register (CHID)
- Channel Interrupt Flag Status and Clear register (CHINTFLAG)

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

Related Links

15. PAC - Peripheral Access Controller

28.5.9 SAM L11 TrustZone Specific Register Access Protection

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

28.5.10 Analog Connections

Not applicable.

Downloaded from Arrow.com.

28.6 Functional Description

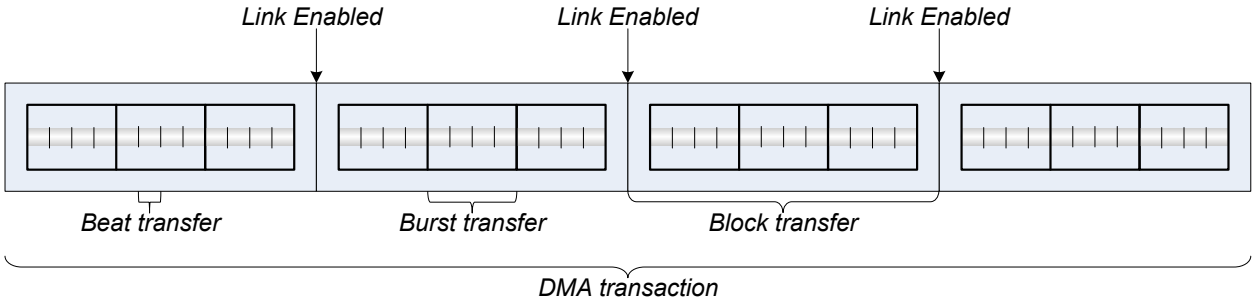
28.6.1 Principle of Operation

The DMAC consists of a DMA module and a CRC module.

28.6.1.1 DMA

The DMAC can transfer data between memories and peripherals without interaction from the CPU. The data transferred by the DMAC are called transactions, and these transactions can be split into smaller data transfers. The following figure shows the relationship between the different transfer sizes:

Figure 28-2. DMA Transfer Sizes



- Beat transfer: The size of one data transfer bus access, and the size is selected by writing the Beat Size bit group in the Block Transfer Control register (BTCTRL.BEATSIZE)
- Block transfer: The amount of data one transfer descriptor can transfer, and the amount can range from 1 to 64k beats. A block transfer can be interrupted.
- Transaction: The DMAC can link several transfer descriptors by having the first descriptor pointing to the second and so forth, as shown in the figure above. A DMA transaction is the complete transfer of all blocks within a linked list.

A transfer descriptor describes how a block transfer should be carried out by the DMAC, and it must remain in SRAM. For further details on the transfer descriptor refer to 28.6.2.3 Transfer Descriptors.

The figure above shows several block transfers linked together, which are called linked descriptors. For further information about linked descriptors, refer to 28.6.3.1 Linked Descriptors.

A DMA transfer is initiated by an incoming transfer trigger on one of the DMA channels. This trigger can be configured to be either a software trigger, an event trigger, or one of the dedicated peripheral triggers. The transfer trigger will result in a DMA transfer request from the specific channel to the arbiter. If there are several DMA channels with pending transfer requests, the arbiter chooses which channel is granted access to become the active channel. The DMA channel granted access as the active channel will carry out the transaction as configured in the transfer descriptor. A current transaction can be interrupted by a higher prioritized channel, but will resume the block transfer when the according DMA channel is granted access as the active channel again.

For each beat transfer, an optional output event can be generated. For each block transfer, optional interrupts and an optional output event can be generated. When a transaction is completed, dependent of the configuration, the DMA channel will either be suspended or disabled.

28.6.1.2 CRC

The internal CRC engine supports two commonly used CRC polynomials: CRC-16 (CRC-CCITT) and CRC-32 (IEEE 802.3). It can be used on a selectable DMA channel, or on the I/O interface. Refer to 28.6.3.7 CRC Operation for details.

Downloaded from Arrow.com.

28.6.2 Basic Operation

28.6.2.1 Initialization

The following DMAC registers are enable-protected, meaning that they can only be written when the DMAC is disabled (CTRL.DMAENABLE=0):

- Descriptor Base Memory Address register (BASEADDR)
- Write-Back Memory Base Address register (WRBADDR)

The following DMAC bit is enable-protected, meaning that it can only be written when both the DMAC and CRC are disabled (CTRL.DMAENABLE=0 and CTRL.CRCENABLE=0):

- Software Reset bit in Control register (CTRL.SWRST)

The following DMA channel register is enable-protected, meaning that it can only be written when the corresponding DMA channel is disabled (CHCTRLA.ENABLE=0):

- Channel Control B (CHCTRLB) register, except the Command bit (CHCTRLB.CMD) and the Channel Arbitration Level bit (CHCTRLB.LVL)

The following DMA channel bit is enable-protected, meaning that it can only be written when the corresponding DMA channel is disabled:

- Channel Software Reset bit in Channel Control A register (CHCTRLA.SWRST)

The following CRC registers are enable-protected, meaning that they can only be written when the CRC is disabled (CTRL.CRCENABLE=0):

- CRC Control register (CRCCTRL)
- CRC Checksum register (CRCCHKSUM)

Enable-protection is denoted by the "Enable-Protected" property in the register description.

Before the DMAC is enabled it must be configured, as outlined by the following steps:

- The SRAM address of where the descriptor memory section is located must be written to the Description Base Address (BASEADDR) register
- The SRAM address of where the write-back section should be located must be written to the Write-Back Memory Base Address (WRBADDR) register
- Priority level x of the arbiter can be enabled by setting the Priority Level x Enable bit in the Control register (CTRL.LVLENx=1)

Before a DMA channel is enabled, the DMA channel and the corresponding first transfer descriptor must be configured, as outlined by the following steps:

- DMA channel configurations
  - The channel number of the DMA channel to configure must be written to the Channel ID (CHID) register
  - Trigger action must be selected by writing the Trigger Action bit group in the Channel Control B register (CHCTRLB.TRIGACT)
  - Trigger source must be selected by writing the Trigger Source bit group in the Channel Control B register (CHCTRLB.TRIGSRC)
- Transfer Descriptor
  - The size of each access of the data transfer bus must be selected by writing the Beat Size bit group in the Block Transfer Control register (BTCTRL.BEATSIZE)

Downloaded from Arrow.com.

- The transfer descriptor must be made valid by writing a one to the Valid bit in the Block Transfer Control register (BTCTRL.VALID)
- Number of beats in the block transfer must be selected by writing the Block Transfer Count (BTCNT) register
- Source address for the block transfer must be selected by writing the Block Transfer Source Address (SRCADDR) register
- Destination address for the block transfer must be selected by writing the Block Transfer Destination Address (DSTADDR) register

If CRC calculation is needed, the CRC engine must be configured before it is enabled, as outlined by the following steps:

- The CRC input source must selected by writing the CRC Input Source bit group in the CRC Control register (CRCCTRL.CRCSRC)
- The type of CRC calculation must be selected by writing the CRC Polynomial Type bit group in the CRC Control register (CRCCTRL.CRCPOLY)
- If I/O is selected as input source, the beat size must be selected by writing the CRC Beat Size bit group in the CRC Control register (CRCCTRL.CRCBEATSIZE)

28.6.2.2 Enabling, Disabling, and Resetting

The DMAC is enabled by writing the DMA Enable bit in the Control register (CTRL.DMAENABLE) to '1'. The DMAC is disabled by writing a '0' to CTRL.DMAENABLE.

A DMA channel is enabled by writing the Enable bit in the Channel Control A register (CHCTRLA.ENABLE) to '1', after writing the corresponding channel id to the Channel ID bit group in the Channel ID register (CHID.ID). A DMA channel is disabled by writing a '0' to CHCTRLA.ENABLE.

The CRC is enabled by writing a '1' to the CRC Enable bit in the Control register (CTRL.CRCENABLE). The CRC is disabled by writing a '0' to CTRL.CRCENABLE.

The DMAC is reset by writing a '1' to the Software Reset bit in the Control register (CTRL.SWRST) while the DMAC and CRC are disabled. All registers in the DMAC except DBGCTRL will be reset to their initial state.

A DMA channel is reset by writing a '1' to the Software Reset bit in the Channel Control A register (CHCTRLA.SWRST), after writing the corresponding channel id to the Channel ID bit group in the Channel ID register (CHID.ID). The channel registers will be reset to their initial state. The corresponding DMA channel must be disabled in order for the reset to take effect.

28.6.2.3 Transfer Descriptors

Together with the channel configurations the transfer descriptors decides how a block transfer should be executed. Before a DMA channel is enabled (CHCTRLA.ENABLE is written to one), and receives a transfer trigger, its first transfer descriptor has to be initialized and valid (BTCTRL.VALID). The first transfer descriptor describes the first block transfer of a transaction.

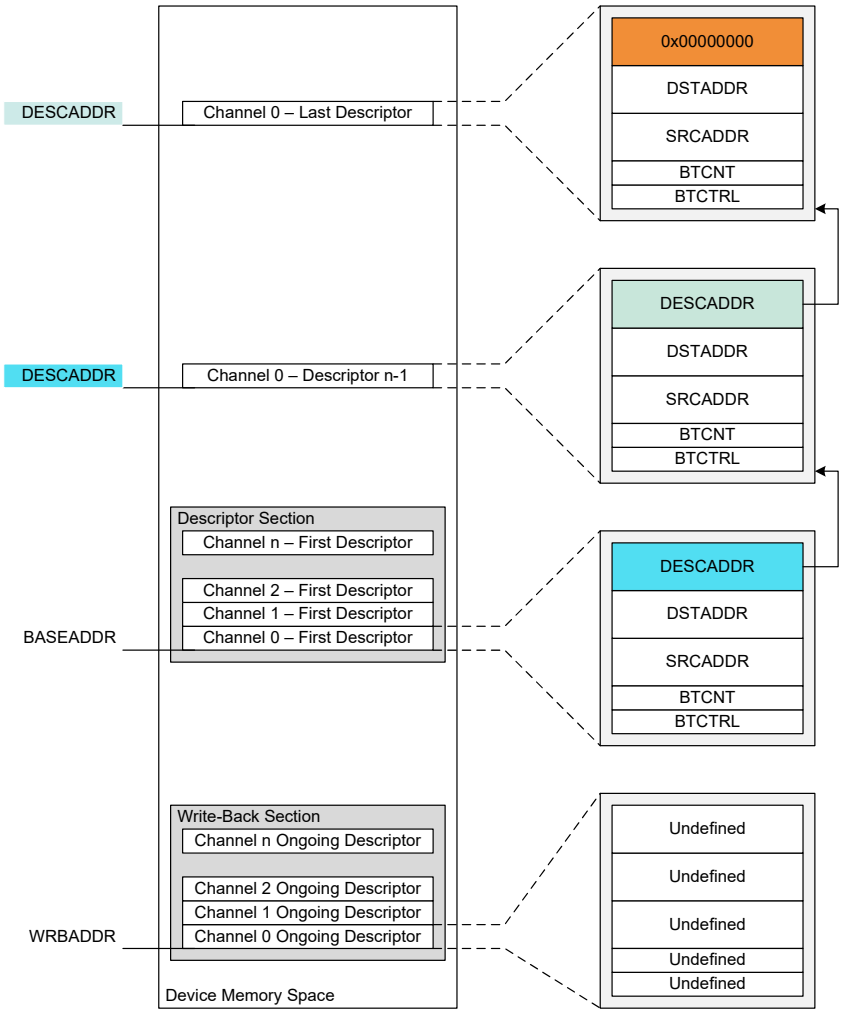
All transfer descriptors must reside in SRAM. The addresses stored in the Descriptor Memory Section Base Address (BASEADDR) and Write-Back Memory Section Base Address (WRBADDR) registers tell the DMAC where to find the descriptor memory section and the write-back memory section.

The descriptor memory section is where the DMAC expects to find the first transfer descriptors for all DMA channels. As BASEADDR points only to the first transfer descriptor of channel 0 (see figure below), all first transfer descriptors must be stored in a contiguous memory section, where the transfer descriptors must be ordered according to their channel number. For further details on linked descriptors, refer to 28.6.3.1 Linked Descriptors.

Downloaded from Arrow.com.

The write-back memory section is the section where the DMAC stores the transfer descriptors for the ongoing block transfers. WRBADDR points to the ongoing transfer descriptor of channel 0. All ongoing transfer descriptors will be stored in a contiguous memory section where the transfer descriptors are ordered according to their channel number. The figure below shows an example of linked descriptors on DMA channel 0. For further details on linked descriptors, refer to 28.6.3.1 Linked Descriptors.

Figure 28-3. Memory Sections



The size of the descriptor and write-back memory sections is dependent on the number of the most significant enabled DMA channel *m*, as shown below:

$$Size = 128\text{bits} \cdot (m + 1)$$

For memory optimization, it is recommended to always use the less significant DMA channels if not all channels are required.

The descriptor and write-back memory sections can either be two separate memory sections, or they can share memory section (BASEADDR=WRBADDR). The benefit of having them in two separate sections, is that the same transaction for a channel can be repeated without having to modify the first transfer descriptor. The benefit of having descriptor memory and write-back memory in the same section is that it requires less SRAM.

Downloaded from Arrow.com.

28.6.2.4 Arbitration

If a DMA channel is enabled and not suspended when it receives a transfer trigger, it will send a transfer request to the arbiter. When the arbiter receives the transfer request it will include the DMA channel in the queue of channels having pending transfers, and the corresponding Pending Channel x bit in the Pending Channels registers (PENDCH.PENDCHx) will be set. Depending on the arbitration scheme, the arbiter will choose which DMA channel will be the next active channel. The active channel is the DMA channel being granted access to perform its next transfer. When the arbiter has granted a DMA channel access to the DMAC, the corresponding bit PENDCH.PENDCHx will be cleared. See also the following figure.

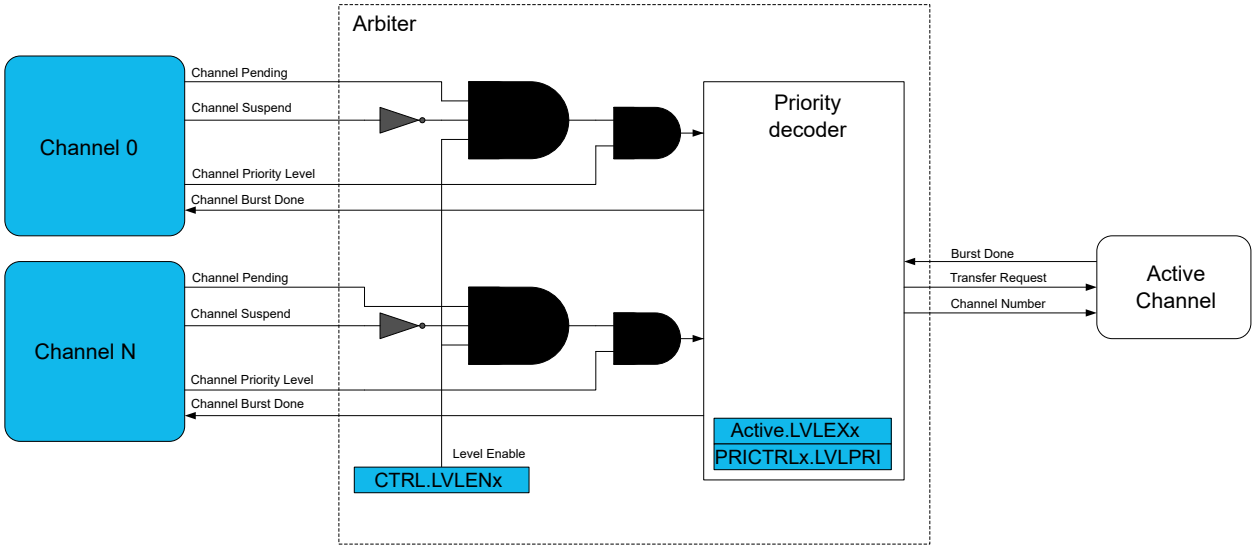
If the upcoming transfer is the first for the transfer request, the corresponding Busy Channel x bit in the Busy Channels register will be set (BUSYCH.BUSYCHx=1), and it will remain '1' for the subsequent granted transfers.

When the channel has performed its granted transfer(s) it will be either fed into the queue of channels with pending transfers, set to be waiting for a new transfer trigger, suspended, or disabled. This depends on the channel and block transfer configuration. If the DMA channel is fed into the queue of channels with pending transfers, the corresponding BUSYCH.BUSYCHx will remain '1'. If the DMA channel is set to wait for a new transfer trigger, suspended, or disabled, the corresponding BUSYCH.BUSYCHx will be cleared.

If a DMA channel is suspended while it has a pending transfer, it will be removed from the queue of pending channels, but the corresponding PENDCH.PENDCHx will remain set. When the same DMA channel is resumed, it will be added to the queue of pending channels again.

If a DMA channel gets disabled (CHCTRLA.ENABLE=0) while it has a pending transfer, it will be removed from the queue of pending channels, and the corresponding PENDCH.PENDCHx will be cleared.

Figure 28-4. Arbiter Overview



Priority Levels

When a channel level is pending or the channel is transferring data, the corresponding Level Executing bit is set in the Active Channel and Levels register (ACTIVE.LVLEXx).

Each DMA channel supports a 4-level priority scheme. The priority level for a channel is configured by writing to the Channel Arbitration Level bit group in the Channel Control B register (CHCTRLB.LVL). As long as all priority levels are enabled, a channel with a higher priority level number will have priority over a channel with a lower priority level number. Each priority level x is enabled by setting the corresponding Priority Level x Enable bit in the Control register (CTRL.LVLENx=1).

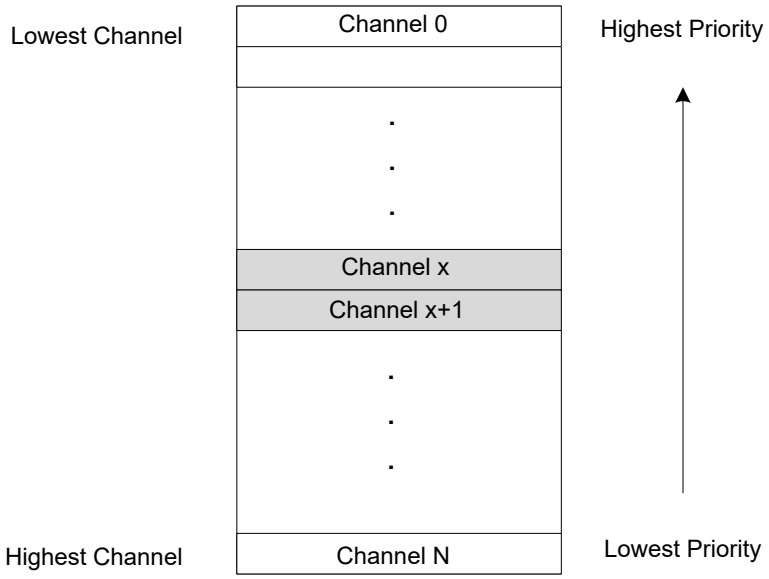
Downloaded from Arrow.com.

Within each priority level the DMAC's arbiter can be configured to prioritize statically or dynamically:

Static Arbitration within a priority level is selected by writing a '0' to the Level x Round-Robin Scheduling Enable bit in the Priority Control 0 register (PRICTRL0.RRLVLENx).

When static arbitration is selected, the arbiter will prioritize a low channel number over a high channel number as shown in the figure below. When using the static arbitration there is a risk of high channel numbers never being granted access as the active channel. This can be avoided using a dynamic arbitration scheme.

Figure 28-5. Static Priority Scheduling

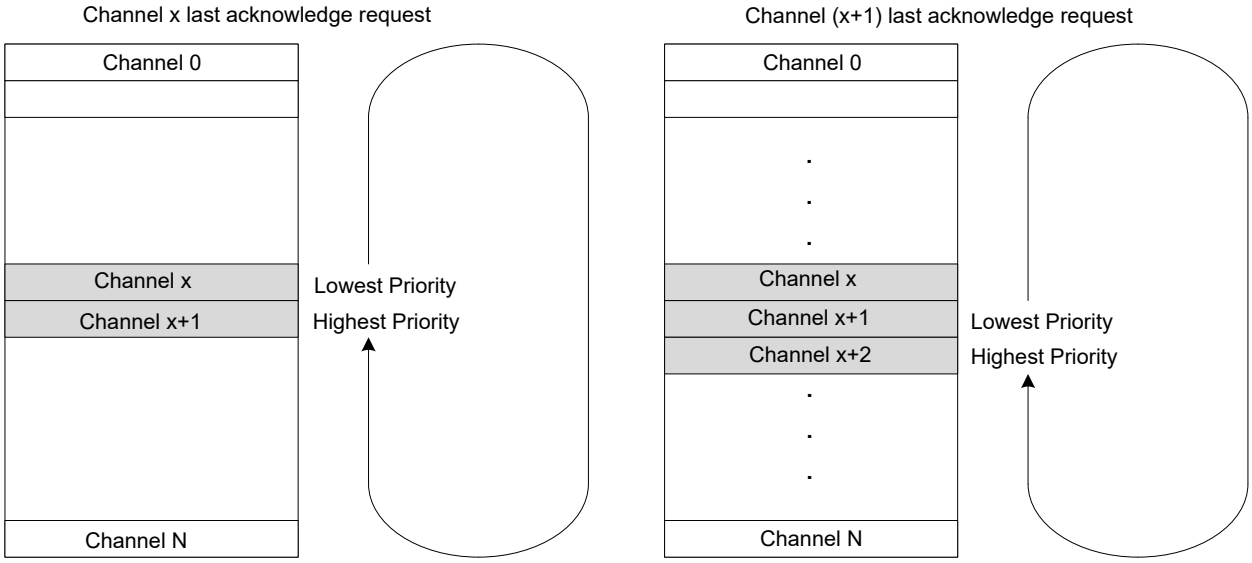


Dynamic Arbitration within a priority level is selected by writing a '1' to PRICTRL0.RRLVLENx.

The dynamic arbitration scheme in the DMAC is round-robin. With the round-robin scheme, the channel number of the last channel being granted access will have the lowest priority the next time the arbiter has to grant access to a channel within the same priority level, as shown in Figure 28-6. The channel number of the last channel being granted access as the active channel is stored in the Level x Channel Priority Number bit group in the Priority Control 0 register (PRICTRL0.LVLPRIx) for the corresponding priority level.

Downloaded from Arrow.com.

Figure 28-6. Dynamic (Round-Robin) Priority Scheduling



28.6.2.5 Data Transmission

Before the DMAC can perform a data transmission, a DMA channel has to be configured and enabled, its corresponding transfer descriptor has to be initialized, and the arbiter has to grant the DMA channel access as the active channel.

Once the arbiter has granted a DMA channel access as the active channel (refer to DMA Block Diagram section) the transfer descriptor for the DMA channel will be fetched from SRAM using the fetch bus, and stored in the internal memory for the active channel. For a new block transfer, the transfer descriptor will be fetched from the descriptor memory section (BASEADDR); For an ongoing block transfer, the descriptor will be fetched from the write-back memory section (WRBADDR). By using the data transfer bus, the DMAC will read the data from the current source address and write it to the current destination address. For further details on how the current source and destination addresses are calculated, refer to the section on Addressing.

The arbitration procedure is performed after each transfer. If the current DMA channel is granted access again, the block transfer counter (BTCNT) of the internal transfer descriptor will be decremented by the number of beats in a transfer, the optional output event Beat will be generated if configured and enabled, and the active channel will perform a new transfer. If a different DMA channel than the current active channel is granted access, the block transfer counter value will be written to the write-back section before the transfer descriptor of the newly granted DMA channel is fetched into the internal memory of the active channel.

When a block transfer has come to its end (BTCNT is zero), the Valid bit in the Block Transfer Control register will be cleared (BTCTRL.VALID=0) before the entire transfer descriptor is written to the write-back memory. The optional interrupts, Channel Transfer Complete and Channel Suspend, and the optional output event Block, will be generated if configured and enabled. After the last block transfer in a transaction, the Next Descriptor Address register (DESCADDR) will hold the value 0x00000000, and the DMA channel will either be suspended or disabled, depending on the configuration in the Block Action bit group in the Block Transfer Control register (BTCTRL.BLOCKACT). If the transaction has further block transfers pending, DESCADDR will hold the SRAM address to the next transfer descriptor to be fetched. The DMAC will fetch the next descriptor into the internal memory of the active channel and write its content to the write-back section for the channel, before the arbiter gets to choose the next active channel.

Downloaded from Arrow.com.

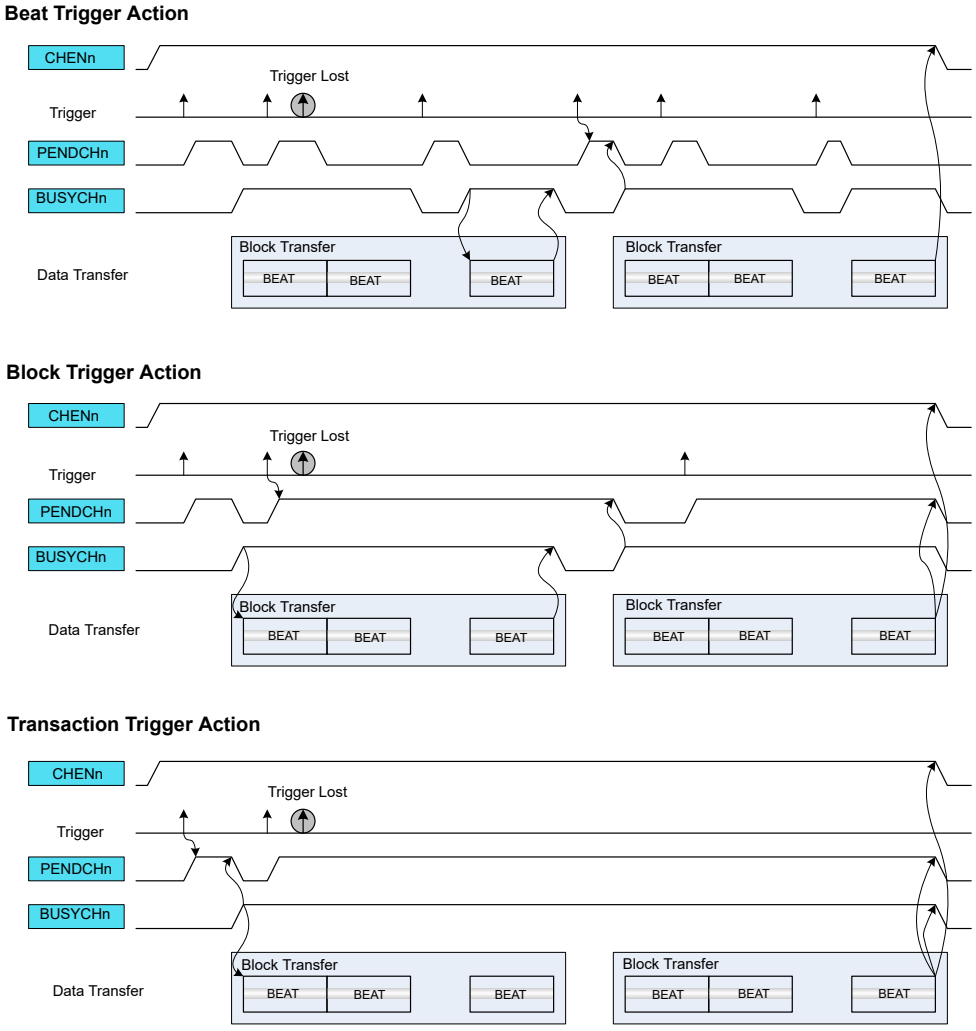
28.6.2.6 Transfer Triggers and Actions

A DMA transfer through a DMA channel can be started only when a DMA transfer request is detected, and the DMA channel has been granted access to the DMA. A transfer request can be triggered from software, from a peripheral, or from an event. There are dedicated Trigger Source selections for each DMA Channel Control B (CHCTRLB.TRIGSRC).

The trigger actions are available in the Trigger Action bit group in the Channel Control B register (CHCTRLB.TRIGACT). By default, a trigger generates a request for a block transfer operation. If a single descriptor is defined for a channel, the channel is automatically disabled when a block transfer has been completed. If a list of linked descriptors is defined for a channel, the channel is automatically disabled when the last descriptor in the list is executed. If the list still has descriptors to execute, the channel will be waiting for the next block transfer trigger. When enabled again, the channel will wait for the next block transfer trigger. The trigger actions can also be configured to generate a request for a beat transfer (CHCTRLB.TRIGACT=0x2) or transaction transfer (CHCTRLB.TRIGACT=0x3) instead of a block transfer (CHCTRLB.TRIGACT=0x0).

Figure 28-7 shows an example where triggers are used with two linked block descriptors.

Figure 28-7. Trigger Action and Transfers



If the trigger source generates a transfer request for a channel during an ongoing transfer, the new transfer request will be kept pending (CHSTATUS.PEND=1), and the new transfer can start after the

Downloaded from Arrow.com.

ongoing one is done. Only one pending transfer can be kept per channel. If the trigger source generates more transfer requests while one is already pending, the additional ones will be lost. All channels pending status flags are also available in the Pending Channels register (PENDCH).

When the transfer starts, the corresponding Channel Busy status flag is set in Channel Status register (CHSTATUS.BUSY). When the trigger action is complete, the Channel Busy status flag is cleared. All channel busy status flags are also available in the Busy Channels register (BUSYCH) in DMAC.

28.6.2.7 Addressing

Each block transfer needs to have both a source address and a destination address defined. The source address is set by writing the Transfer Source Address (SRCADDR) register, the destination address is set by writing the Transfer Destination Address (SRCADDR) register.

The addressing of this DMAC module can be static or incremental, for either source or destination of a block transfer, or both.

Incrementation for the source address of a block transfer is enabled by writing the Source Address Incrementation Enable bit in the Block Transfer Control register (BTCTRL.SRCINC=1). The step size of the incrementation is configurable and can be chosen by writing the Step Selection bit in the Block Transfer Control register (BTCTRL.STEPSEL=1) and writing the desired step size in the Address Increment Step Size bit group in the Block Transfer Control register (BTCTRL.STEPSIZE). If BTCTRL.STEPSEL=0, the step size for the source incrementation will be the size of one beat.

When source address incrementation is configured (BTCTRL.SRCINC=1), SRCADDR is calculated as follows:

If BTCTRL.STEPSEL=1:

$$\mathrm{SRCADDR} = \mathrm{SRCADDR}_{START} + BTCNT \cdot \left(BEATSIZE + 1\right) \cdot 2^{\mathrm{STEPSIZE}}$$

If BTCTRL.STEPSEL=0:

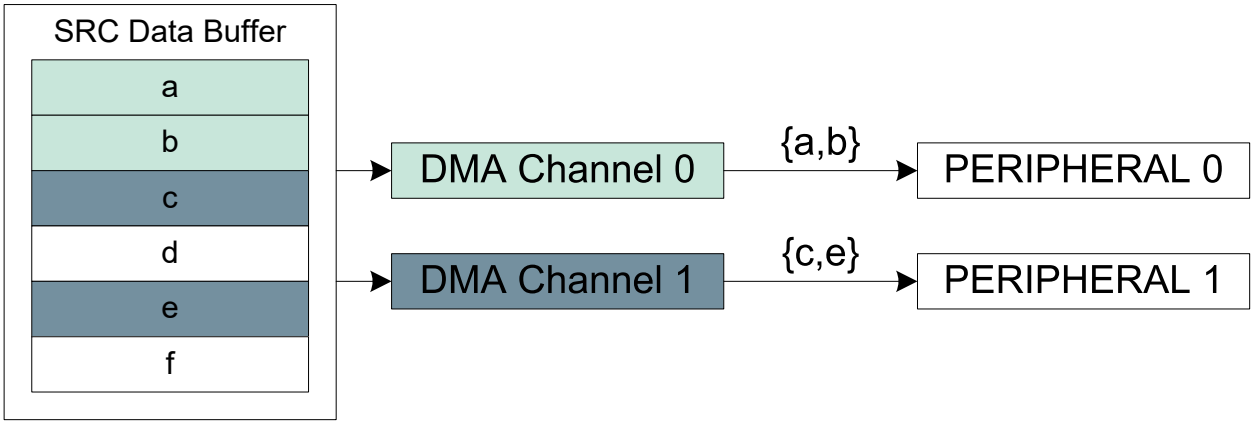
$$\mathrm{SRCADDR} = \mathrm{SRCADDR}_{START} + BTCNT \cdot \left(BEATSIZE + 1\right)$$

- $\mathrm{SRCADDR}_{START}$ is the source address of the first beat transfer in the block transfer
- BTCNT is the initial number of beats remaining in the block transfer
- BEATSIZE is the configured number of bytes in a beat
- STEPSIZE is the configured number of beats for each incrementation

The following figure shows an example where DMA channel 0 is configured to increment the source address by one beat after each beat transfer (BTCTRL.SRCINC=1), and DMA channel 1 is configured to increment the source address by two beats (BTCTRL.SRCINC=1, BTCTRL.STEPSEL=1, and BTCTRL.STEPSIZE=0x1). As the destination address for both channels are peripherals, destination incrementation is disabled (BTCTRL.DSTINC=0).

Downloaded from Arrow.com.

Figure 28-8. Source Address Increment



Incrementation for the destination address of a block transfer is enabled by setting the Destination Address Incrementation Enable bit in the Block Transfer Control register (BTCTRL.DSTINC=1). The step size of the incrementation is configurable by clearing BTCTRL.STEPSEL=0 and writing BTCTRL.STEPSIZE to the desired step size. If BTCTRL.STEPSEL=1, the step size for the destination incrementation will be the size of one beat.

When the destination address incrementation is configured (BTCTRL.DSTINC=1), DSTADDR must be set and calculated as follows:

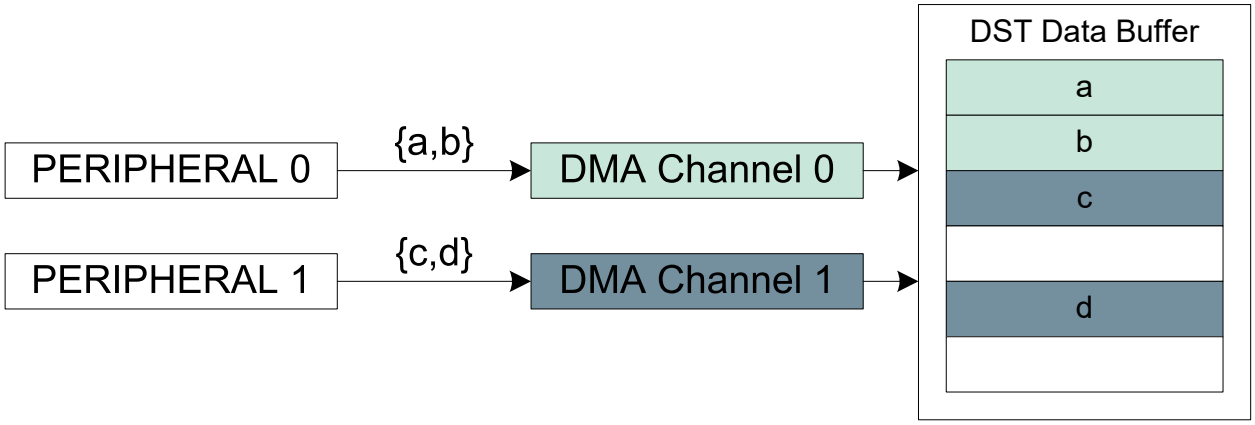
$DSTADDR = DSTADDR_{START} + BTCNT \bullet \left(BEATSIZE + 1\right) \bullet 2^{STEPSIZE}$	where BTCTRL.STEPSEL is zero
$DSTADDR = DSTADDR_{START} + BTCNT \bullet (BEATSIZE + 1)$	where BTCTRL.STEPSEL is one

- $DSTADDR_{START}$ is the destination address of the first beat transfer in the block transfer
- BTCNT is the initial number of beats remaining in the block transfer
- BEATSIZE is the configured number of bytes in a beat
- STEPSIZE is the configured number of beats for each incrementation

The followiong figure shows an example where DMA channel 0 is configured to increment destination address by one beat (BTCTRL.DSTINC=1) and DMA channel 1 is configured to increment destination address by two beats (BTCTRL.DSTINC=1, BTCTRL.STEPSEL=0, and BTCTRL.STEPSIZE=0x1). As the source address for both channels are peripherals, source incrementation is disabled (BTCTRL.SRCINC=0).

Downloaded from Arrow.com.

Figure 28-9. Destination Address Increment



28.6.2.8 Error Handling

If a bus error is received from an AHB slave during a DMA data transfer, the corresponding active channel is disabled and the corresponding Channel Transfer Error Interrupt flag in the Channel Interrupt Status and Clear register (CHINTFLAG.TERR) is set. If enabled, the optional transfer error interrupt is generated. The transfer counter will not be decremented and its current value is written-back in the write-back memory section before the channel is disabled.

When the DMAC fetches an invalid descriptor (BTCTRL.VALID=0) or when the channel is resumed and the DMA fetches the next descriptor with null address (DESCADDR=0x00000000), the corresponding channel operation is suspended, the Channel Suspend Interrupt Flag in the Channel Interrupt Flag Status and Clear register (CHINTFLAG.SUSP) is set, and the Channel Fetch Error bit in the Channel Status register (CHSTATUS.FERR) is set. If enabled, the optional suspend interrupt is generated.

28.6.3 Additional Features

28.6.3.1 Linked Descriptors

A transaction can consist of either a single block transfer or of several block transfers. When a transaction consists of several block transfers it is done with the help of linked descriptors.

Figure 28-3 illustrates how linked descriptors work. When the first block transfer is completed on DMA channel 0, the DMAC fetches the next transfer descriptor, which is pointed to by the value stored in the Next Descriptor Address (DESCADDR) register of the first transfer descriptor. Fetching the next transfer descriptor (DESCADDR) is continued until the last transfer descriptor. When the block transfer for the last transfer descriptor is executed and DESCADDR=0x00000000, the transaction is terminated. For further details on how the next descriptor is fetched from SRAM, refer to section 28.6.2.5 Data Transmission.

28.6.3.1.1 Adding Descriptor to the End of a List

To add a new descriptor at the end of the descriptor list, create the descriptor in SRAM, with DESCADDR=0x00000000 indicating that it is the new last descriptor in the list, and modify the DESCADDR value of the current last descriptor to the address of the newly created descriptor.

28.6.3.1.2 Modifying a Descriptor in a List

In order to add descriptors to a linked list, the following actions must be performed:

1. Enable the Suspend interrupt for the DMA channel.
2. Enable the DMA channel.
3. Reserve memory space in SRAM to configure a new descriptor.
4. Configure the new descriptor:

Downloaded from Arrow.com.

- Set the next descriptor address (DESCADDR)
- Set the destination address (DSTADDR)
- Set the source address (SRCADDR)
- Configure the block transfer control (BTCTRL) including
  - Optionally enable the Suspend block action
  - Set the descriptor VALID bit

5. Clear the VALID bit for the existing list and for the descriptor which has to be updated.
6. Read DESCADDR from the Write-Back memory.
   - If the DMA has not already fetched the descriptor which requires changes (i.e., DESCADDR is wrong):
     - Update the DESCADDR location of the descriptor from the List
     - Optionally clear the Suspend block action
     - Set the descriptor VALID bit to '1'
     - Optionally enable the Resume software command
   - If the DMA is executing the same descriptor as the one which requires changes:
     - Set the Channel Suspend software command and wait for the Suspend interrupt
     - Update the next descriptor address (DESCRADDR) in the write-back memory
     - Clear the interrupt sources and set the Resume software command
     - Update the DESCADDR location of the descriptor from the List
     - Optionally clear the Suspend block action
     - Set the descriptor VALID bit to '1'
7. Go to step 4 if needed.

28.6.3.1.3 Adding a Descriptor Between Existing Descriptors

To insert a new descriptor 'C' between two existing descriptors ('A' and 'B'), the descriptor currently executed by the DMA must be identified.

1. If DMA is executing descriptor B, descriptor C cannot be inserted.
2. If DMA has not started to execute descriptor A, follow the steps:
   2.1. Set the descriptor A VALID bit to '0'.
   2.2. Set the DESCADDR value of descriptor A to point to descriptor C instead of descriptor B.
   2.3. Set the DESCADDR value of descriptor C to point to descriptor B.
   2.4. Set the descriptor A VALID bit to '1'.
3. If DMA is executing descriptor A:
   3.1. Apply the software suspend command to the channel and
   3.2. Perform steps 2.1 through 2.4.
   3.3. Apply the software resume command to the channel.

28.6.3.2 Channel Suspend

The channel operation can be suspended at any time by software by writing a '1' to the Suspend command in the Command bit field of Channel Control B register (CHCTRLB.CMD). After the ongoing burst transfer is completed, the channel operation is suspended and the suspend command is automatically cleared.

When suspended, the Channel Suspend Interrupt flag in the Channel Interrupt Status and Clear register is set (CHINTFLAG.SUSP=1) and the optional suspend interrupt is generated.

Downloaded from Arrow.com.

By configuring the block action to suspend by writing Block Action bit group in the Block Transfer Control register (BTCTRL.BLOCKACT is 0x2 or 0x3), the DMA channel will be suspended after it has completed a block transfer. The DMA channel will be kept enabled and will be able to receive transfer triggers, but it will be removed from the arbitration scheme.

If an invalid transfer descriptor (BTCTRL.VALID=0) is fetched from SRAM, the DMA channel will be suspended, and the Channel Fetch Error bit in the Channel Status register(CHASTATUS.FERR) will be set.

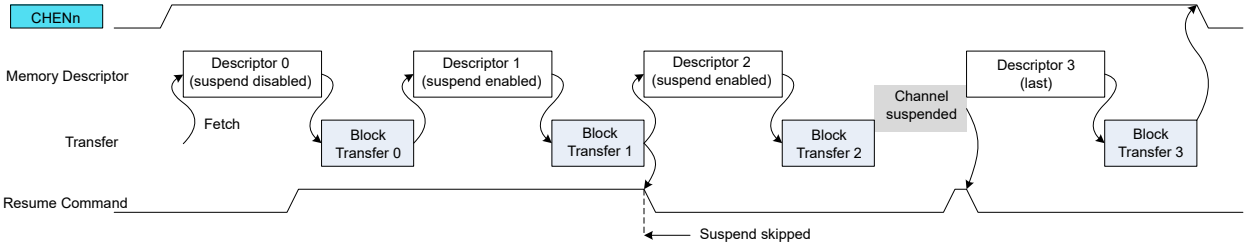
Note: Only enabled DMA channels can be suspended. If a channel is disabled when it is attempted to be suspended, the internal suspend command will be ignored.

For more details on transfer descriptors, refer to section 28.6.2.3 Transfer Descriptors.

28.6.3.3 Channel Resume and Next Suspend Skip

A channel operation can be resumed by software by setting the Resume command in the Command bit field of the Channel Control B register (CHCTRLB.CMD). If the channel is already suspended, the channel operation resumes from where it previously stopped when the Resume command is detected. When the Resume command is issued before the channel is suspended, the next suspend action is skipped and the channel continues the normal operation.

Figure 28-10. Channel Suspend/Resume Operation



28.6.3.4 Event Input Actions

The event input actions are available only on the four least significant DMA channels. For details on channels with event input support, refer to the in the Event system documentation.

Before using event input actions, the event controller must be configured first according to the following table, and the Channel Event Input Enable bit in the Channel Control B register (CHCTRLB.EVIE) must be written to '1'. Refer also to 28.6.6 Events.

Table 28-1. Event Input Action

Action	CHCTRLB.EVACT	CHCTRLB.TRGSRC
None	NOACT	-
Normal Transfer	TRIG	DISABLE
Conditional Transfer on Strobe	TRIG	any peripheral
Conditional Transfer	CTRIG	
Conditional Block Transfer	CBLOCK	
Channel Suspend	SUSPEND	
Channel Resume	RESUME	
Skip Next Block Suspend	SSKIP	

Downloaded from Arrow.com.

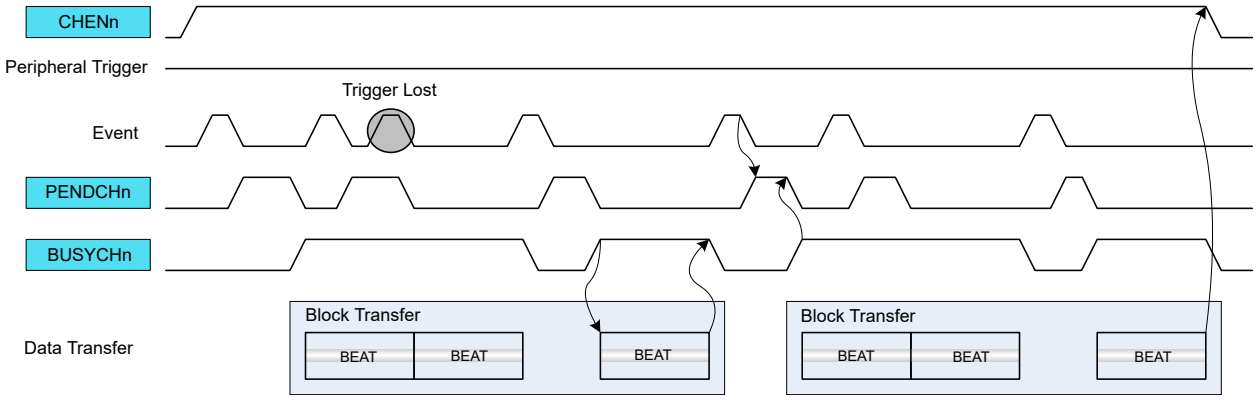
Normal Transfer

The event input is used to trigger a beat or burst transfer on peripherals.

The event is acknowledged as soon as the event is received. When received, both the Channel Pending status bit in the Channel Status register (CHSTATUS.PEND) and the corresponding Channel n bit in the Pending Channels register (28.8.13 PENDCH.PENDCHn) are set. If the event is received while the channel is pending, the event trigger is lost.

The figure below shows an example where beat transfers are enabled by internal events.

Figure 28-11. Beat Event Trigger Action



Conditional Transfer on Strobe

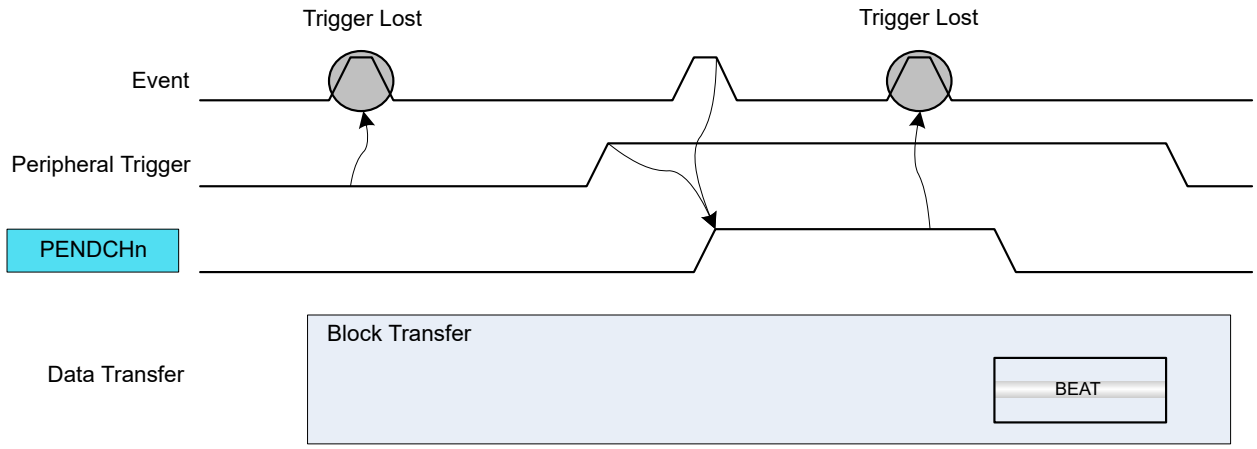
The event input is used to trigger a transfer on peripherals with pending transfer requests. This event action is intended to be used with peripheral triggers, e.g. for timed communication protocols or periodic transfers between peripherals: only when the peripheral trigger coincides with the occurrence of a (possibly cyclic) event the transfer is issued.

The event is acknowledged as soon as the event is received. The peripheral trigger request is stored internally when the previous trigger action is completed (i.e. the channel is not pending) and when an active event is received. If the peripheral trigger is active, the DMA will wait for an event before the peripheral trigger is internally registered. When both event and peripheral transfer trigger are active, both CHSTATUS.PEND and 28.8.13 PENDCH.PENDCHn are set. A software trigger will now trigger a transfer.

The figure below shows an example where the peripheral beat transfer is started by a conditional strobe event action.

Downloaded from Arrow.com.

Figure 28-12. Periodic Event with Beat Peripheral Triggers



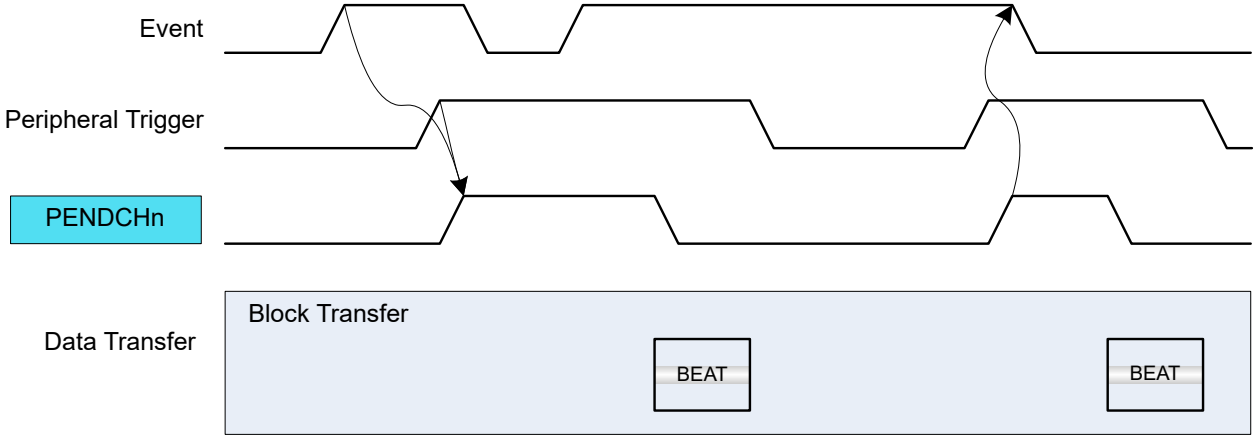
Conditional Transfer

The event input is used to trigger a conditional transfer on peripherals with pending transfer requests. As example, this type of event can be used for peripheral-to-peripheral transfers, where one peripheral is the source of event and the second peripheral is the source of the trigger.

Each peripheral trigger is stored internally when the event is received. When the peripheral trigger is stored internally, the Channel Pending status bit is set (CHSTATUS.PEND), the respective Pending Channel n Bit in the Pending Channels register is set (28.8.13 PENDCH.PENDCHn), and the event is acknowledged. A software trigger will now trigger a transfer.

The figure below shows an example where conditional event is enabled with peripheral beat trigger requests.

Figure 28-13. Conditional Event with Beat Peripheral Triggers



Conditional Block Transfer

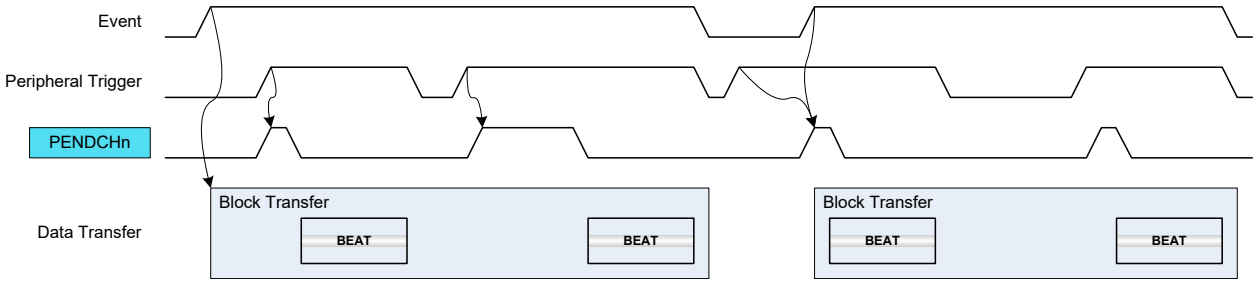
The event input is used to trigger a conditional block transfer on peripherals.

Before starting transfers within a block, an event must be received. When received, the event is acknowledged when the block transfer is completed. A software trigger will trigger a transfer.

The figure below shows an example where conditional event block transfer is started with peripheral beat trigger requests.

Downloaded from Arrow.com.

Figure 28-14. Conditional Block Transfer with Beat Peripheral Triggers



Channel Suspend

The event input is used to suspend an ongoing channel operation. The event is acknowledged when the current AHB access is completed. For further details on Channel Suspend, refer to 28.6.3.2 Channel Suspend.

Channel Resume

The event input is used to resume a suspended channel operation. The event is acknowledged as soon as the event is received and the Channel Suspend Interrupt Flag (CHINTFLAG.SUSP) is cleared. For further details refer to 28.6.3.2 Channel Suspend.

Skip Next Block Suspend

This event can be used to skip the next block suspend action. If the channel is suspended before the event rises, the channel operation is resumed and the event is acknowledged. If the event rises before a suspend block action is detected, the event is kept until the next block suspend detection. When the block transfer is completed, the channel continues the operation (not suspended) and the event is acknowledged.

28.6.3.5 Event Output Selection

Event output selection is available only for the four least significant DMA channels. The pulse width of an event output from a channel is one AHB clock cycle.

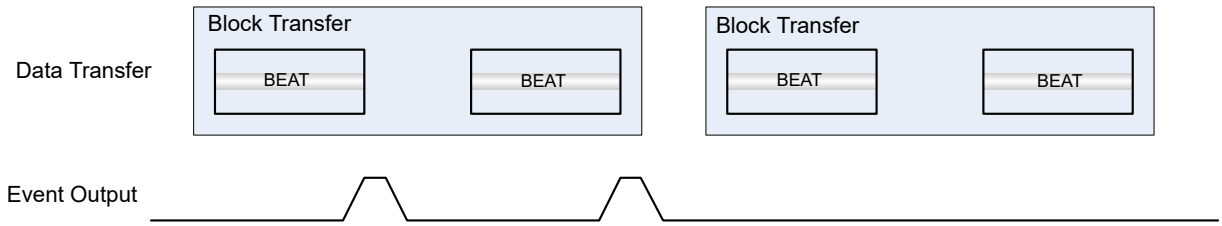
The output of channel events is enabled by writing a '1' to the Channel Event Output Enable bit in the Control B register (CHCTRLB.EVOE). The event output cause is selected by writing to the Event Output Selection bits in the Block Transfer Control register (BTCTRL.EVOSEL). It is possible to generate events after each block transfer (BTCTRL.EVOSEL=0x1) or beat transfer (BTCTRL.EVOSEL=0x3). To enable an event being generated when a transaction is complete, the block event selection must be set in the last transfer descriptor only.

The figure Figure 28-15 shows an example where the event output generation is enabled in the first block transfer, and disabled in the second block.

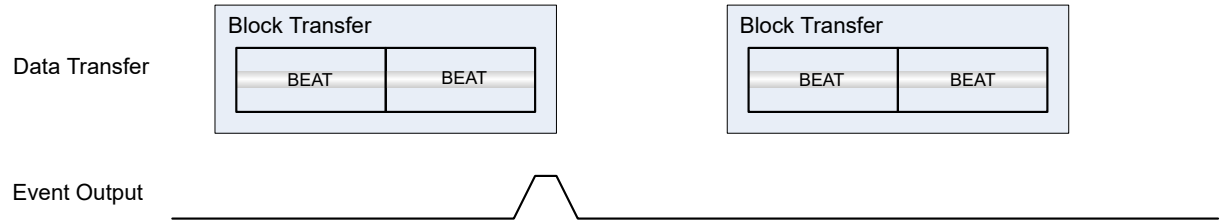
Downloaded from Arrow.com.

Figure 28-15. Event Output Generation

Beat Event Output



Block Event Output



28.6.3.6 Aborting Transfers

Transfers on any channel can be aborted gracefully by software by disabling the corresponding DMA channel. It is also possible to abort all ongoing or pending transfers by disabling the DMAC.

When a DMA channel disable request or DMAC disable request is detected:

- Ongoing transfers of the active channel will be disabled when the ongoing beat transfer is completed and the write-back memory section is updated. This prevents transfer corruption before the channel is disabled.
- All other enabled channels will be disabled in the next clock cycle.

The corresponding Channel Enable bit in the Channel Control A register is cleared (CHCTRLA.ENABLE=0) when the channel is disabled.

The corresponding DMAC Enable bit in the Control register is cleared (CTRL.DMAENABLE=0) when the entire DMAC module is disabled.

28.6.3.7 CRC Operation

A cyclic redundancy check (CRC) is an error detection technique used to find errors in data. It is commonly used to determine whether the data during a transmission, or data present in data and program memories has been corrupted or not. A CRC takes a data stream or a block of data as input and generates a 16- or 32-bit output that can be appended to the data and used as a checksum.

When the data is received, the device or application repeats the calculation: If the new CRC result does not match the one calculated earlier, the block contains a data error. The application will then detect this and may take a corrective action, such as requesting the data to be sent again or simply not using the incorrect data.

The CRC engine in DMAC supports two commonly used CRC polynomials: CRC-16 (CRC-CCITT) and CRC-32 (IEEE 802.3). Typically, applying CRC-n (CRC-16 or CRC-32) to a data block of arbitrary length will detect any single alteration that is ≤n bits in length, and will detect the fraction 1-2-n of all longer error bursts.

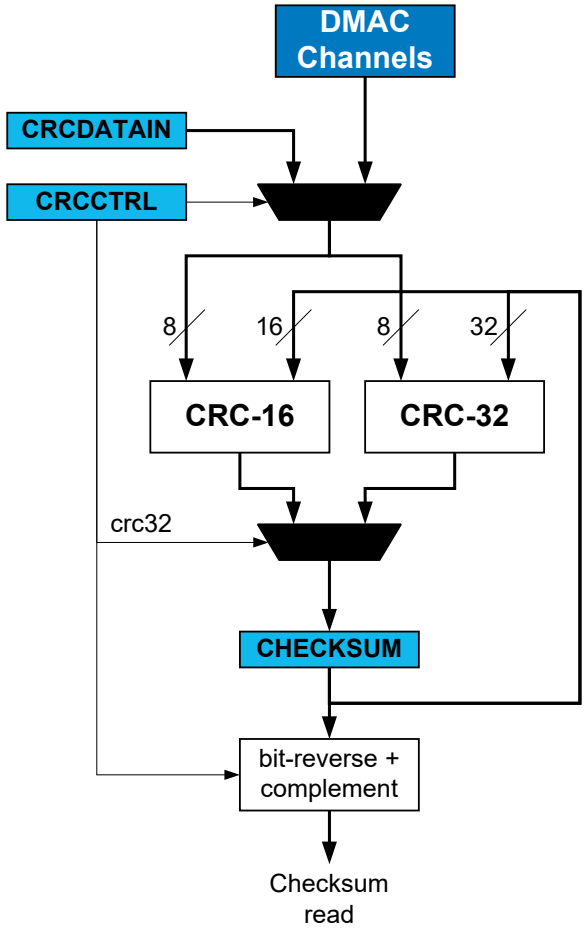
Downloaded from Arrow.com.

- CRC-16:
  - Polynomial: $x^{16}+ x^{12}+ x^5+ 1$
  - Hex value: 0x1021
- CRC-32:
  - Polynomial: $x^{32}+x^{26}+ x^{23}+ x^{22}+x^{16}+ x^{12}+ x^{11}+ x^{10}+ x^8+ x^7+ x^5+ x^4+ x^2+ x + 1$
  - Hex value: 0x04C11DB7

The data source for the CRC engine can either be one of the DMA channels or the APB bus interface, and must be selected by writing to the CRC Input Source bits in the CRC Control register (CRCCTRL.CRCSRC). The CRC engine then takes data input from the selected source and generates a checksum based on these data. The checksum is available in the CRC Checksum register (CRCCHKSUM). When CRC-32 polynomial is used, the final checksum read is bit reversed and complemented, as shown in Figure 28-16.

The CRC polynomial is selected by writing to the CRC Polynomial Type bit in the CRC Control register (CRCCTRL.CRCPOLY), the default is CRC-16. The CRC engine operates on byte only. When the DMA is used as data source for the CRC engine, the DMA channel beat size setting will be used. When used with APB bus interface, the application must select the CRC Beat Size bit field of CRC Control register (CRCCTRL.CRCBEATSIZE). 8-, 16-, or 32-bit bus transfer access type is supported. The corresponding number of bytes will be written in the CRCDATAIN register and the CRC engine will operate on the input data in a byte by byte manner.

Figure 28-16. CRC Generator Block Diagram



Downloaded from Arrow.com.

CRC on DMA data CRC-16 or CRC-32 calculations can be performed on data passing through any DMA channel. Once a DMA channel is selected as the source, the CRC engine will continuously generate the CRC on the data passing through the DMA channel. The checksum is available for readout once the DMA transaction is completed or aborted. A CRC can also be generated on SRAM, Flash, or I/O memory by passing these data through a DMA channel. If the latter is done, the destination register for the DMA data can be the data input (CRCDATAIN) register in the CRC engine.

CRC using the I/O interface Before using the CRC engine with the I/O interface, the application must set the CRC Beat Size bits in the CRC Control register (CRCCTRL.CRCBEATSIZE). 8/16/32-bit bus transfer type can be selected.

CRC can be performed on any data by loading them into the CRC engine using the CPU and writing the data to the CRCDATAIN register. Using this method, an arbitrary number of bytes can be written to the register by the CPU, and CRC is done continuously for each byte. This means if a 32-bit data is written to the CRCDATAIN register the CRC engine takes four cycles to calculate the CRC. The CRC complete is signaled by a set CRCBUSY bit in the CRCSTATUS register. New data can be written only when CRCBUSY flag is not set.

28.6.4 DMA Operation

Not applicable.

28.6.5 Interrupts

The DMAC channels have the following interrupt sources:

- Transfer Complete (TCMPL): Indicates that a block transfer is completed on the corresponding channel. Refer to 28.6.2.5 Data Transmission for details.
- Transfer Error (TERR): Indicates that a bus error has occurred during a burst transfer, or that an invalid descriptor has been fetched. Refer to 28.6.2.8 Error Handling for details.
- Channel Suspend (SUSP): Indicates that the corresponding channel has been suspended. Refer to 28.6.3.2 Channel Suspend and 28.6.2.5 Data Transmission for details.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Channel Interrupt Flag Status and Clear (CHINTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by setting the corresponding bit in the Channel Interrupt Enable Set register (CHINTENSET=1), and disabled by setting the corresponding bit in the Channel Interrupt Enable Clear register (CHINTENCLR=1).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, the DMAC is reset or the corresponding DMA channel is reset. See CHINTFLAG for details on how to clear interrupt flags. All interrupt requests are ORed together on system level to generate one combined interrupt request to the NVIC.

The user must read the Channel Interrupt Status (INTSTATUS) register to identify the channels with pending interrupts and must read the Channel Interrupt Flag Status and Clear (CHINTFLAG) register to determine which interrupt condition is present for the corresponding channel. It is also possible to read the Interrupt Pending register (INTPEND), which provides the lowest channel number with pending interrupt and the respective interrupt flags.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

Downloaded from Arrow.com.

28.6.6 Events

The DMAC can generate the following output events:

- Channel (CH): Generated when a block transfer for a given channel has been completed, or when a beat transfer within a block transfer for a given channel has been completed. Refer to *Event Output Selection* for details.

Setting the Channel Control B Event Output Enable bit (CHCTRLB.EVOE=1) enables the corresponding output event configured in the Event Output Selection bit group in the Block Transfer Control register (BTCTRL.EVOSEL). Clearing CHCTRLB.EVOE=0 disables the corresponding output event.

The DMAC can take the following actions on an input event:

- Transfer and Periodic Transfer Trigger (TRIG): normal transfer or periodic transfers on peripherals are enabled
- Conditional Transfer Trigger (CTRIG): conditional transfers on peripherals are enabled
- Conditional Block Transfer Trigger (CBLOCK): conditional block transfers on peripherals are enabled
- Channel Suspend Operation (SUSPEND): suspend a channel operation
- Channel Resume Operation (RESUME): resume a suspended channel operation
- Skip Next Block Suspend Action (SSKIP): skip the next block suspend transfer condition
- Increase Priority (INCPRI): increase channel priority

Setting the Channel Control B Event Input Enable bit (CHCTRLB.EVIE=1) enables the corresponding action on input event. Clearing this bit disables the corresponding action on input event. Note that several actions can be enabled for incoming events. If several events are connected to the peripheral, any enabled action will be taken for any of the incoming events. For further details on event input actions, refer to *Event Input Actions*.

Note: Event input and outputs are not available for every channel. Refer to 28.2 Features for more information.

Related Links

33. EVSYS – Event System

28.6.7 Sleep Mode Operation

Each DMA channel can be configured to operate in any sleep mode. To be able to run in standby, the RUNSTDBY bit in Channel Control A register (CHCTRLA.RUNSTDBY) must be written to '1'. The DMAC can wake up the device using interrupts from any sleep mode or perform actions through the Event System.

For channels with CHCTRLA.RUNSTDBY = 0, it is up to software to stop DMA transfers on these channels and wait for completion before going to standby mode using the following sequence:

1. Suspend the DMAC channels for which CHCTRLA.RUNSTDBY = 0.
2. Check the SYNCBUSY bits of registers accessed by the DMAC channels being suspended.
3. Go to sleep.
4. When the device wakes up, resume the suspended channels.

28.6.8 Synchronization

Not applicable.

Downloaded from Arrow.com.

28.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRL	7:0						CRCENABLE	DMAENABLE	SWRST
		15:8					LVLENx3	LVLENx2	LVLENx1	LVLENx0
0x02	CRCCTRL	7:0					CRCPOLY[1:0]		CRCBEATSIZE[1:0]	
		15:8			CRCSRC[5:0]					
0x04	CRCDATAIN	7:0	CRCDATAIN[7:0]							
		15:8	CRCDATAIN[15:8]							
		23:16	CRCDATAIN[23:16]							
		31:24	CRCDATAIN[31:24]							
0x08	CRCCHKSUM	7:0	CRCCHKSUM[7:0]							
		15:8	CRCCHKSUM[15:8]							
		23:16	CRCCHKSUM[23:16]							
		31:24	CRCCHKSUM[31:24]							
0x0C	CRCSTATUS	7:0							CRCZERO	CRCBUSY
0x0D	DBGCTRL	7:0								DBGRUN
0x0E	QOSCTRL	7:0			DQOS[1:0]		FQOS[1:0]		WRBQOS[1:0]	
0x0F	Reserved									
0x10	SWTRIGCTRL	7:0		SWTRIGn[6:0]						
		15:8								
		23:16								
		31:24								
0x14	PRICTRL0	7:0	RRLVLEN0				LVLPRI0[3:0]			
		15:8	RRLVLEN1				LVLPRI1[3:0]			
		23:16	RRLVLEN2				LVLPRI2[3:0]			
		31:24	RRLVLEN3				LVLPRI3[3:0]			
0x18 ... 0x1F	Reserved									
0x20	INTPEND	7:0					ID[3:0]			
		15:8	PEND	BUSY	FERR			SUSP	TCMPL	TERR
0x22 ... 0x23	Reserved									
0x24	INTSTATUS	7:0		CHINTn[6:0]						
		15:8								
		23:16								
		31:24								
0x28	BUSYCH	7:0		BUSYCHn[6:0]						
		15:8								
		23:16								
		31:24								
0x2C	PENDCH	7:0	PENDCH7	PENDCH6	PENDCH5	PENDCH4	PENDCH3	PENDCH2	PENDCH1	PENDCH0
		15:8								
		23:16								

Downloaded from Arrow.com.

Offset	Name	Bit Pos.								
		31:24								
0x30	ACTIVE	7:0					LVLEXx	LVLEXx	LVLEXx	LVLEXx
		15:8	ABUSY			ID[4:0]				
		23:16	BTCNT[7:0]							
		31:24	BTCNT[15:8]							
0x34	BASEADDR	7:0	BASEADDR[7:0]							
		15:8			BASEADDR[13:8]					
		23:16								
		31:24								
0x38	WRBADDR	7:0	WRBADDR[7:0]							
		15:8			WRBADDR[13:8]					
		23:16								
		31:24								
0x3C ... 0x3E	Reserved									
0x3F	CHID	7:0					ID[3:0]			
0x40	CHCTRLA	7:0		RUNSTDBY					ENABLE	SWRST
0x41 ... 0x43	Reserved									
0x44	CHCTRLB	7:0		LVL[1:0]		EVOE	EVIE	EVACT[2:0]		
		15:8				TRIGSRC[4:0]				
		23:16	TRIGACT[1:0]							
		31:24							CMD[1:0]	
0x48 ... 0x4B	Reserved									
0x4C	CHINTENCLR	7:0						SUSP	TCMPL	TERR
0x4D	CHINTENSET	7:0						SUSP	TCMPL	TERR
0x4E	CHINTFLAG	7:0						SUSP	TCMPL	TERR
0x4F	CHSTATUS	7:0						FERR	BUSY	PEND

28.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to 28.5.8 Register Access Protection.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

Downloaded from Arrow.com.

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

Downloaded from Arrow.com.

28.8.1 Control

Name: CTRL
Offset: 0x00
Reset: 0x00X0
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
					LVLENx3	LVLENx2	LVLENx1	LVLENx0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
						CRCENABLE	DMAENABLE	SWRST
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 8, 9, 10, 11 – LVLENx Priority Level x Enable
When this bit is set, all requests with the corresponding level will be fed into the arbiter block. When cleared, all requests with the corresponding level will be ignored.

For details on arbitration schemes, refer to the Arbitration section.

These bits are not enable-protected.

Value	Description
0	Transfer requests for Priority level x will not be handled.
1	Transfer requests for Priority level x will be handled.

Bit 2 – CRCENABLE CRC Enable
Writing a '0' to this bit will disable the CRC calculation when the CRC Status Busy flag is cleared (CRCSTATUS. CRCBUSY). The bit is zero when the CRC is disabled.

Writing a '1' to this bit will enable the CRC calculation.

Value	Description
0	The CRC calculation is disabled.
1	The CRC calculation is enabled.

Bit 1 – DMAENABLE DMA Enable
Setting this bit will enable the DMA module.

Writing a '0' to this bit will disable the DMA module. When writing a '0' during an ongoing transfer, the bit will not be cleared until the internal data transfer buffer is empty and the DMA transfer is aborted. The internal data transfer buffer will be empty once the ongoing burst transfer is completed.

This bit is not enable-protected.

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled.

Downloaded from Arrow.com.

Bit 0 – SWRST Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit when both the DMAC and the CRC module are disabled (DMAENABLE and CRCENABLE are '0') resets all registers in the DMAC (except DBGCTRL) to their initial state. If either the DMAC or CRC module is enabled, the Reset request will be ignored and the DMAC will return an access error.

Value	Description
0	There is no Reset operation ongoing.
1	A Reset operation is ongoing.

Downloaded from Arrow.com.

28.8.2 CRC Control

Name: CRCCTRL
Offset: 0x02
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
			CRCSRC[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					CRCPOLY[1:0]		CRCBEATSIZE[1:0]	
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 13:8 – CRCSRC[5:0] CRC Input Source
These bits select the input source for generating the CRC, as shown in the table below. The selected source is locked until either the CRC generation is completed or the CRC module is disabled. This means the CRCSRC cannot be modified when the CRC operation is ongoing. The lock is signaled by the CRCBUSY status bit. CRC generation complete is generated and signaled from the selected source when used with the DMA channel.

Value	Name	Description
0x00	NOACT	No action
0x01	IO	I/O interface
0x02-0x1F	-	Reserved
0x20	CHN	DMA channel 0
0x21	CHN	DMA channel 1
0x22	CHN	DMA channel 2
0x23	CHN	DMA channel 3
0x24	CHN	DMA channel 4
0x25	CHN	DMA channel 5
0x26	CHN	DMA channel 6
0x27	CHN	DMA channel 7
0x28	CHN	DMA channel 8
0x29	CHN	DMA channel 9
0x2A	CHN	DMA channel 10
0x2B	CHN	DMA channel 11
0x2C	CHN	DMA channel 12
0x2D	CHN	DMA channel 13
0x2E	CHN	DMA channel 14
0x2F	CHN	DMA channel 15
0x30	CHN	DMA channel 16
0x31	CHN	DMA channel 17
0x32	CHN	DMA channel 18

Downloaded from Arrow.com.

Value	Name	Description
0x33	CHN	DMA channel 19
0x34	CHN	DMA channel 20
0x35	CHN	DMA channel 21
0x36	CHN	DMA channel 22
0x37	CHN	DMA channel 23
0x38	CHN	DMA channel 24
0x39	CHN	DMA channel 25
0x3A	CHN	DMA channel 26
0x3B	CHN	DMA channel 27
0x3C	CHN	DMA channel 28
0x3D	CHN	DMA channel 29
0x3E	CHN	DMA channel 30
0x3F	CHN	DMA channel 31

Bits 3:2 – CRCPOLY[1:0] CRC Polynomial Type
These bits define the size of the data transfer for each bus access when the CRC is used with I/O interface, as shown in the table below.

Value	Name	Description
0x0	CRC16	CRC-16 (CRC-CCITT)
0x1	CRC32	CRC32 (IEEE 802.3)
0x2-0x3		Reserved

Bits 1:0 – CRCBEATSIZE[1:0] CRC Beat Size
These bits define the size of the data transfer for each bus access when the CRC is used with I/O interface.

Value	Name	Description
0x0	BYTE	8-bit bus transfer
0x1	HWORD	16-bit bus transfer
0x2	WORD	32-bit bus transfer
0x3		Reserved

Downloaded from Arrow.com.

28.8.3 CRC Data Input

Name: CRCDATAIN
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	CRCDATAIN[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CRCDATAIN[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CRCDATAIN[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CRCDATAIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CRCDATAIN[31:0] CRC Data Input
These bits store the data for which the CRC checksum is computed. A new CRC Checksum is ready (CRCBEAT+ 1) clock cycles after the CRCDATAIN register is written.

Downloaded from Arrow.com.

28.8.4 CRC Checksum

Name: CRCCHKSUM
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

The CRCCHKSUM represents the 16- or 32-bit checksum value and the generated CRC. The register is reset to zero by default, but it is possible to reset all bits to one by writing the CRCCHKSUM register directly. It is possible to write this register only when the CRC module is disabled. If CRC-32 is selected and the CRC Status Busy flag is cleared (i.e., CRC generation is completed or aborted), the bit reversed (bit 31 is swapped with bit 0, bit 30 with bit 1, etc.) and complemented result will be read from CRCCHKSUM. If CRC-16 is selected or the CRC Status Busy flag is set (i.e., CRC generation is ongoing), CRCCHKSUM will contain the actual content.

Bit	31	30	29	28	27	26	25	24
	CRCCHKSUM[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CRCCHKSUM[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CRCCHKSUM[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CRCCHKSUM[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CRCCHKSUM[31:0] CRC Checksum
These bits store the generated CRC result. The 16 MSB bits are always read zero when CRC-16 is enabled.

Downloaded from Arrow.com.

28.8.5 CRC Status

Name: CRCSTATUS
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
							CRCZERO	CRCBUSY
Access							R	R/W
Reset							0	0

Bit 1 – CRCZERO CRC Zero
This bit is cleared when a new CRC source is selected.

This bit is set when the CRC generation is complete and the CRC Checksum is zero.

When running CRC-32 and appending the checksum at the end of the packet (as little endian), the final checksum should be 0x2144df1c, and not zero. However, if the checksum is complemented before it is appended (as little endian) to the data, the final result in the checksum register will be zero. See the description of CRCCHKSUM to read out different versions of the checksum.

Bit 0 – CRCBUSY CRC Module Busy
This flag is cleared by writing a one to it when used with I/O interface. When used with a DMA channel, the bit is set when the corresponding DMA channel is enabled, and cleared when the corresponding DMA channel is disabled. This register bit cannot be cleared by the application when the CRC is used with a DMA channel.

This bit is set when a source configuration is selected and as long as the source is using the CRC module.

Downloaded from Arrow.com.

28.8.6 Debug Control

Name: DBGCTRL
Offset: 0x0D
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN Debug Run
This bit is not reset by a software reset.

This bit controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The DMAC is halted when the CPU is halted by an external debugger.
1	The DMAC continues normal operation when the CPU is halted by an external debugger.

Downloaded from Arrow.com.

28.8.7 Quality of Service Control

Name: QOSCTRL
Offset: 0x0E
Reset: 0x2A
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
			DQOS[1:0]		FQOS[1:0]		WRBQOS[1:0]	
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			1	0	1	0	1	0

Bits 5:4 – DQOS[1:0] Data Transfer Quality of Service
These bits define the memory priority access during the data transfer operation.

DQOS[1:0]	Name	Description
0x0	DISABLE	Background (no sensitive operation)
0x1	LOW	Sensitive Bandwidth
0x2	MEDIUM	Sensitive Latency
0x3	HIGH	Critical Latency

Bits 3:2 – FQOS[1:0] Fetch Quality of Service
These bits define the memory priority access during the fetch operation.

FQOS[1:0]	Name	Description
0x0	DISABLE	Background (no sensitive operation)
0x1	LOW	Sensitive Bandwidth
0x2	MEDIUM	Sensitive Latency
0x3	HIGH	Critical Latency

Bits 1:0 – WRBQOS[1:0] Write-Back Quality of Service
These bits define the memory priority access during the write-back operation.

WRBQOS[1:0]	Name	Description
0x0	DISABLE	Background (no sensitive operation)
0x1	LOW	Sensitive Bandwidth
0x2	MEDIUM	Sensitive Latency
0x3	HIGH	Critical Latency

Downloaded from Arrow.com.

28.8.8 Software Trigger Control

Name: SWTRIGCTRL
Offset: 0x10
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		SWTRIGn[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 6:0 – SWTRIGn[6:0] Channel n Software Trigger [n = 7..0]
This bit is cleared when the Channel Pending bit in the Channel Status register (CHSTATUS.PEND) for the corresponding channel is either set, or by writing a '1' to it.

This bit is set if CHSTATUS.PEND is already '1' when writing a '1' to that bit.

Writing a '0' to this bit will clear the bit.

Writing a '1' to this bit will generate a DMA software trigger on channel x, if CHSTATUS.PEND=0 for channel x. CHSTATUS.PEND will be set and SWTRIGn will remain cleared.

Downloaded from Arrow.com.

28.8.9 Priority Control 0

Name: PRICTRL0
Offset: 0x14
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	RRLVLEN3				LVLPRI3[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RRLVLEN2				LVLPRI2[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RRLVLEN1				LVLPRI1[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RRLVLEN0				LVLPRI0[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit 31 – RRLVLEN3 Level 3 Round-Robin Arbitration Enable
This bit controls which arbitration scheme is selected for DMA channels with priority level 3. For details on arbitration schemes, refer to 28.6.2.4 Arbitration.

Value	Description
0	Static arbitration scheme for channels with level 3 priority.
1	Round-robin arbitration scheme for channels with level 3 priority.

Bits 27:24 – LVLPRI3[3:0] Level 3 Channel Priority Number
When round-robin arbitration is enabled (PRICTRL0.RRLVLEN3=1) for priority level 3, this register holds the channel number of the last DMA channel being granted access as the active channel with priority level 3.

When static arbitration is enabled (PRICTRL0.RRLVLEN3=0) for priority level 3, and the value of this bit group is non-zero, it will not affect the static priority scheme.

This bit group is not reset when round-robin arbitration gets disabled (PRICTRL0.RRLVLEN3 written to '0').

Bit 23 – RRLVLEN2 Level 2 Round-Robin Arbitration Enable
This bit controls which arbitration scheme is selected for DMA channels with priority level 2. For details on arbitration schemes, refer to 28.6.2.4 Arbitration.

Downloaded from Arrow.com.

Value	Description
0	Static arbitration scheme for channels with level 2 priority.
1	Round-robin arbitration scheme for channels with level 2 priority.

Bits 19:16 – LVLPRI2[3:0] Level 2 Channel Priority Number
When round-robin arbitration is enabled (PRICTRL0.RRLVLEN2=1) for priority level 2, this register holds the channel number of the last DMA channel being granted access as the active channel with priority level 2.

When static arbitration is enabled (PRICTRL0.RRLVLEN2=0) for priority level 2, and the value of this bit group is non-zero, it will not affect the static priority scheme.

This bit group is not reset when round-robin arbitration gets disabled (PRICTRL0.RRLVLEN2 written to '0').

Bit 15 – RRLVLEN1 Level 1 Round-Robin Scheduling Enable
For details on arbitration schemes, refer to 28.6.2.4 Arbitration.

Value	Description
0	Static arbitration scheme for channels with level 1 priority.
1	Round-robin arbitration scheme for channels with level 1 priority.

Bits 11:8 – LVLPRI1[3:0] Level 1 Channel Priority Number
When round-robin arbitration is enabled (PRICTRL0.RRLVLEN1=1) for priority level 1, this register holds the channel number of the last DMA channel being granted access as the active channel with priority level 1.

When static arbitration is enabled (PRICTRL0.RRLVLEN1=0) for priority level 1, and the value of this bit group is non-zero, it will not affect the static priority scheme.

This bit group is not reset when round-robin arbitration gets disabled (PRICTRL0.RRLVLEN1 written to '0').

Bit 7 – RRLVLEN0 Level 0 Round-Robin Scheduling Enable
For details on arbitration schemes, refer to 28.6.2.4 Arbitration.

Value	Description
0	Static arbitration scheme for channels with level 0 priority.
1	Round-robin arbitration scheme for channels with level 0 priority.

Bits 3:0 – LVLPRI0[3:0] Level 0 Channel Priority Number
When round-robin arbitration is enabled (PRICTRL0.RRLVLEN0=1) for priority level 0, this register holds the channel number of the last DMA channel being granted access as the active channel with priority level 0.

When static arbitration is enabled (PRICTRL0.RRLVLEN0=0) for priority level 0, and the value of this bit group is non-zero, it will not affect the static priority scheme.

This bit group is not reset when round-robin arbitration gets disabled (PRICTRL0.RRLVLEN0 written to '0').

Downloaded from Arrow.com.

28.8.10 Interrupt Pending

Name: INTPEND
Offset: 0x20
Reset: 0x0000
Property: -

This register allows the user to identify the lowest DMA channel with pending interrupt.

Bit	15	14	13	12	11	10	9	8
	PEND	BUSY	FERR			SUSP	TCMPL	TERR
Access	R	R	R			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bit	7	6	5	4	3	2	1	0
					ID[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 15 – PEND Pending
This bit will read '1' when the channel selected by Channel ID field (ID) is pending.

Bit 14 – BUSY Busy
This bit will read '1' when the channel selected by Channel ID field (ID) is busy.

Bit 13 – FERR Fetch Error
This bit will read '1' when the channel selected by Channel ID field (ID) fetched an invalid descriptor.

Bit 10 – SUSP Channel Suspend
This bit will read '1' when the channel selected by Channel ID field (ID) has pending Suspend interrupt.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Channel ID (ID) Suspend interrupt flag.

Bit 9 – TCMPL Transfer Complete
This bit will read '1' when the channel selected by Channel ID field (ID) has pending Transfer Complete interrupt.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Channel ID (ID) Transfer Complete interrupt flag.

Bit 8 – TERR Transfer Error
This bit is read one when the channel selected by Channel ID field (ID) has pending Transfer Error interrupt.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Channel ID (ID) Transfer Error interrupt flag.

Downloaded from Arrow.com.

Bits 3:0 – ID[3:0] Channel ID
These bits store the lowest channel number with pending interrupts. The number is valid if Suspend (SUSP), Transfer Complete (TCMPL) or Transfer Error (TERR) bits are set. The Channel ID field is refreshed when a new channel (with channel number less than the current one) with pending interrupts is detected, or when the application clears the corresponding channel interrupt sources. When no pending channels interrupts are available, these bits will always return zero value when read.

When the bits are written, indirect access to the corresponding Channel Interrupt Flag register is enabled.

Downloaded from Arrow.com.

28.8.11 Interrupt Status

Name: INTSTATUS
Offset: 0x24
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		CHINTn[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bits 6:0 – CHINTn[6:0] Channel n Pending Interrupt [n=7..0]
This bit is set when Channel n has a pending interrupt/the interrupt request is received.

This bit is cleared when the corresponding Channel n interrupts are disabled or the interrupts sources are cleared.

Downloaded from Arrow.com.

28.8.12 Busy Channels

Name: BUSYCH
Offset: 0x28
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		BUSYCHn[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bits 6:0 – BUSYCHn[6:0] Busy Channel n [x=7..0]
This bit is cleared when the channel trigger action for DMA channel n is complete, when a bus error for DMA channel n is detected, or when DMA channel n is disabled.

This bit is set when DMA channel n starts a DMA transfer.

Downloaded from Arrow.com.

28.8.13 Pending Channels

Name: PENDCH
Offset: 0x2C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	PENDCH7	PENDCH6	PENDCH5	PENDCH4	PENDCH3	PENDCH2	PENDCH1	PENDCH0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PENDCH Pending Channel n [n=7..0]
This bit is cleared when trigger execution defined by channel trigger action settings for DMA channel n is started, when a bus error for DMA channel n is detected or when DMA channel n is disabled. For details on trigger action settings, refer to TRIGACT bit in 28.8.19 CHCTRLB.

This bit is set when a transfer is pending on DMA channel n.

Downloaded from Arrow.com.

28.8.14 Active Channel and Levels

Name: ACTIVE
Offset: 0x30
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	BTCNT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BTCNT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ABUSY			ID[4:0]				
Access	R			R	R	R	R	R
Reset	0			0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					LVLEXx	LVLEXx	LVLEXx	LVLEXx
Access					R	R	R	R
Reset					0	0	0	0

Bits 31:16 – BTCNT[15:0] Active Channel Block Transfer Count
These bits hold the 16-bit block transfer count of the ongoing transfer. This value is stored in the active channel and written back in the corresponding Write-Back channel memory location when the arbiter grants a new channel access. The value is valid only when the active channel active busy flag (ABUSY) is set.

Bit 15 – ABUSY Active Channel Busy
This bit is cleared when the active transfer count is written back in the write-back memory section.

This bit is set when the next descriptor transfer count is read from the write-back memory section.

Bits 12:8 – ID[4:0] Active Channel ID
These bits hold the channel index currently stored in the active channel registers. The value is updated each time the arbiter grants a new channel transfer access request.

Bits 3,2,1,0 – LVLEXx Level x Channel Trigger Request Executing [x=3..0]
This bit is set when a level-x channel trigger request is executing or pending.

This bit is cleared when no request is pending or being executed.

Downloaded from Arrow.com.

28.8.15 Descriptor Memory Section Base Address

Name: BASEADDR
Offset: 0x34
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			BASEADDR[13:8]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BASEADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 13:0 – BASEADDR[13:0] Descriptor Memory Base Address
These bits store the Descriptor memory section base address. The value must be 128-bit aligned.

Downloaded from Arrow.com.

28.8.16 Write-Back Memory Section Base Address

Name: WRBADDR
Offset: 0x38
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			WRBADDR[13:8]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	WRBADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 13:0 – WRBADDR[13:0] Write-Back Memory Base Address
These bits store the Write-Back memory base address. The value must be 128-bit aligned.

Downloaded from Arrow.com.

28.8.17 Channel ID

Name: CHID
Offset: 0x3F
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
					ID[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 3:0 – ID[3:0] Channel ID
These bits define the channel number that will be affected by the channel registers (CH*). Before reading or writing a channel register, the channel ID bit group must be written first.

Downloaded from Arrow.com.

28.8.18 Channel Control A

Name: CHCTRLA
Offset: 0x40
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY					ENABLE	SWRST
Access	R	R/W	R	R	R	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 6 – RUNSTDBY Channel run in standby
This bit is used to keep the DMAC channel running in standby mode.

This bit is not enable-protected.

Value	Description
0	The DMAC channel is halted in standby.
1	The DMAC channel continues to run in standby.

Bit 1 – ENABLE Channel Enable
Writing a '0' to this bit during an ongoing transfer, the bit will not be cleared until the internal data transfer buffer is empty and the DMA transfer is aborted. The internal data transfer buffer will be empty once the ongoing burst transfer is completed.

Writing a '1' to this bit will enable the DMA channel.

This bit is not enable-protected.

Value	Description
0	DMA channel is disabled.
1	DMA channel is enabled.

Bit 0 – SWRST Channel Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets the channel registers to their initial state. The bit can be set when the channel is disabled (ENABLE=0). Writing a '1' to this bit will be ignored as long as ENABLE=1. This bit is automatically cleared when the reset is completed.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

Downloaded from Arrow.com.

28.8.19 Channel Control B

Name: CHCTRLB
Offset: 0x44 [ID-00001ece]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Bit	31	30	29	28	27	26	25	24
							CMD[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	TRIGACT[1:0]							
Access	R/W	R/W						
Reset	0	0						

Bit	15	14	13	12	11	10	9	8
				TRIGSRC[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		LVL[1:0]		EVOE	EVIE	EVACT[2:0]		
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 25:24 – CMD[1:0] Software Command
These bits define the software commands. Refer to 28.6.3.2 Channel Suspend and 28.6.3.3 Channel Resume and Next Suspend Skip.

These bits are not enable-protected.

CMD[1:0]	Name	Description
0x0	NOACT	No action
0x1	SUSPEND	Channel suspend operation
0x2	RESUME	Channel resume operation
0x3	-	Reserved

Bits 23:22 – TRIGACT[1:0] Trigger Action
These bits define the trigger action used for a transfer.

TRIGACT[1:0]	Name	Description
0x0	BLOCK	One trigger required for each block transfer
0x1	-	Reserved

Downloaded from Arrow.com.

TRIGACT[1:0]	Name	Description
0x2	BEAT	One trigger required for each beat transfer
0x3	TRANSACTION	One trigger required for each transaction

Bits 12:8 – TRIGSRC[4:0] Trigger Source
These bits define the peripheral trigger which is source of the transfer. For details on trigger selection and trigger modes, refer to Transfer Triggers and Actions and CHCTRLB.TRIGACT.

Value	Name	Description
0x00	DISABLE	Only software/event triggers
0x01	RTC TIMESTAMP	RTC Timestamp Trigger
0x02	DSU DCC0	ID for DCC0 register
0x03	DSU DCC1	ID for DCC1 register
0x04	SERCOM0 RX	SERCOM0 RX Trigger
0x05	SERCOM0 TX	SERCOM0 TX Trigger
0x06	SERCOM1 RX	SERCOM1 RX Trigger
0x07	SERCOM1 TX	SERCOM1 TX Trigger
0x08	SERCOM2 RX	SERCOM2 RX Trigger
0x09	SERCOM2 TX	SERCOM2 TX Trigger
0x0A	TC0 OVF	TC0 Overflow Trigger
0x0B	TC0 MC0	TC0 Match/Compare 0 Trigger
0x0C	TC0 MC1	TC0 Match/Compare 1 Trigger
0x0D	TC1 OVF	TC1 Overflow Trigger
0x0E	TC1 MC0	TC1 Match/Compare 0 Trigger
0x0F	TC1 MC1	TC1 Match/Compare 1 Trigger
0x10	TC2 OVF	TC2 Overflow Trigger
0x11	TC2 MC0	TC2 Match/Compare 0 Trigger
0x12	TC2 MC1	TC2 Match/Compare 1 Trigger
0x13	ADC RESRDY	ADC Result Ready Trigger
0x14	DAC EMPTY	DAC Empty Trigger
0x15	PTC EOC	PTC End of Conversion Trigger
0x16	PTC SEQ	PTC Sequence Trigger
0x17	PTC WCOMP	PTC Window Compare Trigger

Bits 6:5 – LVL[1:0] Channel Arbitration Level
These bits define the arbitration level used for the DMA channel, where a high level has priority over a low level. For further details on arbitration schemes, refer to 28.6.2.4 Arbitration.

These bits are not enable-protected.

TRIGACT[1:0]	Name	Description
0x0	LVL0	Channel Priority Level 0
0x1	LVL1	Channel Priority Level 1
0x2	LVL2	Channel Priority Level 2
0x3	LVL3	Channel Priority Level 3

Downloaded from Arrow.com.

Bit 4 – EVOE Channel Event Output Enable
This bit indicates if the Channel event generation is enabled. The event will be generated for every condition defined in the descriptor Event Output Selection (BTCTRL.EVOSEL).

This bit is available only for the four least significant DMA channels. Refer to table: *User Multiplexer Selection* and *Event Generator Selection* of the Event System for details.

Value	Description
0	Channel event generation is disabled.
1	Channel event generation is enabled.

Bit 3 – EVIE Channel Event Input Enable
This bit is available only for the four least significant DMA channels. Refer to table: *User Multiplexer Selection* and *Event Generator Selection* of the Event System for details.

Value	Description
0	Channel event action will not be executed on any incoming event.
1	Channel event action will be executed on any incoming event.

Bits 2:0 – EVACT[2:0] Event Input Action
These bits define the event input action, as shown below. The action is executed only if the corresponding EVIE bit in the CHCTRLB register of the channel is set.

These bits are available only for the four least significant DMA channels. Refer to table: *User Multiplexer Selection* and *Event Generator Selection* of the Event System for details.

EVACT[2:0]	Name	Description
0x0	NOACT	No action
0x1	TRIG	Normal Transfer and Conditional Transfer on Strobe trigger
0x2	CTRIG	Conditional transfer trigger
0x3	CBLOCK	Conditional block transfer
0x4	SUSPEND	Channel suspend operation
0x5	RESUME	Channel resume operation
0x6	SSKIP	Skip next block suspend action
0x7	-	Reserved

Related Links
33.7.8 CHANNEL

Downloaded from Arrow.com.

28.8.20 Channel Interrupt Enable Clear

Name: CHINTENCLR
Offset: 0x4C
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Channel Interrupt Enable Set (CHINTENSET) register.
This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Bit	7	6	5	4	3	2	1	0
						SUSP	TCMPL	TERR
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – SUSP Channel Suspend Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Channel Suspend Interrupt Enable bit, which disables the Channel Suspend interrupt.

Value	Description
0	The Channel Suspend interrupt is disabled.
1	The Channel Suspend interrupt is enabled.

Bit 1 – TCMPL Channel Transfer Complete Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Channel Transfer Complete Interrupt Enable bit, which disables the Channel Transfer Complete interrupt.

Value	Description
0	The Channel Transfer Complete interrupt is disabled. When block action is set to none, the TCMPL flag will not be set when a block transfer is completed.
1	The Channel Transfer Complete interrupt is enabled.

Bit 0 – TERR Channel Transfer Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Channel Transfer Error Interrupt Enable bit, which disables the Channel Transfer Error interrupt.

Value	Description
0	The Channel Transfer Error interrupt is disabled.
1	The Channel Transfer Error interrupt is enabled.

Downloaded from Arrow.com.

28.8.21 Channel Interrupt Enable Set

Name: CHINTENSET
Offset: 0x4D
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Channel Interrupt Enable Clear (CHINTENCLR) register.
This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Bit	7	6	5	4	3	2	1	0
						SUSP	TCMPL	TERR
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – SUSP Channel Suspend Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Channel Suspend Interrupt Enable bit, which enables the Channel Suspend interrupt.

Value	Description
0	The Channel Suspend interrupt is disabled.
1	The Channel Suspend interrupt is enabled.

Bit 1 – TCMPL Channel Transfer Complete Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Channel Transfer Complete Interrupt Enable bit, which enables the Channel Transfer Complete interrupt.

Value	Description
0	The Channel Transfer Complete interrupt is disabled.
1	The Channel Transfer Complete interrupt is enabled.

Bit 0 – TERR Channel Transfer Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Channel Transfer Error Interrupt Enable bit, which enables the Channel Transfer Error interrupt.

Value	Description
0	The Channel Transfer Error interrupt is disabled.
1	The Channel Transfer Error interrupt is enabled.

Downloaded from Arrow.com.

28.8.22 Channel Interrupt Flag Status and Clear

Name: CHINTFLAG
Offset: 0x4E
Reset: 0x00
Property: -

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Bit	7	6	5	4	3	2	1	0
						SUSP	TCMPL	TERR
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – SUSP Channel Suspend
This flag is cleared by writing a '1' to it.

This flag is set when a block transfer with suspend block action is completed, when a software suspend command is executed, when a suspend event is received or when an invalid descriptor is fetched by the DMA.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Channel Suspend interrupt flag for the corresponding channel.

For details on available software commands, refer to CHCTRLB.CMD.

For details on available event input actions, refer to CHCTRLB.EVACT.

For details on available block actions, refer to BTCTRL.BLOCKACT.

Bit 1 – TCMPL Channel Transfer Complete
This flag is cleared by writing a '1' to it.

This flag is set when a block transfer is completed and the corresponding interrupt block action is enabled.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Transfer Complete interrupt flag for the corresponding channel.

Bit 0 – TERR Channel Transfer Error
This flag is cleared by writing a '1' to it.

This flag is set when a bus error is detected during a beat transfer or when the DMAC fetches an invalid descriptor.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Transfer Error interrupt flag for the corresponding channel.

Downloaded from Arrow.com.

28.8.23 Channel Status

Name: CHSTATUS
Offset: 0x4F
Reset: 0x00
Property: -

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Bit	7	6	5	4	3	2	1	0
						FERR	BUSY	PEND
Access						R	R	R
Reset						0	0	0

Bit 2 – FERR Channel Fetch Error
This bit is cleared when a software resume command is executed.

This bit is set when an invalid descriptor is fetched.

Bit 1 – BUSY Channel Busy
This bit is cleared when the channel trigger action is completed, when a bus error is detected or when the channel is disabled.

This bit is set when the DMA channel starts a DMA transfer.

Bit 0 – PEND Channel Pending
This bit is cleared when the channel trigger action is started, when a bus error is detected or when the channel is disabled. For details on trigger action settings, refer to CHCTRLB.TRIGACT.

This bit is set when a transfer is pending on the DMA channel, as soon as the transfer request is received.

Downloaded from Arrow.com.

28.9 Register Summary - SRAM

Offset	Name	Bit Pos.								
0x00	BTCTRL	7:0				BLOCKACT[1:0]		EVOSEL[1:0]		VALID
		15:8	STEPSIZE[2:0]			STEPSEL	DSTINC	SRCINC	BEATSIZE[1:0]	
0x02	BTCNT	7:0	BTCNT[7:0]							
		15:8	BTCNT[15:8]							
0x04	SRCADDR	7:0	SRCADDR[7:0]							
		15:8	SRCADDR[15:8]							
		23:16	SRCADDR[23:16]							
		31:24	SRCADDR[31:24]							
0x08	DSTADDR	7:0	DSTADDR[7:0]							
		15:8	DSTADDR[15:8]							
		23:16	DSTADDR[23:16]							
		31:24	DSTADDR[31:24]							
0x0C	DESCADDR	7:0	DESCADDR[7:0]							
		15:8	DESCADDR[15:8]							
		23:16	DESCADDR[23:16]							
		31:24	DESCADDR[31:24]							

28.10 Register Description - SRAM

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to 28.5.8 Register Access Protection.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

Downloaded from Arrow.com.

28.10.1 Block Transfer Control

Name: BTCTRL
Offset: 0x00
Property: -

The BTCTRL register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

Bit	15	14	13	12	11	10	9	8
	STEPSIZE[2:0]			STEPSEL	DSTINC	SRCINC	BEATSIZE[1:0]	
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				BLOCKACT[1:0]		EVOSEL[1:0]		VALID
Access								
Reset								

Bits 15:13 – STEPSIZE[2:0] Address Increment Step Size
These bits select the address increment step size. The setting apply to source or destination address, depending on STEPSEL setting.

Value	Name	Description
0x0	X1	Next ADDR = ADDR + (Beat size in byte) * 1
0x1	X2	Next ADDR = ADDR + (Beat size in byte) * 2
0x2	X4	Next ADDR = ADDR + (Beat size in byte) * 4
0x3	X8	Next ADDR = ADDR + (Beat size in byte) * 8
0x4	X16	Next ADDR = ADDR + (Beat size in byte) * 16
0x5	X32	Next ADDR = ADDR + (Beat size in byte) * 32
0x6	X64	Next ADDR = ADDR + (Beat size in byte) * 64
0x7	X128	Next ADDR = ADDR + (Beat size in byte) * 128

Bit 12 – STEPSEL Step Selection
This bit selects if source or destination addresses are using the step size settings.

Value	Name	Description
0x0	DST	Step size settings apply to the destination address
0x1	SRC	Step size settings apply to the source address

Bit 11 – DSTINC Destination Address Increment Enable
Writing a '0' to this bit will disable the destination address incrementation. The address will be kept fixed during the data transfer.

Writing a '1' to this bit will enable the destination address incrementation. By default, the destination address is incremented by 1. If the STEPSEL bit is cleared, flexible step-size settings are available in the STEPSIZE register.

Value	Description
0	The Destination Address Increment is disabled.
1	The Destination Address Increment is enabled.

Downloaded from Arrow.com.

Bit 10 – SRCINC Source Address Increment Enable
Writing a '0' to this bit will disable the source address incrementation. The address will be kept fixed during the data transfer.

Writing a '1' to this bit will enable the source address incrementation. By default, the source address is incremented by 1. If the STEPSEL bit is set, flexible step-size settings are available in the STEPSIZE register.

Value	Description
0	The Source Address Increment is disabled.
1	The Source Address Increment is enabled.

Bits 9:8 – BEATSIZE[1:0] Beat Size
These bits define the size of one beat. A beat is the size of one data transfer bus access, and the setting apply to both read and write accesses.

Value	Name	Description
0x0	BYTE	8-bit bus transfer
0x1	HWORD	16-bit bus transfer
0x2	WORD	32-bit bus transfer
other		Reserved

Bits 4:3 – BLOCKACT[1:0] Block Action
These bits define what actions the DMAC should take after a block transfer has completed.

BLOCKACT[1:0]	Name	Description
0x0	NOACT	Channel will be disabled if it is the last block transfer in the transaction
0x1	INT	Channel will be disabled if it is the last block transfer in the transaction and block interrupt
0x2	SUSPEND	Channel suspend operation is completed
0x3	BOTH	Both channel suspend operation and block interrupt

Bits 2:1 – EVOSEL[1:0] Event Output Selection
These bits define the event output selection.

EVOSEL[1:0]	Name	Description
0x0	DISABLE	Event generation disabled
0x1	BLOCK	Event strobe when block transfer complete
0x2		Reserved
0x3	BEAT	Event strobe when beat transfer complete

Bit 0 – VALID Descriptor Valid
Writing a '0' to this bit in the Descriptor or Write-Back memory will suspend the DMA channel operation when fetching the corresponding descriptor.

The bit is automatically cleared in the Write-Back memory section when channel is aborted, when an error is detected during the block transfer, or when the block transfer is completed.

Downloaded from Arrow.com.

Value	Description
0	The descriptor is not valid.
1	The descriptor is valid.

Downloaded from Arrow.com.

28.10.2 Block Transfer Count

Name: BTCNT
Offset: 0x02
Property: -

The BTCNT register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

Bit	15	14	13	12	11	10	9	8
	BTCNT[15:8]							
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	BTCNT[7:0]							
Access								
Reset								

Bits 15:0 – BTCNT[15:0] Block Transfer Count
This bit group holds the 16-bit block transfer count.

During a transfer, the internal counter value is decremented by one after each beat transfer. The internal counter is written to the corresponding write-back memory section for the DMA channel when the DMA channel loses priority, is suspended or gets disabled. The DMA channel can be disabled by a complete transfer, a transfer error or by software.

Downloaded from Arrow.com.

28.10.3 Block Transfer Source Address

Name: SRCADDR
Offset: 0x04
Property: -

The SRCADDR register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

Bit	31	30	29	28	27	26	25	24
	SRCADDR[31:24]							
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	SRCADDR[23:16]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	SRCADDR[15:8]							
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	SRCADDR[7:0]							
Access								
Reset								

Bits 31:0 – SRCADDR[31:0] Transfer Source Address
This bit group holds the source address corresponding to the last beat transfer address in the block transfer.

Downloaded from Arrow.com.

28.10.4 Block Transfer Destination Address

Name: DSTADDR
Offset: 0x08
Property: -

The DSTADDR register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

Bit	31	30	29	28	27	26	25	24
	DSTADDR[31:24]							
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DSTADDR[23:16]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	DSTADDR[15:8]							
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DSTADDR[7:0]							
Access								
Reset								

Bits 31:0 – DSTADDR[31:0] Transfer Destination Address
This bit group holds the destination address corresponding to the last beat transfer address in the block transfer.

Downloaded from Arrow.com.

28.10.5 Next Descriptor Address

Name: DESCADDR
Offset: 0x0C
Property: -

The DESCADDR register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

Bit	31	30	29	28	27	26	25	24
	DESCADDR[31:24]							
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DESCADDR[23:16]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	DESCADDR[15:8]							
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DESCADDR[7:0]							
Access								
Reset								

Bits 31:0 – DESCADDR[31:0] Next Descriptor Address
This bit group holds the SRAM address of the next descriptor. The value must be 128-bit aligned. If the value of this SRAM register is 0x00000000, the transaction will be terminated when the DMAC tries to load the next transfer descriptor.

Downloaded from Arrow.com.

29. EIC – External Interrupt Controller

29.1 Overview

The External Interrupt Controller (EIC) allows external pins to be configured as interrupt lines. Each interrupt line can be individually masked and can generate an interrupt on rising, falling, or both edges, or on high or low levels. Each external pin has a configurable filter to remove spikes. Each external pin can also be configured to be asynchronous in order to wake up the device from sleep modes where all clocks have been disabled. External pins can also generate an event. Each external pin can be defined as secured or non-secured, where secured pins can only be handled by secure accesses.

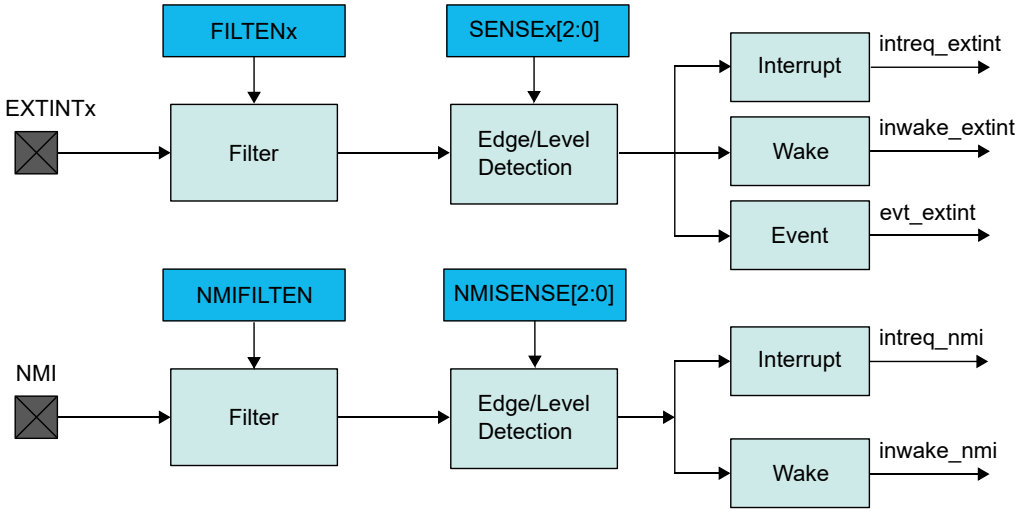
A separate non-maskable interrupt (NMI) is also supported. It has properties similar to the other external interrupts, but is connected to the NMI request of the CPU, enabling it to interrupt any other interrupt mode.

29.2 Features

- Up to 8 external pins (EXTINTx), plus one non-maskable pin (NMI)
- Dedicated, individually maskable interrupt for each pin
- Interrupt on rising, falling, or both edges
- Synchronous or asynchronous edge detection mode
- Interrupt pin debouncing
- Interrupt on high or low levels
- Asynchronous interrupts for sleep modes without clock
- Filtering of external pins
- Event generation from EXTINTx
- Selectable secured or non-secured attribution for each individual external pin **(SAM L11)**

29.3 Block Diagram

Figure 29-1. EIC Block Diagram



Downloaded from Arrow.com.

29.4 Signal Description

Signal Name	Type	Description
EXTINT[7..0]	Digital Input	External interrupt pin
NMI	Digital Input	Non-maskable interrupt pin

One signal may be available on several pins.

29.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

29.5.1 I/O Lines

Using the EIC's I/O lines requires the I/O pins to be configured.

Related Links

32. PORT - I/O Pin Controller

29.5.2 Power Management

All interrupts are available down to STANDBY sleep mode, but the EIC can be configured to automatically mask some interrupts in order to prevent device wake-up.

The EIC will continue to operate in any sleep mode where the selected source clock is running. The EIC's interrupts can be used to wake up the device from sleep modes. Events connected to the Event System can trigger other operations in the system without exiting sleep modes.

Related Links

22. PM – Power Manager

29.5.3 Clocks

The EIC bus clock (CLK_EIC_APB) can be enabled and disabled by the Main Clock Controller, the default state of CLK_EIC_APB can be found in the Peripheral Clock Masking section.

Some optional functions need a peripheral clock, which can either be a generic clock (GCLK_EIC, for wider frequency selection) or a Ultra Low-Power 32 KHz clock (CLK_ULP32K, for highest power efficiency). One of the clock sources must be configured and enabled before using the peripheral:

GCLK_EIC is configured and enabled in the Generic Clock Controller.

CLK_ULP32K is provided by the internal Ultra Low-Power (OSCULP32K) Oscillator in the OSC32KCTRL module.

Both GCLK_EIC and CLK_ULP32K are asynchronous to the user interface clock (CLK_EIC_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

Related Links

19. MCLK – Main Clock

19.6.2.6 Peripheral Clock Masking

18. GCLK - Generic Clock Controller

24. OSC32KCTRL – 32KHz Oscillators Controller

Downloaded from Arrow.com.

29.5.4 DMA

Not applicable.

29.5.5 Interrupts

There are several interrupt request lines, at least one for the external interrupts (EXTINT) and one for non-maskable interrupt (NMI).

The EXTINT interrupt request line is connected to the interrupt controller. Using the EIC interrupt requires the interrupt controller to be configured first.

The NMI interrupt request line is also connected to the interrupt controller, but does not require the interrupt to be configured.

29.5.6 Events

The events are connected to the Event System. Using the events requires the Event System to be configured first.

Related Links

33. EVSYS – Event System

29.5.7 Debug Operation

When the CPU is halted in debug mode, the EIC continues normal operation. If the EIC is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

29.5.8 Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Interrupt Flag Status and Clear register (INTFLAG)
- Non-Maskable Interrupt Flag Status and Clear register (NMIFLAG)

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

Related Links

15. PAC - Peripheral Access Controller

29.5.9 SAM L11 TrustZone Specific Register Access Protection

When the EIC is not PAC secured, non-secure and secure code can both access all functionalities. When the EIC is PAC secured, all registers are by default available in the secure alias only.

A PAC secured EIC can open up individual external interrupts for non-secure access. This is done using the NONSEC and the NONSECNMI registers. When an external interrupt has been set as non-secure, it can be handled from non-secure code, using the EIC module non-secure alias. Since only Secured code has the rights to modify the NONSEC register, an interrupt-based mechanism has been added to let Non Secured code know when these registers have been changed by Secured code. A single flag called NSCHK in the INTFLAG register will rise should changes, conditioned by the NSCHK register, occur in the NONSEC or NONSECNMI registers.

- EIC Security Attribution registers (NONSEC and NONSECNMI) can only be written in the secure alias, otherwise a PAC error results.

Downloaded from Arrow.com.

- The configuration of secured external interrupts can only be changed in the secure alias. Attempt to change the configuration in non-secure mode is silently ignored. Affected configuration registers are: CTRLA, NMICTRL, NMIFLAG, EVCTRL, INTENCLR, INTENSET, INTFLAG, ASYNCH, CONFIGn, DEBOUNCEN, DPRESCALER.

Note: Refer to the *Mix-Secure Peripherals* section in the SAM L11 *Security Features* chapter for more information.

29.5.10 Analog Connections

Not applicable.

29.6 Functional Description

29.6.1 Principle of Operation

The EIC detects edge or level condition to generate interrupts to the CPU interrupt controller or events to the Event System. Each external interrupt pin (EXTINT) can be filtered using majority vote filtering, clocked by GCLK_EIC or by CLK_ULP32K.

29.6.2 Basic Operation

29.6.2.1 Initialization

The EIC must be initialized in the following order:

1. Enable CLK_EIC_APB
2. If required, configure the NMI by writing the Non-Maskable Interrupt Control register (29.8.2 NMICTRL)
3. Enable GCLK_EIC or CLK_ULP32K when one of the following configuration is selected:
   – the NMI uses edge detection or filtering.
   – one EXTINT uses filtering.
   – one EXTINT uses synchronous edge detection.
   – one EXTINT uses debouncing.

   GCLK_EIC is used when a frequency higher than 32KHz is required for filtering.

   CLK_ULP32K is recommended when power consumption is the priority. For CLK_ULP32K write a '1' to the Clock Selection bit in the Control A register (CTRLA.CKSEL).
4. Configure the EIC input sense and filtering by writing the Configuration n register (CONFIG).
5. Optionally, enable the asynchronous mode.
6. Optionally, enable the debouncer mode.
7. Enable the EIC by writing a ‘1’ to CTRLA.ENABLE.

The following bits are enable-protected, meaning that it can only be written when the EIC is disabled (CTRLA.ENABLE=0):

- Clock Selection bit in Control A register (CTRLA.CKSEL)

The following registers are enable-protected:

- Event Control register (29.8.5 EVCTRL)
- Configuration n register (CONFIG).
- External Interrupt Asynchronous Mode register (29.8.9 ASYNCH)

Downloaded from Arrow.com.

- Debouncer Enable register (29.8.11 DEBOUNCEN)
- Debounce Prescaler register (29.8.12 DPRESCALER)

Enable-protected bits in the CTRLA register can be written at the same time when setting CTRLA.ENABLE to '1', but not at the same time as CTRLA.ENABLE is being cleared.

Enable-protection is denoted by the "Enable-Protected" property in the register description.

29.6.2.2 Enabling, Disabling, and Resetting

The EIC is enabled by writing a '1' the Enable bit in the Control A register (CTRLA.ENABLE). The EIC is disabled by writing CTRLA.ENABLE to '0'.

The EIC is reset by setting the Software Reset bit in the Control register (CTRLA.SWRST). All registers in the EIC will be reset to their initial state, and the EIC will be disabled.

Refer to the CTRLA register description for details.

29.6.3 External Pin Processing

Each external pin can be configured to generate an interrupt/event on edge detection (rising, falling or both edges) or level detection (high or low). The sense of external interrupt pins is configured by writing the Input Sense x bits in the Config n register (CONFIG.SENSEx). The corresponding interrupt flag (INTFLAG.EXTINT[x]) in the Interrupt Flag Status and Clear register (29.8.8 INTFLAG) is set when the interrupt condition is met.

When the interrupt flag has been cleared in edge-sensitive mode, INTFLAG.EXTINT[x] will only be set if a new interrupt condition is met.

In level-sensitive mode, when interrupt has been cleared, INTFLAG.EXTINT[x] will be set immediately if the EXTINTx pin still matches the interrupt condition.

Each external pin can be filtered by a majority vote filtering, clocked by GCLK_EIC or CLK_ULP32K. Filtering is enabled if bit Filter Enable x in the Configuration n register (CONFIG.FILTENx) is written to '1'. The majority vote filter samples the external pin three times with GCLK_EIC or CLK_ULP32K and outputs the value when two or more samples are equal.

Table 29-1. Majority Vote Filter

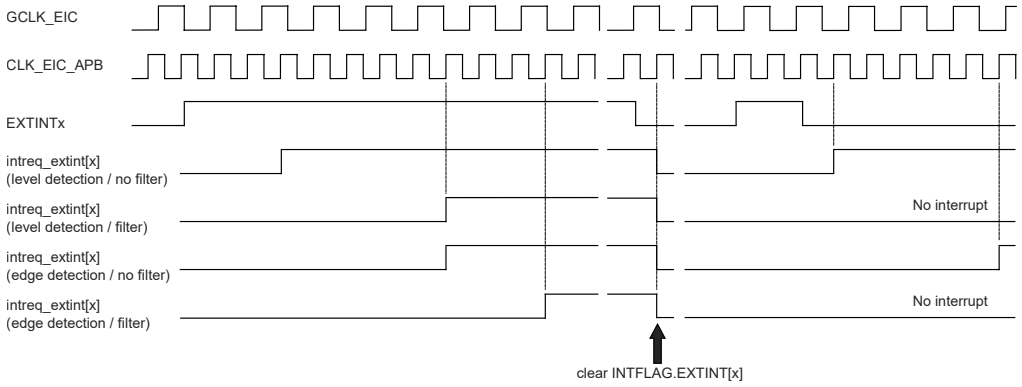
Samples [0, 1, 2]	Filter Output
[0,0,0]	0
[0,0,1]	0
[0,1,0]	0
[0,1,1]	1
[1,0,0]	0
[1,0,1]	1
[1,1,0]	1
[1,1,1]	1

When an external interrupt is configured for level detection and when filtering is disabled, detection is done asynchronously. Level detection and asynchronous edge detection does not require GCLK_EIC or CLK_ULP32K, but interrupt and events can still be generated.

Downloaded from Arrow.com.

If filtering or synchronous edge detection or debouncing is enabled, the EIC automatically requests GCLK_EIC or CLK_ULP32K to operate. The selection between these two clocks is done by writing the Clock Selection bits in the Control A register (CTRLA.CKSEL). GCLK_EIC must be enabled in the GCLK module. In these modes the external pin is sampled at the EIC clock rate, thus pulses with duration lower than two EIC clock periods may not be properly detected.

Figure 29-2. Interrupt Detection Latency by modes (Rising Edge)



The detection latency depends on the detection mode.

Table 29-2. Detection Latency

Detection mode	Latency (worst case)
Level without filter	Five CLK_EIC_APB periods
Level with filter	Four GCLK_EIC/CLK_ULP32K periods + five CLK_EIC_APB periods
Edge without filter	Four GCLK_EIC/CLK_ULP32K periods + five CLK_EIC_APB periods
Edge with filter	Six GCLK_EIC/CLK_ULP32K periods + five CLK_EIC_APB periods

Related Links

18. GCLK - Generic Clock Controller

29.6.4 Additional Features

29.6.4.1 Non-Maskable Interrupt (NMI)

The non-maskable interrupt pin can also generate an interrupt on edge or level detection, but it is configured with the dedicated NMI Control register (NMICTRL). To select the sense for NMI, write to the NMISENSE bit group in the NMI Control register (NMICTRL.NMISENSE). NMI filtering is enabled by writing a '1' to the NMI Filter Enable bit (NMICTRL.NMIFILTEN).

If edge detection or filtering is required, enable GCLK_EIC or CLK_ULP32K.

NMI detection is enabled only by the NMICTRL.NMISENSE value, and the EIC is not required to be enabled.

When an NMI is detected, the non-maskable interrupt flag in the NMI Flag Status and Clear register is set (NMIFLAG.NMI). NMI interrupt generation is always enabled, and NMIFLAG.NMI generates an interrupt request when set.

29.6.4.2 Asynchronous Edge Detection Mode (No Debouncing)

The EXTINT edge detection can be operated synchronously or asynchronously, selected by the Asynchronous Control Mode bit for external pin x in the External Interrupt Asynchronous Mode register

Downloaded from Arrow.com.

(ASYNCH.ASYNCH[x]). The EIC edge detection is operated synchronously when the Asynchronous Control Mode bit (ASYNCH.ASYNCH[x]) is '0' (default value). It is operated asynchronously when ASYNCH.ASYNCH[x] is written to '1'.

In *Synchronous Edge Detection Mode*, the external interrupt (EXTINT) or the non-maskable interrupt (NMI) pins are sampled using the EIC clock as defined by the Clock Selection bit in the Control A register (CTRLA.CKSEL). The External Interrupt flag (INTFLAG.EXTINT[x]) or Non-Maskable Interrupt flag (NMIFLAG.NMI) is set when the last sampled state of the pin differs from the previously sampled state. In this mode, the EIC clock is required.

The Synchronous Edge Detection Mode can be used in Idle and Standby sleep modes.

In *Asynchronous Edge Detection Mode*, the external interrupt (EXTINT) pins or the non-maskable interrupt (NMI) pins set the External Interrupt flag or Non-Maskable Interrupt flag (INTFLAG.EXTINT[x] or NMIFLAG) directly. In this mode, the EIC clock is not requested.

The asynchronous edge detection mode can be used in Idle and Standby sleep modes.

29.6.4.3 Interrupt Pin Debouncing

The external interrupt pin (EXTINT) edge detection can use a debouncer to improve input noise immunity. When selected, the debouncer can work in the synchronous mode or the asynchronous mode, depending on the configuration of the ASYNCH.ASYNCH[x] bit for the pin. The debouncer uses the EIC clock as defined by the bit CTRLA.CKSEL to clock the debouncing circuitry. The debouncing time frame is set with the debouncer prescaler DPRESCALER.DPRESCALERn, which provides the *low frequency clock* tick that is used to reject higher frequency signals.

The debouncing mode for pin EXTINT x can be selected only if the Sense bits in the Configuration y register (CONFIGy.SENSEx) are set to RISE, FALL or BOTH. If the debouncing mode for pin EXTINT x is selected, the filter mode for that pin (CONFIGy.FILTENx) can not be selected.

The debouncer manages an internal “valid pin state” that depends on the external interrupt (EXTINT) pin transitions, the debouncing mode and the debouncer prescaler frequency. The valid pin state reflects the pin value after debouncing. The external interrupt pin (EXTINT) is sampled continously on EIC clock. The sampled value is evaluated on each *low frequency clock* tick to detect a transitional edge when the sampled value is different of the current valid pin state. The sampled value is evaluated on each EIC clock when DPRESCALER.TICKON=0 or on each *low frequency clock* tick when DPRESCALER.TICKON=1, to detect a bounce when the sampled value is equal to the current valid pin state. Transitional edge detection increments the transition counter of the EXTINT pin, while bounce detection resets the transition counter. The transition counter must exceed the transition count threshold as defined by the DPRESCALER.STATESn bitfield. In the synchronous mode the threshold is 4 when DPRESCALER.STATESn=0 or 8 when DPRESCALER.STATESn=1. In the asynchronous mode the threshold is 4.

The valid pin state for the pins can be accessed by reading the register PINSTATE for both synchronous or asynchronous debouncing mode.

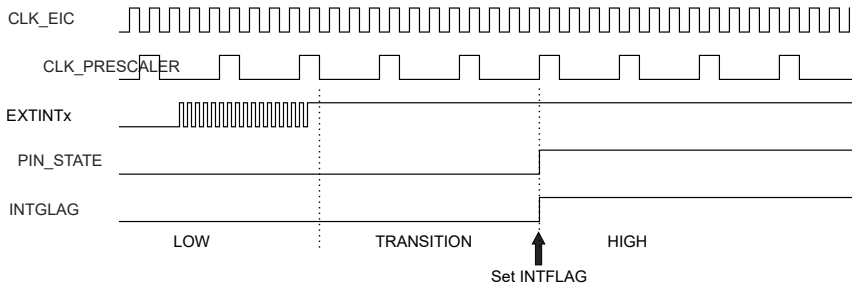
Synchronous edge detection In this mode the external interrupt (EXTINT) pin is sampled continously on EIC clock.

1. A pin edge transition will be validated when the sampled value is consistently different of the current valid pin state for 4 (or 8 depending on bit DPRESCALER.STATESn) consecutive ticks of the low frequency clock.
2. Any pin sample, at the *low frequency clock* tick rate, with a value opposite to the current valid pin state will increment the transition counter.

Downloaded from Arrow.com.

3. Any pin sample, at EIC clock rate (when DPRESCALER.TICKON=0) or the *low frequency clock* tick (when DPRESCALER.TICKON=1), with a value identical to the current valid pin state will return the transition counter to zero.
4. When the transition counter meets the count threshold, the pin edge transition is validated and the pin state PINSTATE.PINSTATE[x] is changed to the detected level.
5. The external interrupt flag (INTFLAG.EXTINT[x]) is set when the pin state PINSTATE.PINSTATE[x] is changed.

Figure 29-3. EXTINT Pin Synchronous Debouncing (Rising Edge)

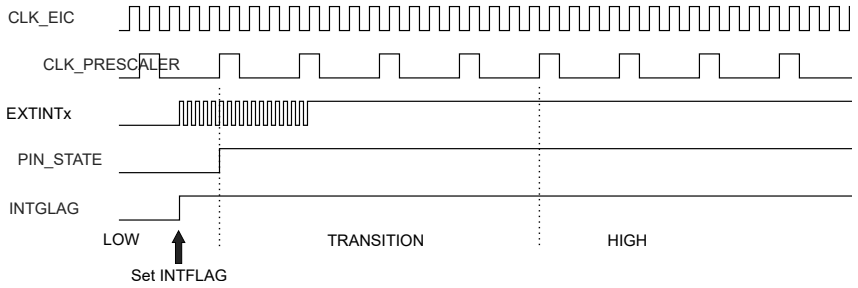


In the synchronous edge detection mode, the EIC clock is required. The synchronous edge detection mode can be used in Idle and Standby sleep modes.

Asynchronous edge detection In this mode, the external interrupt (EXTINT) pin directly drives an asynchronous edges detector which triggers any rising or falling edge on the pin:

1. Any edge detected that indicates a transition from the current valid pin state will immediately set the valid pin state PINSTATE.PINSTATE[x] to the detected level.
2. The external interrupt flag (INTFLAG.EXTINT[x] is immediately changed.
3. The edge detector will then be idle until no other rising or falling edge transition is detected during 4 consecutive ticks of the low frequency clock.
4. Any rising or falling edge transition detected during the idle state will return the transition counter to 0.
5. After 4 consecutive ticks of the low frequency clock without bounce detected, the edge detector is ready for a new detection.

Figure 29-4. EXTINT Pin Asynchronous Debouncing (Rising Edge)



In this mode, the EIC clock is requested. The asynchronous edge detection mode can be used in Idle and Standby sleep modes.

29.6.5 DMA Operation

Not applicable.

29.6.6 Interrupts

The EIC has the following interrupt sources:

Downloaded from Arrow.com.

- External interrupt pins (EXTINTx). See 29.6.2 Basic Operation.
- Non-maskable interrupt pin (NMI). See 29.6.4 Additional Features.
- Non-secure check interrupt pin (NSCHK). See 29.8.8 INTFLAG

Each interrupt source has an associated interrupt flag. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when an interrupt condition occurs (NMIFLAG for NMI). Each interrupt, except NMI, can be individually enabled by setting the corresponding bit in the Interrupt Enable Set register (INTENSET=1), and disabled by setting the corresponding bit in the Interrupt Enable Clear register (INTENCLR=1).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the EIC is reset. See the INTFLAG register for details on how to clear interrupt flags. The EIC has at least one common interrupt request line for all the interrupt sources, and one interrupt request line for the NMI. The user must read the INTFLAG (or NMIFLAG) register to determine which interrupt condition is present.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

29.6.7 Events

The EIC can generate the following output events:

- External event from pin (EXTINTx).

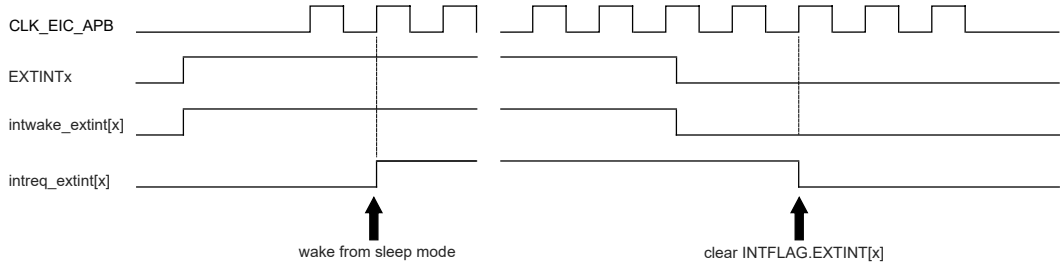
Setting an Event Output Control register (EVCTRL.EXTINTEO) enables the corresponding output event. Clearing this bit disables the corresponding output event. Refer to *Event System* for details on configuring the Event System.

When the condition on pin EXTINTx matches the configuration in the CONFIGn register, the corresponding event is generated, if enabled.

29.6.8 Sleep Mode Operation

In sleep modes, an EXTINTx pin can wake up the device if the corresponding condition matches the configuration in the CONFIG register, and the corresponding bit in the Interrupt Enable Set register (29.8.7 INTENSET) is written to '1'.

Figure 29-5. Wake-up Operation Example (High-Level Detection, No Filter, Interrupt Enable Set)



29.6.9 SAM L11 Secure Access Rights

Non-secure write to CTRLA register or DPRESCALER register is prohibited. Non-secure read to CTRLA or DPRESCALER register or SYNCBUSY register will return zero with no error resulting. Non-secure write to a bit of EVCTRL, ASYNCH, DEBOUNCEN, INTENCLR, INTENSET, INTFLAG and CONFIG registers is prohibited if the related bit of NONSEC.EXTINT is zero. Non-secure write to NMICTRL and NMIFLAG registers is prohibited if NONSECNMI.NMI is zero. Bits relating to secure EXTINT read as zero in non-secure mode with no error resulting.

Downloaded from Arrow.com.

29.6.10 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in control register (CTRLA.SWRST)
- Enable bit in control register (CTRLA.ENABLE)

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Downloaded from Arrow.com.

29.7 Register Summary

Important:
For SAM L11, the EIC register map is automatically duplicated in a Secure and Non-Secure alias:

- The Non-Secure alias is at the peripheral base address
- The Secure alias is located at the peripheral base address + 0x200

Refer to *Mix-Secure Peripherals* for more information on register access rights

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0				CKSEL			ENABLE	SWRST
0x01	NMICTRL	7:0				NMIASYNCH	NMIFILTEN	NMISENSE[2:0]		
0x02	NMIFLAG	7:0								NMI
		15:8								
0x04	SYNCBUSY	7:0							ENABLE	SWRST
		15:8								
		23:16								
		31:24								
0x08	EVCTRL	7:0	EXTINTEO[7:0]							
		15:8								
		23:16								
		31:24								
0x0C	INTENCLR	7:0	EXTINT[7:0]							
		15:8								
		23:16								
		31:24	NSCHK							
0x10	INTENSET	7:0	EXTINT[7:0]							
		15:8								
		23:16								
		31:24	NSCHK							
0x14	INTFLAG	7:0	EXTINT[7:0]							
		15:8								
		23:16								
		31:24	NSCHK							
0x18	ASYNCH	7:0	ASYNCH[7:0]							
		15:8								
		23:16								
		31:24								
0x1C	CONFIG	7:0	FILTENx	SENSEx[2:0]			FILTENx	SENSEx[2:0]		
		15:8	FILTENx	SENSEx[2:0]			FILTENx	SENSEx[2:0]		
		23:16	FILTENx	SENSEx[2:0]			FILTENx	SENSEx[2:0]		
		31:24	FILTENx	SENSEx[2:0]			FILTENx	SENSEx[2:0]		
0x20 ... 0x2F	Reserved									

Downloaded from Arrow.com.

Offset	Name	Bit Pos.								
0x30	DEBOUNCEN	7:0	DEBOUNCEN[7:0]							
		15:8								
		23:16								
		31:24								
0x34	DPRESCALER	7:0					STATES	PRESCALER[2:0]		
		15:8								
		23:16								TICKON
		31:24								
0x38	PINSTATE	7:0	PINSTATE[7:0]							
		15:8								
		23:16								
		31:24								
0x3C	NSCHK	7:0	EXTINT[7:0]							
		15:8								
		23:16								
		31:24	NMI							
0x40	NONSEC	7:0	EXTINT[7:0]							
		15:8								
		23:16								
		31:24	NMI							

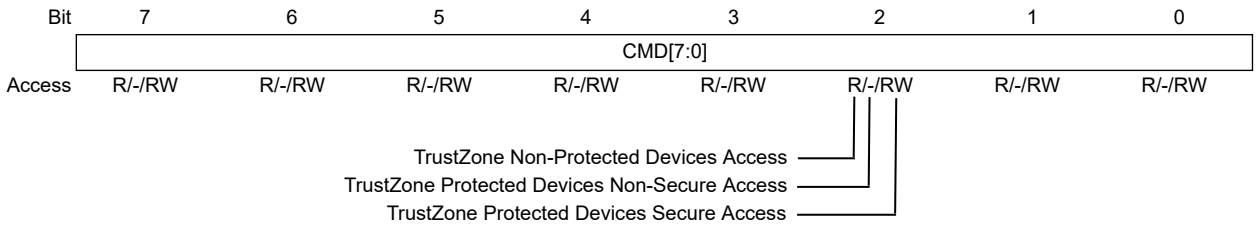
29.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

On SAM L11 devices, the Mix-Secure peripheral has different types of registers (Non-Secure, Secure, Write-Secure, Mix-Secure, and Write-Mix-Secure) with different access permissions for each bitfield. Refer to *Mix-Secure Peripherals* for more details. In the following register descriptions, the access permissions are specified as shown in the following figure.



Downloaded from Arrow.com.

29.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection, Secure

Bit	7	6	5	4	3	2	1	0
				CKSEL			ENABLE	SWRST
Access				RW/-/RW			RW/-/RW	W/-/W
Reset				0			0	0

Bit 4 – CKSEL Clock Selection
The EIC can be clocked either by GCLK_EIC (when a frequency higher than 32KHz is required for filtering) or by CLK_ULP32K (when power consumption is the priority).

This bit is not Write-Synchronized.

Value	Description
0	The EIC is clocked by GCLK_EIC.
1	The EIC is clocked by CLK_ULP32K.

Bit 1 – ENABLE Enable
Due to synchronization there is a delay between writing to CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable bit in the Synchronization Busy register will be set (SYNCBUSY.ENABLE=1). SYNCBUSY.ENABLE will be cleared when the operation is complete.

This bit is not Enable-Protected.

This bit is Write-Synchronized.

Value	Description
0	The EIC is disabled.
1	The EIC is enabled.

Bit 0 – SWRST Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the EIC to their initial state, and the EIC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write operation will be discarded.

Due to synchronization there is a delay from writing CTRLA.SWRST until the Reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the Reset is complete.

This bit is not Enable-Protected.

This bit is Write-Synchronized.

Downloaded from Arrow.com.

Value	Description
0	There is no ongoing reset operation.
1	The reset operation is ongoing.

Downloaded from Arrow.com.

29.8.2 Non-Maskable Interrupt Control

Name: NMICTRL
Offset: 0x01
Reset: 0x00
Property: PAC Write-Protection, Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the NMI interrupt is set as Non-Secure in the NONSEC register (NONSEC.NMI bit).

Bit	7	6	5	4	3	2	1	0
				NMIASYNCH	NMIFILTEN	NMISENSE[2:0]		
Access				RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset				0	0	0	0	0

Bit 4 – NMIASYNCH Non-Maskable Interrupt Asynchronous Edge Detection Mode
The NMI edge detection can be operated synchronously or asynchronously to the EIC clock.

Value	Description
0	The NMI edge detection is synchronously operated.
1	The NMI edge detection is asynchronously operated.

Bit 3 – NMIFILTEN Non-Maskable Interrupt Filter Enable

Value	Description
0	NMI filter is disabled.
1	NMI filter is enabled.

Bits 2:0 – NMISENSE[2:0] Non-Maskable Interrupt Sense Configuration
These bits define on which edge or level the NMI triggers.

Value	Name	Description
0x0	NONE	No detection
0x1	RISE	Rising-edge detection
0x2	FALL	Falling-edge detection
0x3	BOTH	Both-edge detection
0x4	HIGH	High-level detection
0x5	LOW	Low-level detection
0x6 – 0x7	-	Reserved

Downloaded from Arrow.com.

29.8.3 Non-Maskable Interrupt Flag Status and Clear

Name: NMIFLAG
Offset: 0x02
Reset: 0x0000
Property: Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the NMI interrupt is set as Non-Secure in the NONSEC register (NONSEC.NMI bit).

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								NMI
Access								RW/RW*/RW
Reset								0

Bit 0 – NMI Non-Maskable Interrupt
This flag is cleared by writing a '1' to it.

This flag is set when the NMI pin matches the NMI sense configuration, and will generate an interrupt request.

Writing a '0' to this bit has no effect.

Downloaded from Arrow.com.

29.8.4 Synchronization Busy

Name: SYNCBUSY
Offset: 0x04
Reset: 0x00000000
Property: Secure

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							ENABLE	SWRST
Access							R/-/R	R/-/R
Reset							0	0

Bit 1 – ENABLE Enable Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.ENABLE bit is complete.
1	Write synchronization for CTRLA.ENABLE bit is ongoing.

Bit 0 – SWRST Software Reset Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.SWRST bit is complete.
1	Write synchronization for CTRLA.SWRST bit is ongoing.

Downloaded from Arrow.com.

29.8.5 Event Control

Name: EVCTRL
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the external interrupt x (EXTINTx) is set as Non-Secure in the NONSEC register (NONSEC.EXTINTx bit). Some restrictions apply for the Non-Secure accesses to an Enabled-Protected register as it will not be possible for the Non-Secure to configure it once this register is enabled by the Secure application. This will require some veneers to be implemented on Secure side.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	EXTINTEO[7:0]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – EXTINTEO[7:0] External Interrupt Event Output Enable
The bit x of EXTINTEO enables the event associated with the EXTINTx pin.

Value	Description
0	Event from pin EXTINTx is disabled.
1	Event from pin EXTINTx is enabled and will be generated when EXTINTx pin matches the external interrupt sensing configuration.

Downloaded from Arrow.com.

29.8.6 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x0C
Reset: 0x00000000
Property: PAC Write-Protection, Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the external interrupt x (EXTINTx) is set as Non-Secure in the NONSEC register (NONSEC.EXTINTx bit).

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Bit	31	30	29	28	27	26	25	24
	NSCHK							
Access	RW/RW/RW							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	EXTINT[7:0]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit 31 – NSCHK Non-secure Check Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the NSCHK Interrupt Enable bit.

Bits 7:0 – EXTINT[7:0] External Interrupt Enable
The bit x of EXTINT enables the interrupt associated with the EXTINTx pin.

Writing a '0' to bit x has no effect.

Value	Description
0	The external interrupt x is disabled.
1	The external interrupt x is enabled.

Downloaded from Arrow.com.

29.8.7 Interrupt Enable Set

Name: INTENSET
Offset: 0x10
Reset: 0x00000000
Property: PAC Write-Protection, Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the external interrupt x (EXTINTx) is set as Non-Secure in the NONSEC register (NONSEC.EXTINTx bit).

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Bit	31	30	29	28	27	26	25	24
	NSCHK							
Access	RW/RW/RW							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	EXTINT[7:0]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit 31 – NSCHK Non-secure Check Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the NSCHK Interrupt Enable bit.

Bits 7:0 – EXTINT[7:0] External Interrupt Enable
The bit x of EXTINT enables the interrupt associated with the EXTINTx pin.

Writing a '0' to bit x has no effect.

Writing a '1' to bit x will set the External Interrupt Enable bit x, which enables the external interrupt EXTINTx.

Value	Description
0	The external interrupt x is disabled.
1	The external interrupt x is enabled.

Downloaded from Arrow.com.

29.8.8 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x14
Reset: 0x00000000
Property: Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the external interrupt x (EXTINTx) is set as Non-Secure in the NONSEC register (NONSEC.EXTINTx bit).

Bit	31	30	29	28	27	26	25	24
	NSCHK							
Access	RW/RW/RW							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	EXTINT[7:0]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit 31 – NSCHK Non-secure Check Interrupt
The flag is cleared by writing a '1' to it. This flag is set when write to either NONSEC and NSCHK register and if the related bit of NSCHK is enabled and the related bit of NONSEC is zero.

Bits 7:0 – EXTINT[7:0] External Interrupt
The flag bit x is cleared by writing a '1' to it.

This flag is set when EXTINTx pin matches the external interrupt sense configuration and will generate an interrupt request if 29.8.6 INTENCLR.EXTINT[x] or 29.8.7 INTENSET.EXTINT[x] is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the External Interrupt x flag.

Downloaded from Arrow.com.

29.8.9 External Interrupt Asynchronous Mode

Name: ASYNCH
Offset: 0x18
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the external interrupt x (EXTINTx) is set as Non-Secure in the NONSEC register (NONSEC.EXTINTx bit). Some restrictions apply for the Non-Secure accesses to an Enabled-Protected register as it will not be possible for the Non-Secure to configure it once this register is enabled by the Secure application. This will require some veneers to be implemented on Secure side.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ASYNCH[7:0]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – ASYNCH[7:0] Asynchronous Edge Detection Mode
The bit x of ASYNCH set the Asynchronous Edge Detection Mode for the interrupt associated with the EXTINTx pin.

Value	Description
0	The EXTINT x edge detection is synchronously operated.
1	The EXTINT x edge detection is asynchronously operated.

Downloaded from Arrow.com.

29.8.10 External Interrupt Sense Configuration

Name: CONFIG
Offset: 0x1C
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the external interrupt x (EXTINTx) is set as Non-Secure in the NONSEC register (NONSEC.EXTINTx bit). Some restrictions apply for the Non-Secure accesses to an Enabled-Protected register as it will not be possible for the Non-Secure to configure it once this register is enabled by the Secure application. This will require some veneers to be implemented on Secure side.

Bit	31	30	29	28	27	26	25	24
	FILTENx	SENSEx[2:0]			FILTENx	SENSEx[2:0]		
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	FILTENx	SENSEx[2:0]			FILTENx	SENSEx[2:0]		
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FILTENx	SENSEx[2:0]			FILTENx	SENSEx[2:0]		
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FILTENx	SENSEx[2:0]			FILTENx	SENSEx[2:0]		
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bits 3,7,11,15,19,23,27,31 – FILTENx Filter Enable x [x=7..0]

Value	Description
0	Filter is disabled for EXTINT[x] input.
1	Filter is enabled for EXTINT[x] input.

Bits 0:2,4:6,8:10,12:14,16:18,20:22,24:26,28:30 – SENSEx Input Sense Configuration x [x=7..0]
These bits define on which edge or level the interrupt or event for EXTINT[x] will be generated.

Value	Name	Description
0x0	NONE	No detection
0x1	RISE	Rising-edge detection
0x2	FALL	Falling-edge detection

Downloaded from Arrow.com.

Value	Name	Description
0x3	BOTH	Both-edge detection
0x4	HIGH	High-level detection
0x5	LOW	Low-level detection
0x6 – 0x7	-	Reserved

Downloaded from Arrow.com.

29.8.11 Debouncer Enable

Name: DEBOUNCEN
Offset: 0x30
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the external interrupt x (EXTINTx) is set as Non-Secure in the NONSEC register (NONSEC.EXTINTx bit). Some restrictions apply for the Non-Secure accesses to an Enabled-Protected register as it will not be possible for the Non-Secure to configure it once this register is enabled by the Secure application. This will require some veneers to be implemented on Secure side.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DEBOUNCEN[7:0]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – DEBOUNCEN[7:0] Debouncer Enable
The bit x of DEBOUNCEN set the Debounce mode for the interrupt associated with the EXTINTx pin.

Value	Description
0	The EXTINT x edge input is not debounced.
1	The EXTINT x edge input is debounced.

Downloaded from Arrow.com.

29.8.12 Debouncer Prescaler

Name: DPRESCALER
Offset: 0x34
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Secure

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								TICKON
Access								RW/-/RW
Reset								0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					STATES	PRESCALER[2:0]		
Access					RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW
Reset					0	0	0	0

Bit 16 – TICKON Pin Sampler frequency selection
This bit selects the clock used for the sampling of bounce during transition detection.

Value	Description
0	The bounce sampler is using GCLK_EIC.
1	The bounce sampler is using the low frequency clock.

Bit 3 – STATES Debouncer Number of States
This bit selects the number of samples by the debouncer low frequency clock needed to validate a transition from current pin state to next pin state in synchronous debouncing mode for pins EXTINT[7:0].

Value	Description
0	The number of low frequency samples is 3.
1	The number of low frequency samples is 7.

Bits 2:0 – PRESCALER[2:0] Debouncer Prescaler
These bits select the debouncer low frequency clock for pins EXTINT[7:0].

Value	Name	Description
0x0	F/2	EIC clock divided by 2
0x1	F/4	EIC clock divided by 4
0x2	F/8	EIC clock divided by 8

Downloaded from Arrow.com.

Value	Name	Description
0x3	F/16	EIC clock divided by 16
0x4	F/32	EIC clock divided by 32
0x5	F/64	EIC clock divided by 64
0x6	F/128	EIC clock divided by 128
0x7	F/256	EIC clock divided by 256

Downloaded from Arrow.com.

29.8.13 Pin State

Name: PINSTATE
Offset: 0x38
Reset: 0x00000000
Property: PAC Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, read accesses (R*) are allowed only if the external interrupt x (EXTINTx) is set as Non-Secure in the NONSEC register (NONSEC.EXTINTx bit).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	PINSTATE[7:0]							
Access	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – PINSTATE[7:0] Pin State
These bits return the valid pin state of the debounced external interrupt pin EXTINTx.

Downloaded from Arrow.com.

29.8.14 Security Attribution Check

Name: NSCHK
Offset: 0x3C
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to select one or more external pins to check their security attribution as non-secured.

Important: This register is only available for **SAM L11** and has no effect for **SAM L10**.

Bit	31	30	29	28	27	26	25	24
	NMI							
Access	RW/RW/RW							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	EXTINT[7:0]							
Access	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW
Reset	0	0	0	0	0	0	0	0

Bit 31 – NMI Non-Maskable Interrupt Security Attribution Check
This bit selects the Non-Maskable Interrupt pin for security attribution check. If the NMI bit in NONSECNMI is set to the opposite value, then the NSCHK interrupt flag will be set.

Value	Description
0	0-to-1 transition will be detected on corresponding NONSEC bit.
1	1-to-0 transition will be detected on corresponding NONSEC bit.

Bits 7:0 – EXTINT[7:0] External Interrupts Security Attribution Check
These bits select the individual pins for security attribution check. If any pin selected in NSCHK has the corresponding bit in NONSEC set to the opposite value, then the NSCHK interrupt flag will be set.

Downloaded from Arrow.com.

Value	Description
0	0-to-1 transition will be detected on corresponding NONSEC bit.
1	1-to-0 transition will be detected on corresponding NONSEC bit.

Downloaded from Arrow.com.

29.8.15 Non-secure Interrupt

Name: NONSEC
Offset: 0x40
Reset: 0x00000000
Property: PAC Write-Protection, Write-Secure

This register allows to set the NMI or external interrupt control and status registers in non-secure mode, individually per interrupt pin.

Important: This register is only available for **SAM L11** and has no effect for **SAM L10**.

Bit	31	30	29	28	27	26	25	24
	NMI							
Access	RW/R/RW							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	EXTINT[7:0]							
Access	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW
Reset	0	0	0	0	0	0	0	0

Bit 31 – NMI Non-Secure Non-Maskable Interrupt
This bit enables the non-secure mode of NMI.

The registers whose content is set in non-secure mode by NONSEC.NMI are NMICTRL and NMIFLAG registers.

Value	Description
0	NMI is secure.
1	NMI is non-secure.

Bits 7:0 – EXTINT[7:0] Non-Secure External Interrupt
The bit x of EXTINT enables the non-secure mode of EXTINTx.

The registers whose EXTINT bit or bitfield x is set in non-secure mode by NONSEC.EXTINTx are EVCTRL, ASYNCH, IDEBOUNCEN, NTENCLR, INTENSET, INTFLAG and CONFIG registers.

Downloaded from Arrow.com.

Value	Description
0	EXTINTx is secure.
1	EXTINTx is non-secure.

Downloaded from Arrow.com.

30. NVMCTRL – Nonvolatile Memory Controller

30.1 Overview

Non-Volatile Memory (NVM) is a reprogrammable Flash memory that retains program and data storage even with power off. It embeds three separate arrays namely FLASH, Data FLASH and AUX FLASH. The Data FLASH array can be programmed while reading the FLASH array. It is intended to store data while executing from the FLASH without stalling. AUX FLASH stores data needed during the device startup such as calibration and system configuration. The NVM Controller (NVMCTRL) connects to the AHB and APB bus interfaces for system access to the NVM block. The AHB interface is used for reads and writes to the NVM block, while the APB interface is used for commands and configuration.

30.2 Features

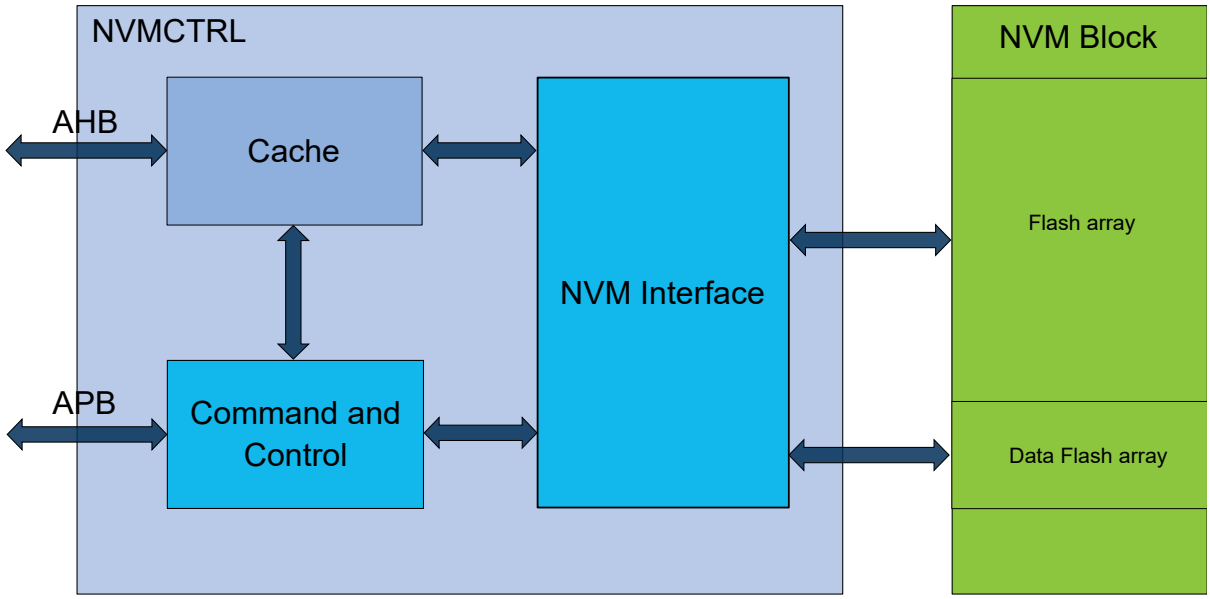
- 32-bit AHB interface for reads and writes
- Write-While-Read (WWR) Data Flash
- All NVM Sections are Memory Mapped to the AHB, Including Calibration and System Configuration
- 32-bit APB Interface for Commands and Control
- Programmable Wait States for Read Optimization
- 6 Regions can be Individually Protected or Unprotected
- Additional Protection for Bootloader
- Interface to Power Manager for Power-Down of Flash Blocks in Sleep Modes
- Can Optionally Wake-up on Exit from Sleep or on First Access
- Direct-mapped Cache
- TrustZone Support **(SAM L11)**

Note: A register with property "Enable-Protected" may contain bits that are *not* enable-protected.

Downloaded from Arrow.com.

30.3 Block Diagram

Figure 30-1. Block Diagram



30.4 Signal Description

Not applicable.

30.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described in the following sections.

30.5.1 Power Management

The NVMCTRL will continue to operate in any sleep mode where the selected source clock is running. The NVMCTRL interrupts can be used to wake up the device from sleep modes.

The Power Manager will automatically put the NVM block into a low-power state when entering sleep mode. This is based on the Control B register (CTRLB) SLEEPPRM bit setting. Refer to the 30.8.2 CTRLB.SLEEPPRM register description for more details. The NVM block goes into low-power mode automatically when the device enters STANDBY mode regardless of SLEEPPRM. The NVM Page Buffer is lost when the NVM goes into low power mode therefore a write command must be issued prior entering the NVM low power mode. NVMCTRL SLEEPPRM can be disabled to avoid such loss when the CPU goes into sleep except if the device goes into STANDBY mode for which there is no way to retain the Page Buffer.

Related Links

22. PM – Power Manager

30.5.2 Clocks

Two synchronous clocks are used by the NVMCTRL. One is provided by the AHB bus (CLK_NVMCTRL_AHB) and the other is provided by the APB bus (CLK_NVMCTRL_APB). For higher

Downloaded from Arrow.com.

system frequencies, a programmable number of wait states can be used to optimize performance. When changing the AHB bus frequency, the user must ensure that the NVM Controller is configured with the proper number of wait states. Refer to the Electrical Characteristics for the exact number of wait states to be used for a particular frequency range.

30.5.3 Interrupts

The NVM Controller interrupt request line is connected to the interrupt controller. Using the NVMCTRL interrupt requires the interrupt controller to be programmed first.

30.5.4 Events

The NVMCTRL can take the following actions on an input event:

- Write zeroes in one Data FLASH row: Refer to 30.6.7 Tamper Erase for details.
- Write a page in the FLASH or in the Data FLASH: Refer to 30.6.6 Event Automatic Write for details.

The NVMCTRL uses only asynchronous events, so the asynchronous Event System channel path must be configured. By default, the NVMCTRL will detect a rising edge on the incoming event. If the NVMCTRL action must be performed on the falling edge of the incoming event, the event line must be inverted first. This is done by setting the corresponding Event Invert Enable bit in Event Control register (NVMCTRL.AUTOWINV=1).

30.5.5 Debug Operation

When an external debugger forces the CPU into debug mode, the peripheral continues normal operation except that FLASH reads are not cached so that the cache state is not altered by debug tools.

30.5.6 Register Access Protection

All registers with write-access are optionally write-protected by the Peripheral Access Controller (PAC), except the following registers:

- Interrupt Flag Status and Clear register (INTFLAG)
- Status register (STATUS)

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

When TrustZone is supported (**SAM L11** only), all register reads are allowed. Non-secure writes to APB registers are limited as follows. Illegal writes will be ignored.

- Some commands written to CTRLA such as Write/Erase and Lock/Unlock are only permitted to non-secure application and data space.
- Writes to all other registers except CTRLC and INTFLAG are not allowed.

Related Links

15. PAC - Peripheral Access Controller

30.5.7 SAM L11 TrustZone Specific Register Access Protection

The NVMCTRL is a split-secure APB module, all registers are available in the secure alias and only a subset of registers is available in the non-secure alias with limited access.

When NONSEC.WRITE is read zero, all APB write accesses to the non-secure APB alias and all non-secure AHB write accesses to the Page Buffer are discarded. The latter returns a hardfault. Any attempt to change the configuration via the non-secure alias is silently ignored.

Downloaded from Arrow.com.

Debug Access to the bus system can be restricted to allow only accesses to non-secure regions or reject all accesses. See the section on the *NVMCTRL Debugger Access Level* for details.

Note: Refer to the *Mix-Secure Peripherals* section in the *SAM L11 Security Features* chapter for more information.

30.5.8 Analog Connections

Not applicable.

30.6 Functional Description

30.6.1 Principle of Operation

The NVM Controller is a slave on the AHB and APB buses. It responds to commands, read requests and write requests, based on user configuration.

30.6.1.1 Initialization

After power up, the NVM Controller goes through a power-up sequence. During this time, access to the NVM Controller from the AHB bus is halted. Upon power-up completion, the NVM Controller is operational without any need for user configuration.

30.6.2 Memory Organization

Refer to the Physical Memory Map for memory sizes and addresses for each device.

The NVM is organized into rows, where each row contains four pages, as shown in the NVM Row Organization figure. The NVM has a row-erase granularity, while the write granularity is by page. In other words, a single row erase will erase all four pages in the row, while four write operations are used to write the complete row.

Figure 30-2. NVM Row Organization

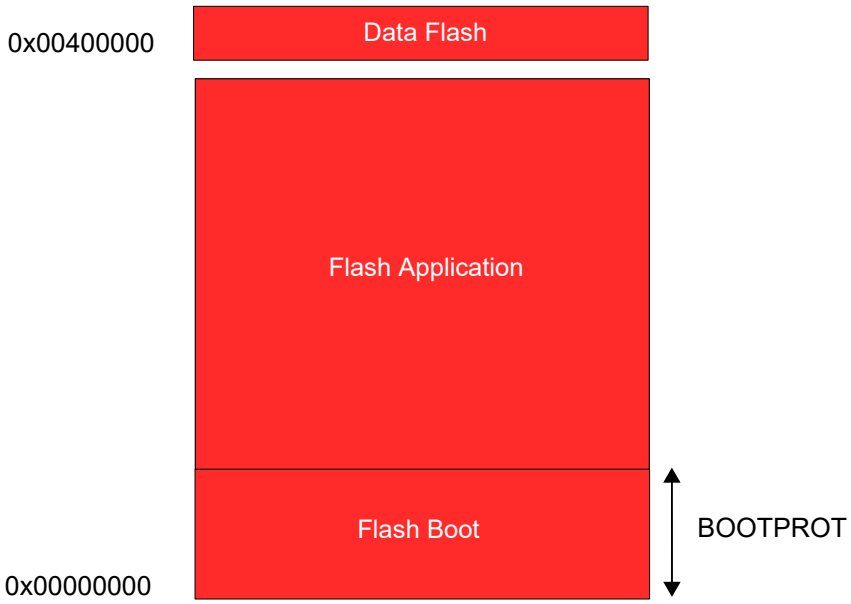
Row n	Page (n*4) + 3	Page (n*4) + 2	Page (n*4) + 1	Page (n*4) + 0

The NVM block contains the AUX FLASH which contain calibration and system configuration, the FLASH area intended to store code and a separate array dedicated to data storage called Data FLASH that can be modified while the FLASH is read (no bus stall). All these areas are memory mapped. Refer to the NVM Organization figure below for details.

The calibration and auxiliary space contains factory calibration and system configuration information. These spaces can be read from the AHB bus in the same way as the FLASH. Note that Data FLASH requires more cycles to be read. The Data FLASH are can be executable, however this is not recommended as it can weaken an application security and also affect performances.

Downloaded from Arrow.com.

Figure 30-3. NVM Memory Organization



The lower rows in the FLASH can be allocated as a boot loader section by using the BOOTPROT fuses.

The boot loader section size is defined by the BOOTPROT fuses expressed in number of rows.

Important: Refer to the Boot ROM section to get Chip Erase commands effects for this specific BOOT area.

30.6.3 Region Unlock Bits

The NVMCTRL has the ability to lock regions defined in the NVM Memory Organization figures.

When a region is locked all modify (i.e. write or erase) commands directed to these regions are discarded. When such an operation occurs a LOCKE error is reported in the INTFLAG register and can generate an interruption.

To lock or unlock a region, write a one to the bitfield corresponding to the selected regions in the SULCK and NSULCK registers with the correct key. Writes to these registers are silently discarded when the key is not correct. Writing these registers with the correct key will temporarily lock/unlock the corresponding regions. The new setting will stay in effect until the next Reset, or until the setting is changed again while writing SULCK and NSULCK. The current status of the lock can be determined by reading the SULCK and NSULCK registers. To change the default lock/unlock setting for a region, the user row of the AUX FLASH must be written using the Write Page command. Writing to the AUX FLASH user row will take effect after the next Reset. Therefore, a boot of the device is needed for changes in the lock/unlock setting to take effect. Refer to the Physical Memory Map for calibration and auxiliary space address mapping.

30.6.4 Command and Data Interface

The NVM Controller is addressable from the APB bus, while the NVM main address space is addressable from the AHB bus. Read and automatic page write operations are performed by addressing the FLASH, Data FLASH and AUX FLASH spaces directly, while other operations such as manual page writes and row erases must be performed by issuing commands through the NVM Controller.

Downloaded from Arrow.com.

To issue a command, the CTRLA.CMD bits must be written along with the CTRLA.CMDEX value. When a command is issued, STATUS.READY is cleared and rises again when the command has completed. INTFLAG.DONE is also set when a command completes. Any commands written while INTFLAG.READY is low will be ignored.

The CTRLB and CTRLC registers must be used to control the power reduction mode, read wait states, and the write mode.

30.6.4.1 NVM Read

Reading from the FLASH is performed via the AHB bus. Read data is available after the configured number of read wait states (CTRLB.RWS) set in the NVM Controller.

The number of cycles data are delayed to the AHB bus is determined by the read wait states.

Reading the NVM main address space while a programming or erase operation is ongoing on the NVM main array results in an AHB bus stall until the end of the operation. Reading the NVM main array does not stall the bus when the Data FLASH is being programmed or erased.

30.6.4.2 DATA FLASH Read

Reading from the Data FLASH is performed via the AHB bus by addressing the Data FLASH address space directly.

Read timings are increased by one cycle compared to regular FLASH read timings when access size is Byte or half-Word. The AHB data phase is twice as long in case of full-Word-size access.

It is not possible to read the Data FLASH while the NVM main array is being written or erased (the read is stalled), whereas the Data FLASH can be written or erased while the main array is being read.

The Data FLASH address space is not cached, therefore it is recommended to limit access to this area for performance and power consumption considerations.

30.6.4.3 NVM Write

Data to be written to the NVM block are first written to and stored in an internal buffer called the page buffer. The page buffer contains the same number of bytes as an NVM page. Writes to the page buffer must be 16 or 32 bits. 8-bit writes to the page buffer are not allowed and will cause a bus error.

Both FLASH and DATA FLASH share the same page buffer. Writing to the NVM block via the AHB bus is performed by a load operation to the page buffer. For each AHB bus write, the address is stored in the ADDR register. After the page buffer has been loaded with the required number of bytes, the page can be written to the array pointed by ADDR by setting CTRLA.CMD to 'Write Page' and setting the key value to CMDEX. The LOAD bit in the STATUS register indicates whether the page buffer has been loaded or not.

If the NVMCTRL is busy processing a write command (STATUS.READY=0), then the AHB bus is stalled upon AHB write until the ongoing command completes.

The NVM Controller requires that an erase must be done before programming. Rows can be individually erased by the Erase Row command to erase a row.

Automatic page writes are enabled by writing the manual write bit to zero (CTRLB.MANW=0). This will trigger a write operation to the page addressed by ADDR when the last location of the page is written.

Because the address is automatically stored in ADDR during the APB bus write operation, the last given address will be present in the ADDR register. There is no need to load the ADDR register manually, unless a different page in memory is to be written. The page buffer is automatically cleared upon a 'Write Page' command completion.

Downloaded from Arrow.com.

30.6.4.3.1 Procedure for Manual Page Writes (CTRLB.MANW=1)

The row to be written to must be erased before the write command is given.

- Write to the page buffer by addressing the NVM main address space directly
- Write the page buffer to memory: CTRL.CMD='Write Page' and CMDEX
- The READY bit in the INTFLAG register will be low while programming is in progress, and access through the AHB will be stalled

30.6.4.3.2 Procedure for Automatic Page Writes (CTRLB.MANW=0)

The row to be written to must be erased before the last write to the page buffer is performed.

Note that partially written pages must be written with a manual write.

- Write to the page buffer by addressing the NVM main address space directly.
  When the last location in the page buffer is written, the page is automatically written to NVM main address space.
- INTFLAG.READY will be zero while programming is in progress and access through the AHB will be stalled.

30.6.4.4 Page Buffer Clear

The page buffer is automatically set to all '1' after a page write is performed. If a partial page has been written and it is desired to clear the contents of the page buffer, the Page Buffer Clear command can be used.

The status of the Page Buffer is reflected by the STATUS.LOAD bitfield, when the PBC command is issued successfuly, STATUS.LOAD reads 0.

30.6.4.5 Erase Row

Before a page can be written, the row containing that page must be erased. The Erase Row command can be used to erase the desired row in the NVM (same command for FLASH, DATA FLASH and AUX FLASH). Erasing the row sets all bits to '1'. If the row resides in a region that is locked, the erase will not be performed and the Lock Error bit in the INTFLAG register (INTFLAG.LOCKE) will be set.

30.6.4.5.1 Procedure for Erase Row

- Write the address of the row to erase to ADDR. Any address within the row can be used.
- Issue an Erase Row command.

Note: The NVM Address bit field in the Address register (ADDR.ADDR) uses 16-bit addressing.

30.6.4.6 Set and Clear Power Reduction Mode

The NVM Controller and block can be taken in and out of power reduction mode through the Set and Clear Power Reduction Mode commands. When the NVM Controller and block are in power reduction mode, the Power Reduction Mode bit in the Status register (STATUS.PRM) is set.

30.6.5 NVM User Configuration

The NVM user configuration resides in the AUX FLASH user row. Refer to the Physical Memory Map of the device for calibration and auxiliary space address mapping.

The NSULCK and SULCK bitfields in the user row define respectively the NSULCK and SULCK register default value after a reset.

30.6.6 Event Automatic Write

The Event Automatic Write feature is enabled by setting EVCTRL.AUTOWEI=1. When enabled, an event input from EVSYS will trigger a page programming command. The polarity of the input event can be inverted by setting EVCTRL.AUTOWINV. The page written is addressed by the address register (ADDR)

Downloaded from Arrow.com.

and can reside in program or data memory. To use this feature, the row must be previously erased and the page buffer must contain the desired data to be written.

As the Page Buffer is lost when the NVM enters low power mode (refer to 30.5.1 Power Management) cannot be used if the device enters STANDBY mode or if the NVM uses power reduction modes.

The cache coherency is not ensured after an Event Automatic Write in a FLASH page. The FLASH region is cacheable, it is the user responsability to clear the cache after such an action. Note that the Data FLASH is not subject to cache coherency issues since it is not cacheble.

30.6.7 Tamper Erase

Tamper Erase ensures rapid overwrite on tamper of a Data FLASH row selected by SECCTRL.TEROW.

When a RTC tamper event occur while tamper erase is enabled (SECCTRL.TAMPEEN=1):

- the Tamper Erase row in data space addressed by SECCTRL.TEROW is written to zero.

This is performed using a special overwrite mechanism in the NVM block that overwrites the complete row with zero. The RTC must be configured to generate the tamper erase event.

Note: Data Flash endurance is affected by the tamper erase feature. Refer to the "NVM Reliability Characteristics" from Electrical Characteristics chapter.

30.6.8 Silent Access

When enabled (SECCTRL.SILACC=1), Silent accesses are performed when accessing one Data Flash row selected by SECCTRL.TEROW. The physical and logical size of the TEROW is divided by two to store each word of Data and its complement to reduce Data Flash reading noise.

When Silent Access is enabled, data in the selected Tamper Erase ROW must be accessed using following address mapping:

TEROW_base_address

W31	W30	W29	…	W3	W2	W1	W0	Page
Data	Data	Data	Data	Data	Data	Data	Data	page0
Data	Data	Data	Data	Data	Data	Data	Data	page1
reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved	page2
reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved	page3

TEROW_base_address + ROW size

All accesses to Reserved area are discarded and generate a bus error.

Note that the physical TEROW mapping in Data Flash is the following:

TEROW_physical_base_address

W31	W30	W29	…	W3	W2	W1	W0	Page
CompData	Data	CompData	Data	CompData	Data	CompData	Data	page0
CompData	Data	CompData	Data	CompData	Data	CompData	Data	page1

Downloaded from Arrow.com.

W31	W30	W29	…	W3	W2	W1	W0	Page
CompData	Data	CompData	Data	CompData	Data	CompData	Data	page2
CompData	Data	CompData	Data	CompData	Data	CompData	Data	page3

TEROW_physical_base_address + ROW size

30.6.9 Chip Erase

The various chip erase operations are managed by the boot ROM code. For more details, refer to the Boot ROM section.

30.6.10 Cache

The NVM Controller cache reduces the device power consumption and improves system performance when wait states are required. Only the Data FLASH area is cached. It is a direct-mapped cache that implements 64 lines of 64 bits (i.e., 512 Bytes). NVM Controller cache can be enabled by writing a '0' to the Cache Disable bit in the Control B register (CTRLB.CACHEDIS).

The cache can be configured to three different modes using the Read Mode bit group in the Control B register (CTRLB.READMODE).

The INVALL command can be issued using the Command bits in the Control A register to invalidate all cache lines (CTRLA.CMD=INVALL). Commands affecting NVM content automatically invalidate cache lines.

30.6.11 Debugger Access Level

The Debugger Access Level (DSU STATUSB.DAL) defines the access rights of a debugger connected to the device.

- 0x0 = Access to very limited features (basically only the DSU external address space)
- 0x1 = Access to all non-secure memory; can debug non-secure CPU code (**SAM L11 only**)
- 0x2 = Access to all memory; can debug secure and non-secure CPU code

DAL can be set to a lower setting using a SDAL command (CTRLA register).

Important: Issuing a SDAL command to set a higher setting for DAL will set INTFLAG.PROGE.

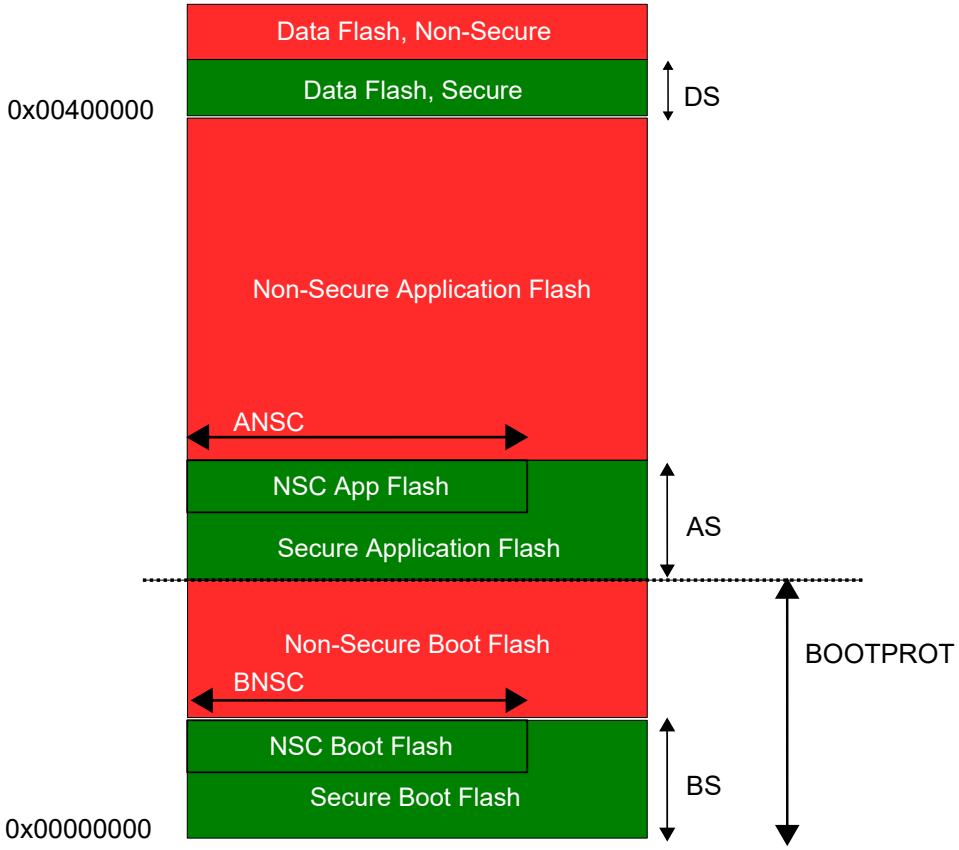
Only a Chip Erase can change DAL to a higher setting.

30.6.12 SAM L11 TrustZone Protection Considerations

On TrustZone protected devices, the FLASH and Data FLASH areas are partitioned into secure, non-secure and non-secure callable sections in order to accommodate with TrustZone core capability.

Downloaded from Arrow.com.

Figure 30-4. NVM Memory Organization



The various ranges and attributes are shown below.

Table 30-1. Memory Regions and Attributes

Memory Region	Base Address	Size	Attribute
FLASH boot secure	0x00000000	BS*ROWSIZE	Secure
FLASH boot non-secure callable	BS*ROWSIZE-BNSC*0x20	BNSC*0x20	Secure (included in Boot secure)
FLASH boot non-secure	BS*ROWSIZE	(remaining boot NVM)	Non-secure
FLASH application secure	BOOTPROT*ROWSIZE	AS*ROWSIZE	Secure
FLASH application non-secure callable	BOOTPROT*ROWSIZE-ANSC*0x20	ANSC*0x20	Secure (included in Application secure)
FLASH application non-secure	(BOOTPROT +AS)*ROWSIZE	(remaining application NVM)	Non-secure
Data FLASH secure	0x00400000	DS*ROWSIZE	Secure
Data FLASH non-secure	0x00400000 + DS*ROWSIZE	(remaining Data NVM area)	Non-secure

Access to the various sections is restricted as shown in the following table. All sections can be read and written without restriction when the access is secure. When the access is non-secure the secure sections

Downloaded from Arrow.com.

are not accessible. When defined non-secure callable sections have the same attributes as the secure sections, therefore the NVMCTRL considers them as secure regions. The system may also have a secure callable boot and application regions. These regions have the same attributes as the secure sections, so there is no special treatment needed in NVMCTRL.

Any illegal access will result in a bus error. The BOOT and Application non-secure callable regions are shown for reference but have no effect on the NVMCTRL. These regions are included in secure regions therefore the NVMCTRL considers them as secure regions.

Table 30-2. Memory Regions AHB Access Limitations

Memory Region	Secure Access	Non-Secure Access	Limitations
FLASH Boot secure	R+W	-	
FLASH Boot non-secure	R+W	R+W	
FLASH Application secure	R+W	-	
FLASH Application non-secure	R+W	R+W	
Data FLASH secure	R+W	-	
Data FLASH non-secure	R+W	R+W	
AUX FLASH Calibration Row	R+W	R	
AUX FLASH User Row (UROW)	R+W	R	
AUX FLASH Boot Configuration (BOCOR)	R+W	-	No read if BCREN is cleared.

The Boot Configuration row (BOCOR) contains information that is read by the boot ROM and written to IDAU and NVMCTRL registers. The BOCOR is read/writable if SCFGB.BCREN/BCWEN are set, respectively.

Important: SCFGB.BCREN/BCWEN are copied from BOCOR by the boot ROM.

Table 30-3. Memory Regions Modify operations Limitations (WP, EP commands)

Memory Region	Secure Access	Non-Secure Access	Limitations
FLASH Boot secure	Y	N	No if SULCK.BS=0
FLASH Boot non-secure	Y	Y	No if NSULCK.BNS=0
FLASH Application secure	Y	N	No if SULCK.AS=0

Downloaded from Arrow.com.

Memory Region	Secure Access	Non-Secure Access	Limitations
FLASH Application non-secure	Y	Y	No if NSULCK.ANS=0
Data FLASH secure	Y	N	No if SULCK.DS=0
Data FLASH non-secure	Y	Y	No if NSULCK.DNS=0
AUX FLASH User Row (UROW)	Y	N	No if BOCOR.URWEN=0
AUX FLASH Boot Configuration (BOCOR)	Y	N	No if BOCOR.BCWEN=0

The NSULCK SULCK bitfields in the user row define respectively the NSULCK and SULCK register default value after a reset.

Special care must be taken when sharing the NVMCTRL between the secure and non-secure domains. When the secure code modifies the NVM it is highly recommended that it disables all write accesses to the APB non-secure alias and writes to AHB non-secure regions by writing a 0 to NONSEC.WRITE. This avoids any interference with non-secure modify operations. Note that in this case even a secure application cannot write the page buffer at a non-secure location since the IDAU changes security attributions of Non-Secure transactions to Non-Secure regions to Non-Secure.

The NONSEC.WRITE reset value is '1', meaning that it is always possible to program a Non-Secure FLASH or Data FLASH region after a debugger probe cold-plugging. But if the debugger connects with the hot-plugging procedure then NONSEC.WRITE must be '1' to let the debugger program Non-Secure regions otherwise the transaction will cause a hardfault (seen as a DAP fault at DAP level).

For applications that don't require Non-Secure regions programming other than from a secure code, it is recommended to always disable Non-Secure writes by disabling NONSEC.WRITE. When disabled secure code needs to enable it to be able to modify Non-Secure regions following this procedure:

- disable interrupts
- write a one to NONSEC.WRITE to allow writes to the non-secure region
- write the page buffer
- write a zero to NONSEC.WRITE
- enable again the interrupts

If the NSCHK interrupt is enabled, a NONSEC.WRITE modification will generate an interrupt so that the non-secure world is aware of this change. Depending on NSCHK.WRITE INTFLAG.NSCHK will rise upon a rising or falling NONSEC.WRITE transition. The interrupt can be configured as secure or non-secure in the NVIC. If secure then a software mechanism can be implemented to call a non-secure NVMCTRL IRQ handler from the secure world.

The NVMCTRL monitors the Page Buffer write accesses and accepts only writes to non-secure regions when the transaction is non-secure. Moreover it checks that any write to the page buffer is in the same page as the previous write when the Page Buffer is not empty. When this check fails, an error is returned to the bus master that initiated the transaction. This ensures that it is not possible to mix different page writes into the Page Buffer. Therefore, any Page Buffer write access must at some point be followed by a manual or automatic Write Page (WP) that automatically clears the page buffer or a Clear Page Buffer (PBC) command.

Downloaded from Arrow.com.

For security reasons, the ADDR register is not accessible from the non-secure alias. The only way to change it is to write a data to the Page Buffer. If the intention is to issue a command that doesn't write the NVM (for instance an Erase Row command (ER)) then the PBC command must be issued to avoid locking further write accesses (even secure writes). The status of the Page Buffer is reflected by the STATUS.LOAD bitfield.

30.6.12.1 Page Buffer Clear

When Page Buffer Clear command is issued from the non-secure APB alias, ADDR must point on a non-secure region otherwise the command is silently discarded. For security reasons, the ADDR register is not accessible from the non-secure alias. The only way to change it is to write a data to the Page Buffer. If the intention is to issue a command that doesn't write the NVM (for instance an Erase Row command (ER)) then the PBC command must be issued to avoid locking further write accesses (even secure writes). ADDR must point to a non-secure NVM region when PBC is issued from the non-secure alias.

30.6.12.2 Page Write

The NVMCTRL monitors the Page Buffer write accesses and accepts only writes to non-secure regions when the transaction is non-secure. Moreover it checks that any write to the page buffer is in the same page as the previous write when the Page Buffer is not empty. When this check fails, an error is returned to the bus master that initiated the transaction. This ensures that it is not possible to mix different page writes into the Page Buffer. Therefore, any Page Buffer write access must at some point be followed by a manual or automatic Write Page (WP) that automatically clears the page buffer or a Clear Page Buffer (PBC) command.

For security reasons, the ADDR register is not accessible from the non-secure alias. The only way to change it is to write a data to the Page Buffer. If the intention is to issue a command that doesn't write the NVM (for instance an Erase Row command (ER)) then the PBC command must be issued to avoid locking further write accesses (even secure writes). The status of the Page Buffer is reflected by the STATUS.LOAD bitfield.

30.6.12.3 Erase Row

ADDR must point to a non-secure region when an ER command is issued from the non-secure APB alias.

30.6.12.4 Lock regions

The NVMCTRL has the ability to lock regions with respect to the IDAU memory mapping:

- FLASH Boot Secure and Non-Secure Callable regions
- FLASH Boot Non-Secure region
- FLASH Application secure region
- FLASH Application non-Secure and Non-Secure Callable regions
- Data FLASH Secure region
- Data FLASH Non-Secure region

When a region is locked, all modify commands (i.e. write or erase) directed to this region are discarded. A LOCKE error is reported in the INTFLAG register and can generate an interrupt.

To lock or unlock a region, write a one to the corresponding bitfield in SULCK and NSULCK registers Writes to these registers are silently discarded if the key is not correct. Writing these registers with the correct key will temporarily lock or unlock the corresponding regions. The new lock setting will stay in effect until the next reset, or until the setting is changed again while writing SULCK and NSULCK.

Note: Writes to these registers are silently discarded if the key is not correct.

The current status of the lock can be determined by reading the SULCK and NSULCK registers. To change the default lock/unlock setting for a region, the user row of the AUX FLASH must be written using

Downloaded from Arrow.com.

the Write Page command. Writing to the NVM User Row (UROW) will take effect after the next Reset. Therefore, a boot of the device is needed for changes in the lock/unlock setting to take effect. Refer to the Physical Memory Map for calibration and auxiliary space address mapping.

SULCK is a Write-Secure register:

- This register can only be written by secure masters from the secure alias
- This register is always readable by secure or non-secure masters from their respective alias

NSULCK is a Write-Mix-Secure register:

- This register can always be written by a secure master from the secure alias
- Or, by a non-secure master from the non-secure alias only if NONSEC.WRITE is set
- This register is always readable by secure or non-secure masters in their respective alias

30.6.12.5 Cache

When a line is cached, the type of transaction is stored in the cache. If the line has been updated upon a secure transaction, only secure transaction can hit, otherwise there is a cache miss and the transaction propagates to the NVMCTRL which enforces the security. If the line has been updated upon a non-secure transaction, it can be hit by both the secure or non-secure transactions. In case of a non-secure transaction cache miss, a line is replaced even if it contained a secure data.

30.6.12.6 Data Scrambling

When data scrambling is enabled (DSCC.DSCEN), data in the secure portion of the Data FLASH (rows below SCFGAD.DS) is scrambled when written and de-scrambled when read.

Scrambling and differential operation can be performed on the same data if the tamper row resides in the secure Data FLASH area and both are enabled. In this case, the process is serial starting with scrambling, followed by Silent Access on writes and the reverse on reads.

30.6.12.7 Tamper Erase

The scrambling key stored in DSCC is written to zero when a RTC tamper event occurs in addition to the erase of the row.

Downloaded from Arrow.com.

30.7 Register Summary

Important:
For SAM L11, the NVMCTRL register map is automatically duplicated in a Secure and Non-Secure alias:

- The Non-Secure alias is at the peripheral base address
- The Secure alias is located at the peripheral base address + 0x1000

Refer to *Mix-Secure Peripherals* for more information on register access rights

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0			CMD[6:0]					
		15:8	CMDEX[7:0]							
0x02 ... 0x03	Reserved									
0x04	CTRLB	7:0				RWS[3:0]				
		15:8					FWUP		SLEEPPRM[1:0]	
		23:16						CACHEDIS	READMODE[1:0]	
		31:24								
0x08	CTRLC	7:0								MANW
0x09	Reserved									
0x0A	EVCTRL	7:0							AUTOWINV	AUTOWEI
0x0B	Reserved									
0x0C	INTENCLR	7:0			NSCHK	KEYE	NVME	LOCKE	PROGE	DONE
0x0D ... 0x0F	Reserved									
0x10	INTENSET	7:0			NSCHK	KEYE	NVME	LOCKE	PROGE	DONE
0x11 ... 0x13	Reserved									
0x14	INTFLAG	7:0			NSCHK	KEYE	NVME	LOCKE	PROGE	DONE
0x15 ... 0x17	Reserved									
0x18	STATUS	7:0				DALFUSE[1:0]		READY	LOAD	PRM
		15:8								
0x1A ... 0x1B	Reserved									
0x1C	ADDR	7:0	AOFFSET[7:0]							
		15:8	AOFFSET[15:8]							
		23:16	ARRAY[1:0]							
		31:24								
0x20	SULCK	7:0						DS	AS	BS

Downloaded from Arrow.com.

Offset	Name	Bit Pos.								
		15:8	SLKEY[7:0]							
0x22	NSULCK	7:0						DNS	ANS	BNS
		15:8	NSLKEY[7:0]							
0x24	PARAM	7:0	FLASHP[7:0]							
		15:8	FLASHP[15:8]							
		23:16	DFLASHP[3:0]					PSZ[2:0]		
		31:24	DFLASHP[11:4]							
0x28 ... 0x2F	Reserved									
0x30	DSCC	7:0	DSCKEY[7:0]							
		15:8	DSCKEY[15:8]							
		23:16	DSCKEY[23:16]							
		31:24			DSCKEY[29:24]					
0x34	SECCTRL	7:0		DXN			DSCEN	SILACC		TAMPEEN
		15:8						TEROW[2:0]		
		23:16								
		31:24	KEY[7:0]							
0x38	SCFGB	7:0							BCWEN	BCREN
		15:8								
		23:16								
		31:24								
0x3C	SCFGAD	7:0								URWEN
		15:8								
		23:16								
		31:24								
0x40	NONSEC	7:0								WRITE
		15:8								
		23:16								
		31:24								
0x44	NSCHK	7:0								WRITE
		15:8								
		23:16								
		31:24								

30.8 Register Description

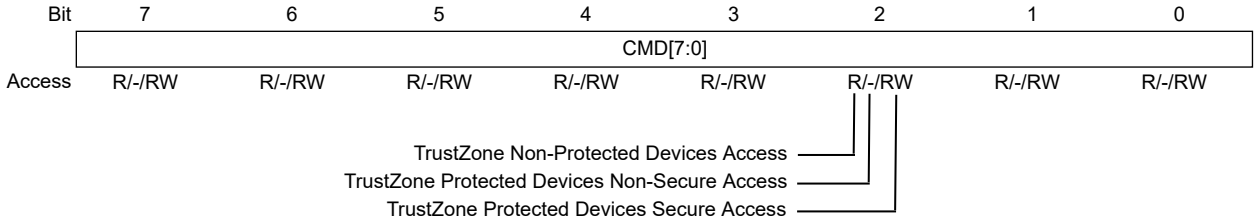
Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

Downloaded from Arrow.com.

On SAM L11 devices, the Mix-Secure peripheral has different types of registers (Non-Secure, Secure, Write-Secure, Mix-Secure, and Write-Mix-Secure) with different access permissions for each bitfield. Refer to *Mix-Secure Peripherals* for more details. In the following register descriptions, the access permissions are specified as shown in the following figure.



Downloaded from Arrow.com.

30.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x0000
Property: PAC Write-Protection, Write-Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, write accesses (W*) are allowed only if Non-Secure Write is set in the NONSEC register.

Bit	15	14	13	12	11	10	9	8
	CMDEX[7:0]							
Access	W/W*/W	W/W*/W	W/W*/W	W/W*/W	W/W*/W	W/W*/W	W/W*/W	W/W*/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		CMD[6:0]						
Access		W/W/W	W/W/W	W/W/W	W/W/W	W/W/W	W/W/W	W/W/W
Reset		0	0	0	0	0	0	0

Bits 15:8 – CMDEX[7:0] Command Execution
When this bit group is written to the key value 0xA5, the command written to CMD will be executed. If a value different from the key value is tried, the write will not be performed and the key error interrupt (INTFLAG.KEYE) will be set. PROGE is set if a previously written command is not completed yet or in case of bad conditions.

The key value must be written at the same time as CMD. If a command is issued through the APB bus on the same cycle as an AHB bus access, the AHB bus access will be given priority. The command will then be executed when the NVM block and the AHB bus are idle.

STATUS.READY must be '1' when the command has issued.

Note: The NVM Address bit field in the Address register (ADDR.ADDR) is driving the hardware (8-bit) address to the NVM when a command is executed using CMDEX.

Bits 6:0 – CMD[6:0] Command
These bits define the command to be executed when the CMDEX key is written.

Important: For **SAM L11**, only ER, WP, PBC, SDAL0 commands are available from the non-secure alias. Non-secure ER, WP, PBC are processed only if ADDR points to a non secure address, otherwise a PROGE error is issued.

Downloaded from Arrow.com.

CMD[6:0]	Group Configuration	Description
0x00-0x01	-	Reserved
0x02	ER	Erase Row - Erases the row addressed by the ADDR register in the NVM main array.
0x03	-	Reserved
0x04	WP	Write Page - Writes the contents of the page buffer to the page addressed by the ADDR register.
0x05-0x41	-	Reserved
0x42	SPRM	Sets the Power Reduction Mode.
0x43	CPRM	Clears the Power Reduction Mode.
0x44	PBC	Page Buffer Clear - Clears the page buffer.
0x45	-	Reserved
0x46	INVALL	Invalidates all cache lines.
0x47-0x4A	-	Reserved
0x4B	SDAL0	Set DAL=0
0x4C (**SAM L10**)	Reserved	Reserved
0x4C (**SAM L11**)	SDAL1	Set DAL=1
0x4D-0x7F	-	Reserved

Downloaded from Arrow.com.

30.8.2 Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000080
Property: PAC Write-Protection, Write-Secure

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						CACHEDIS	READMODE[1:0]	
Access						RW/R/RW	RW/R/RW	RW/R/RW
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
					FWUP		SLEEPPRM[1:0]	
Access					RW/R/RW		RW/R/RW	RW/R/RW
Reset					0		0	0

Bit	7	6	5	4	3	2	1	0
				RWS[3:0]				
Access				RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	
Reset				0	0	0	0	

Bit 18 – CACHEDIS Cache Disable
This bit is used to disable the cache.

Value	Description
0	The cache is enabled
1	The cache is disabled

Bits 17:16 – READMODE[1:0] NVMCTRL Read Mode

Value	Name	Description
0x0	NO_MISS_PENALTY	The NVM Controller (cache system) does not insert wait states on a cache miss. Gives the best system performance.
0x1	LOW_POWER	Reduces power consumption of the cache system, but inserts a wait state each time there is a cache miss. This mode may not be relevant if CPU performance is required, as the application will be stalled and may lead to increased run time.
0x2	DETERMINISTIC	The cache system ensures that a cache hit or miss takes the same amount of time, determined by the number of programmed Flash wait states. This mode can be used for real-time applications that require deterministic execution timings.
0x3	Reserved	

Downloaded from Arrow.com.

Bit 11 – FWUP Cache Disable
This bit is used to disable the cache.

Value	Description
0	Fast wake-up is turned off
1	Fast wake-up is turned on

Bits 9:8 – SLEEPPRM[1:0] Power Reduction Mode during Sleep
Indicates the Power Reduction Mode during sleep.

Value	Name	Description
0x0	WAKEUPACCESS	NVM block enters low-power mode when entering sleep. NVM block exits low-power mode upon first access.
0x1	WAKEUPINSTANT	NVM block enters low-power mode when entering sleep. NVM block exits low-power mode when exiting sleep.
0x2	Reserved	
0x3	DISABLED	Auto power reduction disabled.

Bits 4:1 – RWS[3:0] NVM Read Wait States
These bits control the number of wait states for a read operation. '0' indicates zero wait states, '1' indicates one wait state, etc., up to 15 wait states.
This register is initialized to 0 wait states. Software can change this value based on the NVM access time and system frequency.

Downloaded from Arrow.com.

30.8.3 Control C

Name: CTRLC
Offset: 0x08
Reset: 0x01
Property: PAC Write-Protection, Write-Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, write accesses (W*) are allowed only if Non-Secure Write is set in the NONSEC register.

Bit	7	6	5	4	3	2	1	0
								MANW
Access								RW/RW*/RW
Reset								1

Bit 0 – MANW Manual Write

Value	Description
0	Writing to the last word in the page buffer will initiate a write operation to the page addressed by the last write operation. This includes writes to FLASH, Data FLASH and AUX FLASH.
1	Write commands must be issued through the CTRLA.CMD register.

Downloaded from Arrow.com.

30.8.4 Event Control

Name: EVCTRL
Offset: 0x0A
Reset: 0x00
Property: PAC Write-Protection, Secure

Bit	7	6	5	4	3	2	1	0
							AUTOWINV	AUTOWEI
Access							RW/-/RW	RW/-/RW
Reset							0	0

Bit 1 – AUTOWINV Event Action

Value	Description
0	Input event polarity is not inverted.
1	Input event polarity is inverted.

Bit 0 – AUTOWEI Event Action

Value	Description
0	Input event has no effect.
1	Input event triggers an Automatic Page Write

Downloaded from Arrow.com.

30.8.5 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection, Write-Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, write accesses (W*) are allowed only if Non-Secure Write is set in the NONSEC register.

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Bit	7	6	5	4	3	2	1	0
			NSCHK	KEYE	NVME	LOCKE	PROGE	DONE
Access			RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset			0	0	0	0	0	0

Bit 5 – NSCHK Non-secure Check Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the NSCHK interrupt enable.

This bit will read as the current value of the NSCHK interrupt enable.

Bit 4 – KEYE Key Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the KEYE interrupt enable.

This bit will read as the current value of the KEYE interrupt enable.

Bit 3 – NVME NVM internal Error Interrupt Clear
Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the NVME interrupt enable.

This bit will read as the current value of the NVME interrupt enable.

Bit 2 – LOCKE Lock Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit sets the LOCKE interrupt enable.

This bit will read as the current value of the LOCKE interrupt enable.

Bit 1 – PROGE Programming Error Interrupt Clear
Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the PROGE interrupt enable.

This bit will read as the current value of the PROGE interrupt enable.

Downloaded from Arrow.com.

Bit 0 – DONE NVM Done Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the DONE interrupt enable.

This bit will read as the current value of the DONE interrupt enable.

Downloaded from Arrow.com.

30.8.6 Interrupt Enable Set

Name: INTENSET
Offset: 0x10
Reset: 0x00
Property: PAC Write-Protection, Write-Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, write accesses (W*) are allowed only if Non-Secure Write is set in the NONSEC register.

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Bit	7	6	5	4	3	2	1	0
			NSCHK	KEYE	NVME	LOCKE	PROGE	DONE
Access			RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset			0	0	0	0	0	0

Bit 5 – NSCHK Non-secure Check Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit sets the NSCHK interrupt enable.

This bit will read as the current value of the NSCHK interrupt enable.

Bit 4 – KEYE Key Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit sets the KEYE interrupt enable.

This bit will read as the current value of the KEYE interrupt enable.

Bit 3 – NVME NVM internal Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit sets the NVME interrupt enable.

This bit will read as the current value of the NVME interrupt enable.

Bit 2 – LOCKE Lock Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit sets the LOCKE interrupt enable.

This bit will read as the current value of the LOCKE interrupt enable.

Bit 1 – PROGE Programming Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit sets the PROGE interrupt enable.

This bit will read as the current value of the PROGE interrupt enable.

Downloaded from Arrow.com.

Bit 0 – DONE NVM Done Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit sets the DONE interrupt enable.

This bit will read as the current value of the DONE interrupt enable.

Downloaded from Arrow.com.

30.8.7 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x14
Reset: 0x00
Property: Write-Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, write accesses (W*) are allowed only if Non-Secure Write is set in the NONSEC register.

Bit	7	6	5	4	3	2	1	0
			NSCHK	KEYE	NVME	LOCKE	PROGE	DONE
Access			RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset			0	0	0	0	0	0

Bit 5 – NSCHK Non-Secure Check
This flag is set when the NONSEC register is changed and the new value differs from the NSCHK value.

This bit can be cleared by writing a '1' to its bit location.

Value	Description
0	The NONSEC configuration has not changed since last clear.
1	At least one change has been made to the NONSEC configuration since the last clear.

Bit 4 – KEYE Key Error
This flag is set when a key write-protected register has been accessed in write with a bad key. A one indicates that at least one write access has been discarded.

This bit can be cleared by writing a '1' to its bit location.

Value	Description
0	No key error occured since the last clear.
1	At least one key error occured since the last clear.

Bit 3 – NVME NVM internal Error
This flag is set on the occurrence of a NVM internal error.

This bit can be cleared by writing a '1' to its bit location.

Value	Description
0	No NVM internal error has happened since this bit was last cleared.
1	At least one NVM internal error has happened since this bit was last cleared.

Bit 2 – LOCKE Lock Error
This flag is set on the occurrence of a LOCKE error.

This bit can be cleared by writing a '1' to its bit location.

Downloaded from Arrow.com.

Value	Description
0	No programming of any locked lock region has happened since this bit was last cleared.
1	Programming of at least one locked lock region has happened since this bit was last cleared.

Bit 1 – PROGE Programming Error
This flag is set on the occurrence of a PROGE error.

This bit can be cleared by writing a '1' to its bit location.

Value	Description
0	No invalid commands or bad keywords were written in the NVM Command register since this bit was last cleared.
1	An invalid command and/or a bad keyword was/were written in the NVM Command register since this bit was last cleared.

Bit 0 – DONE NVM Command Done
This bit can be cleared by writing a one to its bit location

Value	Description
0	The NVM controller has not completed any commands since the last clear.
1	At least one command has completed since the last clear.

Downloaded from Arrow.com.

30.8.8 Status

Name: STATUS
Offset: 0x18
Reset: 0x0X00
Property: Write-Secure

Important: For **SAM L11 Non-Secure** accesses, write accesses (W*) are allowed only if Non-Secure Write is set in NONSEC register.

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				DALFUSE[1:0]		READY	LOAD	PRM
Access				R/R/R	R/R/R	R/R/R	R/R/R	R/R/R
Reset				x	x	0	0	0

Bits 4:3 – DALFUSE[1:0] DAL Fuse Value
This field is the current debugger access level fuse value.

Value	Description
0	DAL = 0 : Access to very limited features.
1	DAL = 1 (**SAM L11 only**): Access to all non-secure memory. Can debug non-secure CPU code.
2	DAL = 2 : Access to all memory. Can debug Secure and non-secure CPU code.
3	Reserved

Bit 2 – READY NVM Ready

Value	Description
0	The NVM controller is busy programming or erasing.
1	The NVM controller is ready to accept a new command.

Bit 1 – LOAD NVM Page Buffer Active Loading
This bit indicates that the NVM page buffer has been loaded with one or more words. Immediately after an NVM load has been performed, this flag is set. It remains set until a page write or a page buffer clear (PBC) command is given.

Bit 0 – PRM Power Reduction Mode
This bit indicates the current NVM power reduction state. The NVM block can be set in power reduction mode in two ways: through the command interface or automatically when entering sleep with SLEEPPRM set accordingly.

Downloaded from Arrow.com.

PRM can be cleared in three ways: through AHB access to the NVM block, through the command interface (SPRM and CPRM) or when exiting sleep with SLEEPPRM set accordingly.

Value	Description
0	NVM is not in power reduction mode.
1	NVM is in power reduction mode.

Downloaded from Arrow.com.

30.8.9 Address

Name: ADDR
Offset: 0x1C
Reset: 0x00000000
Property: PAC Write-Protection, Secure

ADDR drives the hardware address to the NVM when a command is executed using CMDEX. This is a Byte aligned address. This register is automatically updated upon AHB writes to the page buffer.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	ARRAY[1:0]							
Access	RW/-/RW	RW/-/RW						
Reset	0	0						

Bit	15	14	13	12	11	10	9	8
	AOFFSET[15:8]							
Access	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	AOFFSET[7:0]							
Access	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW
Reset	0	0	0	0	0	0	0	0

Bits 23:22 – ARRAY[1:0] Array Select

Value	Description
00	Flash
01	Data Flash
10	NVM Rows

Bits 15:0 – AOFFSET[15:0] Array Offset
Address offset

Downloaded from Arrow.com.

30.8.10 Secure Region Unlock Bits

Name: SULCK
Offset: 0x20
Reset: x initially determined from NVM User Row after reset
Property: PAC Write-Protection, Write-Secure

Important: This register is only available for **SAM L11** and has no effect for **SAM L10**.

Bit	15	14	13	12	11	10	9	8
	SLKEY[7:0]							
Access	W/-/W	W/-/W	W/-/W	W/-/W	W/-/W	W/-/W	W/-/W	W/-/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
						DS	AS	BS
Access						RW/R/RW	RW/R/RW	RW/R/RW
Reset						x	x	x

Bits 15:8 – SLKEY[7:0] Secure Unlock Key
When this bit group is written to the key value 0xA5, the write will be performed. If a value different from the key value is tried, the write will be discarded and INTFLAG.KEYE set.

Bit 2 – DS DATA Secure Unlock Bit
Default state after erase will be unlocked (0x1).

Value	Description
0	The DS region is locked.
1	The DS region is not locked.

Bit 1 – AS Application Secure Unlock Bit
Default state after erase will be unlocked (0x1).

Value	Description
0	The AS region is locked.
1	The AS region is not locked.

Bit 0 – BS BOOT Secure Unlock Bit
Default state after erase will be unlocked (0x1).

Value	Description
0	The BS region is locked.
1	The BS region is not locked.

Downloaded from Arrow.com.

30.8.11 Non-Secure Region Unlock Bits

Name: NSULCK
Offset: 0x22
Reset: x initially determined from NVM User Row after reset
Property: PAC Write-Protection, Write-Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, write accesses (W*) are allowed only if Non-Secure Write is set in the NONSEC register.

Bit	15	14	13	12	11	10	9	8
	NSLKEY[7:0]							
Access	W/W*/W	W/W*/W	W/W*/W	W/W*/W	W/W*/W	W/W*/W	W/W*/W	W/W*/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
						DNS	ANS	BNS
Access						RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset						x	x	x

Bits 15:8 – NSLKEY[7:0] Non-Secure Unlock Key
When this bit group is written to the key value 0xA5, the write will be performed. If a value different from the key value is tried, the write will be discarded and INTFLAG.KEYE set.

Bit 2 – DNS Data Non-Secure Unlock Bit
Note: For **SAM L10** devices, the Non-Secure Data Flash region corresponds to the entire Data Flash region.

Value	Description
0	The Non-Secure Data Flash region is locked.
1	The Non-Secure Data Flash region is not locked.

Bit 1 – ANS Application Non-Secure Unlock Bit
Note: For **SAM L10** devices, the Non-Secure APPLICATION region corresponds to the entire APPLICATION Flash region.

Value	Description
0	The Non-Secure APPLICATION region is locked.
1	The Non-Secure APPLICATION region is not locked.

Bit 0 – BNS BOOT Non-Secure Unlock Bit
Note: For **SAM L10** devices, the Non-Secure BOOT region corresponds to the entire BOOT Flash region.

Value	Description
0	The Non-Secure BOOT region is locked.
1	The Non-Secure BOOT region is not locked.

Downloaded from Arrow.com.

30.8.12 NVM Parameter

Name: PARAM
Offset: 0x24
Reset: 0x000XXXXX
Property: Write-Secure

Bit	31	30	29	28	27	26	25	24
	DFLASHP[11:4]							
Access	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DFLASHP[3:0]					PSZ[2:0]		
Access	R/R/R	R/R/R	R/R/R	R/R/R		R/R/R	R/R/R	R/R/R
Reset	0	0	0	0		x	x	x

Bit	15	14	13	12	11	10	9	8
	FLASHP[15:8]							
Access	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R
Reset	x	x	x	x	x	x	x	x

Bit	7	6	5	4	3	2	1	0
	FLASHP[7:0]							
Access	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R
Reset	x	x	x	x	x	x	x	x

Bits 31:20 – DFLASHP[11:0] Data FLASH area Pages
Indicates the number of pages in the Data FLASH array.

Bits 18:16 – PSZ[2:0] Page Size
Indicates the page size. Not all devices of the device families will provide all the page sizes indicated in the table.

Value	Name	Description
0x0	8	8 bytes
0x1	16	16 bytes
0x2	32	32 bytes
0x3	64	64 bytes
0x4	128	128 bytes
0x5	256	256 bytes
0x6	512	512 bytes
0x7	1024	1024 bytes

Bits 15:0 – FLASHP[15:0] FLASH Pages
Indicates the number of pages in the FLASH array.

Downloaded from Arrow.com.

30.8.13 Data Scramble Control

Name: DSCC
Offset: 0x30
Reset: 0x00000000
Property: PAC Write-Protection, Secure, Enable-Protected

Important: This register is only available for **SAM L11** and has no effect for **SAM L10**.

Bit	31	30	29	28	27	26	25	24
			DSCKEY[29:24]					
Access			W/-/W	W/-/W	W/-/W	W/-/W	W/-/W	W/-/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DSCKEY[23:16]							
Access	W/-/W	W/-/W	W/-/W	W/-/W	W/-/W	W/-/W	W/-/W	W/-/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DSCKEY[15:8]							
Access	W/-/W	W/-/W	W/-/W	W/-/W	W/-/W	W/-/W	W/-/W	W/-/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DSCKEY[7:0]							
Access	W/-/W	W/-/W	W/-/W	W/-/W	W/-/W	W/-/W	W/-/W	W/-/W
Reset	0	0	0	0	0	0	0	0

Bits 29:0 – DSCKEY[29:0] Data Scramble Key
This key value is used for data scrambling of the Secure Data Flash. After reset the key is 0. When written, the new value in the register is an XOR of the value written and the previous value of DSCC.DSCKEY.
This register is write only and will always read back as zero.

This register is Enable-Protected with SECCTRL.DSCEN meaning that it can't be modified when DSCEN=1 otherwise a PAC error is generated.

Updated DSCC.DSCKEY contents <- DSCC.DSCKEY XOR value written.

Downloaded from Arrow.com.

30.8.14 Security Control

Name: SECCTRL
Offset: 0x34
Reset: 'x' initially determined from NVM User Row after Reset
Property: PAC Write-Protection, Secure

Bit	31	30	29	28	27	26	25	24
	KEY[7:0]							
Access	W/-/W	W/-/W	W/-/W	W/-/W	W/-/W	W/-/W	W/-/W	W/-/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						TEROW[2:0]		
Access						RW/-/RW	RW/-/RW	RW/-/RW
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
		DXN			DSCEN	SILACC		TAMPEEN
Access		R/-/R			RW/-/RW	RW/-/RW		RW/-/RW
Reset		x			0	0		0

Bits 31:24 – KEY[7:0] Write Key
When this bit group is written to the key value 0xA5, the write will be performed. If a value different from the key value is tried, the write will be discarded and INTFLAG.KEYE set.

Bits 10:8 – TEROW[2:0] Tamper Erase Row
Row address of the row in data space to be erased on RTC tamper event.

Bit 6 – DXN Data eXecute Never
This bit field is only available for SAM L11 and has no effect for SAM L10.

Value	Description
0	Execution out of Data Flash is authorized.
1	Execution out of Data Flash is not authorized.

Bit 3 – DSCEN Data Scramble Enable

Important: This bitfield is only available for **SAM L11** and has no effect for **SAM L10**.

Downloaded from Arrow.com.

Value	Description
0	Secure Data FLASH is not scrambled.
1	Secure Data FLASH is scrambled.

Bit 2 – SILACC Silent Access

Value	Description
0	Data in Tamper Erase Row is not mapped as differential data.
1	Data in Tamper Erase Row is mapped as differential data.

Bit 0 – TAMPEEN Tamper Erase Enable

Value	Description
0	RTC tamper event has no effect.
1	RTC tamper event triggers a Tamper Erase.

Downloaded from Arrow.com.

30.8.15 Secure Boot Configuration

Name: SCFGB
Offset: 0x38
Reset: 0x00000003
Property: PAC Write-Protection, Write-Secure

This register is loaded from BOCOR at boot.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							BCWEN	BCREN
Access							RW/R/RW	RW/R/RW
Reset							1	1

Bit 1 – BCWEN Boot Configuration Row Write Enable

Value	Description
0	BOCOR is not writable.
1	BOCOR is writable.

Bit 0 – BCREN Boot Configuration Row Read Enable

Value	Description
0	BOCOR is not readable.
1	BOCOR is readable.

Downloaded from Arrow.com.

30.8.16 Secure Application and Data Configuration

Name: SCFGAD
Offset: 0x3C
Reset: x initially determined from NVM User Row after reset
Property: PAC Write-Protection, Write-Secure

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								URWEN
Access								RW/R/RW
Reset								x

Bit 0 – URWEN User Row Write Enable

Value	Description
0	UROW is not writable.
1	UROW is writable.

Downloaded from Arrow.com.

30.8.17 Non-secure Write Enable

Name: NONSEC
Offset: 0x40
Reset: 0x00000001
Property: PAC Write-Protection, Write-Secure

This register allows the non-secure writes to the non-secure APB alias and also non-secure AHB writes to the Page Buffer.

Important: This register is only available for **SAM L11** and has no effect for **SAM L10**.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								WRITE
Access								RW/R/RW
Reset								1

Bit 0 – WRITE Non-secure Write Enable
Non-secure APB alias write enable, non-secure AHB writes to non-secure regions enable

Value	Description
0	The non-secure APB alias is not writable, AHB secure or non-secure writes to non-secure regions (Page Buffer) return a hardfault.
1	No restriction.

Downloaded from Arrow.com.

30.8.18 Non-secure Write Enable Check

Name: NSCHK
Offset: 0x44
Reset: 0x00000001
Property: PAC Write-Protection

Important: This register is only available for **SAM L11** and has no effect for **SAM L10**.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								WRITE
Access								RW/RW/RW
Reset								1

Bit 0 – WRITE Non-secure Write Transition Select
This bitfield selects whether to generate a NSCHK interrupt on a NONSEC.WRITE rising of falling transition.

Value	Description
0	INTFLAG.NSCHK rises if NONSEC.WRITE transitions from 0 to 1.
1	INTFLAG.NSCHK rises if NONSEC.WRITE transitions from 1 to 0.

Downloaded from Arrow.com.

31. TRAM - TrustRAM

31.1 Overview

The TrustRAM (TRAM) is the controller interface for a 256-byte security RAM. This RAM is intended for volatile secret data. The TRAM is capable of performing address map scrambling as well as data scrambling for both write and read access to the secure RAM. The TRAM can perform silent access of the data stream to prevent side-channel attacks.

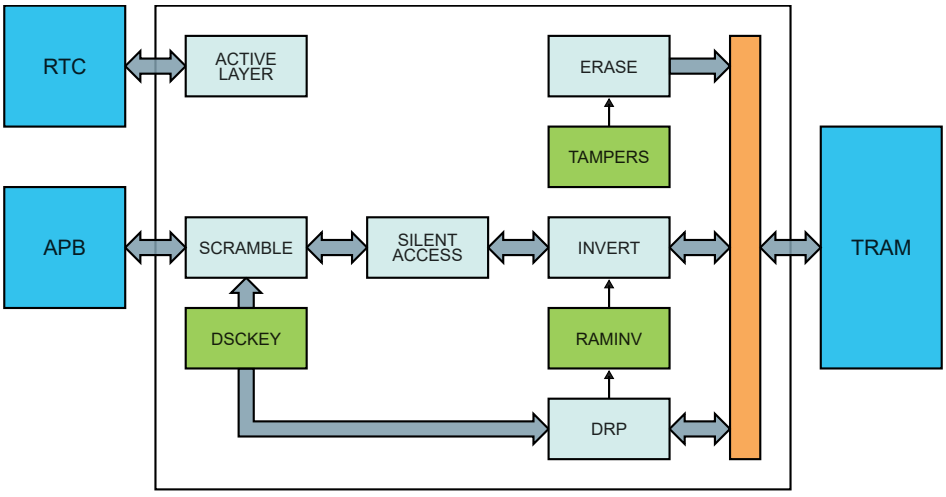
The TRAM can execute two automated tasks that are triggered by external events: remanence prevention and erase. When a remanence periodic event occurs, the physical data stored in the RAM is inverted in order to prevent physical “burn-in” signatures. When a tamper event occurs, the TRAM executes a full erase of the control signals as well as the data in the security RAM. Both automated tasks do not require CPU interaction and can be performed in all sleep modes.

31.2 Features

- Address scrambling to the RAM
- Data scrambling to/from the RAM
- Silent access of data for improved side-channel protection
- Data remanence prevention
- Active shielding to prevent tamper on physical RAM
- Full erasure of scramble key and RAM data on tamper detection

31.3 Block Diagram

Figure 31-1. Block Diagram



31.4 Signal Description

Not applicable.

Downloaded from Arrow.com.

31.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

31.5.1 I/O Lines

Not applicable.

31.5.2 Power Management

The TRAM will continue to operate in any sleep mode, as long as its source clock is running. The TRAM interrupts can be used to wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes. Refer *PM – Power Manager* for details on the different sleep modes.

31.5.3 Clocks

The TRAM bus clock (CLK_TRAM_AHB) can be enabled and disabled by the Main Clock module, and the default state of CLK_TRAM_AHB can be found in the *Peripheral Clock Masking* section.

31.5.4 DMA

Not applicable

31.5.5 Interrupts

The interrupt request lines are connected to the interrupt controller. Using the TRAM interrupts requires the interrupt controller to be configured first. Refer to NVIC - Nested Interrupt *Nested Vector Interrupt Controller* for details.

31.5.6 Events

Data Remanence Prevention and Tamper events are connected directly from the RTC to the TRAM, without going through the Event System.

31.5.7 Debug Operation

Not applicable.

31.5.8 Register Access Protection

All registers with write-access are optionally write-protected by the Peripheral Access Controller (PAC), except for the following registers:

- Interrupt Flag (INTFLAG) register
- Data Scramble Control (DSCC) register
- Permutation Write (PERMW) register
- All RAM (RAM[0:63]) addresses

Write-protection is denoted by the Write-Protected property in the register description.

Write-protection does not apply to accesses through an external debugger. Refer to the Peripheral Access Controller chapter for details.

31.5.9 SAM L11 TrustZone Specific Register Access Protection

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:

Downloaded from Arrow.com.

  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

31.6 Functional Description

31.6.1 Principle of Operation

System bus transactions from the CPU to the security RAM undergoes a scrambling routine. Both address and data buses information are modified through an algorithm determined by a scrambling key. This is performed on both write and read transactions.

31.6.2 Basic Operation

31.6.2.1 Initialization

The following bits are enable-protected, meaning that they can only be written when the TRAM is disabled (CTRLA.ENABLE is zero):

- Tamper Erase bit in the Control A register (CTRLA.TAMPERS)
- Data Remanence Protection bit in the Control A register (CTRLA.DRP)
- Silent Access bit in the Control A register (CTRLA.SILACC)

The following register is enable-protected:

- Data Scramble Control register (DSCC)

Enable-protected bits in the CTRLA register can be written at the same time as CTRLA.ENABLE is written to one, but not at the same time as CTRLA.ENABLE is written to zero.

Enable-protection is denoted by the Enable-Protected property in the register description.

31.6.2.2 Enabling, Disabling and Resetting

The TRAM is enabled by writing a one to the Enable bit in the Control A register (CTRLA.ENABLE). The TRAM is disabled by writing a zero to CTRLA.ENABLE.

The TRAM is reset by writing a one to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the TRAM will be reset to their initial state, and the TRAM will be disabled. All data in the secure RAM will be cleared to ‘0’.

31.6.2.3 Scrambling

The Data Scramble Control (DSCC) must be configured before the CTRLA.ENABLE is set. These settings cannot be changed while the module is enabled.

The scrambling logic is enabled by writing one to the enable bit in the Data Scramble Control register (DSCC.DSCEN). Scrambling is disabled by writing a zero to DSCC.DSCEN. Writing a zero to CTRLA.ENABLE will also disable the scrambling, but will not clear the DSCC.DSCEN bit.

31.6.2.4 Silent Access

Silent access bit (CTRLA.SILACC) must be configured before CTRLA.ENABLE is set. This setting cannot be changed while the module is enabled. When this mode is enabled, only 128 bytes of the security RAM are accessible since the other 128 bytes are reserved to store the 1's complement (bitwise invert) values. The physical access to the RAM is now twice as wide compared to the bus access. Therefore, only 8-bit

Downloaded from Arrow.com.

byte access and 16-bit half-word access are supported in this mode. 32-bit word write accesses are ignored and 32-bit word read accesses return 0.

The TRAM executes the following protocols:

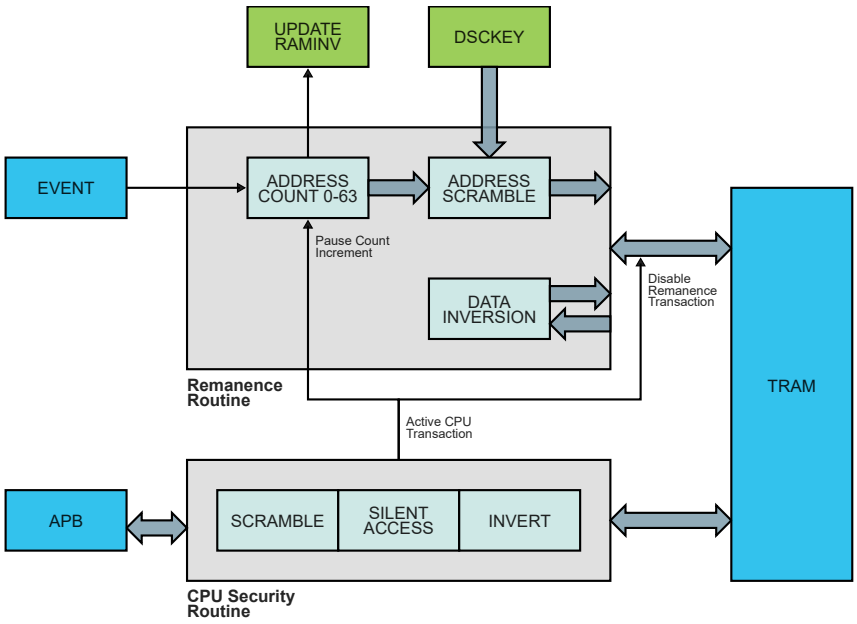
- When the CPU writes to the security RAM, the data and its bitwise invert are stored into the RAM.
- When the CPU reads from the security RAM, both the data and its bitwise invert are retrieved from the RAM. If the TRAM cannot verify that both values complement each other, a bus error is returned.

31.6.2.5 Data Remanence Prevention

Data remanence prevention bit (CTRLA.DRP) must be configured before CTRLA.ENABLE is set. This setting cannot be changed while the module is enabled. When this feature is enabled, the RTC Periodic Interval Daily Event (RTC_PERD) will trigger the automated data remanence routine. An internal counter will count from 0 to 63 and serves as the address access bus to the security RAM. For every address iteration, the TRAM reads the word data from the security RAM, inverts the value and writes back to the same address. To prevent linear access to the security RAM, the remanence address value is scrambled using the same protocols as a CPU address scramble. After remanence has updated all address locations, the routine will end by toggling the RAM inversion status bit (STATUS.RAMINV). See figure.

Data remanence is a low-priority routine. If the CPU attempts to access the security RAM while remanence is active, the routine is temporarily paused until the CPU access is completed. If a tamper full erase event is detected, the remanence routine is aborted and the internal address counter will reset to 0.

Figure 31-2. Remanence Routine



31.6.2.6 Tamper Full Erase

Tamper full erase bit (CTRLA.TAMPERS) must be configured before CTRLA.ENABLE is set. This setting cannot be changed while the module is enabled. When this feature is enabled, the RTC Tamper Event (RTC_TAMPER) will trigger the full erase equivalent to a TRAM software reset and the reset of the Data Scramble Key (DSCC.DSCKEY) register. All TRAM registers are reverted to the default reset value. Data inside the security RAM is written to ‘0’ for all address locations.

Downloaded from Arrow.com.

The tamper full erase routine operates at the highest priority. If a remanence routine executing when a tamper full erase occurs, the remanence routine is immediately terminated. If the CPU attempts to write a new scramble key at the same time the tamper key erase routine is active, the CPU data is ignored, but no bus error will occur. If a CPU security routine access is requested during a tamper full erase, the CPU transaction will be ignored and treated as a bus error similar to accessing the module during a software reset.

Important: In STANDBY low power mode, it is mandatory to enable the dynamic power gating feature (STDBYCFG.DPGPDSW) to ensure TrustRAM erasing when the power domain PDSW is in a retention state.

31.6.3 Interrupts

The TRAM has the following interrupt sources:

- Data Remanence Prevention (DRP): Indicates that the data remanence prevention routine has ended.
- Data Read Error (ERR): Indicates when there is a RAM readout error.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a one to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a one to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the TRAM is reset. See 22.8.6 INTFLAG for details on how to clear interrupt flags. All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. Refer to *Nested Vector Interrupt Controller* for details. The user must read the INTFLAG register to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated. Refer to *Nested Vector Interrupt Controller* for details.

31.6.4 Sleep Mode Operation

The TRAM continues to operate during sleep. When it receives events from the Event System, it will request its own clock in order to perform the requested operation.

An interrupt request will be generated after the wake-up if the Interrupt Controller is configured accordingly. Otherwise the CPU will wake up directly, without triggering an interrupt. In this case, the CPU will continue executing from the instruction following the entry into sleep.

The periodic events can also wake up the CPU through the interrupt function of the Event System. In this case, the event must be enabled and connected to an event channel with its interrupt enabled. See *EVSYS – Event System* for more information.

31.6.5 Synchronization

Due to the asynchronicity between event sources and CLK_TRAM_APB some registers must be synchronized when accessed. A register can require:

- Synchronization when written

Downloaded from Arrow.com.

- No synchronization

When executing an operation that requires synchronization, the corresponding status bit in the Synchronization Busy register (SYNCBUSY.xxx) will be set immediately, and cleared when synchronization is complete.

If an operation that requires synchronization is executed while SYNCBUSY.xxx is one, the operation is discarded and an error is generated. The following bits need synchronization when written:

- Software Reset bit in Control A register (CTRLA.SWRST)
- Enable bit in Control A register (CTRLA.ENABLE)

Downloaded from Arrow.com.

31.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	SILACC	DRP		TAMPERS			ENABLE	SWRST
0x01 ... 0x03	Reserved									
0x04	INTENCLR	7:0							DRP	ERR
0x05	INTENSET	7:0							DRP	ERR
0x06	INTFLAG	7:0							DRP	ERR
0x07	STATUS	7:0							DRP	RAMINV
0x08	SYNCBUSY	7:0							ENABLE	SWRST
		15:8								
		23:16								
		31:24								
0x0C	DSCC	7:0	DSCKEY[7:0]							
		15:8	DSCKEY[15:8]							
		23:16	DSCKEY[23:16]							
		31:24	DSCEN		DSCKEY[29:24]					
0x10	PERMW	7:0						DATA[2:0]		
0x11	PERMR	7:0						DATA[2:0]		
0x12 ... 0xFF	Reserved									
0x0100	RAM0	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0104	RAM1	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0108	RAM2	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x010C	RAM3	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0110	RAM4	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0114	RAM5	7:0	DATA[7:0]							
		15:8	DATA[15:8]							

Downloaded from Arrow.com.

Offset	Name	Bit Pos.								
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0118	RAM6	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x011C	RAM7	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0120	RAM8	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0124	RAM9	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0128	RAM10	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x012C	RAM11	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0130	RAM12	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0134	RAM13	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0138	RAM14	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x013C	RAM15	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0140	RAM16	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							

Downloaded from Arrow.com.

Offset	Name	Bit Pos.								
0x0144	RAM17	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0148	RAM18	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x014C	RAM19	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0150	RAM20	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0154	RAM21	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0158	RAM22	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x015C	RAM23	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0160	RAM24	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0164	RAM25	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0168	RAM26	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x016C	RAM27	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0170	RAM28	7:0	DATA[7:0]							
		15:8	DATA[15:8]							

Downloaded from Arrow.com.

Offset	Name	Bit Pos.								
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0174	RAM29	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0178	RAM30	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x017C	RAM31	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0180	RAM32	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0184	RAM33	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0188	RAM34	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x018C	RAM35	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0190	RAM36	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0194	RAM37	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0198	RAM38	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x019C	RAM39	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							

Downloaded from Arrow.com.

Offset	Name	Bit Pos.								
0x01A0	RAM40	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x01A4	RAM41	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x01A8	RAM42	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x01AC	RAM43	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x01B0	RAM44	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x01B4	RAM45	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x01B8	RAM46	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x01BC	RAM47	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x01C0	RAM48	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x01C4	RAM49	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x01C8	RAM50	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x01CC	RAM51	7:0	DATA[7:0]							
		15:8	DATA[15:8]							

Downloaded from Arrow.com.

Offset	Name	Bit Pos.								
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x01D0	RAM52	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x01D4	RAM53	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x01D8	RAM54	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x01DC	RAM55	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x01E0	RAM56	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x01E4	RAM57	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x01E8	RAM58	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x01EC	RAM59	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x01F0	RAM60	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x01F4	RAM61	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x01F8	RAM62	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							

Downloaded from Arrow.com.

Offset	Name	Bit Pos.								
0x01FC	RAM63	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							

31.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

Refer to *PAC - Peripheral Access Controller* and 39.6.6 Synchronization for details.

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

Downloaded from Arrow.com.

31.8.1 Control A

Name: CTRLA
Offset: 0x000
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	SILACC	DRP		TAMPERS			ENABLE	SWRST
Access	R/W	R/W		R/W			R/W	R/W
Reset	0	0		0			0	0

Bit 7 – SILACC Silent Access
Enables differential storage of data.

Value	Description
0	Silent access is disabled.
1	Silent access is enabled.

Bit 6 – DRP Data Remanence Prevention
Enables periodic DRP in TrustRAM.

Value	Description
0	Data remanence prevention is disabled.
1	Data remanence prevention is enabled.

Bit 4 – TAMPERS Tamper Erase
Auto-erases TrustRAM and DSCC.DSCKEY on tamper event.

Value	Description
0	Tamper erase is disabled.
1	Tamper erase is enabled.

Bit 1 – ENABLE Enable

Value	Description
0	The TRAM is disabled.
1	The TRAM is enabled.

Bit 0 – SWRST Software Reset
Writing a zero to this bit has no effect.

Writing a one to this bit resets all registers in the TRAM to their initial state, and the TRAM will be disabled. This bit can also be set via hardware when a tamper occurs while CTRLA.TAMPERS is set.

Writing a one to CTRLA.SWRST will always take precedence, meaning that all other writes in the same writeoperation will be discarded.

Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

Downloaded from Arrow.com.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

Downloaded from Arrow.com.

31.8.2 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x004
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
							DRP	ERR
Access							R/W	R/W
Reset							0	0

Bit 1 – DRP Data Remanence Prevention Complete Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Data Remanence Prevention Complete Interrupt Enable bit, which disables the data remanence prevention complete interrupt.

Value	Description
0	Data remanence prevention complete interrupt is disabled.
1	Data remanence prevention complete interrupt is enabled.

Bit 0 – ERR TrustRAM Read Error Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the TrustRAM Read Error Interrupt Enable bit, which disables the TrustRAM read error interrupt.

Value	Description
0	TrustRAM read error interrupt is disabled.
1	TrustRAM read error interrupt is enabled.

Downloaded from Arrow.com.

31.8.3 Interrupt Enable Set

Name: INTENSET
Offset: 0x005
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
							DRP	ERR
Access							R/W	R/W
Reset							0	0

Bit 1 – DRP Data Remanence Prevention Complete Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the Data Remanence Prevention Complete Interrupt Enable bit, which enables the data remanence prevention complete interrupt.

Value	Description
0	Data remanence prevention complete interrupt is disabled.
1	Data remanence prevention complete interrupt is enabled.

Bit 0 – ERR TrustRAM Read Error Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the TrustRAM Read Error Interrupt Enable bit, which enables the TrustRAM read error interrupt.

Value	Description
0	TrustRAM read error interrupt is disabled.
1	TrustRAM read error interrupt is enabled.

Downloaded from Arrow.com.

31.8.4 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x006
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
							DRP	ERR
Access							R/W	R/W
Reset							0	0

Bit 1 – DRP Data Remanence Prevention Complete Interrupt
This flag is set when the data remanence prevention routine has completed, and an interrupt request will be generated if INTENCLR.DRP/INTENSET.DRP is one.

Writing a zero to this bit has no effect.

Writing a one to this bit clears the data remanence prevention complete interrupt flag.

Value	Description
0	Data remanence prevention complete interrupt is disabled.
1	Data remanence prevention complete interrupt is enabled.

Bit 0 – ERR TrustRAM Read Error Interrupt
This flag is set when an error is detected in the TrustRAM readout, and an interrupt request will be generated if INTENCLR.ERR/INTENSET.ERR is one.

Writing a zero to this bit has no effect.

Writing a one to this bit clears the TrustRAM read error interrupt flag.

Value	Description
0	TrustRAM read error interrupt is disabled.
1	TrustRAM read error interrupt is enabled.

Downloaded from Arrow.com.

31.8.5 Status

Name: STATUS
Offset: 0x007
Reset: 0x00
Property: Read-Only

Bit	7	6	5	4	3	2	1	0
							DRP	RAMINV
Access							R	R
Reset							0	0

Bit 1 – DRP Data Remanence Prevention Routine
This bit identifies if the data remanence prevention routine is running.

Value	Description
0	The data remanence prevention routine is not running.
1	The data remanence prevention routine is running.

Bit 0 – RAMINV RAM Inversion Bit
This bit identifies if the TrustRAM bit values are inverted.

Value	Description
0	The TrustRAM physical bit information is normal.
1	The TrustRAM physical bit information is inverted.

Downloaded from Arrow.com.

31.8.6 Synchronization Busy

Name: SYNCBUSY
Offset: 0x008
Reset: 0x00000000
Property: Read-Only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							ENABLE	SWRST
Access							R	R
Reset							0	0

Bit 1 – ENABLE Enable

Value	Description
0	Write synchronization for CTRLA.ENABLE bit is complete.
1	Write synchronization for CTRLA.ENABLE bit is ongoing.

Bit 0 – SWRST Software Reset Synchronization Busy Status
This bit will set in two ways:

- Writing ‘1’ to CTRLA.SWRST
- A tamper event occurs when CTRLA.TAMPERS = ‘1’

Value	Description
0	Write synchronization for CTRLA.SWRST bit is complete.
1	Write synchronization for CTRLA.SWRST bit is ongoing.

Downloaded from Arrow.com.

31.8.7 Data Scramble Control

Name: DSCC
Offset: 0x00C
Reset: 0x00000000
Property: PAC Write-Protected, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	DSCEN		DSCKEY[29:24]					
Access	R/W		W	W	W	W	W	W
Reset	0		0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DSCKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DSCKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DSCKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit 31 – DSCEN Data Scramble Enable

Value	Description
0	TrustRAM is not scrambled.
1	TrustRAM is scrambled.

Bits 29:0 – DSCKEY[29:0] Data Scramble Key
The key value used for data scrambling. Any value written to this field is XOR'ed with the previous data. Writing '1' to CTRLA.SWRST will reset this field to 0. These bits will always return zero when read.

Downloaded from Arrow.com.

31.8.8 Permutation Write

Name: PERMW
Offset: 0x010
Reset: 0x00
Property: PAC Write-Protected

Bit	7	6	5	4	3	2	1	0
						DATA[2:0]		
Access						W	W	W
Reset						0	0	0

Bits 2:0 – DATA[2:0] Permutation Write Data
Data is the input value for the scrambler permutation function:

PERMR.DATA = Permutate(PERMW.DATA, DSCC.DSCKEY)

These bits will always return zero when read.

Downloaded from Arrow.com.

31.8.9 Permutation Read

Name: PERMR
Offset: 0x011
Reset: 0x00
Property: Read-Only

Bit	7	6	5	4	3	2	1	0
						DATA[2:0]		
Access						R	R	R
Reset						0	0	0

Bits 2:0 – DATA[2:0] Permutation Write Data
Data is the input value for the scrambler permutation function:

PERMR.DATA = Permutate(PERMW.DATA, DSCC.DSCKEY)

Downloaded from Arrow.com.

31.8.10 Security RAM n

Name: RAM
Offset: 0x0100 + n*0x04 [n=0..63]
Reset: 0x00000000
Property: PAC Write-Protected, Enable-Protected

Access to the Security RAM is only permitted when CTRLA.ENABLE=1.

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0] RAM Data

Downloaded from Arrow.com.

32. PORT - I/O Pin Controller

32.1 Overview

The IO Pin Controller (PORT) controls the I/O pins of the device. The I/O pins are organized in a series of groups, collectively referred to as a PORT group. Each PORT group can have up to 32 pins that can be configured and controlled individually or as a group. The number of PORT groups on a device may depend on the package/number of pins. Each pin may either be used for general-purpose I/O under direct application control or be assigned to an embedded device peripheral. When used for general-purpose I/O, each pin can be configured as input or output, with highly configurable driver and pull settings. Each pin can be defined as secured or non-secured, where secured pins can only be handled by secure accesses.

All I/O pins have true read-modify-write functionality when used for general-purpose I/O; the direction or the output value of one or more pins may be changed (set, reset or toggled) explicitly without unintentionally changing the state of any other pins in the same port group by a single, atomic 8-, 16- or 32-bit write.

The PORT is connected to the high-speed bus matrix through an AHB/APB bridge. The Pin Direction, Data Output Value and Data Input Value registers may also be accessed using the low-latency CPU local bus (IOBUS; ARM® single-cycle I/O port).

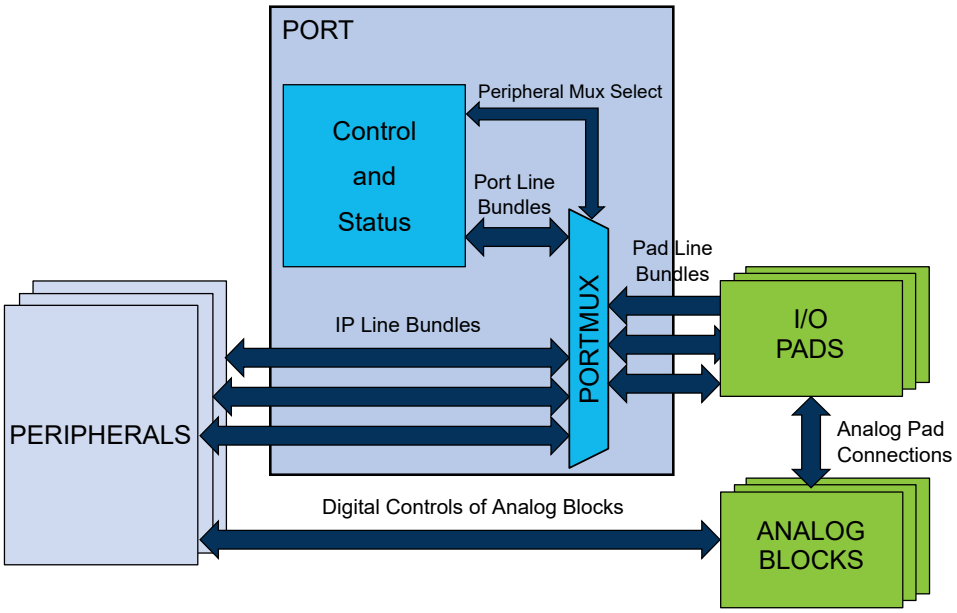
32.2 Features

- Selectable input and output configuration for each individual pin
- Software-controlled multiplexing of peripheral functions on I/O pins
- Flexible pin configuration through a dedicated Pin Configuration register
- Configurable output driver and pull settings:
  - Totem-pole (push-pull)
  - Pull configuration
  - Driver strength
- Configurable input buffer and pull settings:
  - Internal pull-up or pull-down
  - Input sampling criteria
  - Input buffer can be disabled if not needed for lower power consumption
- Input event:
  - Up to four input event pins for each PORT group
  - SET/CLEAR/TOGGLE event actions for each event input on output value of a pin
  - Can be output to pin
- Selectable secured or non-secured attribution for each individual pin **(SAM L11)**

Downloaded from Arrow.com.

32.3 Block Diagram

Figure 32-1. PORT Block Diagram



32.4 Signal Description

Table 32-1. Signal description for PORT

Signal name	Type	Description
Pxy	Digital I/O	General-purpose I/O pin y in group x

Refer to the *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

32.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly as following.

32.5.1 I/O Lines

The I/O lines of the PORT are mapped to pins of the physical device. The following naming scheme is used:

Each line bundle with up to 32 lines is assigned an identifier 'xy', with letter x=A, B, C… and two-digit number y=00, 01, …31. Examples: A24, C03.

PORT pins are labeled 'Pxy' accordingly, for example PA24, PC03. This identifies each pin in the device uniquely.

Each pin may be controlled by one or more peripheral multiplexer settings, which allow the pad to be routed internally to a dedicated peripheral function. When the setting is enabled, the selected peripheral has control over the output state of the pad, as well as the ability to read the current physical pad state. Refer to *I/O Multiplexing and Considerations* for details.

Downloaded from Arrow.com.

Each pin may be secured or non-secured, with secured pins only accessible by secure accesses.

Device-specific configurations may cause some lines (and the corresponding Pxy pin) not to be implemented.

32.5.2 Power Management

During Reset, all PORT lines are configured as inputs with input buffers, output buffers and pull disabled.

The PORT peripheral will continue operating in any sleep mode where its source clock is running.

32.5.3 Clocks

The PORT bus clock (CLK_PORT_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_PORT_APB can be found in the *Peripheral Clock Masking* section in *MCLK – Main Clock*.

The PORT requires an APB clock, which may be divided from the CPU main clock and allows the CPU to access the registers of PORT through the high-speed matrix and the AHB/APB bridge.

The priority of IOBUS accesses is higher than APB accesses. One clock cycle latency can be observed on the APB access in case of concurrent PORT accesses.

Related Links

19. MCLK – Main Clock

32.5.4 DMA

Not applicable.

32.5.5 Interrupts

Not applicable.

32.5.6 Events

The events of this peripheral are connected to the Event System.

Related Links

33. EVSYS – Event System

32.5.7 Debug Operation

When the CPU is halted in debug mode, this peripheral will continue normal operation.

32.5.8 Register Access Protection

All registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC).

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

Write-protection does not apply for accesses through an external debugger.

Related Links

15. PAC - Peripheral Access Controller

32.5.9 Analog Connections

Analog functions are connected directly between the analog blocks and the I/O pads using analog buses. However, selecting an analog peripheral function for a given pin will disable the corresponding digital features of the pad.

Downloaded from Arrow.com.

32.5.10 CPU Local Bus

The CPU local bus (IOBUS) is an interface that connects the CPU directly to the PORT. It is a single-cycle bus interface, which does not support wait states. It supports 8-bit, 16-bit and 32-bit sizes.

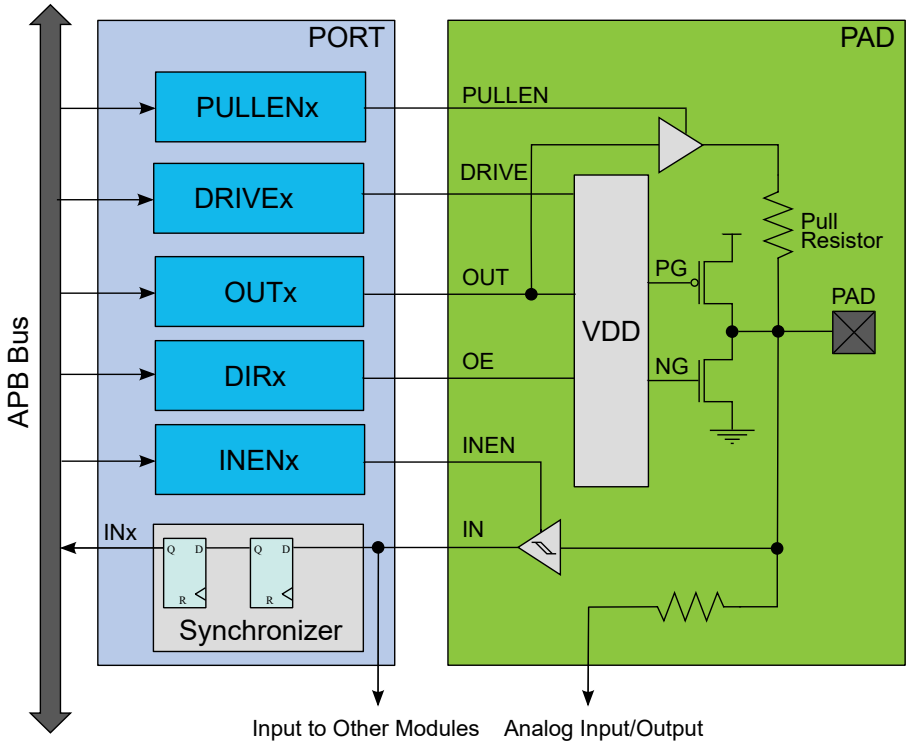
This bus is generally used for low latency operation. The Data Direction (DIR) and Data Output Value (OUT) registers can be read, written, set, cleared or be toggled using this bus, and the Data Input Value (IN) registers can be read.

Since the IOBUS cannot wait for IN register resynchronization, the Control register (CTRL) must be configured to continuous sampling of all pins that need to be read via the IOBUS in order to prevent stale data from being read.

Note: Refer to the *Product Mapping* chapter for the IOBUS address.

32.6 Functional Description

Figure 32-2. Overview of the PORT

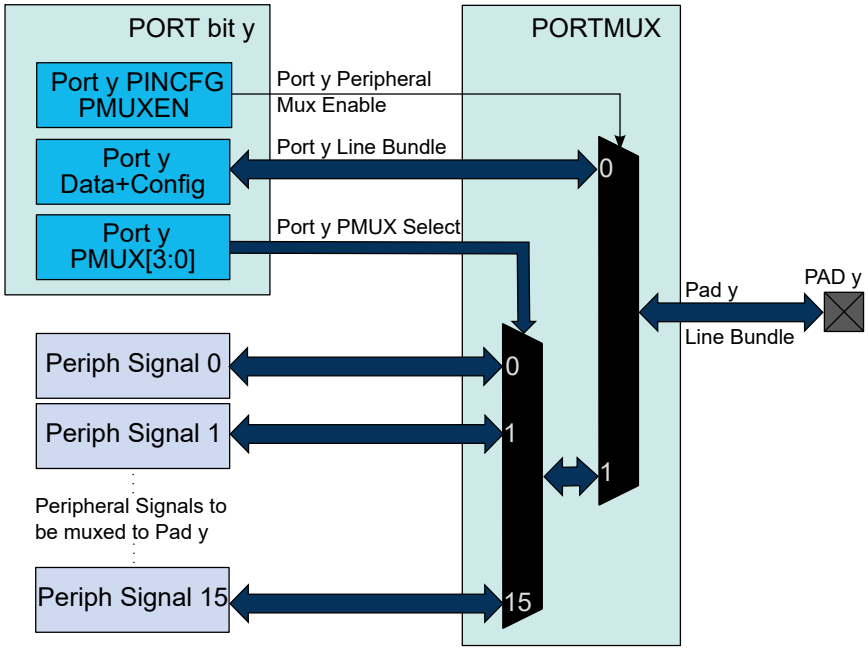


32.6.1 Principle of Operation

Each PORT group of up to 32 pins is controlled by the registers in PORT, as described in the figure. These registers in PORT are duplicated for each PORT group, with increasing base addresses. The number of PORT groups may depend on the package/number of pins.

Downloaded from Arrow.com.

Figure 32-3. Overview of the peripheral functions multiplexing



The I/O pins of the device are controlled by PORT peripheral registers. Each port pin has a corresponding bit in the Data Direction (DIR) and Data Output Value (OUT) registers to enable that pin as an output and to define the output state.

The direction of each pin in a PORT group is configured by the DIR register. If a bit in DIR is set to '1', the corresponding pin is configured as an output pin. If a bit in DIR is set to '0', the corresponding pin is configured as an input pin.

When the direction is set as output, the corresponding bit in the OUT register will set the level of the pin. If bit y in OUT is written to '1', pin y is driven HIGH. If bit y in OUT is written to '0', pin y is driven LOW. Pin configuration can be set by Pin Configuration (PINCFGy) registers, with y=00, 01, ..31 representing the bit position.

The Data Input Value (IN) is set as the input value of a port pin with resynchronization to the PORT clock. To reduce power consumption, these input synchronizers are clocked only when system requires reading the input value. The value of the pin can always be read, whether the pin is configured as input or output. If the Input Enable bit in the Pin Configuration registers (PINCFGy.INEN) is '0', the input value will not be sampled.

In PORT, the Peripheral Multiplexer Enable bit in the PINCFGy register (PINCFGy.PMUXEN) can be written to '1' to enable the connection between peripheral functions and individual I/O pins. The Peripheral Multiplexing n (PMUXn) registers select the peripheral function for the corresponding pin. This will override the connection between the PORT and that I/O pin, and connect the selected peripheral signal to the particular I/O pin instead of the PORT line bundle.

The security attribution of each pin in a PORT group is configured by the NONSEC register. If a bit in the NONSEC register is set to '0', the corresponding pin is configured as a secured pin and can only be handled by secure accesses. If a bit in the NONSEC register is set to '1', the corresponding pin is configured as a non-secured pin. Only secure accesses are allowed to write to the NONSEC register.

Downloaded from Arrow.com.

32.6.2 Basic Operation

32.6.2.1 Initialization

After reset, all standard function device I/O pads are connected to the PORT with outputs tri-stated and input buffers disabled, even if there is no clock running.

However, specific pins, such as those used for connection to a debugger, may be configured differently, as required by their special function.

32.6.2.2 Operation

Each I/O pin y can be controlled by the registers in PORT. Each PORT group has its own set of PORT registers, the base address of the register set for pin y is at byte address PORT + ([y] * 0x4). The index within that register set is [y].

Refer to *I/O Multiplexing and Considerations* for details on available pin configuration and PORT groups.

Configuring Pins as Output

To use pin number y as an *output*, write bit y of the DIR register to '1'. This can also be done by writing bit y in the DIRSET register to '1' - this will avoid disturbing the configuration of other pins in that group. The y bit in the OUT register must be written to the desired output value.

Similarly, writing an OUTSET bit to '1' will set the corresponding bit in the OUT register to '1'. Writing a bit in OUTCLR to '1' will set that bit in OUT to zero. Writing a bit in OUTTGL to '1' will toggle that bit in OUT.

Configuring Pins as Input

To use pin y as an *input*, bit y in the DIR register must be written to '0'. This can also be done by writing bit y in the DIRCLR register to '1' - this will avoid disturbing the configuration of other pins in that group. The input value can be read from bit y in register IN as soon as the INEN bit in the Pin Configuration register (PINCFGy.INEN) is written to '1'.

By default, the input synchronizer is clocked only when an input read is requested. This will delay the read operation by two CLK_PORT cycles. To remove the delay, the input synchronizers for each PORT group of eight pins can be configured to be always active, but this will increase power consumption. This is enabled by writing '1' to the corresponding SAMPLINGn bit field of the CTRL register, see CTRL.SAMPLING for details.

Using Alternative Peripheral Functions

To use pin y as one of the available peripheral functions, the corresponding PMUXEN bit of the PINCFGy register must be '1'. The PINCFGy register for pin y is at byte offset (PINCFG0 + [y]).

The peripheral function can be selected by setting the PMUXO or PMUXE in the PMUXn register. The PMUXO/PMUXE is at byte offset PMUX0 + (y/2). The chosen peripheral must also be configured and enabled.

32.6.3 I/O Pin Configuration

The Pin Configuration register (PINCFGy) is used for additional I/O pin configuration. A pin can be set in a totem-pole or pull configuration.

As pull configuration is done through the Pin Configuration register, all intermediate PORT states during switching of pin direction and pin values are avoided.

The I/O pin configurations are described further in this chapter, and summarized in Table 32-2.

Downloaded from Arrow.com.

32.6.3.1 Pin Configurations Summary

Table 32-2. Pin Configurations Summary

DIR	INEN	PULLEN	OUT	Configuration
0	0	0	X	Reset or analog I/O: all digital disabled
0	0	1	0	Pull-down; input disabled
0	0	1	1	Pull-up; input disabled
0	1	0	X	Input
0	1	1	0	Input with pull-down
0	1	1	1	Input with pull-up
1	0	X	X	Output; input disabled
1	1	X	X	Output; input enabled

32.6.3.2 Input Configuration

Figure 32-4. I/O configuration - Standard Input

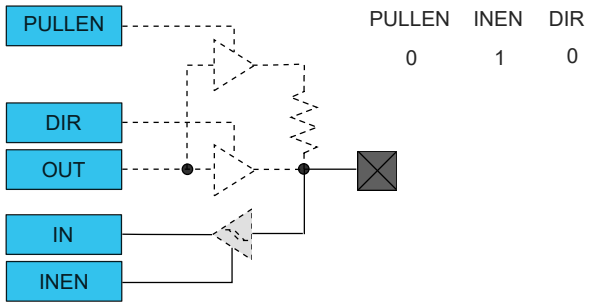
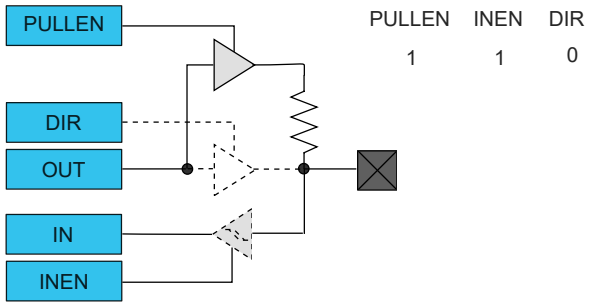


Figure 32-5. I/O Configuration - Input with Pull



Note: When pull is enabled, the pull value is defined by the OUT value.

32.6.3.3 Totem-Pole Output

When configured for totem-pole (push-pull) output, the pin is driven low or high according to the corresponding bit setting in the OUT register. In this configuration there is no current limitation for sink or source other than what the pin is capable of. If the pin is configured for input, the pin will float if no external pull is connected.

Note: Enabling the output driver will automatically disable pull.

Downloaded from Arrow.com.

Figure 32-6. I/O Configuration - Totem-Pole Output with Disabled Input

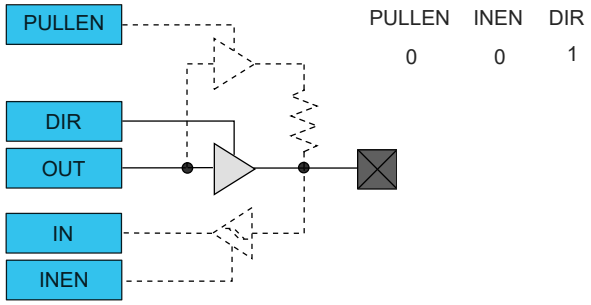


Figure 32-7. I/O Configuration - Totem-Pole Output with Enabled Input

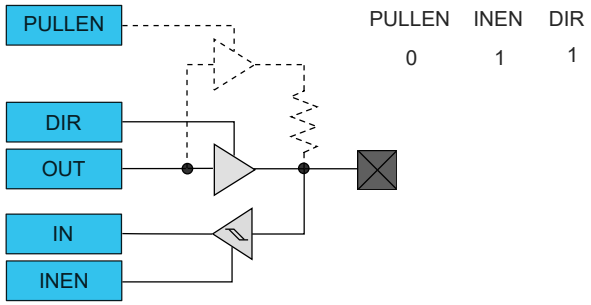
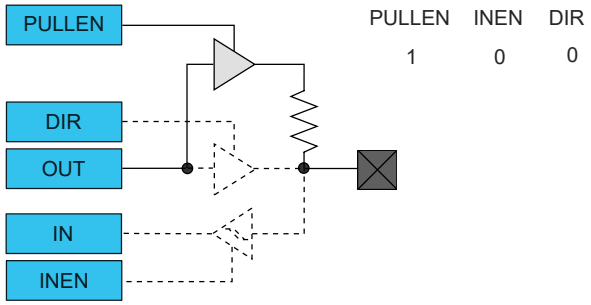


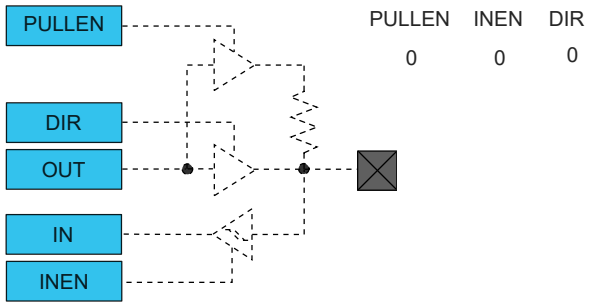
Figure 32-8. I/O Configuration - Output with Pull



32.6.3.4 Digital Functionality Disabled

Neither Input nor Output functionality are enabled.

Figure 32-9. I/O Configuration - Reset or Analog I/O: Digital Output, Input and Pull Disabled



32.6.4 SAM L11 Secure Access Rights

Non-secure write to CTRL, EVCTRL, or NONSEC registers is prohibited.

Non-secure read to CTRL or EVCTRL registers will return zero with no error resulting.

Downloaded from Arrow.com.

Non-secure write to a bit of DIR, DIRCLR, DIRSET, DIRTGL, OUT, OUTCLR, OUTSET, or OUTTGL registers is prohibited if the corresponding bit in NONSEC is zero.

Non-secure write to a PINCFGn register or to a PMUXn register field, either directly or through a write to the WRCONFIG register, is prohibited if the corresponding bit in NONSEC is zero.

DIR, DIRCLR, DIRSET, DIRTGL, IN, OUT, OUTCLR, OUTSET, or OUTTGL bits, PINCFGn registers, or PMUXn register fields relating to secure I/O pins (i.e. the corresponding bits in NONSEC are zero), read as zero in non-secure mode, with no error resulting.

INTFLAG.NSCHK is set to 1 when NSCHK and NONSEC register values are different. Writing a 1 to INTFLAG.NSCHK will clear it and clear the PORT interrupt if enabled.

Secure code should initially write a 1 for all non-secure pins into both NONSEC and NSCHK registers. Then, whenever secure code writes a different value into NONSEC, INTFLAG.NSCHK will be set to 1 and a PORT interrupt will occur, if enabled. The non-secure code can then compare the values of the NONSEC and NSCHK registers to determine which PORT pins have just changed to secure or to non-secure. It should then copy the current NONSEC register value into NSCHK.

32.6.5 Events

The PORT allows input events to control individual I/O pins. These input events are generated by the EVSYS module and can originate from a different clock domain than the PORT module.

The PORT can perform the following actions:

- Output (OUT): I/O pin will be set when the incoming event has a high level ('1') and cleared when the incoming event has a low-level ('0').
- Set (SET): I/O pin will be set when an incoming event is detected.
- Clear (CLR): I/O pin will be cleared when an incoming event is detected.
- Toggle (TGL): I/O pin will toggle when an incoming event is detected.

The event is output to pin without any internal latency. For SET, CLEAR and TOGGLE event actions, the action will be executed up to three clock cycles after a rising edge.

The event actions can be configured with the Event Action m bit group in the Event Input Control register(EVCTRL.EVACTm). Writing a '1' to a PORT Event Enable Input m of the Event Control register (EVCTRL.PORTEIm) enables the corresponding action on input event. Writing '0' to this bit disables the corresponding action on input event. Note that several actions can be enabled for incoming events. If several events are connected to the peripheral, any enabled action will be taken for any of the incoming events. Refer to *EVSYS – Event System*. for details on configuring the Event System.

Each event input can address one and only one I/O pin at a time. The selection of the pin is indicated by the PORT Event Pin Identifier of the Event Input Control register (EVCTR.PIDn). On the other hand, one I/O pin can be addressed by up to four different input events. To avoid action conflict on the output value of the register (OUT) of this particular I/O pin, only one action is performed according to the table below.

Note that this truth table can be applied to any SET/CLR/TGL configuration from two to four active input events.

Downloaded from Arrow.com.

Table 32-3. Priority on Simultaneous SET/CLR/TGL Event Actions

EVACT0	EVACT1	EVACT2	EVACT3	Executed Event Action
SET	SET	SET	SET	SET
CLR	CLR	CLR	CLR	CLR
All Other Combinations				TGL

Be careful when the event is output to pin. Due to the fact the events are received asynchronously, the I/O pin may have unpredictable levels, depending on the timing of when the events are received. When several events are output to the same pin, the lowest event line will get the access. All other events will be ignored.

Related Links

33. EVSYS – Event System

32.6.6 PORT Access Priority

The PORT is accessed by different systems:

- The ARM® CPU through the ARM® single-cycle I/O port (IOBUS)
- The ARM® CPU through the high-speed matrix and the AHB/APB bridge (APB)
- EVSYS through four asynchronous input events

The following priority is adopted:

1. ARM® CPU IOBUS (No wait tolerated)
2. APB
3. EVSYS input events

For input events that require different actions on the same I/O pin, refer to 32.6.5 Events.

Downloaded from Arrow.com.

32.7 Register Summary

Important:
For SAM L11, the PORT register map is automatically duplicated in a Secure and Non-Secure alias:

- The Non-Secure alias is at the peripheral base address
- The Secure alias is located at the peripheral base address + 0x200

Refer to *Mix-Secure Peripherals* for more information on register access rights

Offset	Name	Bit Pos.								
0x00	DIR	7:0	DIR[7:0]							
		15:8	DIR[15:8]							
		23:16	DIR[23:16]							
		31:24	DIR[31:24]							
0x04	DIRCLR	7:0	DIRCLR[7:0]							
		15:8	DIRCLR[15:8]							
		23:16	DIRCLR[23:16]							
		31:24	DIRCLR[31:24]							
0x08	DIRSET	7:0	DIRSET[7:0]							
		15:8	DIRSET[15:8]							
		23:16	DIRSET[23:16]							
		31:24	DIRSET[31:24]							
0x0C	DIRTGL	7:0	DIRTGL[7:0]							
		15:8	DIRTGL[15:8]							
		23:16	DIRTGL[23:16]							
		31:24	DIRTGL[31:24]							
0x10	OUT	7:0	OUT[7:0]							
		15:8	OUT[15:8]							
		23:16	OUT[23:16]							
		31:24	OUT[31:24]							
0x14	OUTCLR	7:0	OUTCLR[7:0]							
		15:8	OUTCLR[15:8]							
		23:16	OUTCLR[23:16]							
		31:24	OUTCLR[31:24]							
0x18	OUTSET	7:0	OUTSET[7:0]							
		15:8	OUTSET[15:8]							
		23:16	OUTSET[23:16]							
		31:24	OUTSET[31:24]							
0x1C	OUTTGL	7:0	OUTTGL[7:0]							
		15:8	OUTTGL[15:8]							
		23:16	OUTTGL[23:16]							
		31:24	OUTTGL[31:24]							
0x20	IN	7:0	IN[7:0]							
		15:8	IN[15:8]							
		23:16	IN[23:16]							

Downloaded from Arrow.com.

Offset	Name	Bit Pos.								
		31:24	IN[31:24]							
0x24	CTRL	7:0	SAMPLING[7:0]							
		15:8	SAMPLING[15:8]							
		23:16	SAMPLING[23:16]							
		31:24	SAMPLING[31:24]							
0x28	WRCONFIG	7:0	PINMASK[7:0]							
		15:8	PINMASK[15:8]							
		23:16		DRVSTR				PULLEN	INEN	PMUXEN
		31:24	HWSEL	WRPINCFG		WRPMUX	PMUX[3:0]			
0x2C	EVCTRL	7:0	PORTEIx	EVACTx[1:0]		PIDx[4:0]				
		15:8	PORTEIx	EVACTx[1:0]		PIDx[4:0]				
		23:16	PORTEIx	EVACTx[1:0]		PIDx[4:0]				
		31:24	PORTEIx	EVACTx[1:0]		PIDx[4:0]				
0x30	PMUX0	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x31	PMUX1	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x32	PMUX2	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x33	PMUX3	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x34	PMUX4	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x35	PMUX5	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x36	PMUX6	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x37	PMUX7	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x38	PMUX8	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x39	PMUX9	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x3A	PMUX10	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x3B	PMUX11	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x3C	PMUX12	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x3D	PMUX13	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x3E	PMUX14	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x3F	PMUX15	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x40	PINCFG0	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x41	PINCFG1	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x42	PINCFG2	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x43	PINCFG3	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x44	PINCFG4	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x45	PINCFG5	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x46	PINCFG6	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x47	PINCFG7	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x48	PINCFG8	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x49	PINCFG9	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x4A	PINCFG10	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x4B	PINCFG11	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x4C	PINCFG12	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x4D	PINCFG13	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x4E	PINCFG14	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x4F	PINCFG15	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x50	PINCFG16	7:0		DRVSTR				PULLEN	INEN	PMUXEN

Downloaded from Arrow.com.

Offset	Name	Bit Pos.								
0x51	PINCFG17	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x52	PINCFG18	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x53	PINCFG19	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x54	PINCFG20	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x55	PINCFG21	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x56	PINCFG22	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x57	PINCFG23	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x58	PINCFG24	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x59	PINCFG25	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x5A	PINCFG26	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x5B	PINCFG27	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x5C	PINCFG28	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x5D	PINCFG29	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x5E	PINCFG30	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x5F	PINCFG31	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x60	INTENCLR	7:0								NSCHK
		15:8								
		23:16								
		31:24								
0x64	INTENSET	7:0								NSCHK
		15:8								
		23:16								
		31:24								
0x68	INTFLAG	7:0								NSCHK
		15:8								
		23:16								
		31:24								
0x6C	NONSEC	7:0	NONSEC[7:0]							
		15:8	NONSEC[15:8]							
		23:16	NONSEC[23:16]							
		31:24	NONSEC[31:24]							
0x70	NSCHK	7:0	NSCHK[7:0]							
		15:8	NSCHK[15:8]							
		23:16	NSCHK[23:16]							
		31:24	NSCHK[31:24]							

32.8 Register Description

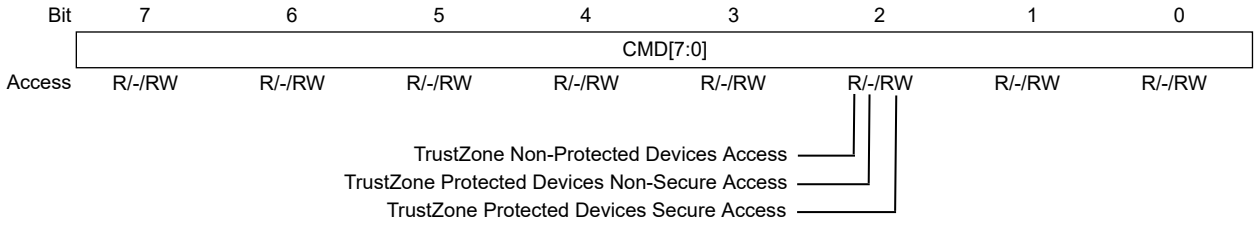
Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to 32.5.8 Register Access Protection.

On SAM L11 devices, the Mix-Secure peripheral has different types of registers (Non-Secure, Secure, Write-Secure, Mix-Secure, and Write-Mix-Secure) with different access permissions for each bitfield.

Downloaded from Arrow.com.

Refer to *Mix-Secure Peripherals* for more details. In the following register descriptions, the access permissions are specified as shown in the following figure.



Downloaded from Arrow.com.

32.8.1 Data Direction

Name: DIR
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, read and write write accesses (RW*) are allowed only if the security attribution for the corresponding I/O pin is set as Non-Secured in the NONSEC register.

This register allows the user to configure one or more I/O pins as an input or output. This register can be manipulated without doing a read-modify-write operation by using the Data Direction Toggle (DIRTGL), Data Direction Clear (DIRCLR) and Data Direction Set (DIRSET) registers.

Tip: The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

Bit	31	30	29	28	27	26	25	24
	DIR[31:24]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DIR[23:16]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DIR[15:8]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DIR[7:0]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DIR[31:0] Port Data Direction
These bits set the data direction for the individual I/O pins in the PORT group.

Downloaded from Arrow.com.

Value	Description
0	The corresponding I/O pin in the PORT group is configured as an input.
1	The corresponding I/O pin in the PORT group is configured as an output.

Downloaded from Arrow.com.

32.8.2 Data Direction Clear

Name: DIRCLR
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the security attribution for the corresponding I/O pin is set as Non-Secured in the NONSEC register.

This register allows the user to set one or more I/O pins as an input, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Direction (DIR), Data Direction Toggle (DIRTGL) and Data Direction Set (DIRSET) registers.

Tip: The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

Bit	31	30	29	28	27	26	25	24
	DIRCLR[31:24]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DIRCLR[23:16]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DIRCLR[15:8]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DIRCLR[7:0]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DIRCLR[31:0] Port Data Direction Clear
Writing a '0' to a bit has no effect.

Downloaded from Arrow.com.

Writing a '1' to a bit will clear the corresponding bit in the DIR register, which configures the I/O pin as an input.

Value	Description
0	The corresponding I/O pin in the PORT group will keep its configuration.
1	The corresponding I/O pin in the PORT group is configured as input.

Downloaded from Arrow.com.

32.8.3 Data Direction Set

Name: DIRSET
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection, Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the security attribution for the corresponding I/O pin is set as Non-Secured in the NONSEC register.

This register allows the user to set one or more I/O pins as an output, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Direction (DIR), Data Direction Toggle (DIRTGL) and Data Direction Clear (DIRCLR) registers.

Tip: The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

Bit	31	30	29	28	27	26	25	24
	DIRSET[31:24]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DIRSET[23:16]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DIRSET[15:8]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DIRSET[7:0]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DIRSET[31:0] Port Data Direction Set
Writing '0' to a bit has no effect.

Downloaded from Arrow.com.

Writing '1' to a bit will set the corresponding bit in the DIR register, which configures the I/O pin as an output.

Value	Description
0	The corresponding I/O pin in the PORT group will keep its configuration.
1	The corresponding I/O pin in the PORT group is configured as an output.

Downloaded from Arrow.com.

32.8.4 Data Direction Toggle

Name: DIRTGL
Offset: 0x0C
Reset: 0x00000000
Property: PAC Write-Protection, Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the security attribution for the corresponding I/O pin is set as Non-Secured in the NONSEC register.

This register allows the user to toggle the direction of one or more I/O pins, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Direction (DIR), Data Direction Set (DIRSET) and Data Direction Clear (DIRCLR) registers.

Tip: The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

Bit	31	30	29	28	27	26	25	24
	DIRTGL[31:24]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DIRTGL[23:16]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DIRTGL[15:8]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DIRTGL[7:0]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DIRTGL[31:0] Port Data Direction Toggle
Writing '0' to a bit has no effect.

Downloaded from Arrow.com.

Writing '1' to a bit will toggle the corresponding bit in the DIR register, which reverses the direction of the I/O pin.

Value	Description
0	The corresponding I/O pin in the PORT group will keep its configuration.
1	The direction of the corresponding I/O pin is toggled.

Downloaded from Arrow.com.

32.8.5 Data Output Value

Name: OUT
Offset: 0x10
Reset: 0x00000000
Property: PAC Write-Protection, Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the security attribution for the corresponding I/O pin is set as Non-Secured in the NONSEC register.

This register sets the data output drive value for the individual I/O pins in the PORT.

This register can be manipulated without doing a read-modify-write operation by using the Data Output Value Clear (OUTCLR), Data Output Value Set (OUTSET), and Data Output Value Toggle (OUTTGL) registers.

Tip: The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

Downloaded from Arrow.com.

Bit	31	30	29	28	27	26	25	24
	OUT[31:24]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	OUT[23:16]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	OUT[15:8]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OUT[7:0]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – OUT[31:0] PORT Data Output Value
For pins configured as outputs via the Data Direction register (DIR), these bits set the logical output drive level.

For pins configured as inputs via the Data Direction register (DIR) and with pull enabled via the Pull Enable bit in the Pin Configuration register (PINCFG.PULLEN), these bits will set the input pull direction.

Value	Description
0	The I/O pin output is driven low, or the input is connected to an internal pull-down.
1	The I/O pin output is driven high, or the input is connected to an internal pull-up.

Downloaded from Arrow.com.

32.8.6 Data Output Value Clear

Name: OUTCLR
Offset: 0x14
Reset: 0x00000000
Property: PAC Write-Protection, Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the security attribution for the corresponding I/O pin is set as Non-Secured in the NONSEC register.

This register allows the user to set one or more output I/O pin drive levels low, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Output Value (OUT), Data Output Value Toggle (OUTTGL) and Data Output Value Set (OUTSET) registers.

Tip: The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

Bit	31	30	29	28	27	26	25	24
	OUTCLR[31:24]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	OUTCLR[23:16]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	OUTCLR[15:8]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OUTCLR[7:0]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – OUTCLR[31:0] PORT Data Output Value Clear
Writing '0' to a bit has no effect.

Downloaded from Arrow.com.

Writing '1' to a bit will clear the corresponding bit in the OUT register. Pins configured as outputs via the Data Direction register (DIR) will be set to low output drive level. Pins configured as inputs via DIR and with pull enabled via the Pull Enable bit in the Pin Configuration register (PINCFG.PULLEN) will set the input pull direction to an internal pull-down.

Value	Description
0	The corresponding I/O pin in the PORT group will keep its configuration.
1	The corresponding I/O pin output is driven low, or the input is connected to an internal pull-down.

Downloaded from Arrow.com.

32.8.7 Data Output Value Set

Name: OUTSET
Offset: 0x18
Reset: 0x00000000
Property: PAC Write-Protection, Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the security attribution for the corresponding I/O pin is set as Non-Secured in the NONSEC register.

This register allows the user to set one or more output I/O pin drive levels high, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Output Value (OUT), Data Output Value Toggle (OUTTGL) and Data Output Value Clear (OUTCLR) registers.

Tip: The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

Bit	31	30	29	28	27	26	25	24
	OUTSET[31:24]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	OUTSET[23:16]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	OUTSET[15:8]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OUTSET[7:0]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – OUTSET[31:0] PORT Data Output Value Set
Writing '0' to a bit has no effect.

Downloaded from Arrow.com.

Writing '1' to a bit will set the corresponding bit in the OUT register, which sets the output drive level high for I/O pins configured as outputs via the Data Direction register (DIR). For pins configured as inputs via Data Direction register (DIR) with pull enabled via the Pull Enable register (PULLEN), these bits will set the input pull direction to an internal pull-up.

Value	Description
0	The corresponding I/O pin in the group will keep its configuration.
1	The corresponding I/O pin output is driven high, or the input is connected to an internal pull-up.

Downloaded from Arrow.com.

32.8.8 Data Output Value Toggle

Name: OUTTGL
Offset: 0x1C
Reset: 0x00000000
Property: PAC Write-Protection, Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the security attribution for the corresponding I/O pin is set as Non-Secured in the NONSEC register.

This register allows the user to toggle the drive level of one or more output I/O pins, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Output Value (OUT), Data Output Value Set (OUTSET) and Data Output Value Clear (OUTCLR) registers.

Tip: The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

Bit	31	30	29	28	27	26	25	24
	OUTTGL[31:24]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	OUTTGL[23:16]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	OUTTGL[15:8]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OUTTGL[7:0]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – OUTTGL[31:0] PORT Data Output Value Toggle
Writing '0' to a bit has no effect.

Downloaded from Arrow.com.

Writing '1' to a bit will toggle the corresponding bit in the OUT register, which inverts the output drive level for I/O pins configured as outputs via the Data Direction register (DIR). For pins configured as inputs via Data Direction register (DIR) with pull enabled via the Pull Enable register (PULLEN), these bits will toggle the input pull direction.

Value	Description
0	The corresponding I/O pin in the PORT group will keep its configuration.
1	The corresponding OUT bit value is toggled.

Downloaded from Arrow.com.

32.8.9 Data Input Value

Name: IN
Offset: 0x20
Reset: 0x40000000
Property: Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, read accesses (R*) are allowed only if the security attribution for the corresponding I/O pin is set as Non-Secured in the NONSEC register.

Tip: The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

Bit	31	30	29	28	27	26	25	24
	IN[31:24]							
Access	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	IN[23:16]							
Access	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	IN[15:8]							
Access	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	IN[7:0]							
Access	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – IN[31:0] PORT Data Input Value
These bits are cleared when the corresponding I/O pin input sampler detects a logical low level on the input pin.

These bits are set when the corresponding I/O pin input sampler detects a logical high level on the input pin.

Downloaded from Arrow.com.

32.8.10 Control

Name: CTRL
Offset: 0x24
Reset: 0x00000000
Property: PAC Write-Protection, Secure

Tip: The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

Bit	31	30	29	28	27	26	25	24
	SAMPLING[31:24]							
Access	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SAMPLING[23:16]							
Access	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SAMPLING[15:8]							
Access	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SAMPLING[7:0]							
Access	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – SAMPLING[31:0] Input Sampling Mode
Configures the input sampling functionality of the I/O pin input samplers, for pins configured as inputs via the Data Direction register (DIR).

The input samplers are enabled and disabled in sub-groups of eight. Thus if any pins within a byte request continuous sampling, all pins in that eight pin sub-group will be continuously sampled.

Value	Description
0	The I/O pin input synchronizer is disabled.
1	The I/O pin input synchronizer is enabled.

Downloaded from Arrow.com.

32.8.11 Write Configuration

Name: WRCONFIG
Offset: 0x28
Reset: 0x00000000
Property: PAC Write-Protection, Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, write accesses (W*) are allowed only if the security attribution for the corresponding I/O pin is set as Non-Secured in the NONSEC register.

Tip: The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

This write-only register is used to configure several pins simultaneously with the same configuration and/or peripheral multiplexing.

In order to avoid side effect of non-atomic access, 8-bit or 16-bit writes to this register will have no effect. Reading this register always returns zero.

Bit	31	30	29	28	27	26	25	24
	HWSEL	WRPINCFG		WRPMUX	PMUX[3:0]			
Access	W/W*/W	W/W*/W		W/W*/W	W/W*/W	W/W*/W	W/W*/W	W/W*/W
Reset	0	0		0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
		DRVSTR				PULLEN	INEN	PMUXEN
Access		W/W*/W				W/W*/W	W/W*/W	W/W*/W
Reset		0				0	0	0

Bit	15	14	13	12	11	10	9	8
	PINMASK[15:8]							
Access	W/W*/W	W/W*/W	W/W*/W	W/W*/W	W/W*/W	W/W*/W	W/W*/W	W/W*/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PINMASK[7:0]							
Access	W/W*/W	W/W*/W	W/W*/W	W/W*/W	W/W*/W	W/W*/W	W/W*/W	W/W*/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – HWSEL Half-Word Select
This bit selects the half-word field of a 32-PORT group to be reconfigured in the atomic write operation.

Downloaded from Arrow.com.

This bit will always read as zero.

Value	Description
0	The lower 16 pins of the PORT group will be configured.
1	The upper 16 pins of the PORT group will be configured.

Bit 30 – WRPINCFG Write PINCFG
This bit determines whether the atomic write operation will update the Pin Configuration register (PINCFGy) or not for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits.

Writing '0' to this bit has no effect.

Writing '1' to this bit updates the configuration of the selected pins with the written WRCONFIG.DRVSTR, WRCONFIG.PULLEN, WRCONFIG.INEN, WRCONFIG.PMUXEN, and WRCONFIG.PINMASK values.

This bit will always read as zero.

Value	Description
0	The PINCFGy registers of the selected pins will not be updated.
1	The PINCFGy registers of the selected pins will be updated.

Bit 28 – WRPMUX Write PMUX
This bit determines whether the atomic write operation will update the Peripheral Multiplexing register (PMUXn) or not for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits.

Writing '0' to this bit has no effect.

Writing '1' to this bit updates the pin multiplexer configuration of the selected pins with the written WRCONFIG. PMUX value.

This bit will always read as zero.

Value	Description
0	The PMUXn registers of the selected pins will not be updated.
1	The PMUXn registers of the selected pins will be updated.

Bits 27:24 – PMUX[3:0] Peripheral Multiplexing
These bits determine the new value written to the Peripheral Multiplexing register (PMUXn) for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPMUX bit is set.

These bits will always read as zero.

Bit 22 – DRVSTR Output Driver Strength Selection
This bit determines the new value written to PINCFGy.DRVSTR for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPINCFG bit is set.

This bit will always read as zero.

Bit 18 – PULLEN Pull Enable
This bit determines the new value written to PINCFGy.PULLEN for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPINCFG bit is set.

This bit will always read as zero.

Downloaded from Arrow.com.

Bit 17 – INEN Input Enable
This bit determines the new value written to PINCFGy.INEN for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPINCFG bit is set.

This bit will always read as zero.

Bit 16 – PMUXEN Peripheral Multiplexer Enable
This bit determines the new value written to PINCFGy.PMUXEN for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPINCFG bit is set.

This bit will always read as zero.

Bits 15:0 – PINMASK[15:0] Pin Mask for Multiple Pin Configuration
These bits select the pins to be configured within the half-word group selected by the WRCONFIG.HWSEL bit.

These bits will always read as zero.

Value	Description
0	The configuration of the corresponding I/O pin in the half-word group will be left unchanged.
1	The configuration of the corresponding I/O pin in the half-word PORT group will be updated.

Downloaded from Arrow.com.

32.8.12 Event Input Control

Name: EVCTRL
Offset: 0x2C
Reset: 0x00000000
Property: PAC Write-Protection, Secure

Tip: The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

There are up to four input event pins for each PORT group. Each byte of this register addresses one Event input pin.

Bit	31	30	29	28	27	26	25	24
	PORTEIx	EVACTx[1:0]		PIDx[4:0]				
Access	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PORTEIx	EVACTx[1:0]		PIDx[4:0]				
Access	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PORTEIx	EVACTx[1:0]		PIDx[4:0]				
Access	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PORTEIx	EVACTx[1:0]		PIDx[4:0]				
Access	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW
Reset	0	0	0	0	0	0	0	0

Bits 31,23,15,7 – PORTEIx PORT Event Input Enable x [x = 3..0]

Value	Description
0	The event action x (EVACTx) will not be triggered on any incoming event.
1	The event action x (EVACTx) will be triggered on any incoming event.

Bits 30:29, 22:21,14:13,6:5 – EVACTx PORT Event Action x [x = 3..0]
These bits define the event action the PORT will perform on event input x. See also Table 32-4.

Bits 28:24,20:16,12:8,4:0 – PIDx PORT Event Pin Identifier x [x = 3..0]
These bits define the I/O pin on which the event action will be performed, according to Table 32-5.

Downloaded from Arrow.com.

Table 32-4. PORT Event x Action (x = [3..0])

Value	Name	Description
0x0	OUT	Output register of pin will be set to level of event.
0x1	SET	Set output register of pin on event.
0x2	CLR	Clear output register of pin on event.
0x3	TGL	Toggle output register of pin on event.

Table 32-5. PORT Event x Pin Identifier (x = [3..0])

Value	Name	Description
0x0	PIN0	Event action to be executed on PIN 0.
0x1	PIN1	Event action to be executed on PIN 1.
...	...	...
0x31	PIN31	Event action to be executed on PIN 31.

Downloaded from Arrow.com.

32.8.13 Peripheral Multiplexing n

Name: PMUX
Offset: 0x30 + n*0x01 [n=0..15]
Reset: 0x00 except PMUX15 = 0x06
Property: PAC Write-Protection, Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the security attribution for the corresponding I/O pin is set as Non-Secured in the NONSEC register.

Tip: The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

There are up to 16 Peripheral Multiplexing registers in each group, one for every set of two subsequent I/O lines. The n denotes the number of the set of I/O lines.

Bit	7	6	5	4	3	2	1	0
	PMUXO[3:0]				PMUXE[3:0]			
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bits 7:4 – PMUXO[3:0] Peripheral Multiplexing for Odd-Numbered Pin
These bits select the peripheral function for odd-numbered pins (2*n + 1) of a PORT group, if the corresponding PINCFGy.PMUXEN bit is '1'.

Not all possible values for this selection may be valid. For more details, refer to the *I/O Multiplexing and Considerations*.

PMUXO[3:0]	Name	Description
0x0	A	Peripheral function A selected
0x1	B	Peripheral function B selected
0x2	C	Peripheral function C selected
0x3	D	Peripheral function D selected
0x4	E	Peripheral function E selected
0x5	-	Reserved
0x6	G	Peripheral function G selected
0x7	H	Peripheral function H selected

Downloaded from Arrow.com.

PMUXO[3:0]	Name	Description
0x8	I	Peripheral function I selected
0x9-0xF	-	Reserved

Bits 3:0 – PMUXE[3:0] Peripheral Multiplexing for Even-Numbered Pin
These bits select the peripheral function for even-numbered pins (2*n) of a PORT group, if the corresponding PINCFGy.PMUXEN bit is '1'.

Not all possible values for this selection may be valid. For more details, refer to the *I/O Multiplexing and Considerations.*

PMUXE[3:0]	Name	Description
0x0	A	Peripheral function A selected
0x1	B	Peripheral function B selected
0x2	C	Peripheral function C selected
0x3	D	Peripheral function D selected
0x4	E	Peripheral function E selected
0x5	-	Reserved
0x6	G	Peripheral function G selected
0x7	H	Peripheral function H selected
0x8	I	Peripheral function I selected
0x9-0xF	-	Reserved

Downloaded from Arrow.com.

32.8.14 Pin Configuration

Name: PINCFG
Offset: 0x40 + n*0x01 [n=0..31]
Reset: 0x00
Property: PAC Write-Protection, Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the security attribution for the corresponding I/O pin is set as Non-Secured in the NONSEC register.

Tip: The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

There are up to 32 Pin Configuration registers in each PORT group, one for each I/O line.

Bit	7	6	5	4	3	2	1	0
		DRVSTR				PULLEN	INEN	PMUXEN
Access		RW/RW*/RW				RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset		0				0	0	0

Bit 6 – DRVSTR Output Driver Strength Selection
This bit controls the output driver strength of an I/O pin configured as an output.

Value	Description
0	Pin drive strength is set to normal drive strength.
1	Pin drive strength is set to stronger drive strength.

Bit 2 – PULLEN Pull Enable
This bit enables the internal pull-up or pull-down resistor of an I/O pin configured as an input.

Value	Description
0	Internal pull resistor is disabled, and the input is in a high-impedance configuration.
1	Internal pull resistor is enabled, and the input is driven to a defined logic level in the absence of external input.

Bit 1 – INEN Input Enable
This bit controls the input buffer of an I/O pin configured as either an input or output.

Writing a zero to this bit disables the input buffer completely, preventing read-back of the physical pin state when the pin is configured as either an input or output.

Downloaded from Arrow.com.

Value	Description
0	Input buffer for the I/O pin is disabled, and the input value will not be sampled.
1	Input buffer for the I/O pin is enabled, and the input value will be sampled when required.

Bit 0 – PMUXEN Peripheral Multiplexer Enable
This bit enables or disables the peripheral multiplexer selection set in the Peripheral Multiplexing register (PMUXn) to enable or disable alternative peripheral control over an I/O pin direction and output drive value.

Writing a zero to this bit allows the PORT to control the pad direction via the Data Direction register (DIR) and output drive value via the Data Output Value register (OUT). The peripheral multiplexer value in PMUXn is ignored. Writing '1' to this bit enables the peripheral selection in PMUXn to control the pad. In this configuration, the physical pin state may still be read from the Data Input Value register (IN) if PINCFGn.INEN is set.

Value	Description
0	The peripheral multiplexer selection is disabled, and the PORT registers control the direction and output drive value.
1	The peripheral multiplexer selection is enabled, and the selected peripheral function controls the direction and output drive value.

Downloaded from Arrow.com.

32.8.15 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x60
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								NSCHK
Access								RW/RW/RW
Reset								0

Bit 0 – NSCHK Non-Secure Check Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Non-Secure Check Interrupt Enable bit, which disables the Non-Secure Check interrupt.

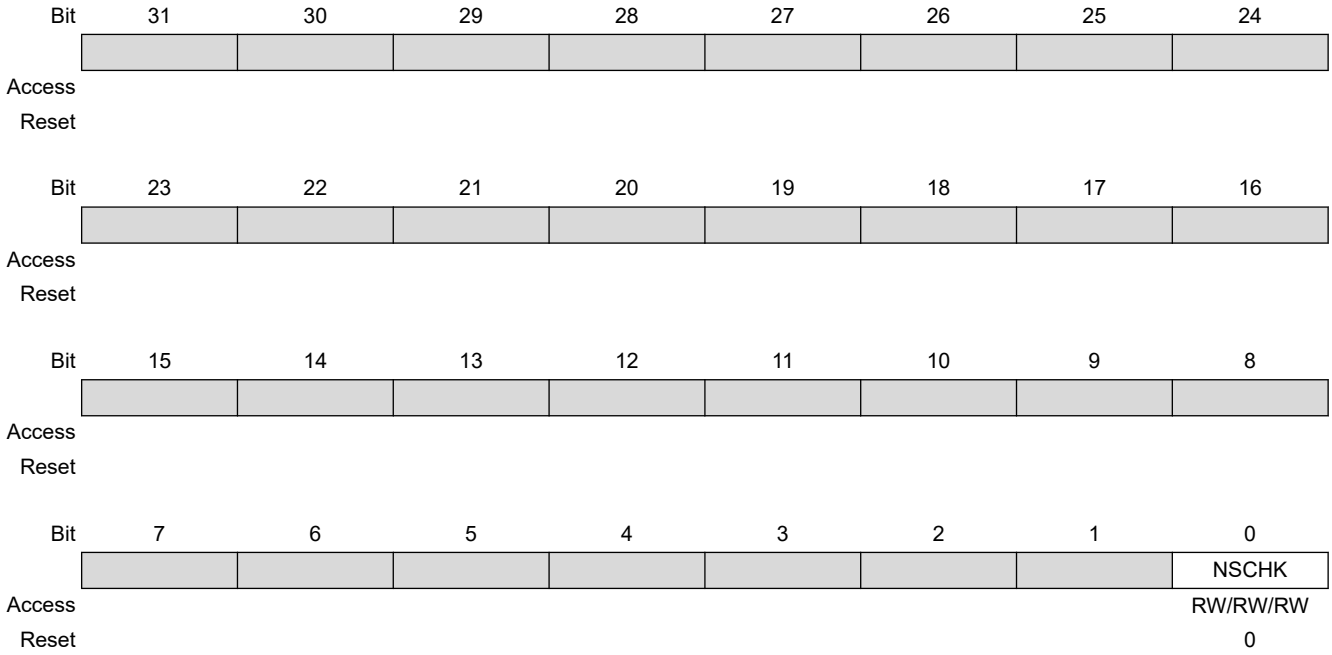
Value	Description
0	The Non-Secure Check interrupt is disabled.
1	The Non-Secure Check interrupt is enabled.

Downloaded from Arrow.com.

32.8.16 Interrupt Enable Set

Name: INTENSET
Offset: 0x64
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).



Bit 0 – NSCHK Non-Secure Check Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Non-Secure Check Interrupt Enable bit, which enables the Non-Secure Check interrupt.

Value	Description
0	The Non-Secure Check interrupt is disabled.
1	The Non-Secure Check interrupt is enabled.

Downloaded from Arrow.com.

32.8.17 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x68
Reset: 0x00000000

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								NSCHK
Access								RW/RW/RW
Reset								0

Bit 0 – NSCHK Non-Secure Check
This flag is set on NONSEC write when a bit in NSCHK is 1 and the corresponding bit in NONSEC is cleared, or when a bit in NSCHK is 0 and the corresponding bit in NONSEC is set.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Non-Secure Check interrupt flag.

Downloaded from Arrow.com.

32.8.18 Security Attribution

Name: NONSEC
Offset: 0x6C
Reset: 0x00000000
Property: PAC Write-Protection, Write-Secure

Important: This register is only available for **SAM L11** and has no effect for **SAM L10**.

This register allows the user to configure one or more I/O pins as secured or non-secured.

Tip: The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

Bit	31	30	29	28	27	26	25	24
	NONSEC[31:24]							
Access	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	NONSEC[23:16]							
Access	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NONSEC[15:8]							
Access	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NONSEC[7:0]							
Access	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NONSEC[31:0] Port Security Attribution
These bits set the security attribution for the individual I/O pins in the PORT group.

Value	Description
0	The corresponding I/O pin in the PORT group is configured as secured. When module is PAC secured, the configuration for this pin is only available through the secure alias. Attempt

Downloaded from Arrow.com.

Value	Description
	to change the pin configuration through the non-secure alias will be silently ignored and reads will return 0.
1	The corresponding I/O pin in the PORT group is configured as non-secured. The I/O line configuration for this pin is available through the non-secure alias.

Downloaded from Arrow.com.

32.8.19 Security Attribution Check

Name: NSCHK
Offset: 0x70
Reset: 0x00000000
Property: PAC Write-Protection

Important: This register is only available for **SAM L11** and has no effect for **SAM L10**.

This register allows the user to select one or more pins to check their security attribution as non-secured.

Tip: The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

Bit	31	30	29	28	27	26	25	24
	NSCHK[31:24]							
Access	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	NSCHK[23:16]							
Access	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NSCHK[15:8]							
Access	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NSCHK[7:0]							
Access	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NSCHK[31:0] Port Security Attribution Check
These bits select the individual pins for security attribution check. If any pin selected in NSCHK has the corresponding bit in NONSEC set to the opposite value, then the NSCHK interrupt flag will be set.

Value	Description
0	0-to-1 transition will be detected on corresponding NONSEC bit.
1	1-to-0 transition will be detected on corresponding NONSEC bit.

Downloaded from Arrow.com.

33. EVSYS – Event System

33.1 Overview

The Event System (EVSYS) allows autonomous, low-latency and configurable communication between peripherals.

Several peripherals can be configured to generate and/or respond to signals known as events. The exact condition to generate an event, or the action taken upon receiving an event, is specific to each peripheral. Peripherals that respond to events are called event users. Peripherals that generate events are called event generators. A peripheral can have one or more event generators and can have one or more event users. Channels and event users can be defined as secured or non-secured, where secured channels or event users can only be handled by secure code.

Communication is made without CPU intervention and without consuming system resources such as bus or RAM bandwidth. This reduces the load on the CPU and other system resources, compared to a traditional interrupt-based system.

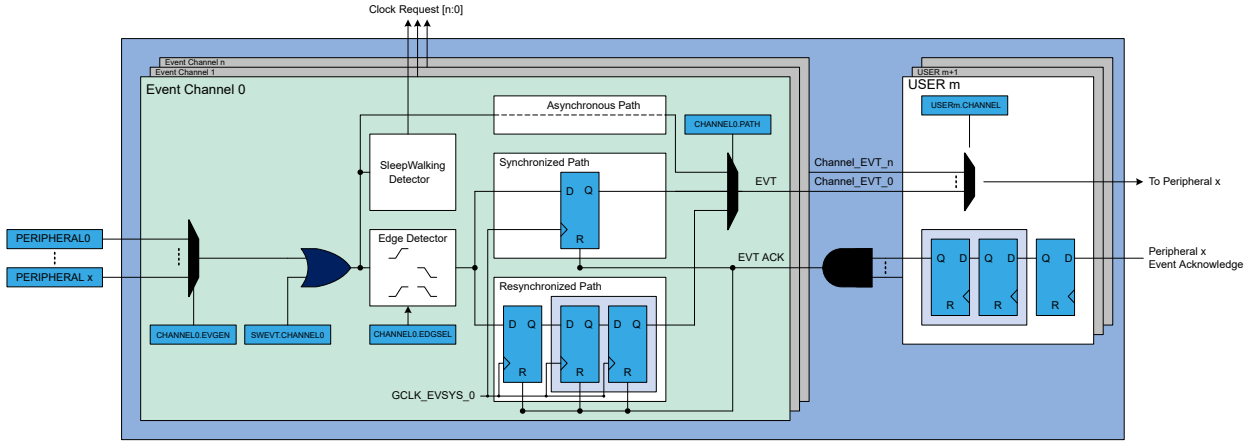
33.2 Features

- 8 configurable event channels:
  - All channels can be connected to any event generator
  - All channels provide a pure asynchronous path
  - 4 channels (CHANNEL0 to CHANNEL3) provide a resynchronized or synchronous path using their dedicated generic clock (GCLK_EVSYS_CHANNEL_n)
- 49 event generators.
- 23 event users.
- Configurable edge detector.
- Peripherals can be event generators, event users, or both.
- SleepWalking and interrupt for operation in sleep modes.
- Software event generation.
- Each event user can choose which channel to respond to.
- Optional Static or Round-Robin interrupt priority arbitration.
- Each channel and each event user can be configured as secured or non-secured (**SAM L11**).

Downloaded from Arrow.com.

33.3 Block Diagram

Figure 33-1. Event System Block Diagram



33.4 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

33.4.1 I/O Lines

Not applicable.

33.4.2 Power Management

The EVSYS can be used to wake up the CPU from all sleep modes (except OFF Mode), even if the clock used by the EVSYS channel and the EVSYS bus clock are disabled. Refer to the *PM – Power Manager* for details on the different sleep modes.

Although the clock for the EVSYS is stopped, the device still can wake up the EVSYS clock. Some event generators can generate an event when their clocks are stopped. The generic clock for the channel (GCLK_EVSYS_CHANNEL_n) will be restarted if that channel uses a synchronized path or a resynchronized path. It does not need to wake the system from sleep.

Important: This generic clock only applies to channels which can be configured as synchronous or resynchronized.

Related Links

22. PM – Power Manager

33.4.3 Clocks

The EVSYS bus clock (CLK_EVSYS_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_EVSYS_APB can be found in *Peripheral Clock Masking*.

Each EVSYS channel which can be configured as synchronous or resynchronized has a dedicated generic clock (GCLK_EVSYS_CHANNEL_n). These are used for event detection and propagation for each channel. These clocks must be configured and enabled in the generic clock controller before using the EVSYS. Refer to *GCLK - Generic Clock Controller* for details.

Downloaded from Arrow.com.

Important: Only EVSYS channel 0 to 3 can be configured as synchronous or resynchronized.

Related Links

19.6.2.6 Peripheral Clock Masking

18. GCLK - Generic Clock Controller

33.4.4 DMA

Not applicable.

33.4.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. Using the EVSYS interrupts requires the interrupt controller to be configured first. Refer to *Nested Vector Interrupt Controller* for details.

33.4.6 Events

Not applicable.

33.4.7 Debug Operation

When the CPU is halted in debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging.

33.4.8 Register Access Protection

Registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC), except for the following:

- Channel Pending Interrupt (INTPEND)
- Channel n Interrupt Flag Status and Clear (CHINTFLAGn)

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

Write-protection does not apply for accesses through an external debugger.

33.4.9 SAM L11 TrustZone Specific Register Access Protection

When the EVSYS is not PAC secured, non-secure and secure code can both access all functionalities. When the EVSYS is PAC secured, all registers are by default available in the secure alias only.

A PAC secured EVSYS can open up individual event channels and event users for non-secure access. This is done using the NONSECCHAN and NONSECUSER registers. When a channel or event user has been configured as non-secure, it can be handled from non-secure code using the EVSYS module non-secure alias. Since only Secured code has the rights to modify the NONSECCHAN and NONSECUSER registers, an interrupt-based mechanism has been added to let Non Secured code know when these registers have been changed by Secured code. A single flag called NSCHK in the INTFLAG register will rise should changes, conditioned by the NSCHKCHAN and NSCHKUSER registers, occur in the NONSECCHAN and NONSECUSER registers.

Note: Refer to the *Mix-Secure Peripherals* section in the *SAM L11 Security Features* chapter.

33.4.10 Analog Connections

Not applicable.

Downloaded from Arrow.com.

33.5 Functional Description

33.5.1 Principle of Operation

The Event System consists of several channels which route the internal events from peripherals (generators) to other internal peripherals or I/O pins (users). Each event generator can be selected as source for multiple channels, but a channel cannot be set to use multiple event generators at the same time.

A channel path can be configured in asynchronous, synchronous or resynchronized mode of operation. The mode of operation must be selected based on the requirements of the application.

When using synchronous or resynchronized path, the Event System includes options to transfer events to users when rising, falling or both edges are detected on event generators.

For further details, refer to the Channel Path section of this chapter.

Related Links

33.5.2.6 Channel Path

33.5.2 Basic Operation

33.5.2.1 Initialization

Before enabling event routing within the system, the Event Users Multiplexer and Event Channels must be selected in the Event System (EVSYS), and the two peripherals that generate and use the event have to be configured. The recommended sequence is:

1. In the event generator peripheral, enable output of event by writing a '1' to the respective Event Output Enable bit ("EO") in the peripheral's Event Control register (e.g., TCC.EVCTRL.MCEO1, AC.EVCTRL.WINEO0, RTC.EVCTRL.OVFEO).
2. Configure the EVSYS:
   2.1. Configure the Event User multiplexer by writing the respective EVSYS.USERm register, see also 33.5.2.3 User Multiplexer Setup.
   2.2. Configure the Event Channel by writing the respective EVSYS.CHANNELn register, see also 33.5.2.4 Event System Channel.
3. Configure the action to be executed by the event user peripheral by writing to the Event Action bits (EVACT) in the respective Event control register (e.g., TC.EVCTRL.EVACT, PDEC.EVCTRL.EVACT). Note: not all peripherals require this step.
4. In the event user peripheral, enable event input by writing a '1' to the respective Event Input Enable bit ("EI") in the peripheral's Event Control register (e.g., AC.EVCTRL.IVEI0, ADC.EVCTRL.STARTEI).

33.5.2.2 Enabling, Disabling, and Resetting

The EVSYS is always enabled.

The EVSYS is reset by writing a ‘1’ to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the EVSYS will be reset to their initial state and all ongoing events will be canceled.

Refer to CTRLA.SWRST register for details.

33.5.2.3 User Multiplexer Setup

The user multiplexer defines the channel to be connected to which event user. Each user multiplexer is dedicated to one event user. A user multiplexer receives all event channels output and must be configured to select one of these channels, as shown in Block Diagram section. The channel is selected with the Channel bit group in the User register (USERm.CHANNEL).

Downloaded from Arrow.com.

The user multiplexer must always be configured before the channel. A list of all user multiplexers is found in the User (USERm) register description.

Related Links

33.3 Block Diagram

33.5.2.4 Event System Channel

An event channel can select one event from a list of event generators. Depending on configuration, the selected event could be synchronized, resynchronized or asynchronously sent to the users. When synchronization or resynchronization is required, the channel includes an internal edge detector, allowing the Event System to generate internal events when rising, falling or both edges are detected on the selected event generator.

An event channel is able to generate internal events for the specific software commands. A channel block diagram is shown in *Block Diagram* section.

Related Links

33.3 Block Diagram

33.5.2.5 Event Generators

Each event channel can receive the events form all event generators. All event generators are listed in the Event Generator bit field in the Channel n register (CHANNELn.EVGEN). For details on event generation, refer to the corresponding module chapter. The channel event generator is selected by the Event Generator bit group in the Channel register (CHANNELn.EVGEN). By default, the channels are not connected to any event generators (ie, CHANNELn.EVGEN = 0)

33.5.2.6 Channel Path

There are different ways to propagate the event from an event generator:

- Asynchronous path
- Synchronous path
- Resynchronized path

The path is decided by writing to the Path Selection bit group of the Channel register (CHANNELn.PATH).

Asynchronous Path

When using the asynchronous path, the events are propagated from the event generator to the event user without intervention from the Event System. The GCLK for this channel (GCLK_EVSYS_CHANNEL_n) is not mandatory, meaning that an event will be propagated to the user without any clock latency.

When the asynchronous path is selected, the channel cannot generate any interrupts, and the Channel x Status register (CHSTATUSx) is always zero. The edge detection is not required and must be disabled by software. Each peripheral event user has to select which event edge must trigger internal actions. For further details, refer to each peripheral chapter description.

Synchronous Path

The synchronous path should be used when the event generator and the event channel share the same generator for the generic clock. If they do not share the same clock, a logic change from the event generator to the event channel might not be detected in the channel, which means that the event will not be propagated to the event user.

Downloaded from Arrow.com.

When using the synchronous path, the channel is able to generate interrupts. The channel status bits in the Channel Status register (CHSTATUS) are also updated and available for use.

Resynchronized Path

The resynchronized path are used when the event generator and the event channel do not share the same generator for the generic clock. When the resynchronized path is used, resynchronization of the event from the event generator is done in the channel.

When the resynchronized path is used, the channel is able to generate interrupts. The channel status bits in the Channel Status register (CHSTATUS) are also updated and available for use.

33.5.2.7 Edge Detection

When synchronous or resynchronized paths are used, edge detection must be enabled. The event system can execute edge detection in three different ways:

- Generate an event only on the rising edge
- Generate an event only on the falling edge
- Generate an event on rising and falling edges.

Edge detection is selected by writing to the Edge Selection bit group of the Channel register (CHANNELn.EDGSEL).

33.5.2.8 Event Latency

An event from an event generator is propagated to an event user with different latency, depending on event channel configuration.

- Asynchronous Path: The maximum routing latency of an external event is related to the internal signal routing and it is device dependent.
- Synchronous Path: The maximum routing latency of an external event is one GCLK_EVSYS_CHANNEL_n clock cycle.
- Resynchronized Path: The maximum routing latency of an external event is three GCLK_EVSYS_CHANNEL_n clock cycles.

The maximum propagation latency of a user event to the peripheral clock core domain is three peripheral clock cycles.

The event generators, event channel and event user clocks ratio must be selected in relation with the internal event latency constraints. Events propagation or event actions in peripherals may be lost if the clock setup violates the internal latencies.

33.5.2.9 The Overrun Channel n Interrupt

The Overrun Channel n interrupt flag in the Interrupt Flag Status and Clear register (CHINTFLAGn.OVR) will be set, and the optional interrupt will be generated in the following cases:

- One or more event users on channel n is not ready when there is a new event.
- An event occurs when the previous event on channel m has not been handled by all event users connected to that channel.

The flag will only be set when using synchronous or resynchronized paths. In the case of asynchronous path, the CHINTFLAGn.OVR is always read as zero.

33.5.2.10 The Event Detected Channel n Interrupt

The Event Detected Channel n interrupt flag in the Interrupt Flag Status and Clear register (CHINTFLAGn.EVD) is set when an event coming from the event generator configured on channel n is detected.

Downloaded from Arrow.com.

The flag will only be set when using a synchronous or resynchronized path. In the case of asynchronous path, the CHINTFLAGn.EVD is always zero.

33.5.2.11 Channel Status

The Channel Status register (CHSTATUS) shows the status of the channels when using a synchronous or resynchronized path. There are two different status bits in CHSTATUS for each of the available channels:

- The CHSTATUSn.BUSYCH bit will be set when an event on the corresponding channel n has not been handled by all event users connected to that channel.
- The CHSTATUSn.RDYUSR bit will be set when all event users connected to the corresponding channel are ready to handle incoming events on that channel.

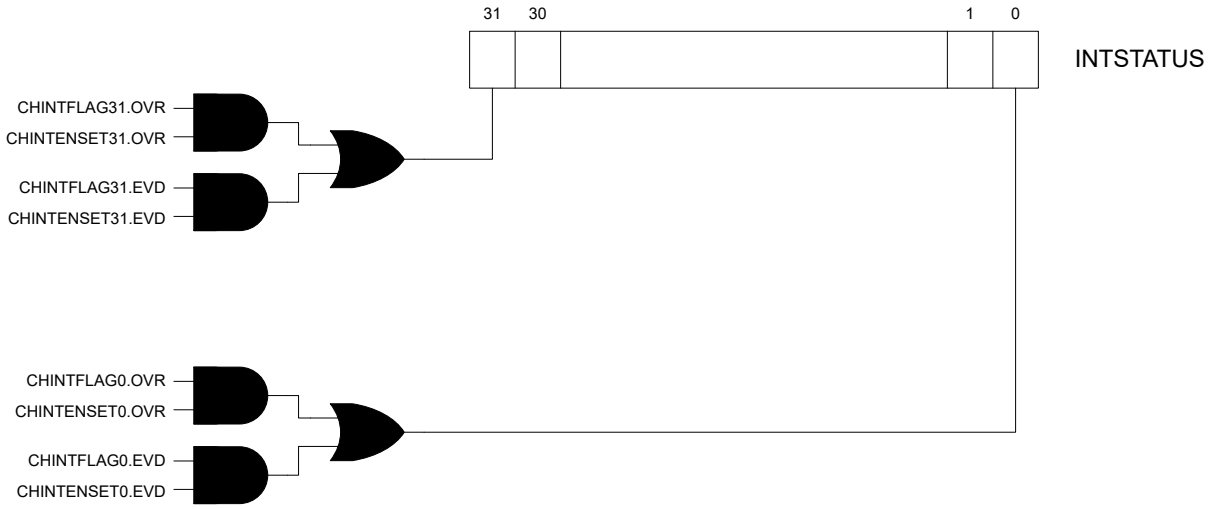
33.5.2.12 Software Event

A software event can be initiated on a channel by writing a '1' to the Software Event bit in the Channel register (CHANNELm.SWEVT). Then the software event can be serviced as any event generator; i.e., when the bit is set to ‘1’, an event will be generated on the respective channel.

33.5.2.13 Interrupt Status and Interrupts Arbitration

The Interrupt Status register stores all channels with pending interrupts, as shown below.

Figure 33-2. Interrupt Status Register

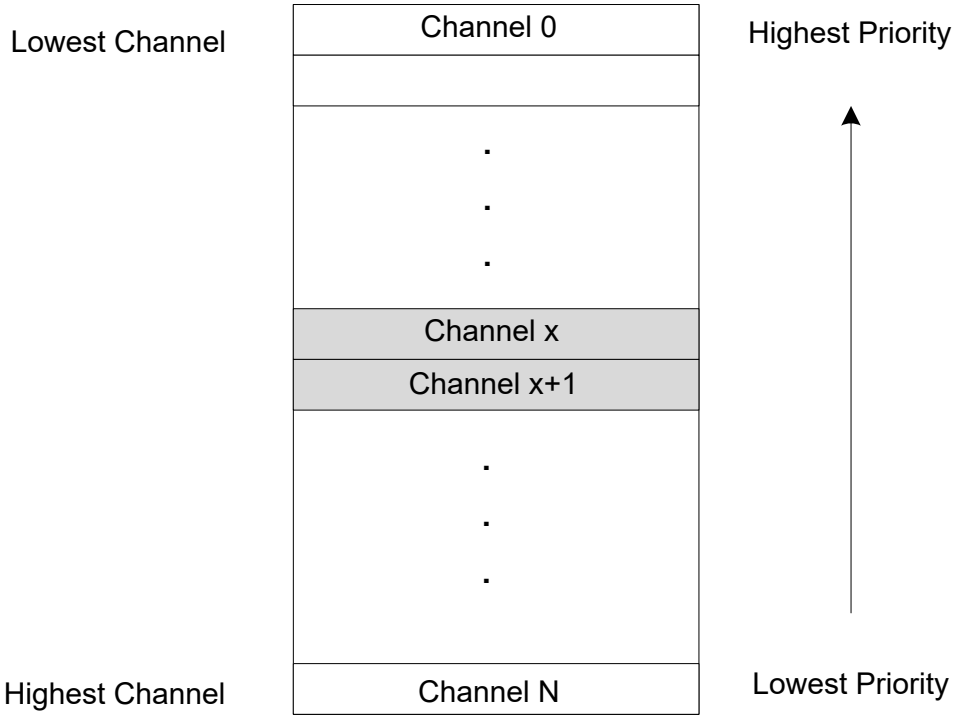


The Event System can arbitrate between all channels with pending interrupts. The arbiter can be configured to prioritize statically or dynamically the incoming events. The priority is evaluated each time a new channel has an interrupt pending, or an interrupt has been cleared. The Channel Pending Interrupt register (INTPEND) will provide the channel number with the highest interrupt priority, and the corresponding channel interrupt flags and status bits.

By default, static arbitration is enabled (PRICTRL.RRENx is '0'), the arbiter will prioritize a low channel number over a high channel number as shown below. When using the status scheme, there is a risk of high channel numbers never being granted access by the arbiter. This can be avoided using a dynamic arbitration scheme.

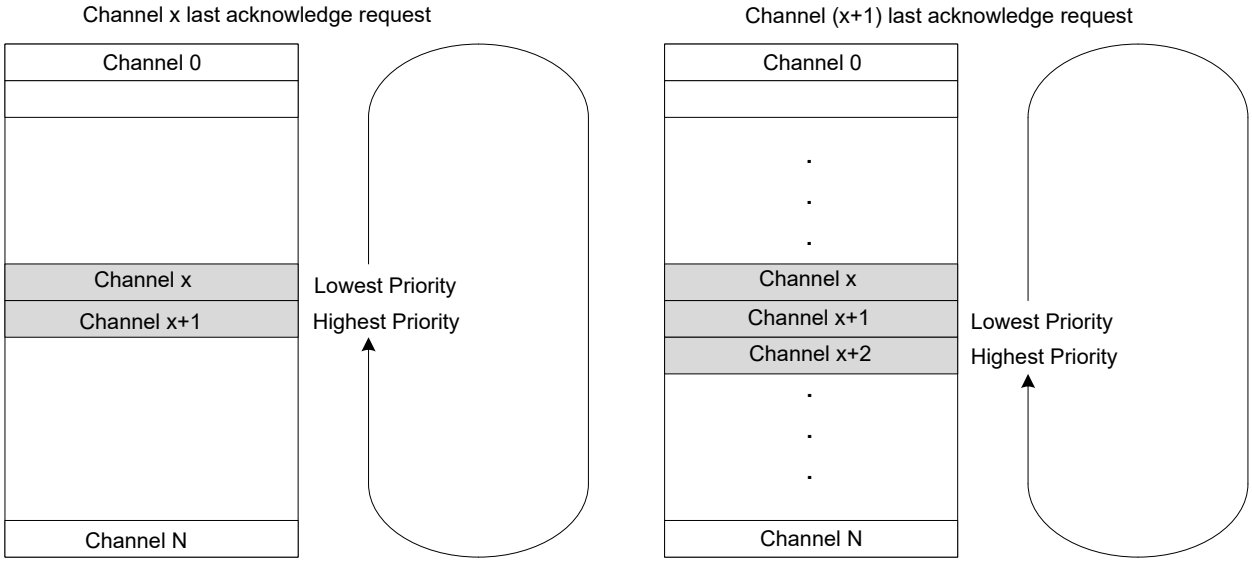
Downloaded from Arrow.com.

Figure 33-3. Static Priority



The dynamic arbitration scheme available in the Event System is round-robin. Round-robin arbitration is enabled by writing PRICTRL.RREN to one. With the round-robin scheme, the channel number of the last channel being granted access will have the lowest priority the next time the arbiter has to grant access to a channel, as shown below. The channel number of the last channel being granted access, will be stored in the Channel Priority Number bit group in the Priority Control register (PRICTRL.PRI).

Figure 33-4. Round-Robin Scheduling



The Channel Pending Interrupt register (INTPEND) also offers the possibility to indirectly clear the interrupt flags of a specific channel. Writing a flag to one in this register, will clear the corresponding interrupt flag of the channel specified by the INTPEND.ID bits.

Downloaded from Arrow.com.

33.5.3 Interrupts

The EVSYS has the following interrupt sources for each channel:

- Overrun Channel n interrupt (OVR)
- Event Detected Channel n interrupt (EVD)

These interrupts events are asynchronous wake-up sources.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the corresponding Channel n Interrupt Flag Status and Clear (CHINTFLAG) register is set when the interrupt condition occurs.

Note: Interrupts must be globally enabled to allow the generation of interrupt requests.

Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Channel n Interrupt Enable Set (CHINTENSET) register, and disabled by writing a '1' to the corresponding bit in the Channel n Interrupt Enable Clear (CHINTENCLR) register. An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled or the Event System is reset. All interrupt requests are ORed together on system level to generate one combined interrupt request to the NVIC.

The user must read the Channel Interrupt Status (INTSTATUS) register to identify the channels with pending interrupts, and must read the Channel n Interrupt Flag Status and Clear (CHINTFLAG) register to determine which interrupt condition is present for the corresponding channel. It is also possible to read the Interrupt Pending register (INTPEND), which provides the highest priority channel with pending interrupt and the respective interrupt flags.

33.5.4 Sleep Mode Operation

The Event System can generate interrupts to wake up the device from IDLE or STANDBY sleep mode.

To be able to run in standby, the Run in Standby bit in the Channel register (CHANNELn.RUNSTDBY) must be set to '1'. When the Generic Clock On Demand bit in Channel register (CHANNELn.ONDEMAND) is set to '1' and the event generator is detected, the event channel will request its clock (GCLK_EVSYS_CHANNEL_n). The event latency for a resynchronized channel path will increase by two GCLK_EVSYS_CHANNEL_n clock (i.e., up to five GCLK_EVSYS_CHANNEL_n clock cycles).

A channel will behave differently in different sleep modes regarding to CHANNELn.RUNSTDBY and CHANNELn.ONDEMAND:

Table 33-1. Event Channel Sleep Behavior

CHANNELn.PATH	CHANNELn.ONDEMAND	CHANNELn.RUNSTDBY	Sleep Behavior
ASYNC	0	0	Only run in IDLE sleep modes if an event must be propagated. Disabled in STANDBY sleep mode.
SYNC/RESYNC	0	1	Run in both IDLE and STANDBY sleep modes.
SYNC/RESYNC	1	0	Only run in IDLE sleep modes if an event must be propagated. Disabled in STANDBY sleep mode. Two GCLK_EVSYS_n latency

Downloaded from Arrow.com.

CHANNELn.PATH	CHANNELn.ONDEMAND	CHANNELn.RUNSTDBY	Sleep Behavior
			added in RESYNC path before the event is propagated internally.
SYNC/RESYNC	1	1	Run in both IDLE and STANDBY sleep modes. Two GCLK_EVSYS_n latency added in RESYNC path before the event is propagated internally.

Downloaded from Arrow.com.

33.6 Register Summary

Important:
For SAM L11, the EVSYS register map is automatically duplicated in a Secure and Non-Secure alias:

- The Non-Secure alias is at the peripheral base address
- The Secure alias is located at the peripheral base address + 0x200

Refer to *Mix-Secure Peripherals* for more information on register access rights

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0								SWRST
0x01 ... 0x03	Reserved									
0x04	SWEVT	7:0	CHANNEL[7:0]							
		15:8								
		23:16								
		31:24								
0x08	PRICTRL	7:0	RREN						PRI[1:0]	
0x09 ... 0x0F	Reserved									
0x10	INTPEND	7:0							ID[1:0]	
		15:8	BUSY	READY					EVD	OVR
0x12 ... 0x13	Reserved									
0x14	INTSTATUS	7:0					CHINT3	CHINT2	CHINT1	CHINT0
		15:8								
		23:16								
		31:24								
0x18	BUSYCH	7:0					BUSYCHx3	BUSYCHx2	BUSYCHx1	BUSYCHx0
		15:8								
		23:16								
		31:24								
0x1C	READYUSR	7:0					READYUSR3	READYUSR2	READYUSR1	READYUSR0
		15:8								
		23:16								
		31:24								
0x20	CHANNEL0	7:0			EVGEN[5:0]					
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x24	CHINTENCLR0	7:0							EVD	OVR
0x25	CHINTENSET0	7:0							EVD	OVR

Downloaded from Arrow.com.

Offset	Name	Bit Pos.								
0x26	CHINTFLAG0	7:0							EVD	OVR
0x27	CHSTATUS0	7:0							BUSYCH	RDYUSR
0x28	CHANNEL1	7:0			EVGEN[5:0]					
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x2C	CHINTENCLR1	7:0							EVD	OVR
0x2D	CHINTENSET1	7:0							EVD	OVR
0x2E	CHINTFLAG1	7:0							EVD	OVR
0x2F	CHSTATUS1	7:0							BUSYCH	RDYUSR
0x30	CHANNEL2	7:0			EVGEN[5:0]					
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x34	CHINTENCLR2	7:0							EVD	OVR
0x35	CHINTENSET2	7:0							EVD	OVR
0x36	CHINTFLAG2	7:0							EVD	OVR
0x37	CHSTATUS2	7:0							BUSYCH	RDYUSR
0x38	CHANNEL3	7:0			EVGEN[5:0]					
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x3C	CHINTENCLR3	7:0							EVD	OVR
0x3D	CHINTENSET3	7:0							EVD	OVR
0x3E	CHINTFLAG3	7:0							EVD	OVR
0x3F	CHSTATUS3	7:0							BUSYCH	RDYUSR
0x40	CHANNEL4	7:0			EVGEN[5:0]					
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x44	CHINTENCLR4	7:0							EVD	OVR
0x45	CHINTENSET4	7:0							EVD	OVR
0x46	CHINTFLAG4	7:0							EVD	OVR
0x47	CHSTATUS4	7:0							BUSYCH	RDYUSR
0x48	CHANNEL5	7:0			EVGEN[5:0]					
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x4C	CHINTENCLR5	7:0							EVD	OVR
0x4D	CHINTENSET5	7:0							EVD	OVR
0x4E	CHINTFLAG5	7:0							EVD	OVR
0x4F	CHSTATUS5	7:0							BUSYCH	RDYUSR
0x50	CHANNEL6	7:0			EVGEN[5:0]					
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								

Downloaded from Arrow.com.

Offset	Name	Bit Pos.								
0x54	CHINTENCLR6	7:0							EVD	OVR
0x55	CHINTENSET6	7:0							EVD	OVR
0x56	CHINTFLAG6	7:0							EVD	OVR
0x57	CHSTATUS6	7:0							BUSYCH	RDYUSR
0x58	CHANNEL7	7:0			EVGEN[5:0]					
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x5C	CHINTENCLR7	7:0							EVD	OVR
0x5D	CHINTENSET7	7:0							EVD	OVR
0x5E	CHINTFLAG7	7:0							EVD	OVR
0x5F	CHSTATUS7	7:0							BUSYCH	RDYUSR
0x60 ... 0x011F	Reserved									
0x0120	USER0	7:0					CHANNEL[3:0]			
0x0121	USER1	7:0					CHANNEL[3:0]			
0x0122	USER2	7:0					CHANNEL[3:0]			
0x0123	USER3	7:0					CHANNEL[3:0]			
0x0124	USER4	7:0					CHANNEL[3:0]			
0x0125	USER5	7:0					CHANNEL[3:0]			
0x0126	USER6	7:0					CHANNEL[3:0]			
0x0127	USER7	7:0					CHANNEL[3:0]			
0x0128	USER8	7:0					CHANNEL[3:0]			
0x0129	USER9	7:0					CHANNEL[3:0]			
0x012A	USER10	7:0					CHANNEL[3:0]			
0x012B	USER11	7:0					CHANNEL[3:0]			
0x012C	USER12	7:0					CHANNEL[3:0]			
0x012D	USER13	7:0					CHANNEL[3:0]			
0x012E	USER14	7:0					CHANNEL[3:0]			
0x012F	USER15	7:0					CHANNEL[3:0]			
0x0130	USER16	7:0					CHANNEL[3:0]			
0x0131	USER17	7:0					CHANNEL[3:0]			
0x0132	USER18	7:0					CHANNEL[3:0]			
0x0133	USER19	7:0					CHANNEL[3:0]			
0x0134	USER20	7:0					CHANNEL[3:0]			
0x0135	USER21	7:0					CHANNEL[3:0]			
0x0136	USER22	7:0					CHANNEL[3:0]			
0x0137 ... 0x01D3	Reserved									
0x01D4	INTENCLR	7:0								NSCHK
0x01D5	INTENSET	7:0								NSCHK
0x01D6	INTFLAG	7:0								NSCHK
0x01D7	Reserved									
0x01D8	NONSECCHAN	7:0	CHANNELn[7:0]							

Downloaded from Arrow.com.

Offset	Name	Bit Pos.								
		15:8								
		23:16								
		31:24								
0x01DC	NSCHKCHAN	7:0	CHANNELn[7:0]							
		15:8								
		23:16								
		31:24								
0x01E0	NONSECUSER0	7:0	USERn[7:0]							
		15:8	USERn[15:8]							
		23:16		USERn[22:16]						
		31:24								
0x01E4	NONSECUSER1	7:0	USERn[7:0]							
		15:8	USERn[15:8]							
		23:16		USERn[22:16]						
		31:24								
0x01E8 ... 0x01EF	Reserved									
0x01F0	NSCHKUSER0	7:0	USERn[7:0]							
		15:8	USERn[15:8]							
		23:16		USERn[22:16]						
		31:24								
0x01F4	NSCHKUSER1	7:0	USERn[7:0]							
		15:8	USERn[15:8]							
		23:16		USERn[22:16]						
		31:24								

33.7 Register Description

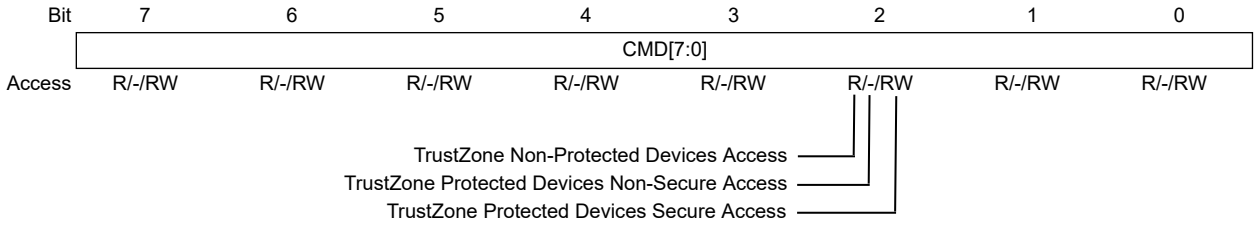
Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Refer to *Register Access Protection* and *PAC - Peripheral Access Controller*.

On SAM L11 devices, the Mix-Secure peripheral has different types of registers (Non-Secure, Secure, Write-Secure, Mix-Secure, and Write-Mix-Secure) with different access permissions for each bitfield. Refer to *Mix-Secure Peripherals* for more details. In the following register descriptions, the access permissions are specified as shown in the following figure.

Downloaded from Arrow.com.



Related Links

15. PAC - Peripheral Access Controller

33.4.8 Register Access Protection

Downloaded from Arrow.com.

33.7.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection , Secure

Bit	7	6	5	4	3	2	1	0
								SWRST
Access								W/-/W
Reset								0

Bit 0 – SWRST Software Reset
Writing '0' to this bit has no effect.
Writing '1' to this bit resets all registers in the EVSYS to their initial state.

Note: Before applying a Software Reset it is recommended to disable the event generators.

Downloaded from Arrow.com.

33.7.2 Software Event

Name: SWEVT
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, write accesses (W*) are allowed only if the security attribution for the corresponding channel (CHANNELx) is set as Non-Secured in the NONSECCHAN register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CHANNEL[7:0]							
Access	W/W*/W	W/W*/W	W/W*/W	W/W*/W	W/W*/W	W/W*/W	W/W*/W	W/W*/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – CHANNEL[7:0] Channel x Software Selection
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will trigger a software event for channel x.

These bits always return '0' when read.

Downloaded from Arrow.com.

33.7.3 Priority Control

Name: PRICTRL
Offset: 0x08
Reset: 0x00
Property: PAC Write-Protection, Secure

Bit	7	6	5	4	3	2	1	0
	RREN						PRI[1:0]	
Access	RW/RW/RW						RW/-/RW	RW/-/RW
Reset	0						0	0

Bit 7 – RREN Round-Robin Scheduling Enable
For details on scheduling schemes, refer to Interrupt Status and Interrupts Arbitration

Value	Description
0	Static scheduling scheme for channels with level priority
1	Round-robin scheduling scheme for channels with level priority

Bits 1:0 – PRI[1:0] Channel Priority Number
When round-robin arbitration is enabled (PRICTRL.RREN=1) for priority level, this register holds the channel number of the last EVSYS channel being granted access as the active channel with priority level. The value of this bit group is updated each time the INTPEND or any of CHINTFLAG registers are written.

When static arbitration is enabled (PRICTRL.RREN=0) for priority level, and the value of this bit group is nonzero, it will not affect the static priority scheme.

This bit group is not reset when round-robin scheduling gets disabled (PRICTRL.RREN written to zero).

Downloaded from Arrow.com.

33.7.4 Channel Pending Interrupt

Name: INTPEND
Offset: 0x10
Reset: 0x4000
Property: Secure

An interrupt that handles several channels should consult the INTPEND register to find out which channel number has priority (ignoring/filtering each channel that has its own interrupt line). An interrupt dedicated to only one channel must not use the INTPEND register.

Bit	15	14	13	12	11	10	9	8
	BUSY	READY					EVD	OVR
Access	R/-/R	R/-/R					RW/-/RW	RW/-/RW
Reset	0	1					0	0

Bit	7	6	5	4	3	2	1	0
							ID[1:0]	
Access							RW/-/RW	RW/-/RW
Reset							0	0

Bit 15 – BUSY Busy
This bit is read '1' when the event on a channel selected by Channel ID field (ID) has not been handled by all the event users connected to this channel.

Bit 14 – READY Ready
This bit is read '1' when all event users connected to the channel selected by Channel ID field (ID) are ready to handle incoming events on this channel.

Bit 9 – EVD Channel Event Detected
This flag is set on the next CLK_EVSYS_APB cycle when an event is being propagated through the channel, and an interrupt request will be generated if CHINTENCLR/SET.EVD is '1'.

When the event channel path is asynchronous, the EVD bit will not be set.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear it. It will also clear the corresponding flag in the Channel n Interrupt Flag Status and Clear register (CHINTFLAGn) of this peripheral, where n is determined by the Channel ID bit field (ID) in this register.

Bit 8 – OVR Channel Overrun
This flag is set on the next CLK_EVSYS cycle after an overrun channel condition occurs, and an interrupt request will be generated if CHINTENCLR/SET.OVRx is '1'.

There are two possible overrun channel conditions:

- One or more of the event users on channel selected by Channel ID field (ID) are not ready when a new event occurs
- An event happens when the previous event on channel selected by Channel ID field (ID) has not yet been handled by all event users

Downloaded from Arrow.com.

When the event channel path is asynchronous, the OVR interrupt flag will not be set.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear it. It will also clear the corresponding flag in the Channel n Interrupt Flag Status and Clear register (CHINTFLAGn) of this peripheral, where n is determined by the Channel ID bit field (ID) in this register.

Bits 1:0 – ID[1:0] Channel ID
These bits store the channel number of the highest priority.

When the bits are written, indirect access to the corresponding Channel Interrupt Flag register is enabled.

Downloaded from Arrow.com.

33.7.5 Interrupt Status

Name: INTSTATUS
Offset: 0x14
Reset: 0x00000000
Property: Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, read accesses (R*) are allowed only if the security attribution for the corresponding channel (CHANNELx) is set as Non-Secured in the NONSECCHAN register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					CHINT3	CHINT2	CHINT1	CHINT0
Access					R/R*/R	R/R*/R	R/R*/R	R/R*/R
Reset					0	0	0	0

Bits 0, 1, 2, 3 – CHINT Channel x Pending Interrupt
This bit is set when Channel x has a pending interrupt.

This bit is cleared when the corresponding Channel x interrupts are disabled, or the source interrupt sources are cleared.

Downloaded from Arrow.com.

33.7.6 Busy Channels

Name: BUSYCH
Offset: 0x18
Reset: 0x00000000
Property: Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, read accesses (R*) are allowed only if the security attribution for the corresponding channel (CHANNELx) is set as Non-Secured in the NONSECCHAN register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					BUSYCHx3	BUSYCHx2	BUSYCHx1	BUSYCHx0
Access					R/R*/R	R/R*/R	R/R*/R	R/R*/R
Reset					0	0	0	0

Bits 0, 1, 2, 3 – BUSYCHx Busy Channel x
This bit is set if an event occurs on channel x has not been handled by all event users connected to channel x.

This bit is cleared when channel x is idle.

When the event channel x path is asynchronous, this bit is always read '0'.

Downloaded from Arrow.com.

33.7.7 Ready Users

Name: READYUSR
Offset: 0x1C
Reset: 0x0000000F
Property: Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, read accesses (R*) are allowed only if the security attribution for the corresponding channel (CHANNELx) is set as Non-Secured in the NONSECCHAN register.

Bit	31	30	29	28	27	26	25	24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					READYUSR3	READYUSR2	READYUSR1	READYUSR0
Access	R	R	R	R	R/R*/R	R/R*/R	R/R*/R	R/R*/R
Reset	0	0	0	0	1	1	1	1

Bits 0, 1, 2, 3 – READYUSR Ready User for Channel n
This bit is set when all event users connected to channel n are ready to handle incoming events on channel n.

This bit is cleared when at least one of the event users connected to the channel is not ready.

When the event channel n path is asynchronous, this bit is always read zero.

Downloaded from Arrow.com.

33.7.8 Channel n Control

Name: CHANNEL
Offset: 0x20 + n*0x08 [n=0..7]
Reset: 0x00008000
Property: PAC Write-Protection, Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the security attribution for the corresponding channel (CHANNELx) is set as Non-Secured in the NONSECCHAN register.

This register allows the user to configure channel n. To write to this register, do a single, 32-bit write of all the configuration data.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
Access	RW/RW*/RW	RW/RW*/RW			RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	1	0			0	0	0	0

Bit	7	6	5	4	3	2	1	0
			EVGEN[5:0]					
Access			RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset			0	0	0	0	0	0

Bit 15 – ONDEMAND Generic Clock On Demand

Value	Description
0	Generic clock for a channel is always on, if the channel is configured and generic clock source is enabled.
1	Generic clock is requested on demand while an event is handled

Bit 14 – RUNSTDBY Run in Standby
This bit is used to define the behavior during standby sleep mode.

Value	Description
0	The channel is disabled in standby sleep mode.
1	The channel is not stopped in standby sleep mode and depends on the CHANNEL.ONDEMAND bit.

Downloaded from Arrow.com.

Bits 11:10 – EDGSEL[1:0] Edge Detection Selection
These bits set the type of edge detection to be used on the channel.

These bits must be written to zero when using the asynchronous path.

Value	Name	Description
0x0	NO_EVT_OUTPUT	No event output when using the resynchronized or synchronous path
0x1	RISING_EDGE	Event detection only on the rising edge of the signal from the event generator
0x2	FALLING_EDGE	Event detection only on the falling edge of the signal from the event generator
0x3	BOTH_EDGES	Event detection on rising and falling edges of the signal from the event generator

Bits 9:8 – PATH[1:0] Path Selection
These bits are used to choose which path will be used by the selected channel.

Note: The path choice can be limited by the channel source, see the table in 33.7.13 USERm.

Important: Only EVSYS channel 0 to 3 can be configured as synchronous or resynchronized.

Value	Name	Description
0x0	SYNCHRONOUS	Synchronous path
0x1	RESYNCHRONIZED	Resynchronized path
0x2	ASYNCHRONOUS	Asynchronous path
Other	-	Reserved

Bits 5:0 – EVGEN[5:0] Event Generator Selection
These bits are used to choose the event generator to connect to the selected channel.

Table 33-2. Event Generators

Value	Event Generator	Description
0x00	NONE	No event generator selected
0x01	OSCCTRL_XOSC_FAIL	XOSC fail detection
0x02	OSC32KCTRL_XOSC32K_FAIL	XOSC32K fail detection
0x03	SUPC_BOD33DET	SUPC BOD33 detection
0x04-0x0B	RTC_PER	RTC period
0x0C-0x0D	RTC_CMP	RTC comparison
0x0E	RTC_TAMPER	RTC tamper detection
0x0F	RTC_OVF	RTC overflow
0x10	RTC_PERD	RTC periodic interval daily
0x11-0x18	EIC_EXTINT	EIC external interrupt

Downloaded from Arrow.com.

Value	Event Generator	Description
0x19-0x1C	DMAC_CH	DMAC channel
0x1D	TC0_OVF	TC0 overflow
0x1E-0x1F	TC0_MCX	TC0 match/compare
0x20	TC1_OVF	TC1 overflow
0x21-0x22	TC1_MCX	TC1 match/compare
0x23	TC2_OVF	TC2 overflow
0x24-0x25	TC2_MCX	TC2 match/compare
0x26	ADC_RESRDY	ADC resolution ready
0x27	ADC_WINMON	ADC window monitor
0x28-0x29	AC_COMP	AC comparator
0x2A	AC_WIN	AC window
0x2B	DAC_EMPTY	DAC empty
0x2C	PTC_EOC	PTC end of conversion
0x2D	PTC_WCOMP	PTC window comparator
0x2E	TRNG_READY	Data ready
0x2F-0x30	CCL_LUTOUT	CCL output
0x31	PAC_ERR	PAC access error

Downloaded from Arrow.com.

33.7.9 Channel n Interrupt Enable Clear

Name: CHINTENCLR
Offset: 0x24 + n*0x08 [n=0..7]
Reset: 0x00
Property: PAC Write-Protection, Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the security attribution for the corresponding channel (CHANNELx) is set as Non-Secured in the NONSECCHAN register.

Bit	7	6	5	4	3	2	1	0
							EVD	OVR
Access							RW/RW*/RW	RW/RW*/RW
Reset							0	0

Bit 1 – EVD Channel Event Detected Interrupt Disable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Event Detected Channel Interrupt Enable bit, which disables the Event Detected Channel interrupt.

Value	Description
0	The Event Detected Channel interrupt is disabled.
1	The Event Detected Channel interrupt is enabled.

Bit 0 – OVR Channel Overrun Interrupt Disable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Overrun Channel Interrupt Enable bit, which disables the Overrun Channel interrupt.

Value	Description
0	The Overrun Channel interrupt is disabled.
1	The Overrun Channel interrupt is enabled.

Downloaded from Arrow.com.

33.7.10 Channel n Interrupt Enable Set

Name: CHINTENSET
Offset: 0x25 + n*0x08 [n=0..7]
Reset: 0x00
Property: PAC Write-Protection, Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the security attribution for the corresponding channel (CHANNELx) is set as Non-Secured in the NONSECCHAN register.

Bit	7	6	5	4	3	2	1	0
							EVD	OVR
Access							RW/RW*/RW	RW/RW*/RW
Reset							0	0

Bit 1 – EVD Channel Event Detected Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Event Detected Channel Interrupt Enable bit, which enables the Event Detected Channel interrupt.

Value	Description
0	The Event Detected Channel interrupt is disabled.
1	The Event Detected Channel interrupt is enabled.

Bit 0 – OVR Channel Overrun Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Overrun Channel Interrupt Enable bit, which enables the Overrun Channel interrupt.

Value	Description
0	The Overrun Channel interrupt is disabled.
1	The Overrun Channel interrupt is enabled.

Downloaded from Arrow.com.

33.7.11 Channel n Interrupt Flag Status and Clear

Name: CHINTFLAG
Offset: 0x26 + n*0x08 [n=0..7]
Reset: 0x00
Property: Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the security attribution for the corresponding channel (CHANNELx) is set as Non-Secured in the NONSECCHAN register.

Bit	7	6	5	4	3	2	1	0
							EVD	OVR
Access							RW/RW*/RW	RW/RW*/RW
Reset							0	0

Bit 1 – EVD Channel Event Detected
This flag is set on the next CLK_EVSYS_APB cycle when an event is being propagated through the channel, and an interrupt request will be generated if CHINTENCLR/SET.EVD is '1'.

When the event channel path is asynchronous, the EVD interrupt flag will not be set.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Event Detected Channel interrupt flag.

Bit 0 – OVR Channel Overrun
This flag is set on the next CLK_EVSYS cycle after an overrun channel condition occurs, and an interrupt request will be generated if CHINTENCLR/SET.OVRx is '1'.

There are two possible overrun channel conditions:

- One or more of the event users on the channel are not ready when a new event occurs.
- An event happens when the previous event on channel has not yet been handled by all event users.

When the event channel path is asynchronous, the OVR interrupt flag will not be set.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Overrun Channel interrupt flag.

Downloaded from Arrow.com.

33.7.12 Channel n Status

Name: CHSTATUSn
Offset: 0x27 + n*0x08 [n=0..7]
Reset: 0x01
Property: Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, read accesses (R*) are allowed only if the security attribution for the corresponding channel (CHANNELx) is set as Non-Secured in the NONSECCHAN register.

Bit	7	6	5	4	3	2	1	0
							BUSYCH	RDYUSR
Access							R/R*/R	R/R*/R
Reset							0	0

Bit 1 – BUSYCH Busy Channel
This bit is cleared when channel is idle.

This bit is set if an event on channel has not been handled by all event users connected to channel.

When the event channel path is asynchronous, this bit is always read '0'.

Bit 0 – RDYUSR Ready User
This bit is cleared when at least one of the event users connected to the channel is not ready.

This bit is set when all event users connected to channel are ready to handle incoming events on the channel.

When the event channel path is asynchronous, this bit is always read zero.

Downloaded from Arrow.com.

33.7.13 Event User m

Name: USERm
Offset: 0x0120 + m*0x01 [m=0..22]
Reset: 0x0
Property: PAC Write-Protection, Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the security attribution for the corresponding event user (USERx) is set as Non-Secured in the NONSECUSER register.

Bit	7	6	5	4	3	2	1	0
					CHANNEL[3:0]			
Access					R/W/RW*/RW	R/W/RW*/RW	R/W/RW*/RW	R/W/RW*/RW
Reset					0	0	0	0

Bits 3:0 – CHANNEL[3:0] Channel Event Selection
These bits select channel n to connect to the event user m.

Note: A value x of this bit field selects channel n = x-1.

Table 33-3. User Multiplexer Number m

USERm	User Multiplexer	Description	Path Type
m=0	OSCCTRL_TUNE	DFLLULP Tune	A
m=1	RTC_TAMPER	RTC Tamper	A
m=2	NWMCTRL_PAGEW	NVMCTRL Auto-Write	A,S,R
m=3..6	PORT_EV[0..3]	Port Event 0..3	A
m=7..10	DMAC_CH[0..3]	Channel 0...3	S,R
m=11	TC0_EVU	TC0 EVU	A,S,R
m=12	TC1_EVU	TC1 EVU	A,S,R
m=13	TC2_EVU	TC2 EVU	A,S,R
m=14	ADC_START	ADC Start Conversion	A,S,R
m=15	ADC_SYNC	Flush ADC	A,S,R
m=16..17	AC_COMP[0..1]	Start Comparator 0..1	A
m=18	DAC_START	DAC Start Conversion	A
m=19	PTC_STCONV	PTC Start Conversion	A,S,R
m=20	PTC_DSEQR	PTC Sequencing	A,S,R
m=21..22	CCL_LUTIN[0..1]	CCL Input 0..1	A

1) A = Asynchronous path, S = Synchronous path, R = Resynchronized path

Downloaded from Arrow.com.

Value	Description
0x00	No channel selected
0x01	Channel 0 selected
0x02	Channel 1 selected
0x03	Channel 2 selected
0x04	Channel 3 selected
0x05	Channel 4 selected
0x06	Channel 5 selected
0x07	Channel 6 selected
0x08	Channel 7 selected

Downloaded from Arrow.com.

33.7.14 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x1D4
Reset: 0x0
Property: PAC Write-Protection

Important: This register is only available for **SAM L11** and has no effect for **SAM L10**.

Bit	7	6	5	4	3	2	1	0
								NSCHK
Access								RW/RW/RW
Reset								0

Bit 0 – NSCHK Non-Secure Check Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Non-Secure Check Interrupt Enable bit, which disables the Non-Secure Check interrupt.

Value	Description
0	The Non-Secure Check interrupt is disabled.
1	The Non-Secure Check interrupt is enabled.

Downloaded from Arrow.com.

33.7.15 Interrupt Enable Set

Name: INTENSET
Offset: 0x1D5
Reset: 0x0
Property: PAC Write-Protection

Important: This register is only available for **SAM L11** and has no effect for **SAM L10**.

Bit	7	6	5	4	3	2	1	0
								NSCHK
Access								RW/RW/RW
Reset								0

Bit 0 – NSCHK Non-Secure Check Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Non-Secure Check Interrupt Enable bit, which disables the Non-Secure Check interrupt.

Value	Description
0	The Non-Secure Check interrupt is disabled.
1	The Non-Secure Check interrupt is enabled.

Downloaded from Arrow.com.

33.7.16 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x1D6
Reset: 0x0
Property: -

Important: This register is only available for **SAM L11** and has no effect for **SAM L10**.

Bit	7	6	5	4	3	2	1	0
								NSCHK
Access								RW/RW/RW
Reset								0

Bit 0 – NSCHK Non-Secure Check
This flag is set when a bit in NSCHKCHAN is 1 and the corresponding bit in NONSECCHAN is cleared, or when a bit in NSCHKCHAN is 0 and the corresponding bit in NONSECCHAN is set, or when a bit in NSCHKUSER is 1 and the corresponding bit in NONSECUSER is cleared, or when a bit in NSCHKUSER is 0 and the corresponding bit in NONSECUSER is set.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Non-Secure Check interrupt flag.

Downloaded from Arrow.com.

33.7.17 Channel Security Attribution

Name: NONSECCHAN
Offset: 0x1D8
Reset: 0x00000000
Property: PAC Write-Protection, Write-Secure

This register allows the user to configure one or more channels as secured or non-secured.

Important: This register is only available for **SAM L11** and has no effect for **SAM L10**.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CHANNELn[7:0]							
Access	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – CHANNELn[7:0] Channel n Security Attribution [n=7..0]The bit n of CHANNEL enables the non-secure mode of CHANNELn. The registers whose CHANNEL bit or bitfield n is set in non-secure mode by NONSECCHAN.CHANNELn are CHANNELn, CHINTENCLRn, CHINTENSETn, CHINTFLAGn and CHSTATUSx registers.
These bits set the security attribution for the individual channels.

Value	Description
0	The corresponding channel is secured. When the module is PAC secured, the configuration and status bits for this channel are only available through the secure alias. Attempts to change the channel configuration through the non-secure alias will be silently ignored and reads will return 0.
1	The corresponding channel is non-secured. The configuration and status bits for this channel are available through the non-secure alias.

Downloaded from Arrow.com.

33.7.18 Channel Security Attribution Check

Name: NSCHKCHAN
Offset: 0x1DC
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to select one or more channels to check their security attribution as non-secured.

Important: This register is only available for **SAM L11** and has no effect for **SAM L10**.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CHANNELn[7:0]							
Access	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – CHANNELn[7:0] Channel n Selection [n=7..0]
These bits selects the individual channels for security attribution check. If any channel selected in NSCHKCHAN has the corresponding bit in NONSECCHAN set to the opposite value, then the NSCHK interrupt flag will be set.

Value	Description
0	0-to-1 transition will be detected on corresponding NONSECCHAN bit.
1	1-to-0 transition will be detected on corresponding NONSECCHAN bit.

Downloaded from Arrow.com.

33.7.19 Event User Security Attribution

Name: NONSECUSERm
Offset: 0x01E0 + m*0x04 [m=0..1]
Reset: 0x00000000
Property: PAC Write-Protection, Write-Secure

This register allows the user to configure one or more event users as secured or non-secured.

Important: This register is only available for **SAM L11** and has no effect for **SAM L10**.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		USERn[22:16]						
Access		RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	USERn[15:8]							
Access	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	USERn[7:0]							
Access	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW
Reset	0	0	0	0	0	0	0	0

Bits 22:0 – USERn[22:0] Event User n Security Attribution [n=22..0]The bit n of USER enables the non-secure mode of USERn. The registers whose USER bit or bitfield n is set in non-secure mode by NONSECUSER.USERn are USERn registers.
These bits set the security attribution for the individual event users.

Value	Description
0	The corresponding event user is secured. When the module is PAC secured, the configuration and status bits for this event user are only available through the secure alias. Attempts to change the event user configuration through the non-secure alias will be silently ignored and reads will return 0.
1	The corresponding event user is non-secured. The configuration and status bits for this event user are available through the non-secure alias.

Downloaded from Arrow.com.

33.7.20 Event User Security Attribution Check

Name: NSCHKUSERm
Offset: 0x01F0 + m*0x04 [m=0..1]
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to select one or more event users to check their security attribution as non-secured.

Important: This register is only available for **SAM L11** and has no effect for **SAM L10**.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		USERn[22:16]						
Access		RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	USERn[15:8]							
Access	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	USERn[7:0]							
Access	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW
Reset	0	0	0	0	0	0	0	0

Bits 22:0 – USERn[22:0] Event User n Selection [n=22..0]
These bits selects the individual event users for security attribution check. If any event user selected in NSCHKUSER has the corresponding bit in NONSECUSER set to the opposite value, then the NSCHK interrupt flag will be set.

Value	Description
0	0-to-1 transition will be detected on corresponding NONSECUSER bit.
1	1-to-0 transition will be detected on corresponding NONSECUSER bit.

Downloaded from Arrow.com.

34. SERCOM – Serial Communication Interface

34.1 Overview

There are up to three instances of the serial communication interface (SERCOM) peripheral.

A SERCOM can be configured to support a number of modes: I^2C, SPI, and USART. When an instance of SERCOM is configured and enabled, all of the resources of that SERCOM instance will be dedicated to the selected mode.

The SERCOM serial engine consists of a transmitter and receiver, baud-rate generator and address matching functionality. It can use the internal generic clock or an external clock. Using an external clock allows the SERCOM to be operated in all Sleep modes.

Related Links

35. SERCOM USART - SERCOM Synchronous and Asynchronous Receiver and Transmitter
36. SERCOM SPI – SERCOM Serial Peripheral Interface
37. SERCOM I2C – SERCOM Inter-Integrated Circuit

34.2 Features

- Interface for configuring into one of the following:
  - Inter-Integrated Circuit (I^2C) Two-wire Serial Interface
  - System Management Bus (SMBus™) compatible
  - Serial Peripheral Interface (SPI)
  - Universal Synchronous/Asynchronous Receiver/Transmitter (USART)
- Single transmit buffer and double receive buffer
- Baud-rate generator
- Address match/mask logic
- Operational in all Sleep modes with an external clock source
- Can be used with DMA

See the Related Links for full feature lists of the interface configurations.

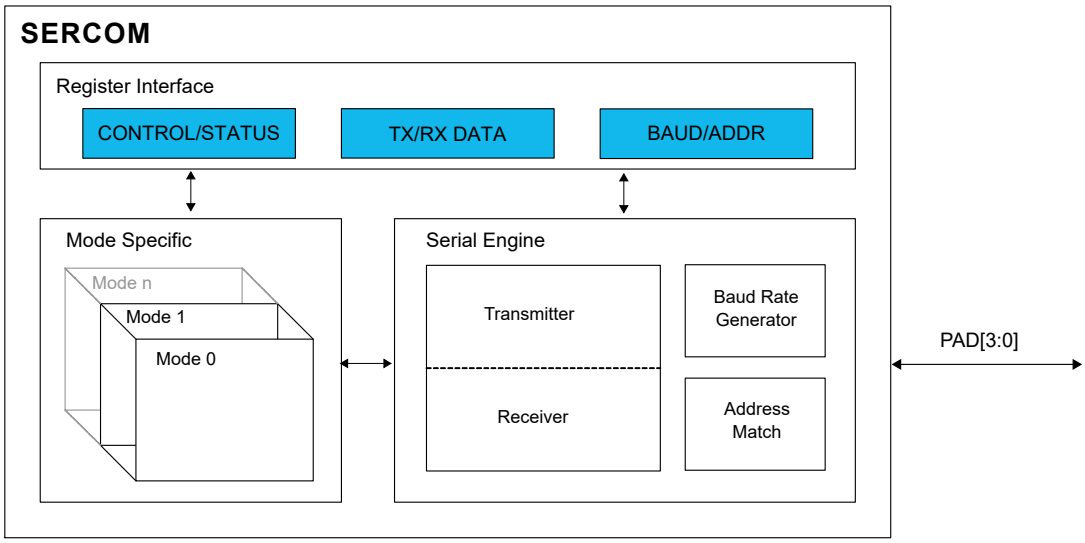
Related Links

35. SERCOM USART - SERCOM Synchronous and Asynchronous Receiver and Transmitter
36. SERCOM SPI – SERCOM Serial Peripheral Interface
37. SERCOM I2C – SERCOM Inter-Integrated Circuit

Downloaded from Arrow.com.

34.3 Block Diagram

Figure 34-1. SERCOM Block Diagram



34.4 Signal Description

See the respective SERCOM mode chapters for details.

Related Links

35. SERCOM USART - SERCOM Synchronous and Asynchronous Receiver and Transmitter
36. SERCOM SPI – SERCOM Serial Peripheral Interface
37. SERCOM I2C – SERCOM Inter-Integrated Circuit

34.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

34.5.1 I/O Lines

Using the SERCOM I/O lines requires the I/O pins to be configured using port configuration (PORT).

The SERCOM has four internal pads, PAD[3:0], and the signals from I2C, SPI and USART are routed through these SERCOM pads via a multiplexer. The configuration of the multiplexer is available from the different SERCOM modes. Refer to the mode specific chapters for details.

Related Links

35. SERCOM USART - SERCOM Synchronous and Asynchronous Receiver and Transmitter
36. SERCOM SPI – SERCOM Serial Peripheral Interface
37. SERCOM I2C – SERCOM Inter-Integrated Circuit
32. PORT - I/O Pin Controller
35.3 Block Diagram

34.5.2 Power Management

The SERCOM can operate in any Sleep mode provided the selected clock source is running. SERCOM interrupts can be configured to wake the device from Sleep modes.

Downloaded from Arrow.com.

Related Links

22. PM – Power Manager

34.5.3 Clocks

The SERCOM bus clock (CLK_SERCOMx_APB) can be enabled and disabled in the Main Clock Controller. Refer to *Peripheral Clock Masking* for details and default status of this clock.

The SERCOM uses two generic clocks: GCLK_SERCOMx_CORE and GCLK_SERCOMx_SLOW. The core clock (GCLK_SERCOMx_CORE) is required to clock the SERCOM while working as a master. The slow clock (GCLK_SERCOMx_SLOW) is only required for certain functions. See specific mode chapters for details.

These clocks must be configured and enabled in the Generic Clock Controller (GCLK) before using the SERCOM.

The generic clocks are asynchronous to the user interface clock (CLK_SERCOMx_APB). Due to this asynchronicity, writing to certain registers will require synchronization between the clock domains. Refer to 34.6.8 Synchronization for details.

Related Links

18. GCLK - Generic Clock Controller
19. MCLK – Main Clock

34.5.4 DMA

The DMA request lines are connected to the DMA Controller (DMAC). The DMAC must be configured before the SERCOM DMA requests are used.

Related Links

28. DMAC – Direct Memory Access Controller

34.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller (NVIC). The NVIC must be configured before the SERCOM interrupts are used.

34.5.6 Events

Not applicable.

34.5.7 Debug Operation

When the CPU is halted in debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

34.5.8 Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Interrupt Flag Clear and Status register (INTFLAG)
- Status register (STATUS)
- Data register (DATA)
- Address register (ADDR)

Downloaded from Arrow.com.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

Related Links

15. PAC - Peripheral Access Controller

34.5.9 SAM L11 TrustZone Specific Register Access Protection

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

34.5.10 Analog Connections

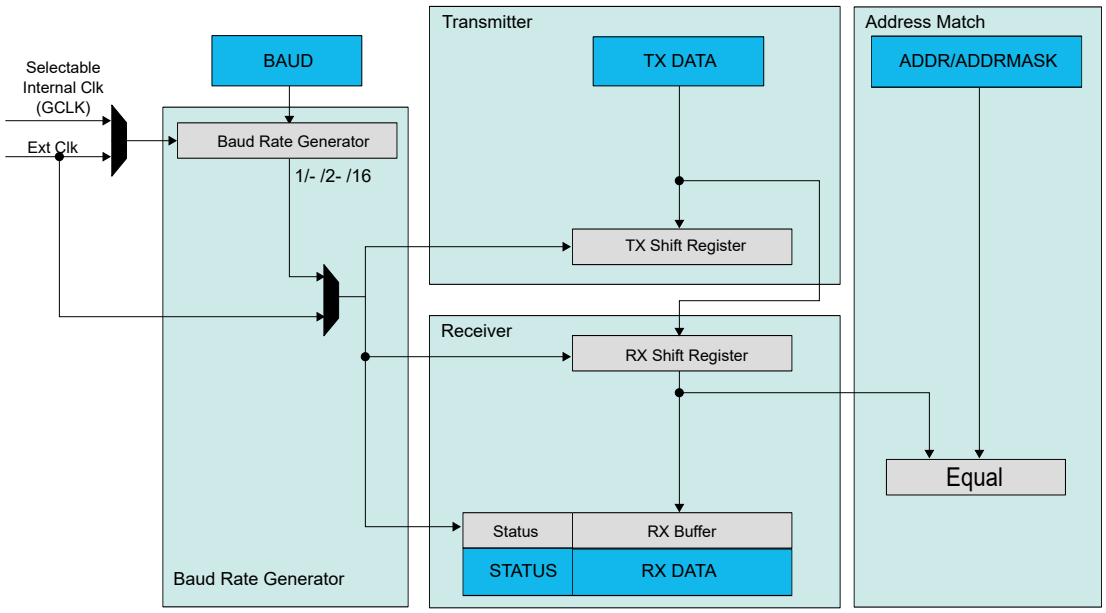
Not applicable.

34.6 Functional Description

34.6.1 Principle of Operation

The basic structure of the SERCOM serial engine is shown in Figure 34-2. Labels in capital letters are synchronous to the system clock and accessible by the CPU; labels in lowercase letters can be configured to run on the GCLK_SERCOMx_CORE clock or an external clock.

Figure 34-2. SERCOM Serial Engine



The transmitter consists of a single write buffer and a shift register.

Downloaded from Arrow.com.

The receiver consists of a one-level (I^2C), two-level or four-level (USART, SPI) receive buffer and a shift register.

The baud-rate generator is capable of running on the GCLK_SERCOMx_CORE clock or an external clock.

Address matching logic is included for SPI and I^2C operation.

34.6.2 Basic Operation

34.6.2.1 Initialization

The SERCOM must be configured to the desired mode by writing the Operating Mode bits in the Control A register (CTRLA.MODE). Refer to table SERCOM Modes for details.

Table 34-1. SERCOM Modes

CTRLA.MODE	Description
0x0	USART with external clock
0x1	USART with internal clock
0x2	SPI in slave operation
0x3	SPI in master operation
0x4	I^2C slave operation
0x5	I^2C master operation
0x6-0x7	Reserved

For further initialization information, see the respective SERCOM mode chapters:

Related Links

35. SERCOM USART - SERCOM Synchronous and Asynchronous Receiver and Transmitter
36. SERCOM SPI – SERCOM Serial Peripheral Interface
37. SERCOM I2C – SERCOM Inter-Integrated Circuit

34.6.2.2 Enabling, Disabling, and Resetting

This peripheral is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE), and disabled by writing '0' to it.

Writing ‘1’ to the Software Reset bit in the Control A register (CTRLA.SWRST) will reset all registers of this peripheral to their initial states, except the DBGCTRL register, and the peripheral is disabled.

Refer to the CTRLA register description for details.

34.6.2.3 Clock Generation – Baud-Rate Generator

The baud-rate generator, as shown in Figure 34-3, generates internal clocks for asynchronous and synchronous communication. The output frequency (f_{BAUD}) is determined by the Baud register (BAUD) setting and the baud reference frequency (f_{ref}). The baud reference clock is the serial engine clock, and it can be internal or external.

For asynchronous communication, the /16 (divide-by-16) output is used when transmitting, whereas the /1 (divide-by-1) output is used while receiving.

For synchronous communication, the /2 (divide-by-2) output is used.

Downloaded from Arrow.com.

This functionality is automatically configured, depending on the selected operating mode.

Figure 34-3. Baud Rate Generator

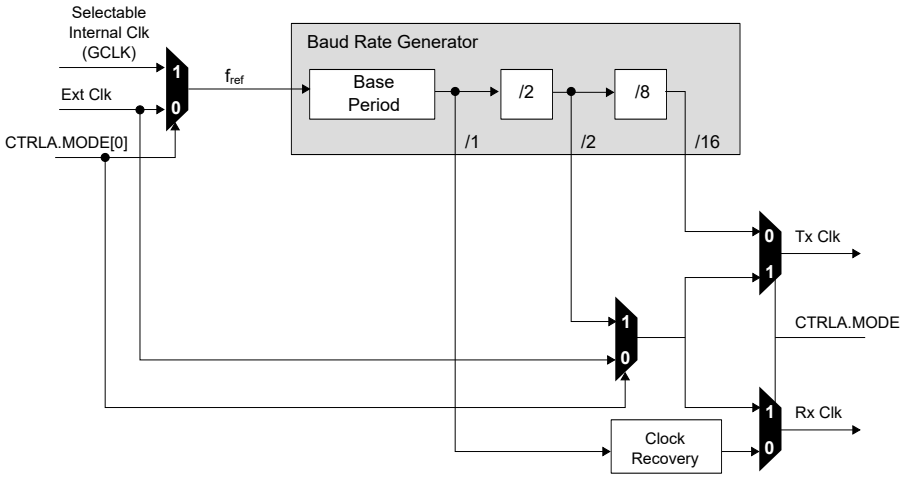


Table 34-2 contains equations for the baud rate (in bits per second) and the BAUD register value for each operating mode.

For asynchronous operation, there is one mode: *arithmetic mode*, the BAUD register value is 16 bits (0 to 65,535).

For synchronous operation, the BAUD register value is 8 bits (0 to 255).

Table 34-2. Baud Rate Equations

Operating Mode	Condition	Baud Rate (Bits Per Second)	BAUD Register Value Calculation
Asynchronous Arithmetic	$f_{BAUD} \leq \frac{f_{ref}}{16}$	$f_{BAUD} = \frac{f_{ref}}{16}\left(1 - \frac{BAUD}{65536}\right)$	$BAUD = 65536 \cdot \left(1 - 16 \cdot \frac{f_{BAUD}}{f_{ref}}\right)$
Asynchronous Fractional	$f_{BAUD} \leq \frac{f_{ref}}{S}$	$f_{BAUD} = \frac{f_{ref}}{S \cdot \left(BAUD + \frac{FP}{8}\right)}$	$BAUD = \frac{f_{ref}}{S \cdot f_{BAUD}} - \frac{FP}{8}$
Synchronous	$f_{BAUD} \leq \frac{f_{ref}}{2}$	$f_{BAUD} = \frac{f_{ref}}{2 \cdot (BAUD + 1)}$	$BAUD = \frac{f_{ref}}{2 \cdot f_{BAUD}} - 1$

S - Number of samples per bit, which can be 16, 8, or 3.

The Asynchronous Fractional option is used for auto-baud detection.

The baud rate error is represented by the following formula:

$$\text{Error} = 1 - \left(\frac{\text{ExpectedBaudRate}}{\text{ActualBaudRate}}\right)$$

34.6.2.3.1 Asynchronous Arithmetic Mode BAUD Value Selection

The formula given for f_{BAUD} calculates the average frequency over 65536 f_{ref} cycles. Although the BAUD register can be set to any value between 0 and 65536, the actual average frequency of f_{BAUD} over a single frame is more granular. The BAUD register values that will affect the average frequency over a single frame lead to an integer increase in the cycles per frame (CPF)

$$CPF = \frac{f_{ref}}{f_{BAUD}}(D + S)$$

Downloaded from Arrow.com.

where

- *D* represent the data bits per frame
- *S* represent the sum of start and first stop bits, if present.

Table 34-3 shows the BAUD register value versus baud frequency f_{BAUD} at a serial engine frequency of 48MHz. This assumes a *D* value of 8 bits and an *S* value of 2 bits (10 bits, including start and stop bits).

Table 34-3. BAUD Register Value vs. Baud Frequency

BAUD Register Value	Serial Engine CPF	f_{BAUD} at 48MHz Serial Engine Frequency (f_{REF})
0 – 406	160	3MHz
407 – 808	161	2.981MHz
809 – 1205	162	2.963MHz
...	...	...
65206	31775	15.11kHz
65207	31871	15.06kHz
65208	31969	15.01kHz

34.6.3 Additional Features

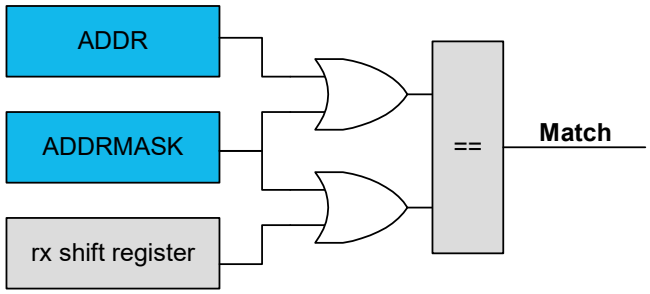
34.6.3.1 Address Match and Mask

The SERCOM address match and mask feature is capable of matching either one address, two unique addresses, or a range of addresses with a mask, based on the mode selected. The match uses seven or eight bits, depending on the mode.

34.6.3.1.1 Address With Mask

An address written to the Address bits in the Address register (ADDR.ADDR), and a mask written to the Address Mask bits in the Address register (ADDR.ADDRMASK) will yield an address match. All bits that are masked are not included in the match. Note that writing the ADDR.ADDRMASK to 'all zeros' will match a single unique address, while writing ADDR.ADDRMASK to 'all ones' will result in all addresses being accepted.

Figure 34-4. Address With Mask

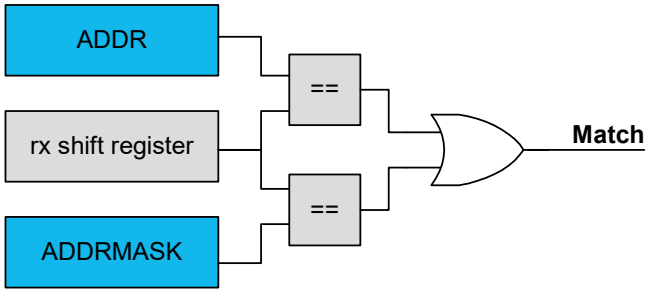


34.6.3.1.2 Two Unique Addresses

The two addresses written to ADDR and ADDRMASK will cause a match.

Downloaded from Arrow.com.

Figure 34-5. Two Unique Addresses



34.6.3.1.3 Address Range

The range of addresses between and including ADDR.ADDR and ADDR.ADDRMASK will cause a match. ADDR.ADDR and ADDR.ADDRMASK can be set to any two addresses, with ADDR.ADDR acting as the upper limit and ADDR.ADDRMASK acting as the lower limit.

Figure 34-6. Address Range

34.6.4 DMA Operation

The available DMA interrupts and their depend on the operation mode of the SERCOM peripheral. Refer to the Functional Description sections of the respective SERCOM mode.

Related Links

35. SERCOM USART - SERCOM Synchronous and Asynchronous Receiver and Transmitter
36. SERCOM SPI – SERCOM Serial Peripheral Interface
37. SERCOM I2C – SERCOM Inter-Integrated Circuit

34.6.5 Interrupts

Interrupt sources are mode-specific. See the respective SERCOM mode chapters for details.

Each interrupt source has its own interrupt flag.

The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) will be set when the interrupt condition is met.

Each interrupt can be individually enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is disabled, or the SERCOM is reset. For details on clearing interrupt flags, refer to the INTFLAG register description.

The value of INTFLAG indicates which interrupt condition occurred. The user must read the INTFLAG register to determine which interrupt condition is present.

Note: Interrupts must be globally enabled for interrupt requests.

34.6.6 Events

Not applicable.

Downloaded from Arrow.com.

34.6.7 Sleep Mode Operation

The peripheral can operate in any sleep mode where the selected serial clock is running. This clock can be external or generated by the internal baud-rate generator.

The SERCOM interrupts can be used to wake up the device from sleep modes. Refer to the different SERCOM mode chapters for details.

34.6.8 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Required read-synchronization is denoted by the "Read-Synchronized" property in the register description.

Downloaded from Arrow.com.

35. SERCOM USART - SERCOM Synchronous and Asynchronous Receiver and Transmitter

35.1 Overview

The Universal Synchronous and Asynchronous Receiver and Transmitter (USART) is one of the available modes in the Serial Communication Interface (SERCOM).

The USART uses the SERCOM transmitter and receiver, see 35.3 Block Diagram. Labels in uppercase letters are synchronous to CLK_SERCOMx_APB and accessible for CPU. Labels in lowercase letters can be programmed to run on the internal generic clock or an external clock.

The transmitter consists of a single write buffer, a shift register, and control logic for different frame formats. The write buffer support data transmission without any delay between frames. The receiver consists of a two-level or four-level receive buffer and a shift register. Status information of the received data is available for error checking. Data and clock recovery units ensure robust synchronization and noise filtering during asynchronous data reception.

Related Links

34. SERCOM – Serial Communication Interface

35.2 USART Features

- Full-duplex operation
- Asynchronous (with clock reconstruction) or synchronous operation
- Internal or external clock source for asynchronous and synchronous operation
- Baud-rate generator
- Supports serial frames with 5, 6, 7, 8 or 9 data bits and 1 or 2 stop bits
- Odd or even parity generation and parity check
- Selectable LSB- or MSB-first data transfer
- Buffer overflow and frame error detection
- Noise filtering, including false start-bit detection and digital low-pass filter
- Collision detection
- Can operate in all sleep modes
- Operation at speeds up to half the system clock for internally generated clocks
- Operation at speeds up to the system clock for externally generated clocks
- RTS and CTS flow control
- IrDA modulation and demodulation up to 115.2kbps
- LIN slave support
  - Auto-baud and break character detection
- ISO 7816 T=0 or T=1 protocols for Smart Card interfacing
- RS485 Support
- Start-of-frame detection
- Two- or Four-Level Receive Buffer
- Can work with DMA

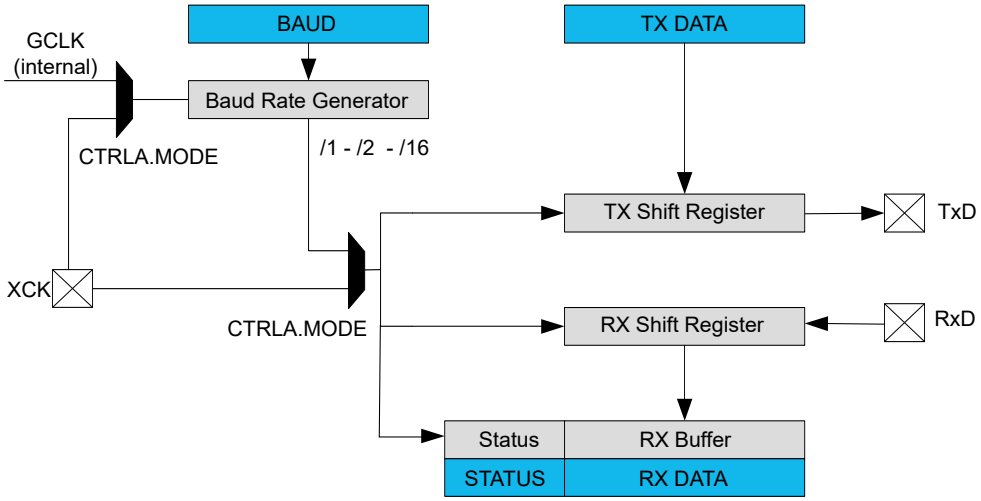
Downloaded from Arrow.com.

Related Links

34.2 Features

35.3 Block Diagram

Figure 35-1. USART Block Diagram



35.4 Signal Description

Table 35-1. SERCOM USART Signals

Signal Name	Type	Description
PAD[3:0]	Digital I/O	General SERCOM pins

One signal can be mapped to one of several pins.

35.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

35.5.1 I/O Lines

Using the USART's I/O lines requires the I/O pins to be configured using the I/O Pin Controller (PORT).

When the SERCOM is used in USART mode, the SERCOM controls the direction and value of the I/O pins according to the table below. Both PORT control bits PINCFGn.PULLEN and PINCFGn.DRVSTR are still effective. If the receiver or transmitter is disabled, these pins can be used for other purposes.

Table 35-2. USART Pin Configuration

Pin	Pin Configuration
TxD	Output
RxD	Input
XCK	Output or input

Downloaded from Arrow.com.

The combined configuration of PORT and the Transmit Data Pinout and Receive Data Pinout bit fields in the Control A register (CTRLA.TXPO and CTRLA.RXPO, respectively) will define the physical position of the USART signals in Table 35-2.

Related Links

32. PORT - I/O Pin Controller

35.5.2 Power Management

This peripheral can continue to operate in any sleep mode where its source clock is running. The interrupts can wake up the device from sleep modes.

Related Links

22. PM – Power Manager

35.5.3 Clocks

The SERCOM bus clock (CLK_SERCOMx_APB) can be enabled and disabled in the Main Clock Controller. Refer to *Peripheral Clock Masking* for details and default status of this clock.

A generic clock (GCLK_SERCOMx_CORE) is required to clock the SERCOMx_CORE. This clock must be configured and enabled in the Generic Clock Controller before using the SERCOMx_CORE. Refer to *GCLK - Generic Clock Controller* for details.

This generic clock is asynchronous to the bus clock (CLK_SERCOMx_APB). Therefore, writing to certain registers will require synchronization to the clock domains. Refer to *Synchronization* for further details.

Related Links

19.6.2.6 Peripheral Clock Masking

35.6.6 Synchronization

18. GCLK - Generic Clock Controller

35.5.4 DMA

The DMA request lines are connected to the DMA Controller (DMAC). In order to use DMA requests with this peripheral the DMAC must be configured first. Refer to *DMAC – Direct Memory Access Controller* for details.

Related Links

28. DMAC – Direct Memory Access Controller

35.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. Refer to *Nested Vector Interrupt Controller* for details.

35.5.6 Events

Not applicable.

35.5.7 Debug Operation

When the CPU is halted in debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

Related Links

Downloaded from Arrow.com.

35.8.13 DBGCTRL

35.5.8 Register Access Protection

Registers with write-access can be write-protected optionally by the peripheral access controller (PAC).

PAC Write-Protection is not available for the following registers:

- Interrupt Flag Clear and Status register (INTFLAG)
- Status register (STATUS)
- Data register (DATA)

Optional PAC Write-Protection is denoted by the "PAC Write-Protection" property in each individual register description.

Write-protection does not apply to accesses through an external debugger.

Related Links

15. PAC - Peripheral Access Controller

35.5.9 SAM L11 TrustZone Specific Register Access Protection

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

35.5.10 Analog Connections

Not applicable.

35.6 Functional Description

35.6.1 Principle of Operation

The USART uses the following lines for data transfer:

- RxD for receiving
- TxD for transmitting
- XCK for the transmission clock in synchronous operation

USART data transfer is frame based. A serial frame consists of:

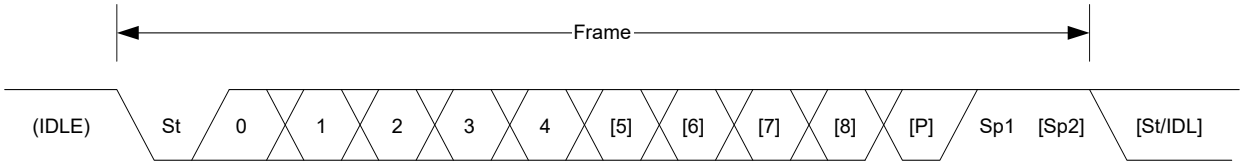
- 1 start bit
- From 5 to 9 data bits (MSB or LSB first)
- No, even or odd parity bit
- 1 or 2 stop bits

A frame starts with the start bit followed by one character of data bits. If enabled, the parity bit is inserted after the data bits and before the first stop bit. After the stop bit(s) of a frame, either the next frame can

Downloaded from Arrow.com.

follow immediately, or the communication line can return to the idle (high) state. The figure below illustrates the possible frame formats. Brackets denote optional bits.

Figure 35-2. Frame Formats



St Start bit. Signal is always low.

n, [n] Data bits. 0 to [5..9]

[P] Parity bit. Either odd or even.

Sp, [Sp] Stop bit. Signal is always high.

IDLE No frame is transferred on the communication line. Signal is always high in this state.

35.6.2 Basic Operation

35.6.2.1 Initialization

The following registers are enable-protected, meaning they can only be written when the USART is disabled (CTRL.ENABLE=0):

- Control A register (CTRLA), except the Enable (ENABLE) and Software Reset (SWRST) bits.
- Control B register (CTRLB), except the Receiver Enable (RXEN) and Transmitter Enable (TXEN) bits.
- Baud register (BAUD)

When the USART is enabled or is being enabled (CTRLA.ENABLE=1), any writing attempt to these registers will be discarded. If the peripheral is being disabled, writing to these registers will be executed after disabling is completed. Enable-protection is denoted by the "Enable-Protection" property in the register description.

Before the USART is enabled, it must be configured by these steps:

1. Select either external (0x0) or internal clock (0x1) by writing the Operating Mode value in the CTRLA register (CTRLA.MODE).
2. Select either asynchronous (0) or or synchronous (1) communication mode by writing the Communication Mode bit in the CTRLA register (CTRLA.CMODE).
3. Select pin for receive data by writing the Receive Data Pinout value in the CTRLA register (CTRLA.RXPO).
4. Select pads for the transmitter and external clock by writing the Transmit Data Pinout bit in the CTRLA register (CTRLA.TXPO).
5. Configure the Character Size field in the CTRLB register (CTRLB.CHSIZE) for character size.
6. Set the Data Order bit in the CTRLA register (CTRLA.DORD) to determine MSB- or LSB-first data transmission.
7. To use parity mode:
   7.1. Enable parity mode by writing 0x1 to the Frame Format field in the CTRLA register (CTRLA.FORM).

Downloaded from Arrow.com.

7.2. Configure the Parity Mode bit in the CTRLB register (CTRLB.PMODE) for even or odd parity.

8. Configure the number of stop bits in the Stop Bit Mode bit in the CTRLB register (CTRLB.SBMODE).
9. When using an internal clock, write the Baud register (BAUD) to generate the desired baud rate.
10. Enable the transmitter and receiver by writing '1' to the Receiver Enable and Transmitter Enable bits in the CTRLB register (CTRLB.RXEN and CTRLB.TXEN).

35.6.2.2 Enabling, Disabling, and Resetting

This peripheral is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE), and disabled by writing '0' to it.

Writing '1' to the Software Reset bit in the Control A register (CTRLA.SWRST) will reset all registers of this peripheral to their initial states, except the DBGCTRL register, and the peripheral is disabled.

Refer to the CTRLA register description for details.

35.6.2.3 Clock Generation and Selection

For both synchronous and asynchronous modes, the clock used for shifting and sampling data can be generated internally by the SERCOM baud-rate generator or supplied externally through the XCK line.

The synchronous mode is selected by writing a '1' to the Communication Mode bit in the Control A register (CTRLA.CMODE), the asynchronous mode is selected by writing a zero to CTRLA.CMODE.

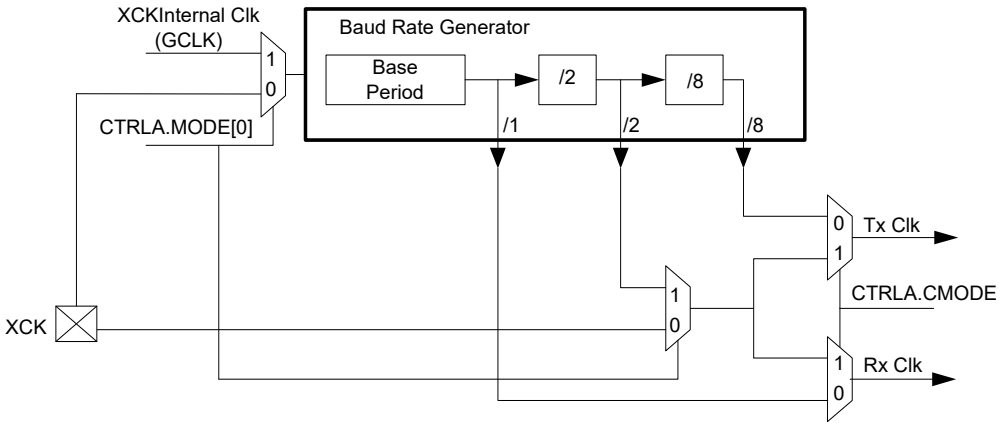
The internal clock source is selected by writing 0x1 to the Operation Mode bit field in the Control A register (CTRLA.MODE), the external clock source is selected by writing 0x0 to CTRLA.MODE.

The SERCOM baud-rate generator is configured as in the figure below.

In asynchronous mode (CTRLA.CMODE=0), the 16-bit Baud register value is used.

In synchronous mode (CTRLA.CMODE=1), the eight LSBs of the Baud register are used. Refer to *Clock Generation – Baud-Rate Generator* for details on configuring the baud rate.

Figure 35-3. Clock Generation



Related Links

34.6.2.3 Clock Generation – Baud-Rate Generator

34.6.2.3.1 Asynchronous Arithmetic Mode BAUD Value Selection

35.6.2.3.1 Synchronous Clock Operation

In synchronous mode, the CTRLA.MODE bit field determines whether the transmission clock line (XCK) serves either as input or output. The dependency between clock edges, data sampling, and data change

Downloaded from Arrow.com.

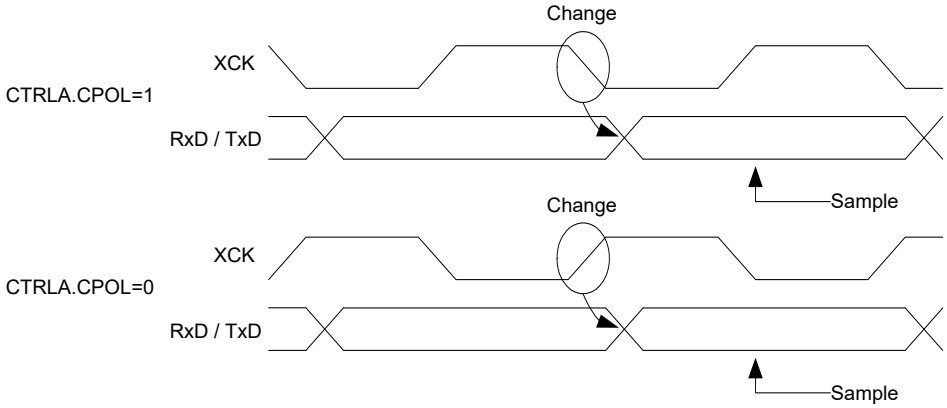
is the same for internal and external clocks. Data input on the RxD pin is sampled at the opposite XCK clock edge when data is driven on the TxD pin.

The Clock Polarity bit in the Control A register (CTRLA.CPOL) selects which XCK clock edge is used for RxD sampling, and which is used for TxD change:

When CTRLA.CPOL is '0', the data will be changed on the rising edge of XCK, and sampled on the falling edge of XCK.

When CTRLA.CPOL is '1', the data will be changed on the falling edge of XCK, and sampled on the rising edge of XCK.

Figure 35-4. Synchronous Mode XCK Timing



When the clock is provided through XCK (CTRLA.MODE=0x0), the shift registers operate directly on the XCK clock. This means that XCK is not synchronized with the system clock and, therefore, can operate at frequencies up to the system frequency.

35.6.2.4 Data Register

The USART Transmit Data register (TxDATA) and USART Receive Data register (RxDATA) share the same I/O address, referred to as the Data register (DATA). Writing the DATA register will update the TxDATA register. Reading the DATA register will return the contents of the RxDATA register.

35.6.2.5 Data Transmission

Data transmission is initiated by writing the data to be sent into the DATA register. Then, the data in TxDATA will be moved to the shift register when the shift register is empty and ready to send a new frame. After the shift register is loaded with data, the data frame will be transmitted.

When the entire data frame including stop bit(s) has been transmitted and no new data was written to DATA, the Transmit Complete interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.TXC) will be set, and the optional interrupt will be generated.

The Data Register Empty flag in the Interrupt Flag Status and Clear register (INTFLAG.DRE) indicates that the register is empty and ready for new data. The DATA register should only be written to when INTFLAG.DRE is set.

35.6.2.5.1 Disabling the Transmitter

The transmitter is disabled by writing '0' to the Transmitter Enable bit in the CTRLB register (CTRLB.TXEN).

Disabling the transmitter will complete only after any ongoing and pending transmissions are completed, i.e., there is no data in the transmit shift register and TxDATA to transmit.

Downloaded from Arrow.com.

35.6.2.6 Data Reception

The receiver accepts data when a valid start bit is detected. Each bit following the start bit will be sampled according to the baud rate or XCK clock, and shifted into the receive shift register until the first stop bit of a frame is received. The second stop bit will be ignored by the receiver.

When the first stop bit is received and a complete serial frame is present in the receive shift register, the contents of the shift register will be moved into the two-level or four-level receive buffer. Then, the Receive Complete interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) will be set, and the optional interrupt will be generated.

The received data can be read from the DATA register when the Receive Complete interrupt flag is set.

35.6.2.6.1 Disabling the Receiver

Writing '0' to the Receiver Enable bit in the CTRLB register (CTRLB.RXEN) will disable the receiver, flush the two-level or four-level receive buffer, and data from ongoing receptions will be lost.

35.6.2.6.2 Error Bits

The USART receiver has three error bits in the Status (STATUS) register: Frame Error (FERR), Buffer Overflow (BUFOVF), and Parity Error (PERR). Once an error happens, the corresponding error bit will be set until it is cleared by writing ‘1’ to it. These bits are also cleared automatically when the receiver is disabled.

There are two methods for buffer overflow notification, selected by the Immediate Buffer Overflow Notification bit in the Control A register (CTRLA.IBON):

When CTRLA.IBON=1, STATUS.BUFOVF is raised immediately upon buffer overflow. Software can then empty the receive FIFO by reading RxDATA, until the receiver complete interrupt flag (INTFLAG.RXC) is cleared.

When CTRLA.IBON=0, the buffer overflow condition is attending data through the receive FIFO. After the received data is read, STATUS.BUFOVF will be set along with INTFLAG.RXC.

35.6.2.6.3 Asynchronous Data Reception

The USART includes a clock recovery and data recovery unit for handling asynchronous data reception.

The clock recovery logic can synchronize the incoming asynchronous serial frames at the RxD pin to the internally generated baud-rate clock.

The data recovery logic samples and applies a low-pass filter to each incoming bit, thereby improving the noise immunity of the receiver.

35.6.2.6.4 Asynchronous Operational Range

The operational range of the asynchronous reception depends on the accuracy of the internal baud-rate clock, the rate of the incoming frames, and the frame size (in number of bits). In addition, the operational range of the receiver is depending on the difference between the received bit rate and the internally generated baud rate. If the baud rate of an external transmitter is too high or too low compared to the internally generated baud rate, the receiver will not be able to synchronize the frames to the start bit.

There are two possible sources for a mismatch in baud rate: First, the reference clock will always have some minor instability. Second, the baud-rate generator cannot always do an exact division of the reference clock frequency to get the baud rate desired. In this case, the BAUD register value should be set to give the lowest possible error. Refer to *Clock Generation – Baud-Rate Generator* for details.

Recommended maximum receiver baud-rate errors for various character sizes are shown in the table below.

Downloaded from Arrow.com.

Table 35-3. Asynchronous Receiver Error for 16-fold Oversampling

D (Data bits+Parity)	R_{SLOW} [%]	R_{FAST} [%]	Max. total error [%]	Recommended max. Rx error [%]
5	94.12	107.69	+5.88/-7.69	±2.5
6	94.92	106.67	+5.08/-6.67	±2.0
7	95.52	105.88	+4.48/-5.88	±2.0
8	96.00	105.26	+4.00/-5.26	±2.0
9	96.39	104.76	+3.61/-4.76	±1.5
10	96.70	104.35	+3.30/-4.35	±1.5

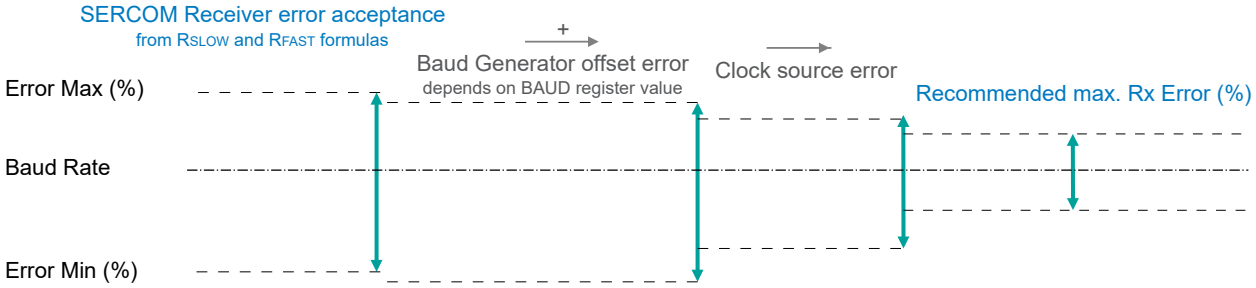
The following equations calculate the ratio of the incoming data rate and internal receiver baud rate:

$$R_{\text{SLOW}} = \frac{(D+1)S}{S-1+D\cdot S+S_F} \quad , \quad R_{\text{FAST}} = \frac{(D+2)S}{(D+1)S+S_M}$$

- R_{SLOW} is the ratio of the slowest incoming data rate that can be accepted in relation to the receiver baud rate
- R_{FAST} is the ratio of the fastest incoming data rate that can be accepted in relation to the receiver baud rate
- D is the sum of character size and parity size (D = 5 to 10 bits)
- S is the number of samples per bit (S = 16, 8 or 3)
- S_F is the first sample number used for majority voting (S_F = 7, 3, or 2) when CTRLA.SAMPA=0.
- S_M is the middle sample number used for majority voting (S_M = 8, 4, or 2) when CTRLA.SAMPA=0.

The recommended maximum Rx Error assumes that the receiver and transmitter equally divide the maximum total error. Its connection to the SERCOM Receiver error acceptance is depicted in this figure:

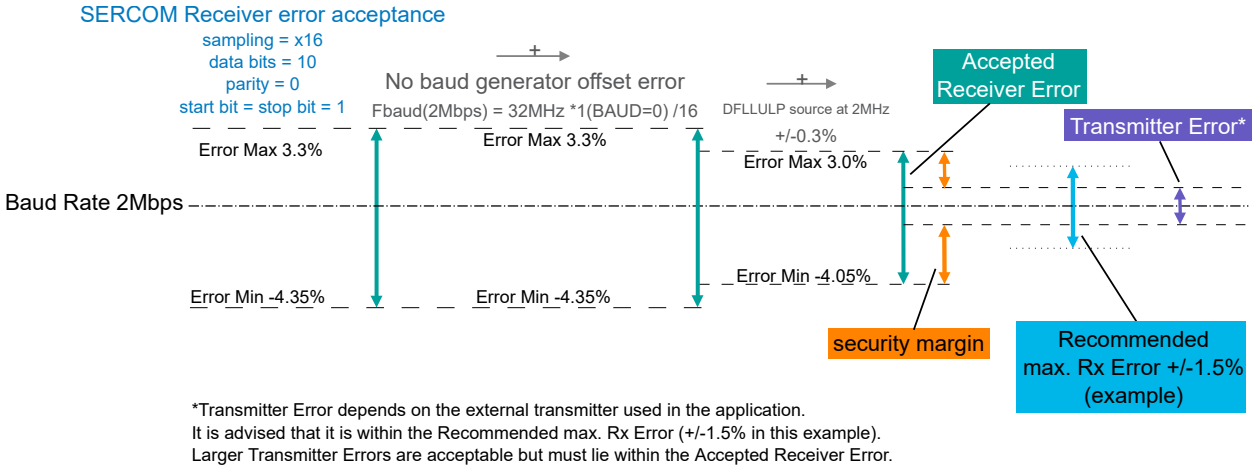
Figure 35-5. USART Rx Error Calculation



The recommendation values in the table above accommodate errors of the clock source and the baud generator. The following figure gives an example for a baud rate of 3Mbps:

Downloaded from Arrow.com.

Figure 35-6. USART Rx Error Calculation Example



*Transmitter Error depends on the external transmitter used in the application.
It is advised that it is within the Recommended max. Rx Error (+/-1.5% in this example).
Larger Transmitter Errors are acceptable but must lie within the Accepted Receiver Error.

Related Links

34.6.2.3 Clock Generation – Baud-Rate Generator

34.6.2.3.1 Asynchronous Arithmetic Mode BAUD Value Selection

35.6.3 Additional Features

35.6.3.1 Parity

Even or odd parity can be selected for error checking by writing 0x1 to the Frame Format bit field in the Control A register (CTRLA.FORM).

If *even parity* is selected (CTRLB.PMODE=0), the parity bit of an outgoing frame is '1' if the data contains an odd number of bits that are '1', making the total number of '1' even.

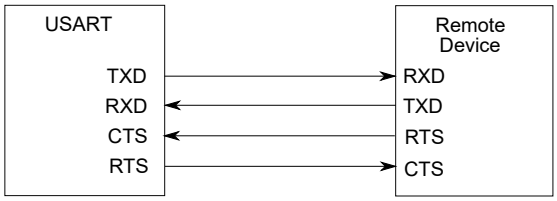
If *odd parity* is selected (CTRLB.PMODE=1), the parity bit of an outgoing frame is '1' if the data contains an even number of bits that are '0', making the total number of '1' odd.

When parity checking is enabled, the parity checker calculates the parity of the data bits in incoming frames and compares the result with the parity bit of the corresponding frame. If a parity error is detected, the Parity Error bit in the Status register (STATUS.PERR) is set.

35.6.3.2 Hardware Handshaking

The USART features an out-of-band hardware handshaking flow control mechanism, implemented by connecting the RTS and CTS pins with the remote device, as shown in the figure below.

Figure 35-7. Connection with a Remote Device for Hardware Handshaking



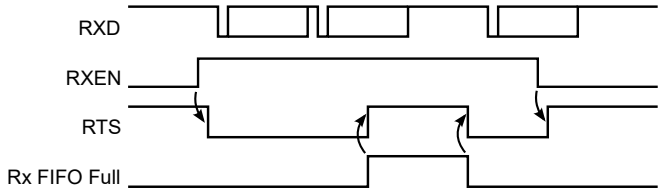
Hardware handshaking is only available in the following configuration:

- USART with internal clock (CTRLA.MODE=1),
- Asynchronous mode (CTRLA.CMODE=0),
- and Flow control pinout (CTRLA.TXPO=2).

Downloaded from Arrow.com.

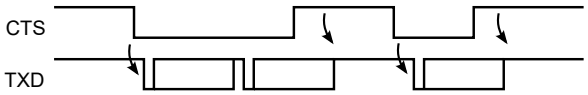
When the receiver is disabled or the receive FIFO is full, the receiver will drive the RTS pin high. This notifies the remote device to stop transfer after the ongoing transmission. Enabling and disabling the receiver by writing to CTRLB.RXEN will set/clear the RTS pin after a synchronization delay. When the receive FIFO goes full, RTS will be set immediately and the frame being received will be stored in the shift register until the receive FIFO is no longer full.

Figure 35-8. Receiver Behavior when Operating with Hardware Handshaking



The current CTS Status is in the STATUS register (STATUS.CTS). Character transmission will start only if STATUS.CTS=0. When CTS is set, the transmitter will complete the ongoing transmission and stop transmitting.

Figure 35-9. Transmitter Behavior when Operating with Hardware Handshaking



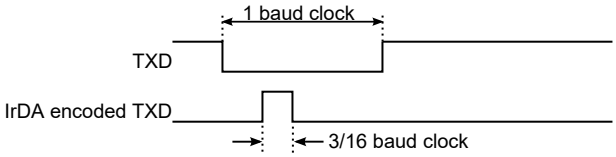
35.6.3.3 IrDA Modulation and Demodulation

Transmission and reception can be encoded IrDA compliant up to 115.2 kb/s. IrDA modulation and demodulation work in the following configuration:

- IrDA encoding enabled (CTRLB.ENC=1),
- Asynchronous mode (CTRLA.CMODE=0),
- and 16x sample rate (CTRLA.SAMPR[0]=0).

During transmission, each low bit is transmitted as a high pulse. The pulse width is 3/16 of the baud rate period, as illustrated in the figure below.

Figure 35-10. IrDA Transmit Encoding



The reception decoder has two main functions.

The first is to synchronize the incoming data to the IrDA baud rate counter. Synchronization is performed at the start of each zero pulse.

The second main function is to decode incoming Rx data. If a pulse width meets the minimum length set by configuration (RXPL.RXPL), it is accepted. When the baud rate counter reaches its middle value (1/2 bit length), it is transferred to the receiver.

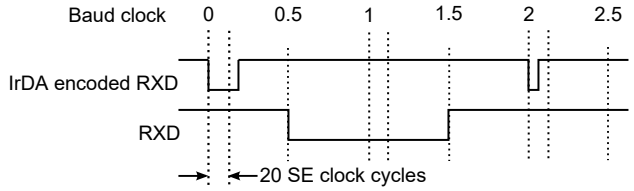
Note: Note that the polarity of the transmitter and receiver are opposite: During transmission, a '0' bit is transmitted as a '1' pulse. During reception, an accepted '0' pulse is received as a '0' bit.

> **Example:** The figure below illustrates reception where RXPL.RXPL is set to 19. This indicates that the pulse width should be at least 20 SE clock cycles. When using BAUD=0xE666 or 160 SE cycles per bit, this corresponds to 2/16 baud clock as

Downloaded from Arrow.com.

minimum pulse width required. In this case the first bit is accepted as a '0', the second bit is a '1', and the third bit is also a '1'. A low pulse is rejected since it does not meet the minimum requirement of 2/16 baud clock.

Figure 35-11. IrDA Receive Decoding



35.6.3.4 Break Character Detection and Auto-Baud/LIN Slave

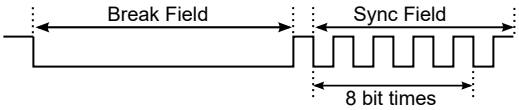
Break character detection and auto-baud are available in this configuration:

- Auto-baud frame format (CTRLA.FORM = 0x04 or 0x05),
- Asynchronous mode (CTRLA.CMODE = 0),
- and 16x sample rate using fractional baud rate generation (CTRLA.SAMPR = 1).

The USART uses a break detection threshold of greater than 11 nominal bit times at the configured baud rate. At any time, if more than 11 consecutive dominant bits are detected on the bus, the USART detects a Break Field. When a Break Field has been detected, the Receive Break interrupt flag (INTFLAG.RXBRK) is set and the USART expects the Sync Field character to be 0x55. This field is used to update the actual baud rate in order to stay synchronized. If the received Sync character is not 0x55, then the Inconsistent Sync Field error flag (STATUS.ISF) is set along with the Error interrupt flag (INTFLAG.ERROR), and the baud rate is unchanged.

The auto-baud follows the LIN format. All LIN Frames start with a Break Field followed by a Sync Field.

Figure 35-12. LIN Break and Sync Fields



After a break field is detected and the start bit of the Sync Field is detected, a counter is started. The counter is then incremented for the next 8 bit times of the Sync Field. At the end of these 8 bit times, the counter is stopped. At this moment, the 13 most significant bits of the counter (value divided by 8) give the new clock divider (BAUD.BAUD), and the 3 least significant bits of this value (the remainder) give the new Fractional Part (BAUD.FP).

When the Sync Field has been received, the clock divider (BAUD.BAUD) and the Fractional Part (BAUD.FP) are updated after a synchronization delay. After the Break and Sync Fields are received, multiple characters of data can be received.

35.6.3.5 RS485

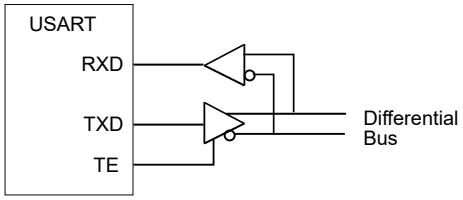
RS485 is available with the following configuration:

- USART frame format (CTRLA.FORM = 0x00 or 0x01)
- RS485 pinout (CTRLA.TXPO=0x3).

The RS485 feature enables control of an external line driver as shown in the figure below. While operating in RS485 mode, the transmit enable pin (TE) is driven high when the transmitter is active.

Downloaded from Arrow.com.

Figure 35-13. RS485 Bus Connection



The TE pin will remain high for the complete frame including stop bit(s). If a Guard Time is programmed in the Control C register (CTRLC.GTIME), the line will remain driven after the last character completion. The following figure shows a transfer with one stop bit and CTRLC.GTIME=3.

Figure 35-14. Example of TE Drive with Guard Time

The Transmit Complete interrupt flag (INTFLAG.TXC) will be raised after the guard time is complete and TE goes low.

35.6.3.6 ISO 7816 for Smart Card Interfacing

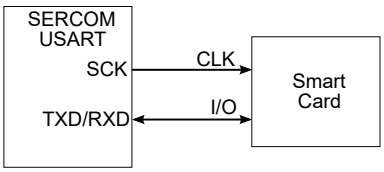
The SERCOM USART features an ISO/IEC 7816-compatible operating mode. This mode permits interfacing with smart cards and Security Access Modules (SAM) communicating through an ISO 7816 link. Both T=0 and T=1 protocols defined by the ISO 7816 specification are supported.

ISO 7816 is available with the following configuration:

- ISO 7816 format (CTRLA.FORM = 0x07)
- Inverse transmission and reception (CTRLA.RXINV=1 and CTRLA.TXINV=1)
- Single bidirectional data line (CTRLA.TXPO and CTRLA.RXPO configured to use the same data pin)
- Even parity (CTRLB.PMODE=0)
- 8-bit character size (CTRLB.CHSIZE=0)
- T=0 (CTRLA.CMODE=1) or T=1 (CTRLA.CMODE=0)

ISO 7816 is a half duplex communication on a single bidirectional line. The USART connects to a smart card as shown below. The output is only driven when the USART is transmitting. The USART is considered as the master of the communication as it generates the clock.

Figure 35-15. Connection of a Smart Card to the SERCOM USART



ISO 7816 characters are specified as 8 bits with even parity. The USART must be configured accordingly.

The USART cannot operate concurrently in both receiver and transmitter modes as the communication is unidirectional. It has to be configured according to the required mode by enabling or disabling either the receiver or the transmitter as desired. Enabling both the receiver and the transmitter at the same time in ISO 7816 mode may lead to unpredictable results.

The ISO 7816 specification defines an inverse transmission format. Data bits of the character must be transmitted on the I/O line at their negative value (CTRLA.RXINV=1 and CTRLA.TXINV=1).

Downloaded from Arrow.com.

Protocol T=0

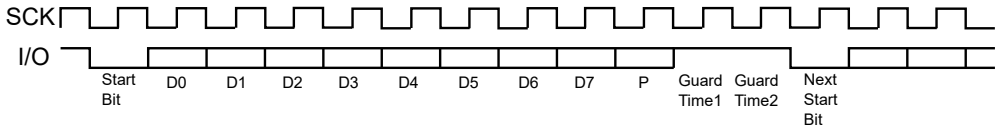
In T=0 protocol, a character is made up of:

- one start bit,
- eight data bits,
- one parity bit
- and one guard time, which lasts two bit times.

The transfer is synchronous (CTRLA.CMODE=1). The transmitter shifts out the bits and does not drive the I/O line during the guard time. Additional guard time can be added by programming the Guard Time (CTRLC.GTIME).

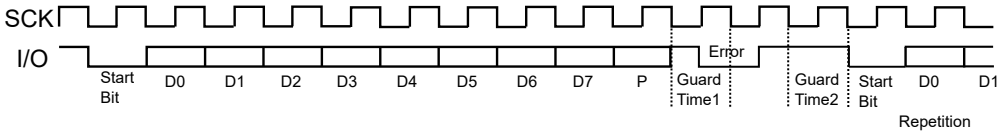
If no parity error is detected, the I/O line remains during the guard time and the transmitter can continue with the transmission of the next character, as shown in the figure below.

Figure 35-16. T=0 Protocol without Parity Error



If a parity error is detected by the receiver, it drives the I/O line to 0 during the guard time, as shown in the next figure. This error bit is also named NACK, for Non Acknowledge. In this case, the character lasts 1 bit time more, as the guard time length is the same and is added to the error bit time, which lasts 1 bit time.

Figure 35-17. T=0 Protocol with Parity Error



When the USART is the receiver and it detects a parity error, the parity error bit in the Status Register (STATUS.PERR) is set and the character is not written to the receive FIFO.

Receive Error Counter

The receiver also records the total number of errors (receiver parity errors and NACKs from the remote transmitter) up to a maximum of 255. This can be read in the Receive Error Count (RXERRCNT) register. RXERRCNT is automatically cleared on read.

Receive NACK Inhibit

The receiver can also be configured to inhibit error generation. This can be achieved by setting the Inhibit Not Acknowledge (CTRLC.INACK) bit. If CTRLC.INACK is 1, no error signal is driven on the I/O line even if a parity error is detected. Moreover, if CTRLC.INACK is set, the erroneous received character is stored in the receive FIFO, and the STATUS.PERR bit is set. Inhibit not acknowledge (CTRLC.INACK) takes priority over disable successive receive NACK (CTRLC.DSNACK).

Transmit Character Repetition

When the USART is transmitting a character and gets a NACK, it can automatically repeat the character before moving on to the next character. Repetition is enabled by writing the Maximum Iterations register (CTRLC.MAXITER) to a non-zero value. The USART repeats the character the number of times specified in CTRLC.MAXITER.

Downloaded from Arrow.com.

When the USART repetition number reaches the programmed value in CTRLC.MAXITER, the STATUS.ITER bit is set and the internal iteration counter is reset. If the repetition of the character is acknowledged by the receiver before the maximum iteration is reached, the repetitions are stopped and the iteration counter is cleared.

Disable Successive Receive NACK

The receiver can limit the number of successive NACKs sent back to the remote transmitter. This is programmed by setting the Disable Successive NACK bit (CTRLC.DSNACK). The maximum number of NACKs transmitted is programmed in the CTRLC.MAXITER field. As soon as the maximum is reached, the character is considered as correct, an acknowledge is sent on the line, the STATUS.ITER bit is set and the internal iteration counter is reset.

Protocol T=1

When operating in ISO7816 protocol T=1, the transmission is asynchronous (CTRL1.CMODE=0) with one or two stop bits. After the stop bits are sent, the transmitter does not drive the I/O line.

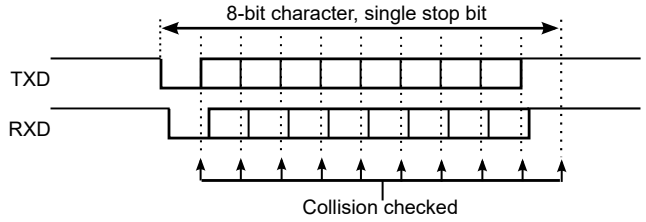
Parity is generated when transmitting and checked when receiving. Parity error detection sets the STATUS.PERR bit, and the erroneous character is written to the receive FIFO. When using T=1 protocol, the receiver does not signal errors on the I/O line and the transmitter does not retransmit.

35.6.3.7 Collision Detection

When the receiver and transmitter are connected either through pin configuration or externally, transmit collision can be detected after selecting the Collision Detection Enable bit in the CTRLB register (CTRLB.COLDEN=1). To detect collision, the receiver and transmitter must be enabled (CTRLB.RXEN=1 and CTRLB.TXEN=1).

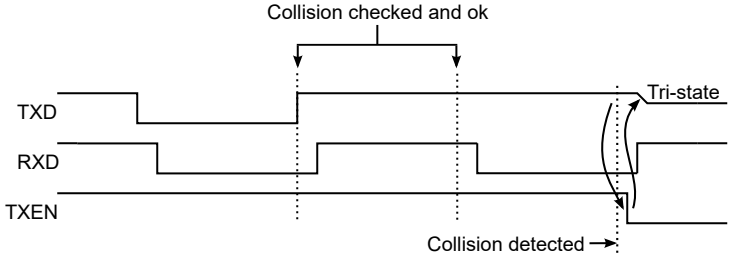
Collision detection is performed for each bit transmitted by comparing the received value with the transmit value, as shown in the figure below. While the transmitter is idle (no transmission in progress), characters can be received on RxD without triggering a collision.

Figure 35-18. Collision Checking



The next figure shows the conditions for a collision detection. In this case, the start bit and the first data bit are received with the same value as transmitted. The second received data bit is found to be different than the transmitted bit at the detection point, which indicates a collision.

Figure 35-19. Collision Detected



When a collision is detected, the USART follows this sequence:

Downloaded from Arrow.com.

1. Abort the current transfer.
2. Flush the transmit buffer.
3. Disable transmitter (CTRLB.TXEN=0)
   - This is done after a synchronization delay. The CTRLB Synchronization Busy bit (SYNCBUSY.CTRLB) will be set until this is complete.
   - After disabling, the TxD pin will be tri-stated.
4. Set the Collision Detected bit (STATUS.COLL) along with the Error interrupt flag (INTFLAG.ERROR).
5. Set the Transmit Complete interrupt flag (INTFLAG.TXC), since the transmit buffer no longer contains data.

After a collision, software must manually enable the transmitter again before continuing, after assuring that the CTRLB Synchronization Busy bit (SYNCBUSY.CTRLB) is not set.

35.6.3.8 Loop-Back Mode

For loop-back mode, configure the Receive Data Pinout (CTRLA.RXPO) and Transmit Data Pinout (CTRLA.TXPO) to use the same data pins for transmit and receive. The loop-back is through the pad, so the signal is also available externally.

35.6.3.9 Start-of-Frame Detection

The USART start-of-frame detector can wake-up the CPU when it detects a Start bit. In Standby Sleep mode, the internal fast start-up oscillator must be selected as the GCLK_SERCOMx_CORE source.

When a 1-to-0 transition is detected on RxD, the 8 MHz Internal Oscillator is powered up and the USART clock is enabled. After start-up, the rest of the data frame can be received, provided that the baud rate is slow enough in relation to the fast start-up internal oscillator start-up time. Refer to the *Electrical Characteristics* chapters for details. The start-up time of this oscillator varies with supply voltage and temperature.

The USART start-of-frame detection works both in Asynchronous and Synchronous modes. It is enabled by writing '1' to the Start of Frame Detection Enable bit in the Control B register (CTRLB.SFDE).

If the Receive Start Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.RXS) is set, the Receive Start interrupt is generated immediately when a start is detected.

When using start-of-frame detection without the Receive Start interrupt, start detection will force the 8 MHz internal oscillator and USART clock active while the frame is being received. In this case, the CPU will not wake up until the receive complete interrupt is generated.

35.6.3.10 Sample Adjustment

In asynchronous mode (CTRLA.CMODE=0), three samples in the middle are used to determine the value based on majority voting. The three samples used for voting can be selected using the Sample Adjustment bit field in Control A register (CTRLA.SAMPA). When CTRLA.SAMPA=0, samples 7-8-9 are used for 16x oversampling, and samples 3-4-5 are used for 8x oversampling.

35.6.4 DMA, Interrupts and Events

Table 35-4. Module Request for SERCOM USART

Condition	Request		
	DMA	Interrupt	Event
Data Register Empty (DRE)	Yes	Yes	NA

Downloaded from Arrow.com.

Condition	Request		
	DMA	Interrupt	Event
	(request cleared when data is written)		
Receive Complete (RXC)	Yes (request cleared when data is read)	Yes	
Transmit Complete (TXC)	NA	Yes	
Receive Start (RXS)	NA	Yes	
Clear to Send Input Change (CTSIC)	NA	Yes	
Receive Break (RXBRK)	NA	Yes	
Error (ERROR)	NA	Yes	

35.6.4.1 DMA Operation

The USART generates the following DMA requests:

- Data received (RX): The request is set when data is available in the receive FIFO. The request is cleared when DATA is read.
- Data transmit (TX): The request is set when the transmit buffer (TX DATA) is empty. The request is cleared when DATA is written.

35.6.4.2 Interrupts

The USART has the following interrupt sources. These are asynchronous interrupts, and can wake up the device from any sleep mode:

- Data Register Empty (DRE)
- Receive Complete (RXC)
- Transmit Complete (TXC)
- Receive Start (RXS)
- Clear to Send Input Change (CTSIC)
- Received Break (RXBRK)
- Error (ERROR)

Each interrupt source has its own interrupt flag. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) will be set when the interrupt condition is met. Each interrupt can be individually enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the interrupt flag is set and if the corresponding interrupt is enabled. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is disabled, or the USART is reset. For details on clearing interrupt flags, refer to the INTFLAG register description.

The value of INTFLAG indicates which interrupt is executed. Note that interrupts must be globally enabled for interrupt requests. Refer to *Nested Vector Interrupt Controller* for details.

35.6.4.3 Events

Not applicable.

Downloaded from Arrow.com.

35.6.5 Sleep Mode Operation

The behavior in sleep mode is depending on the clock source and the Run In Standby bit in the Control A register (CTRLA.RUNSTDBY):

- Internal clocking, CTRLA.RUNSTDBY=1: GCLK_SERCOMx_CORE can be enabled in all sleep modes. Any interrupt can wake up the device.
- External clocking, CTRLA.RUNSTDBY=1: The Receive Complete interrupt(s) can wake up the device.
- Internal clocking, CTRLA.RUNSTDBY=0: Internal clock will be disabled, after any ongoing transfer was completed. The Receive Complete interrupt(s) can wake up the device.
- External clocking, CTRLA.RUNSTDBY=0: External clock will be disconnected, after any ongoing transfer was completed. All reception will be dropped.

35.6.6 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in the CTRLA register (CTRLA.SWRST)
- Enable bit in the CTRLA register (CTRLA.ENABLE)
- Receiver Enable bit in the CTRLB register (CTRLB.RXEN)
- Transmitter Enable bit in the Control B register (CTRLB.TXEN)

Note: CTRLB.RXEN is write-synchronized somewhat differently. See also 35.8.2 CTRLB for details.

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Downloaded from Arrow.com.

35.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
		15:8	SAMPR[2:0]					RXINV	TXINV	IBON
		23:16	SAMPA[1:0]		RXPO[1:0]				TXPO[1:0]	
		31:24		DORD	CPOL	CMODE	FORM[3:0]			
0x04	CTRLB	7:0		SBMODE				CHSIZE[2:0]		
		15:8			PMODE			ENC	SFDE	COLDEN
		23:16							RXEN	TXEN
		31:24								
0x08	CTRLC	7:0						GTIME[2:0]		
		15:8								
		23:16		MAXITER[2:0]					DSNACK	INACK
		31:24								
0x0C	BAUD	7:0	BAUD[7:0]							
		15:8	BAUD[15:8]							
0x0E	RXPL	7:0	RXPL[7:0]							
0x0F ... 0x13	Reserved									
0x14	INTENCLR	7:0	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
0x15	Reserved									
0x16	INTENSET	7:0	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
0x17	Reserved									
0x18	INTFLAG	7:0	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
0x19	Reserved									
0x1A	STATUS	7:0	ITER	TXE	COLL	ISF	CTS	BUFOVF	FERR	PERR
		15:8								
0x1C	SYNCBUSY	7:0					RXERRCNT	CTRLB	ENABLE	SWRST
		15:8								
		23:16								
		31:24								
0x20	RXERRCNT	7:0	RXERRCNT[7:0]							
0x21 ... 0x27	Reserved									
0x28	DATA	7:0	DATA[7:0]							
		15:8								DATA[8:8]
0x2A ... 0x2F	Reserved									
0x30	DBGCTRL	7:0								DBGSTOP

Downloaded from Arrow.com.

35.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

Downloaded from Arrow.com.

35.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
		DORD	CPOL	CMODE	FORM[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SAMPA[1:0]		RXPO[1:0]				TXPO[1:0]	
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Bit	15	14	13	12	11	10	9	8
	SAMPR[2:0]					RXINV	TXINV	IBON
Access	R/W	R/W	R/W			R/W	R/W	R
Reset	0	0	0			0	0	0

Bit	7	6	5	4	3	2	1	0
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit 30 – DORD Data Order
This bit selects the data order when a character is shifted out from the Data register.

This bit is not synchronized.

Value	Description
0	MSB is transmitted first.
1	LSB is transmitted first.

Bit 29 – CPOL Clock Polarity
This bit selects the relationship between data output change and data input sampling in synchronous mode.

This bit is not synchronized.

CPOL	TxD Change	RxD Sample
0x0	Rising XCK edge	Falling XCK edge
0x1	Falling XCK edge	Rising XCK edge

Bit 28 – CMODE Communication Mode
This bit selects asynchronous or synchronous communication.

Downloaded from Arrow.com.

This bit is not synchronized.

Value	Description
0	Asynchronous communication.
1	Synchronous communication.

Bits 27:24 – FORM[3:0] Frame Format
These bits define the frame format.

These bits are not synchronized.

FORM[3:0]	Description
0x0	USART frame
0x1	USART frame with parity
0x2-0x3	Reserved
0x4	Auto-baud (LIN Slave) - break detection and auto-baud.
0x5	Auto-baud - break detection and auto-baud with parity
0x6	Reserved
0x7	ISO 7816
0x8-0xF	Reserved

Bits 23:22 – SAMPA[1:0] Sample Adjustment
These bits define the sample adjustment.

These bits are not synchronized.

SAMPA[1:0]	16x Over-sampling (CTRLA.SAMPR=0 or 1)	8x Over-sampling (CTRLA.SAMPR=2 or 3)
0x0	7-8-9	3-4-5
0x1	9-10-11	4-5-6
0x2	11-12-13	5-6-7
0x3	13-14-15	6-7-8

Bits 21:20 – RXPO[1:0] Receive Data Pinout
These bits define the receive data (RxD) pin configuration.

These bits are not synchronized.

RXPO[1:0]	Name	Description
0x0	PAD[0]	SERCOM PAD[0] is used for data reception
0x1	PAD[1]	SERCOM PAD[1] is used for data reception
0x2	PAD[2]	SERCOM PAD[2] is used for data reception
0x3	PAD[3]	SERCOM PAD[3] is used for data reception

Downloaded from Arrow.com.

Bits 17:16 – TXPO[1:0] Transmit Data Pinout
These bits define the transmit data (TxD) and XCK pin configurations.

This bit is not synchronized.

TXPO	TxD Pin Location	XCK Pin Location (When Applicable)	RTS/TE	CTS
0x0	SERCOM PAD[0]	SERCOM PAD[1]	N/A	N/A
0x1	SERCOM PAD[2]	SERCOM PAD[3]	N/A	N/A
0x2	SERCOM PAD[0]	N/A	SERCOM PAD[2]	SERCOM PAD[3]
0x3	SERCOM_PAD[0]	SERCOM_PAD[1]	SERCOM_PAD[2]	N/A

Bits 15:13 – SAMPR[2:0] Sample Rate
These bits select the sample rate.

These bits are not synchronized.

SAMPR[2:0]	Description
0x0	16x over-sampling using arithmetic baud rate generation.
0x1	16x over-sampling using fractional baud rate generation.
0x2	8x over-sampling using arithmetic baud rate generation.
0x3	8x over-sampling using fractional baud rate generation.
0x4	3x over-sampling using arithmetic baud rate generation.
0x5-0x7	Reserved

Bit 10 – RXINV Receive Data Invert
This bit controls whether the receive data (RxD) is inverted or not.
Note: Start, parity and stop bit(s) are unchanged. When enabled, parity is calculated on the inverted data.

Value	Description
0	RxD is not inverted.
1	RxD is inverted.

Bit 9 – TXINV Transmit Data Invert
This bit controls whether the transmit data (TxD) is inverted or not.
Note: Start, parity and stop bit(s) are unchanged. When enabled, parity is calculated on the inverted data.

Value	Description
0	TxD is not inverted.
1	TxD is inverted.

Bit 8 – IBON Immediate Buffer Overflow Notification
This bit controls when the buffer overflow status bit (STATUS.BUFOVF) is asserted when a buffer overflow occurs.

Downloaded from Arrow.com.

Value	Description
0	STATUS.BUFOVF is asserted when it occurs in the data stream.
1	STATUS.BUFOVF is asserted immediately upon buffer overflow.

Bit 7 – RUNSTDBY Run In Standby
This bit defines the functionality in standby sleep mode.

This bit is not synchronized.

RUNSTDBY	External Clock	Internal Clock
0x0	External clock is disconnected when ongoing transfer is finished. All reception is dropped.	Generic clock is disabled when ongoing transfer is finished. The device can wake up on Transfer Complete interrupt.
0x1	Wake on Receive Complete interrupt.	Generic clock is enabled in all sleep modes. Any interrupt can wake up the device.

Bits 4:2 – MODE[2:0] Operating Mode
These bits select the USART serial communication interface of the SERCOM.

These bits are not synchronized.

Value	Description
0x0	USART with external clock
0x1	USART with internal clock

Bit 1 – ENABLE Enable
Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable Synchronization Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE is cleared when the operation is complete.

This bit is not enable-protected.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled or being enabled.

Bit 0 – SWRST Software Reset
Writing '0' to this bit has no effect.

Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state, and the SERCOM will be disabled.

Writing '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded. Any register write access during the ongoing reset will result in an APB error. Reading any register will return the reset value of the register.

Due to synchronization, there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

This bit is not enable-protected.

Downloaded from Arrow.com.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

Downloaded from Arrow.com.

35.8.2 Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							RXEN	TXEN
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
			PMODE			ENC	SFDE	COLDEN
Access			R/W			R/W	R/W	R/W
Reset			0			0	0	0

Bit	7	6	5	4	3	2	1	0
		SBMODE				CHSIZE[2:0]		
Access		R/W				R/W	R/W	R/W
Reset		0				0	0	0

Bit 17 – RXEN Receiver Enable
Writing '0' to this bit will disable the USART receiver. Disabling the receiver will flush the receive buffer and clear the FERR, PERR and BUFOVF bits in the STATUS register.

Writing '1' to CTRLB.RXEN when the USART is disabled will set CTRLB.RXEN immediately. When the USART is enabled, CTRLB.RXEN will be cleared, and SYNCBUSY.CTRLB will be set and remain set until the receiver is enabled. When the receiver is enabled, CTRLB.RXEN will read back as '1'.

Writing '1' to CTRLB.RXEN when the USART is enabled will set SYNCBUSY.CTRLB, which will remain set until the receiver is enabled, and CTRLB.RXEN will read back as '1'.

This bit is not enable-protected.

Value	Description
0	The receiver is disabled or being enabled.
1	The receiver is enabled or will be enabled when the USART is enabled.

Bit 16 – TXEN Transmitter Enable
Writing '0' to this bit will disable the USART transmitter. Disabling the transmitter will not become effective until ongoing and pending transmissions are completed.

Writing '1' to CTRLB.TXEN when the USART is disabled will set CTRLB.TXEN immediately. When the USART is enabled, CTRLB.TXEN will be cleared, and SYNCBUSY.CTRLB will be set and remain set until the transmitter is enabled. When the transmitter is enabled, CTRLB.TXEN will read back as '1'.

Downloaded from Arrow.com.

Writing '1' to CTRLB.TXEN when the USART is enabled will set SYNCBUSY.CTRLB, which will remain set until the transmitter is enabled, and CTRLB.TXEN will read back as '1'.

This bit is not enable-protected.

Value	Description
0	The transmitter is disabled or being enabled.
1	The transmitter is enabled or will be enabled when the USART is enabled.

Bit 13 – PMODE Parity Mode
This bit selects the type of parity used when parity is enabled (CTRLA.FORM is '1'). The transmitter will automatically generate and send the parity of the transmitted data bits within each frame. The receiver will generate a parity value for the incoming data and parity bit, compare it to the parity mode and, if a mismatch is detected, STATUS.PERR will be set.

This bit is not synchronized.

Value	Description
0	Even parity.
1	Odd parity.

Bit 10 – ENC Encoding Format
This bit selects the data encoding format.

This bit is not synchronized.

Value	Description
0	Data is not encoded.
1	Data is IrDA encoded.

Bit 9 – SFDE Start of Frame Detection Enable
This bit controls whether the start-of-frame detector will wake up the device when a start bit is detected on the RxD line.

This bit is not synchronized.

SFDE	INTENSET.RXS	INTENSET.RXC	Description
0	X	X	Start-of-frame detection disabled.
1	0	0	Reserved
1	0	1	Start-of-frame detection enabled. RXC wakes up the device from all sleep modes.
1	1	0	Start-of-frame detection enabled. RXS wakes up the device from all sleep modes.
1	1	1	Start-of-frame detection enabled. Both RXC and RXS wake up the device from all sleep modes.

Bit 8 – COLDEN Collision Detection Enable
This bit enables collision detection.

This bit is not synchronized.

Downloaded from Arrow.com.

Value	Description
0	Collision detection is not enabled.
1	Collision detection is enabled.

Bit 6 – SBMODE Stop Bit Mode
This bit selects the number of stop bits transmitted.

This bit is not synchronized.

Value	Description
0	One stop bit.
1	Two stop bits.

Bits 2:0 – CHSIZE[2:0] Character Size
These bits select the number of bits in a character.

These bits are not synchronized.

CHSIZE[2:0]	Description
0x0	8 bits
0x1	9 bits
0x2-0x4	Reserved
0x5	5 bits
0x6	6 bits
0x7	7 bits

Downloaded from Arrow.com.

35.8.3 Control C

Name: CTRLC
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		MAXITER[2:0]					DSNACK	INACK
Access		R/W	R/W	R/W			R/W	R/W
Reset		0	0	0			0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						GTIME[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 22:20 – MAXITER[2:0] Maximum Iterations
These bits define the maximum number of retransmit iterations.

These bits also define the successive NACKs sent to the remote transmitter when CTRLC.DSNACK is set.

This field is only valid when using ISO7816 T=0 mode (CTRLA.MODE=0x7 and CTRLA.CMODE=0).

Bit 17 – DSNACK Disable Successive Not Acknowledge
This bit controls how many times NACK will be sent on parity error reception.

This bit is only valid in ISO7816 T=0 mode and when CTRLC.INACK=0.

Value	Description
0	NACK is sent on the ISO line for every parity error received.
1	Successive parity errors are counted up to the value specified in CTRLC.MAXITER. These parity errors generate a NACK on the ISO line. As soon as this value is reached, no additional NACK is sent on the ISO line.

Bit 16 – INACK Inhibit Not Acknowledge
This bit controls whether a NACK is transmitted when a parity error is received.

This bit is only valid in ISO7816 T=0 mode.

Downloaded from Arrow.com.

Value	Description
0	NACK is transmitted when a parity error is received.
1	NACK is not transmitted when a parity error is received.

Bits 2:0 – GTIME[2:0] Guard Time
These bits define the guard time when using RS485 mode (CTRLA.FORM=0x0 or CTRLA.FORM=0x1, and CTRLA.TXPO=0x3) or ISO7816 mode (CTRLA.FORM=0x7).

For RS485 mode, the guard time is programmable from 0-7 bit times and defines the time that the transmit enable pin (TE) remains high after the last stop bit is transmitted and there is no remaining data to be transmitted.

For ISO7816 T=0 mode, the guard time is programmable from 2-9 bit times and defines the guard time between each transmitted byte.

Downloaded from Arrow.com.

35.8.4 Baud

Name: BAUD
Offset: 0x0C
Reset: 0x0000
Property: Enable-Protected, PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
	BAUD[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BAUD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – BAUD[15:0] Baud Value
Arithmetic Baud Rate Generation (`CTRLA.SAMPR[0]=0`):

These bits control the clock generation, as described in the SERCOM Baud Rate section.

If Fractional Baud Rate Generation (`CTRLA.SAMPR[0]=1`) bit positions 15 to 13 are replaced by FP[2:0] Fractional Part:

- **Bits 15:13 - FP[2:0]: Fractional Part**

  These bits control the clock generation, as described in the *SERCOM Clock Generation – Baud-Rate Generator* section.
- **Bits 12:0 - BAUD[12:0]: Baud Value**

  These bits control the clock generation, as described in the *SERCOM Clock Generation – Baud-Rate Generator* section.

Related Links

34.6.2.3 Clock Generation – Baud-Rate Generator
34.6.2.3.1 Asynchronous Arithmetic Mode BAUD Value Selection

Downloaded from Arrow.com.

35.8.5 Receive Pulse Length Register

Name: RXPL
Offset: 0x0E
Reset: 0x00
Property: Enable-Protected, PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	RXPL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – RXPL[7:0] Receive Pulse Length
When the encoding format is set to IrDA (CTRLB.ENC=1), these bits control the minimum pulse length that is required for a pulse to be accepted by the IrDA receiver with regards to the serial engine clock period SE_{per}.

$PULSE \geq (\mathrm{RXPL} + 2) \cdot SE_{per}$

Downloaded from Arrow.com.

35.8.6 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Bit	7	6	5	4	3	2	1	0
	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bit 7 – ERROR Error Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 5 – RXBRK Receive Break Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Receive Break Interrupt Enable bit, which disables the Receive Break interrupt.

Value	Description
0	Receive Break interrupt is disabled.
1	Receive Break interrupt is enabled.

Bit 4 – CTSIC Clear to Send Input Change Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Clear To Send Input Change Interrupt Enable bit, which disables the Clear To Send Input Change interrupt.

Value	Description
0	Clear To Send Input Change interrupt is disabled.
1	Clear To Send Input Change interrupt is enabled.

Bit 3 – RXS Receive Start Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Receive Start Interrupt Enable bit, which disables the Receive Start interrupt.

Value	Description
0	Receive Start interrupt is disabled.
1	Receive Start interrupt is enabled.

Downloaded from Arrow.com.

Bit 2 – RXC Receive Complete Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Receive Complete Interrupt Enable bit, which disables the Receive Complete interrupt.

Value	Description
0	Receive Complete interrupt is disabled.
1	Receive Complete interrupt is enabled.

Bit 1 – TXC Transmit Complete Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Transmit Complete Interrupt Enable bit, which disables the Receive Complete interrupt.

Value	Description
0	Transmit Complete interrupt is disabled.
1	Transmit Complete interrupt is enabled.

Bit 0 – DRE Data Register Empty Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Data Register Empty Interrupt Enable bit, which disables the Data Register Empty interrupt.

Value	Description
0	Data Register Empty interrupt is disabled.
1	Data Register Empty interrupt is enabled.

Downloaded from Arrow.com.

35.8.7 Interrupt Enable Set

Name: INTENSET
Offset: 0x16
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Bit	7	6	5	4	3	2	1	0
	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bit 7 – ERROR Error Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 5 – RXBRK Receive Break Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Receive Break Interrupt Enable bit, which enables the Receive Break interrupt.

Value	Description
0	Receive Break interrupt is disabled.
1	Receive Break interrupt is enabled.

Bit 4 – CTSIC Clear to Send Input Change Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Clear To Send Input Change Interrupt Enable bit, which enables the Clear To Send Input Change interrupt.

Value	Description
0	Clear To Send Input Change interrupt is disabled.
1	Clear To Send Input Change interrupt is enabled.

Bit 3 – RXS Receive Start Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Receive Start Interrupt Enable bit, which enables the Receive Start interrupt.

Value	Description
0	Receive Start interrupt is disabled.
1	Receive Start interrupt is enabled.

Downloaded from Arrow.com.

Bit 2 – RXC Receive Complete Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Receive Complete Interrupt Enable bit, which enables the Receive Complete interrupt.

Value	Description
0	Receive Complete interrupt is disabled.
1	Receive Complete interrupt is enabled.

Bit 1 – TXC Transmit Complete Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Transmit Complete Interrupt Enable bit, which enables the Transmit Complete interrupt.

Value	Description
0	Transmit Complete interrupt is disabled.
1	Transmit Complete interrupt is enabled.

Bit 0 – DRE Data Register Empty Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Data Register Empty Interrupt Enable bit, which enables the Data Register Empty interrupt.

Value	Description
0	Data Register Empty interrupt is disabled.
1	Data Register Empty interrupt is enabled.

Downloaded from Arrow.com.

35.8.8 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
Access	R/W		R/W	R/W	R/W	R	R/W	R
Reset	0		0	0	0	0	0	0

Bit 7 – ERROR Error
This flag is cleared by writing '1' to it.

This bit is set when any error is detected. Errors that will set this flag have corresponding status flags in the STATUS register. Errors that will set this flag are COLL, ISF, BUFOVF, FERR, and PERR.Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 5 – RXBRK Receive Break
This flag is cleared by writing '1' to it.

This flag is set when auto-baud is enabled (CTRLA.FORM) and a break character is received.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 4 – CTSIC Clear to Send Input Change
This flag is cleared by writing a '1' to it.

This flag is set when a change is detected on the CTS pin.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 3 – RXS Receive Start
This flag is cleared by writing '1' to it.

This flag is set when a start condition is detected on the RxD line and start-of-frame detection is enabled (CTRLB.SFDE is '1').

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Receive Start interrupt flag.

Bit 2 – RXC Receive Complete
This flag is cleared by reading the Data register (DATA) or by disabling the receiver.

This flag is set when there are unread data in DATA.

Writing '0' to this bit has no effect.

Downloaded from Arrow.com.

Writing '1' to this bit has no effect.

Bit 1 – TXC Transmit Complete
This flag is cleared by writing '1' to it or by writing new data to DATA.

This flag is set when the entire frame in the transmit shift register has been shifted out and there are no new data in DATA.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 0 – DRE Data Register Empty
This flag is cleared by writing new data to DATA.

This flag is set when DATA is empty and ready to be written.

Writing '0' to this bit has no effect.

Writing '1' to this bit has no effect.

Downloaded from Arrow.com.

35.8.9 Status

Name: STATUS
Offset: 0x1A
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ITER	TXE	COLL	ISF	CTS	BUFOVF	FERR	PERR
Access	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7 – ITER Maximum Number of Repetitions Reached
This bit is set when the maximum number of NACK repetitions or retransmissions is met in ISO7816 T=0 mode.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

Bit 6 – TXE Transmitter Empty
This bit will always read back as zero.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

Bit 5 – COLL Collision Detected
This bit is cleared by writing '1' to the bit or by disabling the receiver.

This bit is set when collision detection is enabled (CTRLB.COLDEN) and a collision is detected.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

Bit 4 – ISF Inconsistent Sync Field
This bit is cleared by writing '1' to the bit or by disabling the receiver.

This bit is set when the frame format is set to auto-baud (CTRLA.FORM) and a sync field not equal to 0x55 is received.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

Bit 3 – CTS Clear to Send
This bit indicates the current level of the CTS pin when flow control is enabled (CTRLA.TXPO).

Downloaded from Arrow.com.

Writing '0' to this bit has no effect.

Writing '1' to this bit has no effect.

Bit 2 – BUFOVF Buffer Overflow
Reading this bit before reading the Data register will indicate the error status of the next character to be read.

This bit is cleared by writing '1' to the bit or by disabling the receiver.

This bit is set when a buffer overflow condition is detected. A buffer overflow occurs when the receive buffer is full, there is a new character waiting in the receive shift register and a new start bit is detected.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

Bit 1 – FERR Frame Error
Reading this bit before reading the Data register will indicate the error status of the next character to be read.

This bit is cleared by writing '1' to the bit or by disabling the receiver.

This bit is set if the received character had a frame error, i.e., when the first stop bit is zero.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

Bit 0 – PERR Parity Error
Reading this bit before reading the Data register will indicate the error status of the next character to be read.

This bit is cleared by writing '1' to the bit or by disabling the receiver.

This bit is set if parity checking is enabled (CTRLA.FORM is 0x1, 0x5, or 0x7) and a parity error is detected.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

Downloaded from Arrow.com.

35.8.10 Synchronization Busy

Name: SYNCBUSY
Offset: 0x1C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					RXERRCNT	CTRLB	ENABLE	SWRST
Access					R	R	R	R
Reset					0	0	0	0

Bit 3 – RXERRCNT Receive Error Count Synchronization Busy
The RXERRCNT register is automatically synchronized to the APB domain upon error. When returning from sleep, this bit will be raised until the new value is available to be read.

Value	Description
0	RXERRCNT synchronization is not busy.
1	RXERRCNT synchronization is busy.

Bit 2 – CTRLB CTRLB Synchronization Busy
Writing to the CTRLB register when the SERCOM is enabled requires synchronization. When writing to CTRLB the SYNCBUSY.CTRLB bit will be set until synchronization is complete. If CTRLB is written while SYNCBUSY.CTRLB is asserted, an APB error will be generated.

Value	Description
0	CTRLB synchronization is not busy.
1	CTRLB synchronization is busy.

Bit 1 – ENABLE SERCOM Enable Synchronization Busy
Enabling and disabling the SERCOM (CTRLA.ENABLE) requires synchronization. When written, the SYNCBUSY.ENABLE bit will be set until synchronization is complete.

Downloaded from Arrow.com.

Value	Description
0	Enable synchronization is not busy.
1	Enable synchronization is busy.

Bit 0 – SWRST Software Reset Synchronization Busy
Resetting the SERCOM (CTRLA.SWRST) requires synchronization. When written, the SYNCBUSY.SWRST bit will be set until synchronization is complete.

Value	Description
0	SWRST synchronization is not busy.
1	SWRST synchronization is busy.

Downloaded from Arrow.com.

35.8.11 Receive Error Count

Name: RXERRCNT
Offset: 0x20
Reset: 0x00
Property: Read-Synchronized

Bit	7	6	5	4	3	2	1	0
	RXERRCNT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – RXERRCNT[7:0] Receive Error Count
This register records the total number of parity errors and NACK errors combined in ISO7816 mode (CTRLA.FORM=0x7).

This register is automatically cleared on read.

Downloaded from Arrow.com.

35.8.12 Data

Name: DATA
Offset: 0x28
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
								DATA[8:8]
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 8:0 – DATA[8:0] Data
Reading these bits will return the contents of the Receive Data register. The register should be read only when the Receive Complete Interrupt Flag bit in the Interrupt Flag Status and Clear register (INTFLAG.RXC) is set. The status bits in STATUS should be read before reading the DATA value in order to get any corresponding error.

Writing these bits will write the Transmit Data register. This register should be written only when the Data Register Empty Interrupt Flag bit in the Interrupt Flag Status and Clear register (INTFLAG.DRE) is set.

Downloaded from Arrow.com.

35.8.13 Debug Control

Name: DBGCTRL
Offset: 0x30
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGSTOP
Access								R/W
Reset								0

Bit 0 – DBGSTOP Debug Stop Mode
This bit controls the baud-rate generator functionality when the CPU is halted by an external debugger.

Value	Description
0	The baud-rate generator continues normal operation when the CPU is halted by an external debugger.
1	The baud-rate generator is halted when the CPU is halted by an external debugger.

Downloaded from Arrow.com.

36. SERCOM SPI – SERCOM Serial Peripheral Interface

36.1 Overview

The serial peripheral interface (SPI) is one of the available modes in the Serial Communication Interface (SERCOM).

The SPI uses the SERCOM transmitter and receiver configured as shown in 36.3 Block Diagram. Each side, master and slave, depicts a separate SPI containing a shift register, a transmit buffer and a two-level or four-level receive buffer. In addition, the SPI master uses the SERCOM baud-rate generator, while the SPI slave can use the SERCOM address match logic. Labels in capital letters are synchronous to CLK_SERCOMx_APB and accessible by the CPU, while labels in lowercase letters are synchronous to the SCK clock.

Related Links

34. SERCOM – Serial Communication Interface

36.2 Features

SERCOM SPI includes the following features:

- Full-duplex, four-wire interface (MISO, MOSI, SCK, $\overline{SS}$)
- One-level transmit buffer, two-level or four-level receive buffer
- Supports all four SPI modes of operation
- Single data direction operation allows alternate function on MISO or MOSI pin
- Selectable LSB- or MSB-first data transfer
- Can be used with DMA
- Master operation:
  - Serial clock speed, $f_{SCK}=1/t_{SCK}$(1)
  - 8-bit clock generator
  - Hardware controlled $\overline{SS}$

1. For t_{SCK} and t_{SSCK} values, refer to SPI Timing Characteristics.

Related Links

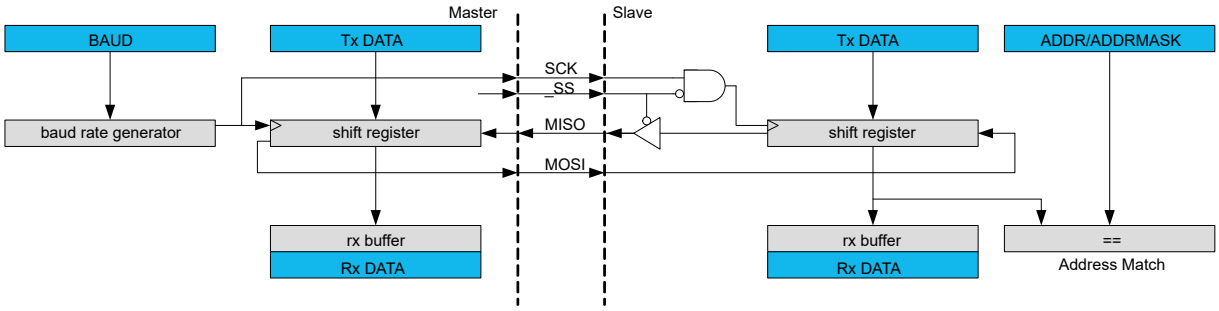
34. SERCOM – Serial Communication Interface

34.2 Features

Downloaded from Arrow.com.

36.3 Block Diagram

Figure 36-1. Full-Duplex SPI Master Slave Interconnection



36.4 Signal Description

Table 36-1. SERCOM SPI Signals

Signal Name	Type	Description
PAD[3:0]	Digital I/O	General SERCOM pins

One signal can be mapped to one of several pins.

36.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

36.5.1 I/O Lines

In order to use the SERCOM's I/O lines, the I/O pins must be configured using the IO Pin Controller (PORT).

When the SERCOM is configured for SPI operation, the SERCOM controls the direction and value of the I/O pins according to the table below. Both PORT control bits PINCFGn.PULLEN and PINCFGn.DRVSTR are still effective. If the receiver is disabled, the data input pin can be used for other purposes. In master mode, the slave select line ($\overline{SS}$) is hardware controlled when the Master Slave Select Enable bit in the Control B register (CTRLB.MSSEN) is '1'.

Table 36-2. SPI Pin Configuration

Pin	Master SPI	Slave SPI
MOSI	Output	Input
MISO	Input	Output
SCK	Output	Input
$\overline{SS}$	Output (CTRLB.MSSEN=1)	Input

The combined configuration of PORT, the Data In Pinout and the Data Out Pinout bit groups in the Control A register (CTRLA.DIPO and CTRLA.DOPO) define the physical position of the SPI signals in the table above.

Related Links

Downloaded from Arrow.com.

32. PORT - I/O Pin Controller

36.5.2 Power Management

This peripheral can continue to operate in any sleep mode where its source clock is running. The interrupts can wake up the device from sleep modes.

Related Links

22. PM – Power Manager

36.5.3 Clocks

The SERCOM bus clock (CLK_SERCOMx_APB) can be enabled and disabled in the Main Clock Controller. Refer to *Peripheral Clock Masking* for details and default status of this clock.

A generic clock (GCLK_SERCOMx_CORE) is required to clock the SPI. This clock must be configured and enabled in the Generic Clock Controller before using the SPI.

This generic clock is asynchronous to the bus clock (CLK_SERCOMx_APB). Therefore, writes to certain registers will require synchronization to the clock domains.

Related Links

18. GCLK - Generic Clock Controller
19.6.2.6 Peripheral Clock Masking
36.6.6 Synchronization

36.5.4 DMA

The DMA request lines are connected to the DMA Controller (DMAC). In order to use DMA requests with this peripheral the DMAC must be configured first. Refer to *DMAC – Direct Memory Access Controller* for details.

Related Links

28. DMAC – Direct Memory Access Controller

36.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. Refer to *Nested Vector Interrupt Controller* for details.

36.5.6 Events

Not applicable.

36.5.7 Debug Operation

When the CPU is halted in debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

36.5.8 Register Access Protection

Registers with write-access can be write-protected optionally by the peripheral access controller (PAC).

PAC Write-Protection is not available for the following registers:

- Interrupt Flag Clear and Status register (INTFLAG)

Downloaded from Arrow.com.

- Status register (STATUS)
- Data register (DATA)

Optional PAC Write-Protection is denoted by the "PAC Write-Protection" property in each individual register description.

Write-protection does not apply to accesses through an external debugger.

Related Links

15. PAC - Peripheral Access Controller

36.5.9 SAM L11 TrustZone Specific Register Access Protection

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

36.5.10 Analog Connections

Not applicable.

36.6 Functional Description

36.6.1 Principle of Operation

The SPI is a high-speed synchronous data transfer interface It allows high-speed communication between the device and peripheral devices.

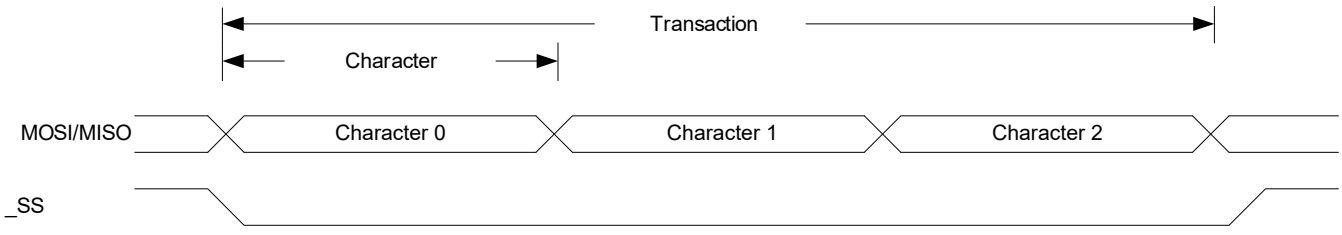
The SPI can operate as master or slave. As master, the SPI initiates and controls all data transactions. The SPI is single buffered for transmitting and double buffered for receiving.

When transmitting data, the Data register can be loaded with the next character to be transmitted during the current transmission.

When receiving, the data is transferred to the two-level or four-level receive buffer, and the receiver is ready for a new character.

The SPI transaction format is shown in SPI Transaction Format. Each transaction can contain one or more characters. The character size is configurable, and can be either 8 or 9 bits.

Figure 36-2. SPI Transaction Format



Downloaded from Arrow.com.

The SPI master must pull the slave select line ($\overline{SS}$) of the desired slave low to initiate a transaction. The master and slave prepare data to send via their respective shift registers, and the master generates the serial clock on the SCK line.

Data are always shifted from master to slave on the Master Output Slave Input line (MOSI); data is shifted from slave to master on the Master Input Slave Output line (MISO).

Each time character is shifted out from the master, a character will be shifted out from the slave simultaneously. To signal the end of a transaction, the master will pull the $\overline{SS}$ line high

36.6.2 Basic Operation

36.6.2.1 Initialization

The following registers are enable-protected, meaning that they can only be written when the SPI is disabled (CTRL.ENABLE=0):

- Control A register (CTRLA), except Enable (CTRLA.ENABLE) and Software Reset (CTRLA.SWRST)
- Control B register (CTRLB), except Receiver Enable (CTRLB.RXEN)
- Baud register (BAUD)
- Address register (ADDR)

When the SPI is enabled or is being enabled (CTRLA.ENABLE=1), any writing to these registers will be discarded.

when the SPI is being disabled, writing to these registers will be completed after the disabling.

Enable-protection is denoted by the Enable-Protection property in the register description.

Initialize the SPI by following these steps:

1. Select SPI mode in master / slave operation in the Operating Mode bit group in the CTRLA register (CTRLA.MODE= 0x2 or 0x3).
2. Select transfer mode for the Clock Polarity bit and the Clock Phase bit in the CTRLA register (CTRLA.CPOL and CTRLA.CPHA) if desired.
3. Select the Frame Format value in the CTRLA register (CTRLA.FORM).
4. Configure the Data In Pinout field in the Control A register (CTRLA.DIPO) for SERCOM pads of the receiver.
5. Configure the Data Out Pinout bit group in the Control A register (CTRLA.DOPO) for SERCOM pads of the transmitter.
6. Select the Character Size value in the CTRLB register (CTRLB.CHSIZE).
7. Write the Data Order bit in the CTRLA register (CTRLA.DORD) for data direction.
8. If the SPI is used in master mode:
    8.1. Select the desired baud rate by writing to the Baud register (BAUD).
    8.2. If Hardware SS control is required, write '1' to the Master Slave Select Enable bit in CTRLB register (CTRLB.MSSEN).
9. Enable the receiver by writing the Receiver Enable bit in the CTRLB register (CTRLB.RXEN=1).

36.6.2.2 Enabling, Disabling, and Resetting

This peripheral is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE), and disabled by writing '0' to it.

Writing ‘1’ to the Software Reset bit in the Control A register (CTRLA.SWRST) will reset all registers of this peripheral to their initial states, except the DBGCTRL register, and the peripheral is disabled.

Downloaded from Arrow.com.

Refer to the CTRLA register description for details.

36.6.2.3 Clock Generation

In SPI master operation (CTRLA.MODE=0x3), the serial clock (SCK) is generated internally by the SERCOM baud-rate generator.

In SPI mode, the baud-rate generator is set to synchronous mode. The 8-bit Baud register (BAUD) value is used for generating SCK and clocking the shift register. Refer to *Clock Generation – Baud-Rate Generator* for more details.

In SPI slave operation (CTRLA.MODE is 0x2), the clock is provided by an external master on the SCK pin. This clock is used to directly clock the SPI shift register.

Related Links

34.6.2.3 Clock Generation – Baud-Rate Generator
34.6.2.3.1 Asynchronous Arithmetic Mode BAUD Value Selection

36.6.2.4 Data Register

The SPI Transmit Data register (TxDATA) and SPI Receive Data register (RxDATA) share the same I/O address, referred to as the SPI Data register (DATA). Writing DATA register will update the Transmit Data register. Reading the DATA register will return the contents of the Receive Data register.

36.6.2.5 SPI Transfer Modes

There are four combinations of SCK phase and polarity to transfer serial data. The SPI data transfer modes are shown in SPI Transfer Modes (Table) and SPI Transfer Modes (Figure).

SCK phase is configured by the Clock Phase bit in the CTRLA register (CTRLA.CPHA). SCK polarity is programmed by the Clock Polarity bit in the CTRLA register (CTRLA.CPOL). Data bits are shifted out and latched in on opposite edges of the SCK signal. This ensures sufficient time for the data signals to stabilize.

Table 36-3. SPI Transfer Modes

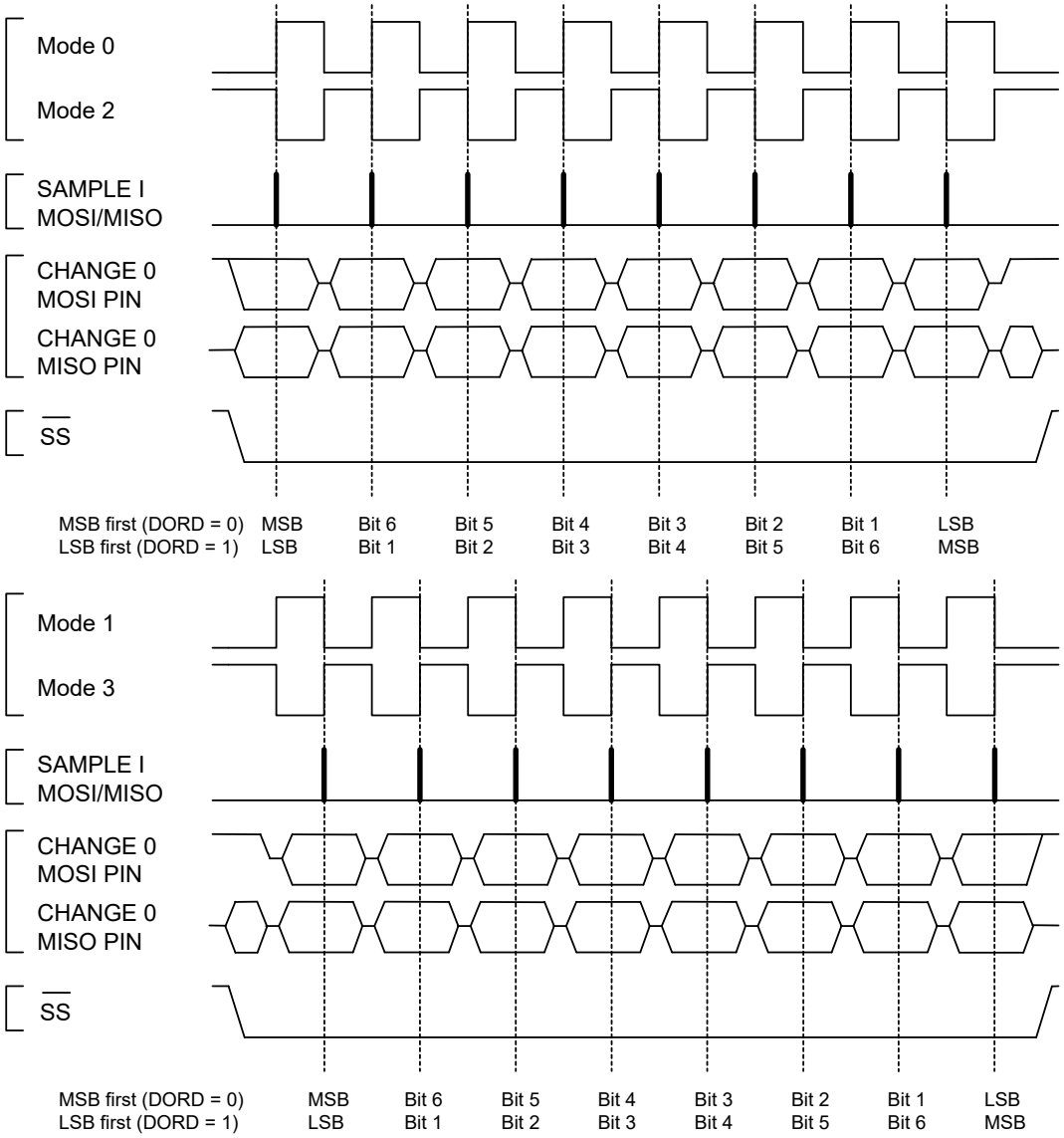
Mode	CPOL	CPHA	Leading Edge	Trailing Edge
0	0	0	Rising, sample	Falling, setup
1	0	1	Rising, setup	Falling, sample
2	1	0	Falling, sample	Rising, setup
3	1	1	Falling, setup	Rising, sample

Note:
Leading edge is the first clock edge in a clock cycle.

Trailing edge is the second clock edge in a clock cycle.

Downloaded from Arrow.com.

Figure 36-3. SPI Transfer Modes



36.6.2.6 Transferring Data

36.6.2.6.1 Master

In master mode (CTRLA.MODE=0x3), when Master Slave Enable Select (CTRLB.MSSEN) is ‘1’, hardware will control the $\overline{SS}$ line.

When Master Slave Select Enable (CTRLB.MSSEN) is '0', the $\overline{SS}$ line must be configured as an output. $\overline{SS}$ can be assigned to any general purpose I/O pin. When the SPI is ready for a data transaction, software must pull the $\overline{SS}$ line low.

When writing a character to the Data register (DATA), the character will be transferred to the shift register. Once the content of TxDATA has been transferred to the shift register, the Data Register Empty flag in the Interrupt Flag Status and Clear register (INTFLAG.DRE) will be set. And a new character can be written to DATA.

Each time one character is shifted out from the master, another character will be shifted in from the slave simultaneously. If the receiver is enabled (CTRLA.RXEN=1), the contents of the shift register will be

Downloaded from Arrow.com.

transferred to the two-level or four-level receive buffer. The transfer takes place in the same clock cycle as the last data bit is shifted in. And the Receive Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) will be set. The received data can be retrieved by reading DATA.

When the last character has been transmitted and there is no valid data in DATA, the Transmit Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.TXC) will be set. When the transaction is finished, the master must pull the $\overline{SS}$ line high to notify the slave. If Master Slave Select Enable (CTRLB.MSSEN) is set to '0', the software must pull the $\overline{SS}$ line high.

36.6.2.6.2 Slave

In slave mode (CTRLA.MODE=0x2), the SPI interface will remain inactive with the MISO line tri-stated as long as the $\overline{SS}$ pin is pulled high. Software may update the contents of DATA at any time as long as the Data Register Empty flag in the Interrupt Status and Clear register (INTFLAG.DRE) is set.

When $\overline{SS}$ is pulled low and SCK is running, the slave will sample and shift out data according to the transaction mode set. When the content of TxDATA has been loaded into the shift register, INTFLAG.DRE will be set, and new data can be written to DATA.

Similar to the master, the slave will receive one character for each character transmitted. A character will be transferred into the two-level or four-level receive buffer within the same clock cycle its last data bit is received. The received character can be retrieved from DATA when the Receive Complete interrupt flag (INTFLAG.RXC) is set.

When the master pulls the $\overline{SS}$ line high, the transaction is done and the Transmit Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.TXC) will be set.

After DATA is written it takes up to three SCK clock cycles until the content of DATA is ready to be loaded into the shift register on the next character boundary. As a consequence, the first character transferred in a SPI transaction will not be the content of DATA. This can be avoided by using the preloading feature. Refer to 36.6.3.2 Preloading of the Slave Shift Register.

When transmitting several characters in one SPI transaction, the data has to be written into DATA register with at least three SCK clock cycles left in the current character transmission. If this criteria is not met, the previously received character will be transmitted.

Once the DATA register is empty, it takes three CLK_SERCOM_APB cycles for INTFLAG.DRE to be set.

36.6.2.7 Receiver Error Bit

The SPI receiver has one error bit: the Buffer Overflow bit (BUFOVF), which can be read from the Status register (STATUS). Once an error happens, the bit will stay set until it is cleared by writing '1' to it. The bit is also automatically cleared when the receiver is disabled.

There are two methods for buffer overflow notification, selected by the immediate buffer overflow notification bit in the Control A register (CTRLA.IBON):

If CTRLA.IBON=1, STATUS.BUFOVF is raised immediately upon buffer overflow. Software can then empty the receive FIFO by reading RxDATA until the receiver complete interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) goes low.

If CTRLA.IBON=0, the buffer overflow condition travels with data through the receive FIFO. After the received data is read, STATUS.BUFOVF and INTFLAG.ERROR will be set along with INTFLAG.RXC, and RxDATA will be zero.

Downloaded from Arrow.com.

36.6.3 Additional Features

36.6.3.1 Address Recognition

When the SPI is configured for slave operation (CTRLA.MODE=0x2) with address recognition (CTRLA.FORM is 0x2), the SERCOM address recognition logic is enabled: the first character in a transaction is checked for an address match.

If there is a match, the Receive Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) is set, the MISO output is enabled, and the transaction is processed. If the device is in sleep mode, an address match can wake up the device in order to process the transaction.

If there is no match, the complete transaction is ignored.

If a 9-bit frame format is selected, only the lower 8 bits of the shift register are checked against the Address register (ADDR).

Preload must be disabled (CTRLB.PLOADEN=0) in order to use this mode.

Related Links

34.6.3.1 Address Match and Mask

36.6.3.2 Preloading of the Slave Shift Register

When starting a transaction, the slave will first transmit the contents of the shift register before loading new data from DATA. The first character sent can be either the reset value of the shift register (if this is the first transmission since the last reset) or the last character in the previous transmission.

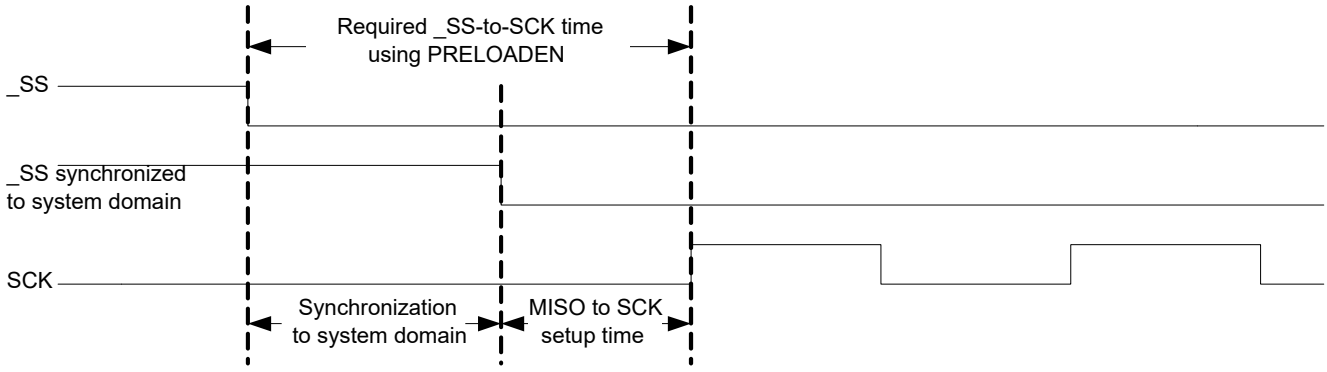
Preloading can be used to preload data into the shift register while $\overline{SS}$ is high: this eliminates sending a dummy character when starting a transaction. If the shift register is not preloaded, the current contents of the shift register will be shifted out.

Only one data character will be preloaded into the shift register while the synchronized $\overline{SS}$ signal is high. If the next character is written to DATA before $\overline{SS}$ is pulled low, the second character will be stored in DATA until transfer begins.

For proper preloading, sufficient time must elapse between $\overline{SS}$ going low and the first SCK sampling edge, as in Timing Using Preloading. See also the *Electrical Characteristics* chapters for timing details.

Preloading is enabled by writing '1' to the Slave Data Preload Enable bit in the CTRLB register (CTRLB.PLOADEN).

Figure 36-4. Timing Using Preloading



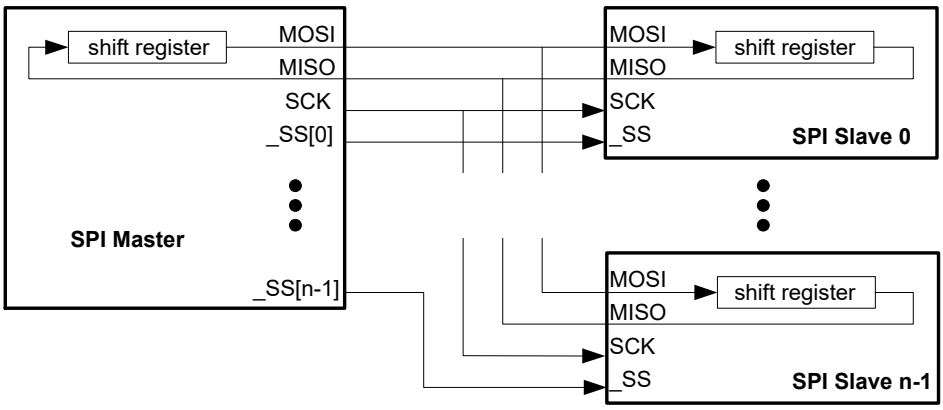
36.6.3.3 Master with Several Slaves

Master with multiple slaves in parallel is only available when Master Slave Select Enable (CTRLB.MSSEN) is set to zero and hardware $\overline{SS}$ control is disabled. If the bus consists of several SPI

Downloaded from Arrow.com.

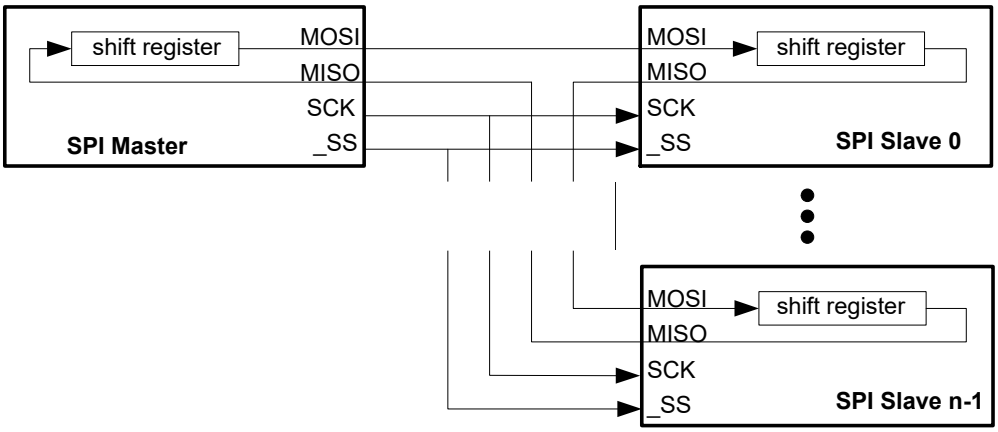
slaves, an SPI master can use general purpose I/O pins to control the $\overline{SS}$ line to each of the slaves on the bus, as shown in Multiple Slaves in Parallel. In this configuration, the single selected SPI slave will drive the tri-state MISO line.

Figure 36-5. Multiple Slaves in Parallel



Another configuration is multiple slaves in series, as in Multiple Slaves in Series. In this configuration, all n attached slaves are connected in series. A common $\overline{SS}$ line is provided to all slaves, enabling them simultaneously. The master must shift n characters for a complete transaction. Depending on the Master Slave Select Enable bit (CTRLB.MSSEN), the $\overline{SS}$ line can be controlled either by hardware or user software and normal GPIO.

Figure 36-6. Multiple Slaves in Series



36.6.3.4 Loop-Back Mode

For loop-back mode, configure the Data In Pinout (CTRLA.DIPO) and Data Out Pinout (CTRLA.DOPO) to use the same data pins for transmit and receive. The loop-back is through the pad, so the signal is also available externally.

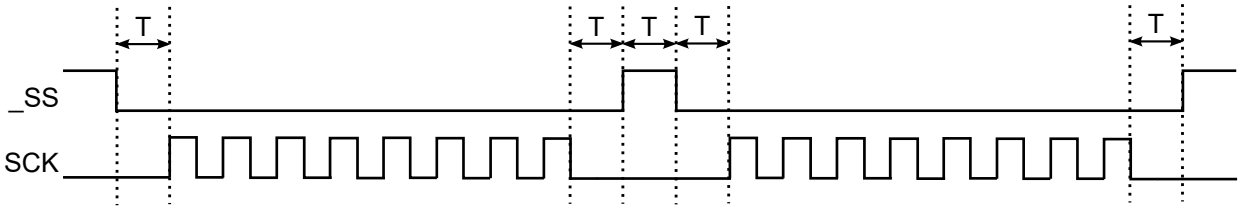
36.6.3.5 Hardware Controlled $\overline{SS}$

In master mode, a single $\overline{SS}$ chip select can be controlled by hardware by writing the Master Slave Select Enable (CTRLB.MSSEN) bit to '1'. In this mode, the $\overline{SS}$ pin is driven low for a minimum of one baud cycle before transmission begins, and stays low for a minimum of one baud cycle after transmission completes. If back-to-back frames are transmitted, the $\overline{SS}$ pin will always be driven high for a minimum of one baud cycle between frames.

In Hardware Controlled $\overline{SS}$, the time T is between one and two baud cycles depending on the SPI transfer mode.

Downloaded from Arrow.com.

Figure 36-7. Hardware Controlled $\overline{SS}$



T = 1 to 2 baud cycles

When CTRLB.MSSEN=0, the $\overline{SS}$ pin(s) is/are controlled by user software and normal GPIO.

36.6.3.6 Slave Select Low Detection

In slave mode, the SPI can wake the CPU when the slave select ($\overline{SS}$) goes low. When the Slave Select Low Detect is enabled (CTRLB.SSDE=1), a high-to-low transition will set the Slave Select Low interrupt flag (INTFLAG.SSL) and the device will wake up if applicable.

36.6.4 DMA, Interrupts, and Events

Table 36-4. Module Request for SERCOM SPI

Condition	Request		
	DMA	Interrupt	Event
Data Register Empty (DRE)	Yes (request cleared when data is written)	Yes	NA
Receive Complete (RXC)	Yes (request cleared when data is read)	Yes	
Transmit Complete (TXC)	NA	Yes	
Slave Select low (SSL)	NA	Yes	
Error (ERROR)	NA	Yes	

36.6.4.1 DMA Operation

The SPI generates the following DMA requests:

- Data received (RX): The request is set when data is available in the receive FIFO. The request is cleared when DATA is read.
- Data transmit (TX): The request is set when the transmit buffer (TX DATA) is empty. The request is cleared when DATA is written.

36.6.4.2 Interrupts

The SPI has the following interrupt sources. These are asynchronous interrupts, and can wake up the device from any sleep mode:

- Data Register Empty (DRE)
- Receive Complete (RXC)
- Transmit Complete (TXC)
- Slave Select Low (SSL)
- Error (ERROR)

Downloaded from Arrow.com.

Each interrupt source has its own interrupt flag. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) will be set when the interrupt condition is met. Each interrupt can be individually enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the interrupt flag is set and if the corresponding interrupt is enabled. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is disabled, or the SPI is reset. For details on clearing interrupt flags, refer to the INTFLAG register description.

The value of INTFLAG indicates which interrupt is executed. Note that interrupts must be globally enabled for interrupt requests. Refer to *Nested Vector Interrupt Controller* for details.

36.6.4.3 Events

Not applicable.

36.6.5 Sleep Mode Operation

The behavior in Sleep mode is depending on the master/slave configuration and the Run In Standby bit in the Control A register (CTRLA.RUNSTDBY):

- Master operation, CTRLA.RUNSTDBY=1: The peripheral clock GCLK_SERCOM_CORE will continue to run in idle sleep mode and in Standby Sleep mode. Any interrupt can wake up the device.
- Master operation, CTRLA.RUNSTDBY=0: GLK_SERCOMx_CORE will be disabled after the ongoing transaction is finished. Any interrupt can wake up the device.
- Slave operation, CTRLA.RUNSTDBY=1: The Receive Complete interrupt can wake up the device.
- Slave operation, CTRLA.RUNSTDBY=0: All reception will be dropped, including the ongoing transaction.

36.6.6 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in the CTRLA register (CTRLA.SWRST)
- Enable bit in the CTRLA register (CTRLA.ENABLE)
- Receiver Enable bit in the CTRLB register (CTRLB.RXEN)

Note: CTRLB.RXEN is write-synchronized somewhat differently. See also *CTRLB* register for details.

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Downloaded from Arrow.com.

36.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
		15:8								IBON
		23:16			DIPO[1:0]				DOPO[1:0]	
		31:24		DORD	CPOL	CPHA	FORM[3:0]			
0x04	CTRLB	7:0		PLOADEN				CHSIZE[2:0]		
		15:8	AMODE[1:0]		MSSEN				SSDE	
		23:16							RXEN	
		31:24								
0x08 ... 0x0B	Reserved									
0x0C	BAUD	7:0	BAUD[7:0]							
0x0D ... 0x13	Reserved									
0x14	INTENCLR	7:0	ERROR				SSL	RXC	TXC	DRE
0x15	Reserved									
0x16	INTENSET	7:0	ERROR				SSL	RXC	TXC	DRE
0x17	Reserved									
0x18	INTFLAG	7:0	ERROR				SSL	RXC	TXC	DRE
0x19	Reserved									
0x1A	STATUS	7:0						BUFOVF		
		15:8								
0x1C	SYNCBUSY	7:0						CTRLB	ENABLE	SWRST
		15:8								
		23:16								
		31:24								
0x20 ... 0x23	Reserved									
0x24	ADDR	7:0	ADDR[7:0]							
		15:8								
		23:16	ADDRMASK[7:0]							
		31:24								
0x28	DATA	7:0	DATA[7:0]							
		15:8								DATA[8:8]
0x2A ... 0x2F	Reserved									
0x30	DBGCTRL	7:0								DBGSTOP

Downloaded from Arrow.com.

36.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Refer to 36.6.6 Synchronization

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Refer to 36.5.8 Register Access Protection.

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

Downloaded from Arrow.com.

36.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
		DORD	CPOL	CPHA	FORM[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
			DIPO[1:0]				DOPO[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	15	14	13	12	11	10	9	8
								IBON
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit 30 – DORD Data Order
This bit selects the data order when a character is shifted out from the shift register.

This bit is not synchronized.

Value	Description
0	MSB is transferred first.
1	LSB is transferred first.

Bit 29 – CPOL Clock Polarity
In combination with the Clock Phase bit (CPHA), this bit determines the SPI transfer mode.

This bit is not synchronized.

Value	Description
0	SCK is low when idle. The leading edge of a clock cycle is a rising edge, while the trailing edge is a falling edge.
1	SCK is high when idle. The leading edge of a clock cycle is a falling edge, while the trailing edge is a rising edge.

Bit 28 – CPHA Clock Phase
In combination with the Clock Polarity bit (CPOL), this bit determines the SPI transfer mode.

Downloaded from Arrow.com.

This bit is not synchronized.

Mode	CPOL	CPHA	Leading Edge	Trailing Edge
0x0	0	0	Rising, sample	Falling, change
0x1	0	1	Rising, change	Falling, sample
0x2	1	0	Falling, sample	Rising, change
0x3	1	1	Falling, change	Rising, sample

Value	Description
0	The data is sampled on a leading SCK edge and changed on a trailing SCK edge.
1	The data is sampled on a trailing SCK edge and changed on a leading SCK edge.

Bits 27:24 – FORM[3:0] Frame Format
This bit field selects the various frame formats supported by the SPI in slave mode. When the 'SPI frame with address' format is selected, the first byte received is checked against the ADDR register.

FORM[3:0]	Name	Description
0x0	SPI	SPI frame
0x1	-	Reserved
0x2	SPI_ADDR	SPI frame with address
0x3-0xF	-	Reserved

Bits 21:20 – DIPO[1:0] Data In Pinout
These bits define the data in (DI) pad configurations.

In master operation, DI is MISO.

In slave operation, DI is MOSI.

These bits are not synchronized.

DIPO[1:0]	Name	Description
0x0	PAD[0]	SERCOM PAD[0] is used as data input
0x1	PAD[1]	SERCOM PAD[1] is used as data input
0x2	PAD[2]	SERCOM PAD[2] is used as data input
0x3	PAD[3]	SERCOM PAD[3] is used as data input

Bits 17:16 – DOPO[1:0] Data Out Pinout
This bit defines the available pad configurations for data out (DO) and the serial clock (SCK). In slave operation, the slave select line ($\overline{SS}$) is controlled by DOPO, while in master operation the $\overline{SS}$ line is controlled by the port configuration.

In master operation, DO is MOSI.

In slave operation, DO is MISO.

These bits are not synchronized.

Downloaded from Arrow.com.

DOPO	DO	SCK	Slave $\overline{SS}$	Master $\overline{SS}$
0x0	PAD[0]	PAD[1]	PAD[2]	System configuration
0x1	PAD[2]	PAD[3]	PAD[1]	System configuration
0x2	PAD[3]	PAD[1]	PAD[2]	System configuration
0x3	PAD[0]	PAD[3]	PAD[1]	System configuration

Bit 8 – IBON Immediate Buffer Overflow Notification
This bit controls when the buffer overflow status bit (STATUS.BUFOVF) is set when a buffer overflow occurs.

This bit is not synchronized.

Value	Description
0	STATUS.BUFOVF is set when it occurs in the data stream.
1	STATUS.BUFOVF is set immediately upon buffer overflow.

Bit 7 – RUNSTDBY Run In Standby
This bit defines the functionality in standby sleep mode.

These bits are not synchronized.

RUNSTDBY	Slave	Master
0x0	Disabled. All reception is dropped, including the ongoing transaction.	Generic clock is disabled when ongoing transaction is finished. All interrupts can wake up the device.
0x1	Ongoing transaction continues, wake on Receive Complete interrupt.	Generic clock is enabled while in sleep modes. All interrupts can wake up the device.

Bits 4:2 – MODE[2:0] Operating Mode
These bits must be written to 0x2 or 0x3 to select the SPI serial communication interface of the SERCOM.

0x2: SPI slave operation

0x3: SPI master operation

These bits are not synchronized.

Bit 1 – ENABLE Enable
Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRL.ENABLE will read back immediately and the Synchronization Enable Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE is cleared when the operation is complete.

This bit is not enable-protected.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled or being enabled.

Downloaded from Arrow.com.

Bit 0 – SWRST Software Reset
Writing '0' to this bit has no effect.

Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state, and the SERCOM will be disabled.

Writing "1' to CTRL.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded. Any register write access during the ongoing reset will result in an APB error. Reading any register will return the reset value of the register.

Due to synchronization, there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY. SWRST will both be cleared when the reset is complete.

This bit is not enable-protected.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

Downloaded from Arrow.com.

36.8.2 Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							RXEN	
Access							R/W	
Reset							0	

Bit	15	14	13	12	11	10	9	8
	AMODE[1:0]		MSSEN				SSDE	
Access	R/W	R/W	R/W				R/W	
Reset	0	0	0				0	

Bit	7	6	5	4	3	2	1	0
		PLOADEN				CHSIZE[2:0]		
Access		R/W				R/W	R/W	R/W
Reset		0				0	0	0

Bit 17 – RXEN Receiver Enable
Writing '0' to this bit will disable the SPI receiver immediately. The receive buffer will be flushed, data from ongoing receptions will be lost and STATUS.BUFOVF will be cleared.

Writing '1' to CTRLB.RXEN when the SPI is disabled will set CTRLB.RXEN immediately. When the SPI is enabled, CTRLB.RXEN will be cleared, SYNCBUSY.CTRLB will be set and remain set until the receiver is enabled. When the receiver is enabled CTRLB.RXEN will read back as '1'.

Writing '1' to CTRLB.RXEN when the SPI is enabled will set SYNCBUSY.CTRLB, which will remain set until the receiver is enabled, and CTRLB.RXEN will read back as '1'.

This bit is not enable-protected.

Value	Description
0	The receiver is disabled or being enabled.
1	The receiver is enabled or it will be enabled when SPI is enabled.

Bits 15:14 – AMODE[1:0] Address Mode
These bits set the slave addressing mode when the frame format (CTRLA.FORM) with address is used. They are unused in master mode.

Downloaded from Arrow.com.

AMODE[1:0]	Name	Description
0x0	MASK	ADDRMASK is used as a mask to the ADDR register
0x1	2_ADDRS	The slave responds to the two unique addresses in ADDR and ADDRMASK
0x2	RANGE	The slave responds to the range of addresses between and including ADDR and ADDRMASK. ADDR is the upper limit
0x3	-	Reserved

Bit 13 – MSSEN Master Slave Select Enable
This bit enables hardware slave select ($\overline{SS}$) control.

Value	Description
0	Hardware $\overline{SS}$ control is disabled.
1	Hardware $\overline{SS}$ control is enabled.

Bit 9 – SSDE Slave Select Low Detect Enable
This bit enables wake up when the slave select ($\overline{SS}$) pin transitions from high to low.

Value	Description
0	$\overline{SS}$ low detector is disabled.
1	$\overline{SS}$ low detector is enabled.

Bit 6 – PLOADEN Slave Data Preload Enable
Setting this bit will enable preloading of the slave shift register when there is no transfer in progress. If the $\overline{SS}$ line is high when DATA is written, it will be transferred immediately to the shift register.

Bits 2:0 – CHSIZE[2:0] Character Size

CHSIZE[2:0]	Name	Description
0x0	8BIT	8 bits
0x1	9BIT	9 bits
0x2-0x7	-	Reserved

Downloaded from Arrow.com.

36.8.3 Baud Rate

Name: BAUD
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
	BAUD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – BAUD[7:0] Baud Register
These bits control the clock generation, as described in the *SERCOM Clock Generation – Baud-Rate Generator*.

Related Links

34.6.2.3 Clock Generation – Baud-Rate Generator
34.6.2.3.1 Asynchronous Arithmetic Mode BAUD Value Selection

Downloaded from Arrow.com.

36.8.4 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Bit	7	6	5	4	3	2	1	0
	ERROR				SSL	RXC	TXC	DRE
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit 7 – ERROR Error Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 3 – SSL Slave Select Low Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Slave Select Low Interrupt Enable bit, which disables the Slave Select Low interrupt.

Value	Description
0	Slave Select Low interrupt is disabled.
1	Slave Select Low interrupt is enabled.

Bit 2 – RXC Receive Complete Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Receive Complete Interrupt Enable bit, which disables the Receive Complete interrupt.

Value	Description
0	Receive Complete interrupt is disabled.
1	Receive Complete interrupt is enabled.

Bit 1 – TXC Transmit Complete Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Transmit Complete Interrupt Enable bit, which disable the Transmit Complete interrupt.

Value	Description
0	Transmit Complete interrupt is disabled.
1	Transmit Complete interrupt is enabled.

Downloaded from Arrow.com.

Bit 0 – DRE Data Register Empty Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Data Register Empty Interrupt Enable bit, which disables the Data Register Empty interrupt.

Value	Description
0	Data Register Empty interrupt is disabled.
1	Data Register Empty interrupt is enabled.

Downloaded from Arrow.com.

36.8.5 Interrupt Enable Set

Name: INTENSET
Offset: 0x16
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Bit	7	6	5	4	3	2	1	0
	ERROR				SSL	RXC	TXC	DRE
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit 7 – ERROR Error Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 3 – SSL Slave Select Low Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Slave Select Low Interrupt Enable bit, which enables the Slave Select Low interrupt.

Value	Description
0	Slave Select Low interrupt is disabled.
1	Slave Select Low interrupt is enabled.

Bit 2 – RXC Receive Complete Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Receive Complete Interrupt Enable bit, which enables the Receive Complete interrupt.

Value	Description
0	Receive Complete interrupt is disabled.
1	Receive Complete interrupt is enabled.

Bit 1 – TXC Transmit Complete Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Transmit Complete Interrupt Enable bit, which enables the Transmit Complete interrupt.

Value	Description
0	Transmit Complete interrupt is disabled.
1	Transmit Complete interrupt is enabled.

Downloaded from Arrow.com.

Bit 0 – DRE Data Register Empty Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Data Register Empty Interrupt Enable bit, which enables the Data Register Empty interrupt.

Value	Description
0	Data Register Empty interrupt is disabled.
1	Data Register Empty interrupt is enabled.

Downloaded from Arrow.com.

36.8.6 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	ERROR				SSL	RXC	TXC	DRE
Access	R/W				R/W	R	R/W	R
Reset	0				0	0	0	0

Bit 7 – ERROR Error
This flag is cleared by writing '1' to it.

This bit is set when any error is detected. Errors that will set this flag have corresponding status flags in the STATUS register. The BUFOVF error will set this interrupt flag.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 3 – SSL Slave Select Low
This flag is cleared by writing '1' to it.

This bit is set when a high to low transition is detected on the _SS pin in slave mode and Slave Select Low Detect (CTRLB.SSDE) is enabled.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 2 – RXC Receive Complete
This flag is cleared by reading the Data (DATA) register or by disabling the receiver.

This flag is set when there are unread data in the receive buffer. If address matching is enabled, the first data received in a transaction will be an address.

Writing '0' to this bit has no effect.

Writing '1' to this bit has no effect.

Bit 1 – TXC Transmit Complete
This flag is cleared by writing '1' to it or by writing new data to DATA.

In master mode, this flag is set when the data have been shifted out and there are no new data in DATA.

In slave mode, this flag is set when the _SS pin is pulled high. If address matching is enabled, this flag is only set if the transaction was initiated with an address match.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Downloaded from Arrow.com.

Bit 0 – DRE Data Register Empty
This flag is cleared by writing new data to DATA.

This flag is set when DATA is empty and ready for new data to transmit.

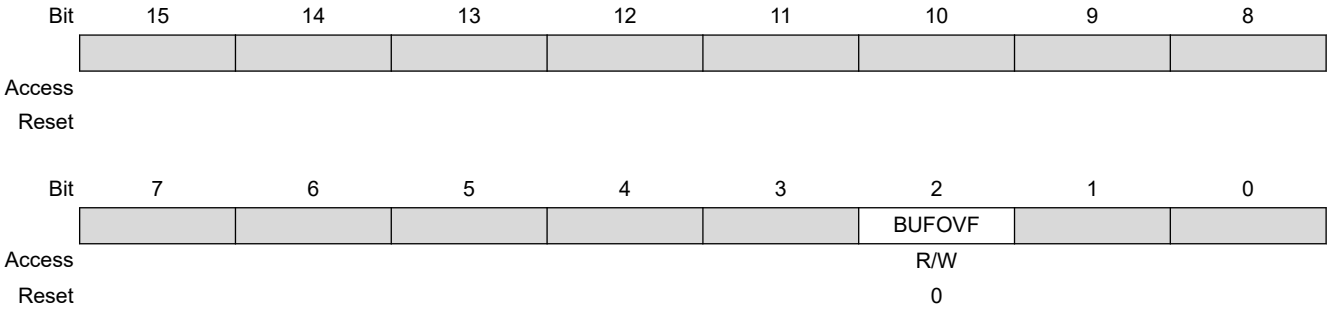
Writing '0' to this bit has no effect.

Writing '1' to this bit has no effect.

Downloaded from Arrow.com.

36.8.7 Status

Name: STATUS
Offset: 0x1A
Reset: 0x0000
Property: –



Bit 2 – BUFOVF Buffer Overflow
Reading this bit before reading DATA will indicate the error status of the next character to be read.

This bit is cleared by writing '1' to the bit or by disabling the receiver.

This bit is set when a buffer overflow condition is detected. See also CTRLA.IBON for overflow handling.

When set, the corresponding RxDATA will be zero.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

Value	Description
0	No Buffer Overflow has occurred.
1	A Buffer Overflow has occurred.

Downloaded from Arrow.com.

36.8.8 Synchronization Busy

Name: SYNCBUSY
Offset: 0x1C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						CTRLB	ENABLE	SWRST
Access						R	R	R
Reset						0	0	0

Bit 2 – CTRLB CTRLB Synchronization Busy
Writing to the CTRLB when the SERCOM is enabled requires synchronization. Ongoing synchronization is indicated by SYNCBUSY.CTRLB=1 until synchronization is complete. If CTRLB is written while SYNCBUSY.CTRLB=1, an APB error will be generated.

Value	Description
0	CTRLB synchronization is not busy.
1	CTRLB synchronization is busy.

Bit 1 – ENABLE SERCOM Enable Synchronization Busy
Enabling and disabling the SERCOM (CTRLA.ENABLE) requires synchronization. Ongoing synchronization is indicated by SYNCBUSY.ENABLE=1 until synchronization is complete.

Value	Description
0	Enable synchronization is not busy.
1	Enable synchronization is busy.

Bit 0 – SWRST Software Reset Synchronization Busy
Resetting the SERCOM (CTRLA.SWRST) requires synchronization. Ongoing synchronization is indicated by SYNCBUSY.SWRST=1 until synchronization is complete.

Downloaded from Arrow.com.

Value	Description
0	SWRST synchronization is not busy.
1	SWRST synchronization is busy.

Downloaded from Arrow.com.

36.8.9 Address

Name: ADDR
Offset: 0x24
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	ADDRMASK[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – ADDRMASK[7:0] Address Mask
These bits hold the address mask when the transaction format with address is used (CTRLA.FORM, CTRLB.AMODE).

Bits 7:0 – ADDR[7:0] Address
These bits hold the address when the transaction format with address is used (CTRLA.FORM, CTRLB.AMODE).

Downloaded from Arrow.com.

36.8.10 Data

Name: DATA
Offset: 0x28
Reset: 0x0000
Property: –

Bit	15	14	13	12	11	10	9	8
								DATA[8:8]
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

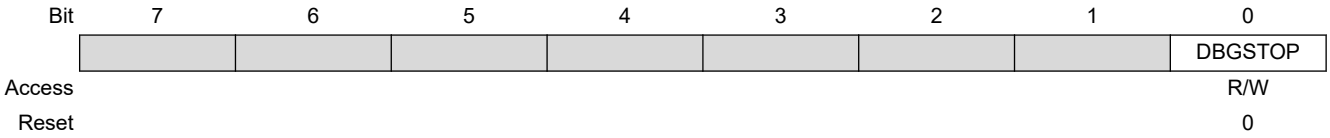
Bits 8:0 – DATA[8:0] Data
Reading these bits will return the contents of the receive data buffer. The register should be read only when the Receive Complete Interrupt Flag bit in the Interrupt Flag Status and Clear register (INTFLAG.RXC) is set.

Writing these bits will write the transmit data buffer. This register should be written only when the Data Register Empty Interrupt Flag bit in the Interrupt Flag Status and Clear register (INTFLAG.DRE) is set.

Downloaded from Arrow.com.

36.8.11 Debug Control

Name: DBGCTRL
Offset: 0x30
Reset: 0x00
Property: PAC Write-Protection



Bit 0 – DBGSTOP Debug Stop Mode
This bit controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The baud-rate generator continues normal operation when the CPU is halted by an external debugger.
1	The baud-rate generator is halted when the CPU is halted by an external debugger.

Downloaded from Arrow.com.

37. SERCOM I²C – SERCOM Inter-Integrated Circuit

37.1 Overview

The inter-integrated circuit (I²C) interface is one of the available modes in the serial communication interface (SERCOM).

The I²C interface uses the SERCOM transmitter and receiver configured as shown in Figure 37-1. Labels in capital letters are registers accessible by the CPU, while lowercase labels are internal to the SERCOM.

A SERCOM instance can be configured to be either an I²C master or an I²C slave. Both master and slave have an interface containing a shift register, a transmit buffer and a receive buffer. In addition, the I²C master uses the SERCOM baud-rate generator, while the I²C slave uses the SERCOM address match logic.

Related Links

34. SERCOM – Serial Communication Interface

37.2 Features

SERCOM I²C includes the following features:

- Master or slave operation
- Can be used with DMA
- Philips I²C compatible
- SMBus™ compatible
- PMBus compatible
- Support of 100kHz and 400kHz, 1MHz and 3.4MHz I²C mode
- 4-Wire operation supported
- Physical interface includes:
  - Slew-rate limited outputs
  - Filtered inputs
- Slave operation:
  - Operation in all sleep modes
  - Wake-up on address match
  - 7-bit and 10-bit Address match in hardware for:
  -
    - Unique address and/or 7-bit general call address
    - Address range
    - Two unique addresses can be used with DMA

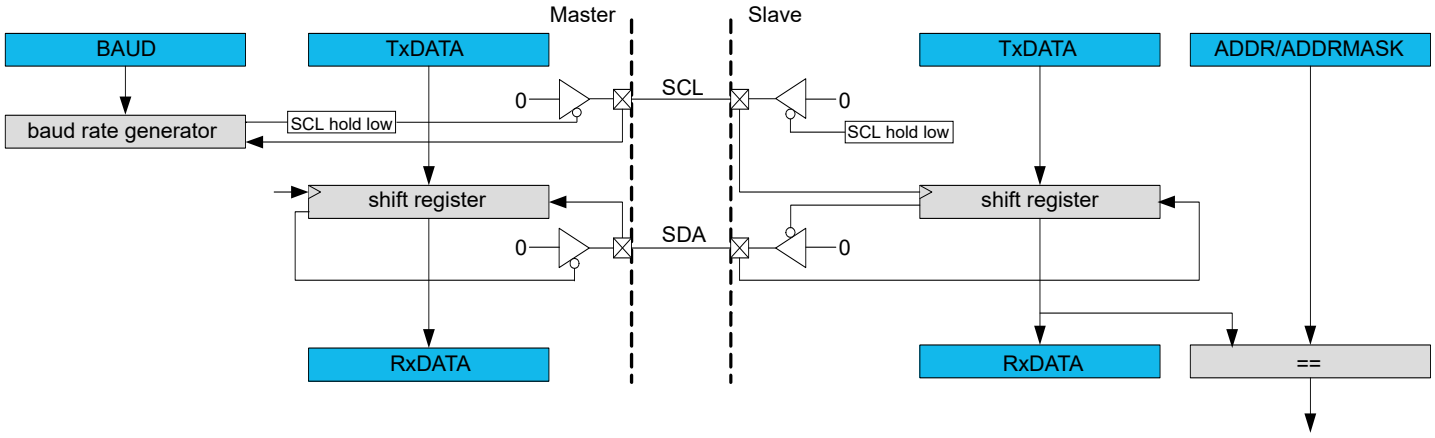
Related Links

34.2 Features

Downloaded from Arrow.com.

37.3 Block Diagram

Figure 37-1. I²C Single-Master Single-Slave Interconnection



37.4 Signal Description

Signal Name	Type	Description
PAD[0]	Digital I/O	SDA
PAD[1]	Digital I/O	SCL
PAD[2]	Digital I/O	SDA_OUT (4-wire operation)
PAD[3]	Digital I/O	SCL_OUT (4-wire operation)

One signal can be mapped on several pins.

Not all the pins are I²C pins.

Related Links

37.6.3.3 4-Wire Mode

37.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

37.5.1 I/O Lines

In order to use the I/O lines of this peripheral, the I/O pins must be configured using the I/O Pin Controller (PORT).

When the SERCOM is used in I²C mode, the SERCOM controls the direction and value of the I/O pins. Both PORT control bits PINCFGn.PULLEN and PINCFGn.DRVSTR are still effective. If the receiver or transmitter is disabled, these pins can be used for other purposes.

Related Links

32. PORT - I/O Pin Controller

Downloaded from Arrow.com.

37.5.2 Power Management

This peripheral can continue to operate in any sleep mode where its source clock is running. The interrupts can wake up the device from sleep modes.

Related Links

22. PM – Power Manager

37.5.3 Clocks

The SERCOM bus clock (CLK_SERCOMx_APB) can be enabled and disabled in the Main Clock Controller. Refer to *Peripheral Clock Masking* for details and default status of this clock.

Two generic clocks are used by SERCOM, GCLK_SERCOMx_CORE and GCLK_SERCOM_SLOW. The core clock (GCLK_SERCOMx_CORE) can clock the I^2C when working as a master. The slow clock (GCLK_SERCOM_SLOW) is required only for certain functions, e.g. SMBus timing. These two clocks must be configured and enabled in the Generic Clock Controller (GCLK) before using the I^2C.

These generic clocks are asynchronous to the bus clock (CLK_SERCOMx_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to 37.6.6 Synchronization for further details.

Related Links

18. GCLK - Generic Clock Controller

19.6.2.6 Peripheral Clock Masking

22. PM – Power Manager

37.5.4 DMA

The DMA request lines are connected to the DMA Controller (DMAC). In order to use DMA requests with this peripheral the DMAC must be configured first. Refer to *DMAC – Direct Memory Access Controller* for details.

Related Links

28. DMAC – Direct Memory Access Controller

37.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. Refer to *Nested Vector Interrupt Controller* for details.

37.5.6 Events

Not applicable.

37.5.7 Debug Operation

When the CPU is halted in debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

37.5.8 Register Access Protection

Registers with write-access can be write-protected optionally by the peripheral access controller (PAC).

PAC Write-Protection is not available for the following registers:

Downloaded from Arrow.com.

- Interrupt Flag Clear and Status register (INTFLAG)
- Status register (STATUS)
- Data register (DATA)
- Address register (ADDR)

Optional PAC Write-Protection is denoted by the "PAC Write-Protection" property in each individual register description.

Write-protection does not apply to accesses through an external debugger.

Related Links

15. PAC - Peripheral Access Controller

37.5.9 SAM L11 TrustZone Specific Register Access Protection

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

37.5.10 Analog Connections

Not applicable.

37.6 Functional Description

37.6.1 Principle of Operation

The I^2C interface uses two physical lines for communication:

- Serial Data Line (SDA) for data transfer
- Serial Clock Line (SCL) for the bus clock

A transaction starts with the I^2C master sending the start condition, followed by a 7-bit address and a direction bit (read or write to/from the slave).

The addressed I^2C slave will then acknowledge (ACK) the address, and data packet transactions can begin. Every 9-bit data packet consists of 8 data bits followed by a one-bit reply indicating whether the data was acknowledged or not.

If a data packet is not acknowledged (NACK), whether by the I^2C slave or master, the I^2C master takes action by either terminating the transaction by sending the stop condition, or by sending a repeated start to transfer more data.

The figure below illustrates the possible transaction formats and Transaction Diagram Symbols explains the transaction symbols. These symbols will be used in the following descriptions.

Downloaded from Arrow.com.

Figure 37-2. Transaction Diagram Symbols

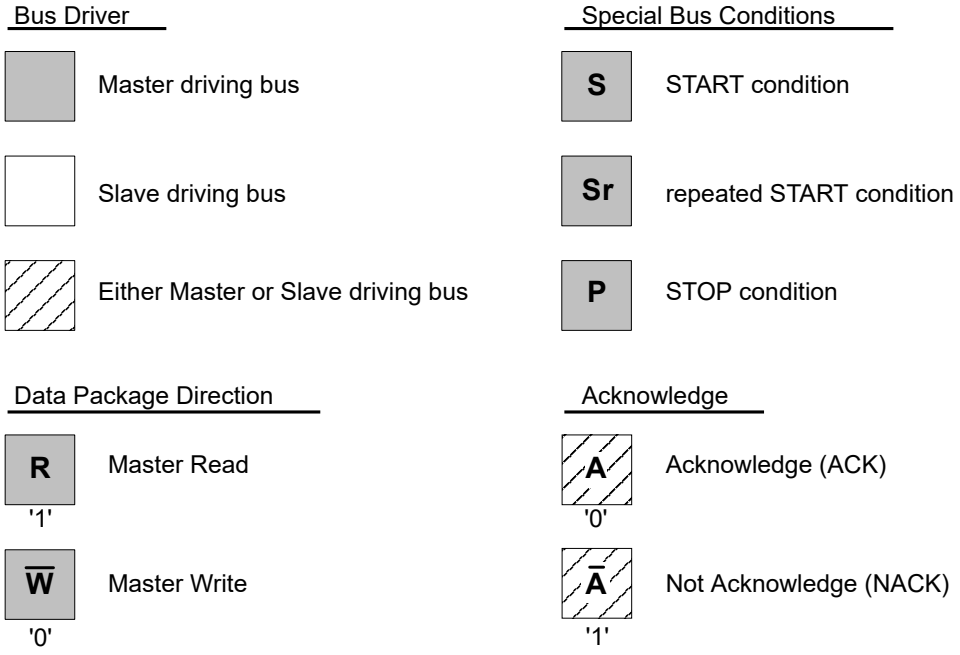
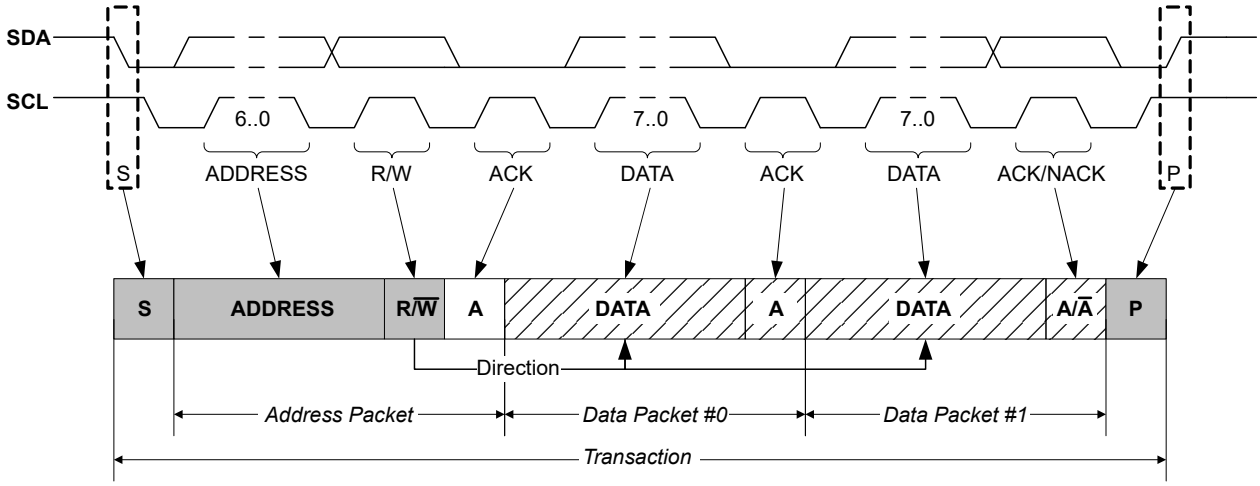


Figure 37-3. Basic I²C Transaction Diagram



37.6.2 Basic Operation

37.6.2.1 Initialization

The following registers are enable-protected, meaning they can be written only when the I²C interface is disabled (CTRLA.ENABLE is '0'):

- Control A register (CTRLA), except Enable (CTRLA.ENABLE) and Software Reset (CTRLA.SWRST) bits
- Control B register (CTRLB), except Acknowledge Action (CTRLB.ACKACT) and Command (CTRLB.CMD) bits
- Baud register (BAUD)
- Address register (ADDR) in slave operation.

When the I²C is enabled or is being enabled (CTRLA.ENABLE=1), writing to these registers will be discarded. If the I²C is being disabled, writing to these registers will be completed after the disabling.

Downloaded from Arrow.com.

Enable-protection is denoted by the "Enable-Protection" property in the register description.

Before the I²C is enabled it must be configured as outlined by the following steps:

1. Select I²C Master or Slave mode by writing 0x4 (Slave mode) or 0x5 (Master mode) to the Operating Mode bits in the CTRLA register (CTRLA.MODE).
2. If desired, select the SDA Hold Time value in the CTRLA register (CTRLA.SDAHOLD).
3. If desired, enable smart operation by setting the Smart Mode Enable bit in the CTRLB register (CTRLB.SMEN).
4. If desired, enable SCL low time-out by setting the SCL Low Time-Out bit in the Control A register (CTRLA.LOWTOUT).
5. In Master mode:
   - 5.1. Select the inactive bus time-out in the Inactive Time-Out bit group in the CTRLA register (CTRLA.INACTOUT).
   - 5.2. Write the Baud Rate register (BAUD) to generate the desired baud rate.

   In Slave mode:
   - 5.1. Configure the address match configuration by writing the Address Mode value in the CTRLB register (CTRLB.AMODE).
   - 5.2. Set the Address and Address Mask value in the Address register (ADDR.ADDR and ADDR.ADDRMASK) according to the address configuration.

37.6.2.2 Enabling, Disabling, and Resetting

This peripheral is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE), and disabled by writing '0' to it.

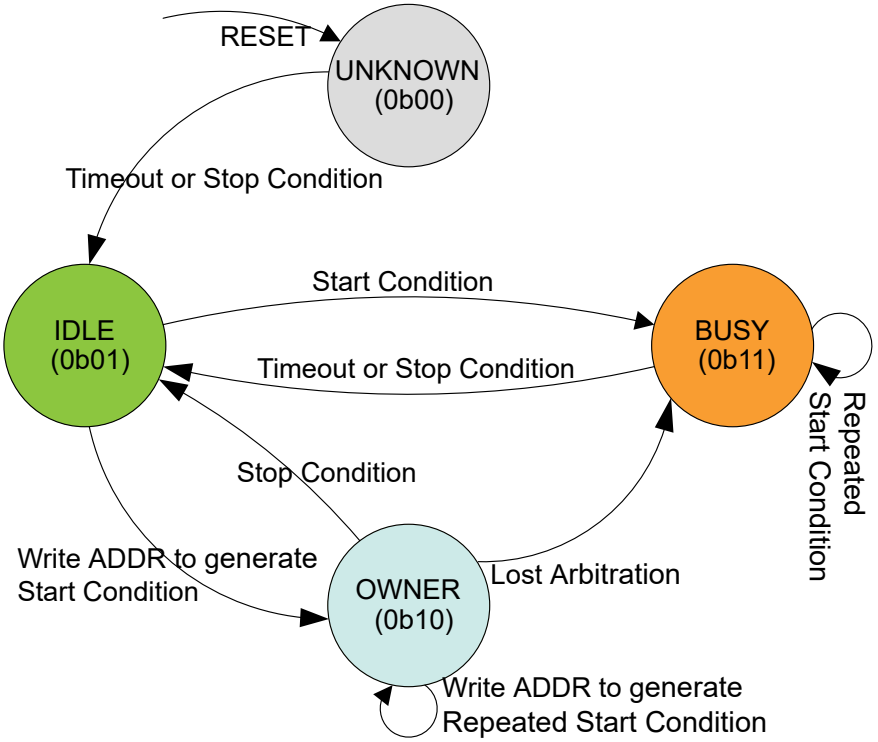
Writing ‘1’ to the Software Reset bit in the Control A register (CTRLA.SWRST) will reset all registers of this peripheral to their initial states, except the DBGCTRL register, and the peripheral is disabled.

37.6.2.3 I²C Bus State Logic

The bus state logic includes several logic blocks that continuously monitor the activity on the I²C bus lines in all sleep modes with running GCLK_SERCOM_x clocks. The start and stop detectors and the bit counter are all essential in the process of determining the current bus state. The bus state is determined according to Bus State Diagram. Software can get the current bus state by reading the Master Bus State bits in the Status register (STATUS.BUSSTATE). The value of STATUS.BUSSTATE in the figure is shown in binary.

Downloaded from Arrow.com.

Figure 37-4. Bus State Diagram



The bus state machine is active when the I²C master is enabled.

After the I²C master has been enabled, the bus state is UNKNOWN (0b00). From the UNKNOWN state, the bus will transition to IDLE (0b01) by either:

- Forcing by writing 0b01 to STATUS.BUSSTATE
- A stop condition is detected on the bus
- If the inactive bus time-out is configured for SMBus compatibility (CTRLA.INACTOUT) and a time-out occurs.

Note: Once a known bus state is established, the bus state logic will not re-enter the UNKNOWN state.

When the bus is IDLE it is ready for a new transaction. If a start condition is issued on the bus by another I²C master in a multi-master setup, the bus becomes BUSY (0b11). The bus will re-enter IDLE either when a stop condition is detected, or when a time-out occurs (inactive bus time-out needs to be configured).

If a start condition is generated internally by writing the Address bit group in the Address register (ADDR.ADDR) while IDLE, the OWNER state (0b10) is entered. If the complete transaction was performed without interference, i.e., arbitration was not lost, the I²C master can issue a stop condition, which will change the bus state back to IDLE.

However, if a packet collision is detected while in OWNER state, the arbitration is assumed lost and the bus state becomes BUSY until a stop condition is detected. A repeated start condition will change the bus state only if arbitration is lost while issuing a repeated start.

Note: Violating the protocol may cause the I²C to hang. If this happens it is possible to recover from this state by a software reset (CTRLA.SWRST='1').

Related Links

37.10.1 CTRLA

Downloaded from Arrow.com.

37.6.2.4 I²C Master Operation

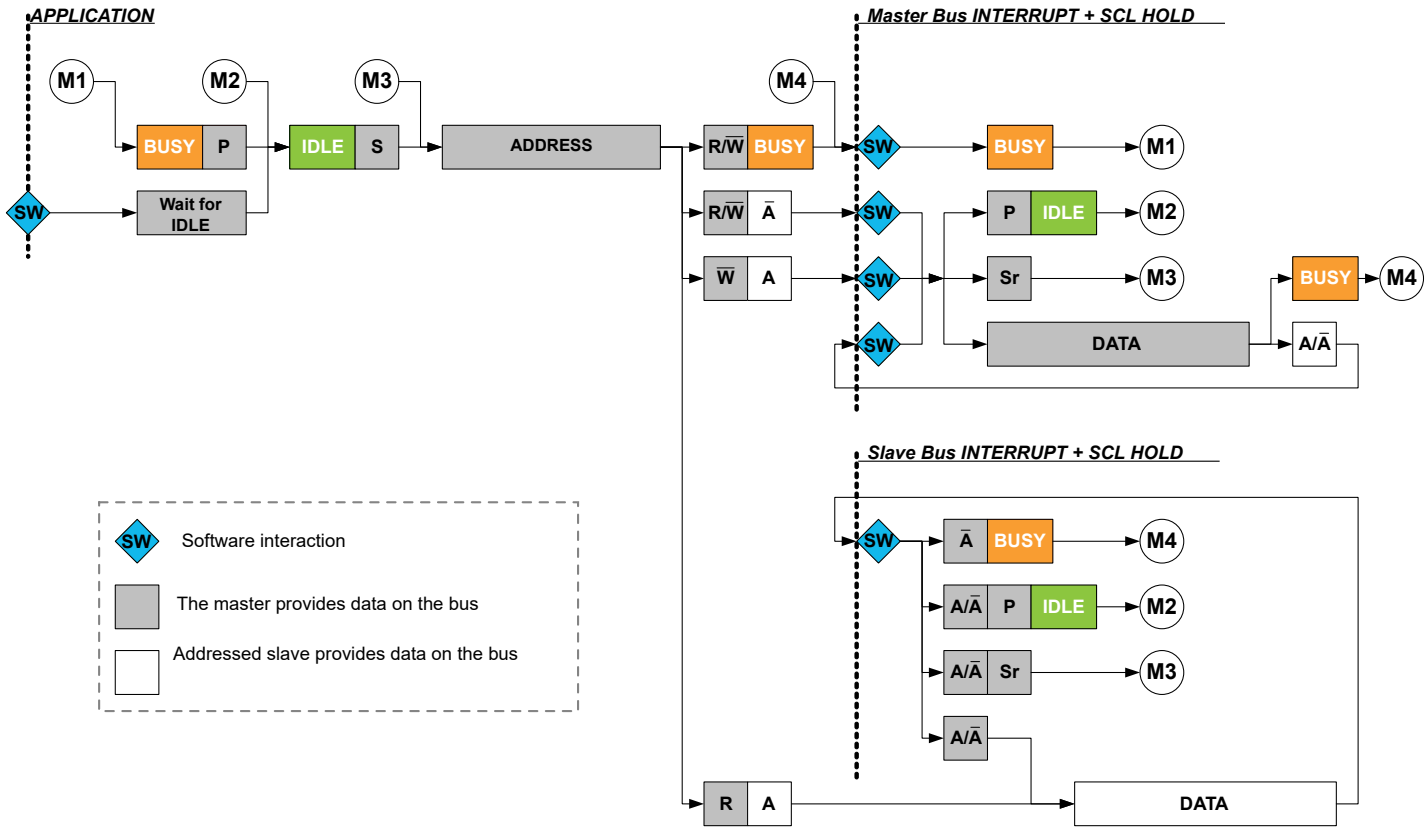
The I²C master is byte-oriented and interrupt based. The number of interrupts generated is kept at a minimum by automatic handling of most incidents. The software driver complexity and code size are reduced by auto-triggering of operations, and a special smart mode, which can be enabled by the Smart Mode Enable bit in the Control A register (CTRLA.SMEN).

The I²C master has two interrupt strategies.

When SCL Stretch Mode (CTRLA.SCLSM) is '0', SCL is stretched before or after the acknowledge bit . In this mode the I²C master operates according to Master Behavioral Diagram (SCLSM=0). The circles labelled "M*n*" (M1, M2..) indicate the nodes the bus logic can jump to, based on software or hardware interaction.

This diagram is used as reference for the description of the I²C master operation throughout the document.

Figure 37-5. I²C Master Behavioral Diagram (SCLSM=0)

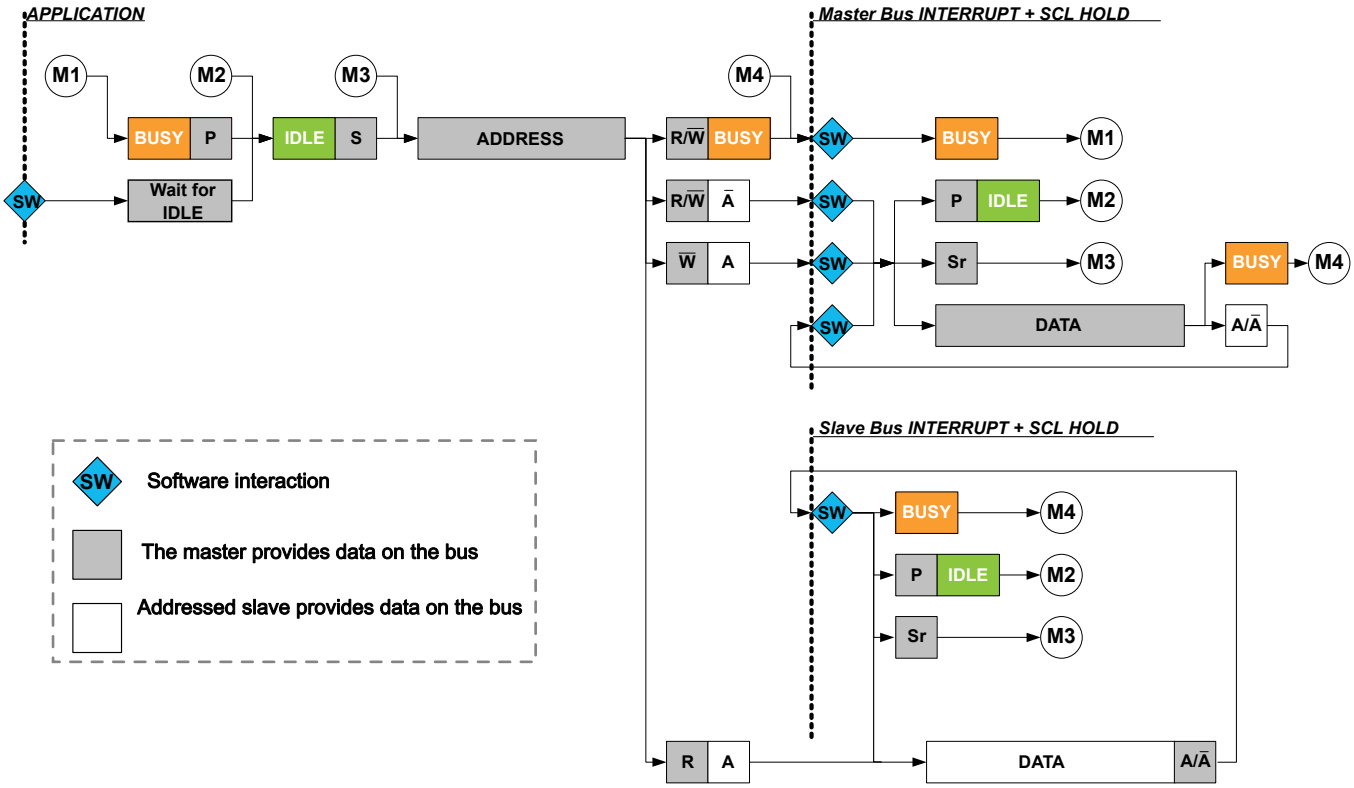


In the second strategy (CTRLA.SCLSM=1), interrupts only occur after the ACK bit, as in Master Behavioral Diagram (SCLSM=1). This strategy can be used when it is not necessary to check DATA before acknowledging.

Note: I²C High-speed (*Hs*) mode requires CTRLA.SCLSM=1.

Downloaded from Arrow.com.

Figure 37-6. I²C Master Behavioral Diagram (SCLSM=1)



37.6.2.4.1 Master Clock Generation

The SERCOM peripheral supports several I²C bidirectional modes:

- Standard mode (*Sm*) up to 100 kHz
- Fast mode (*Fm*) up to 400 kHz
- Fast mode Plus (*Fm+*) up to 1 MHz
- High-speed mode (*Hs*) up to 3.4 MHz

The Master clock configuration for *Sm, Fm,* and *Fm+* are described in Clock Generation (Standard-Mode, Fast-Mode, and Fast-Mode Plus). For *Hs*, refer to Master Clock Generation (High-Speed Mode).

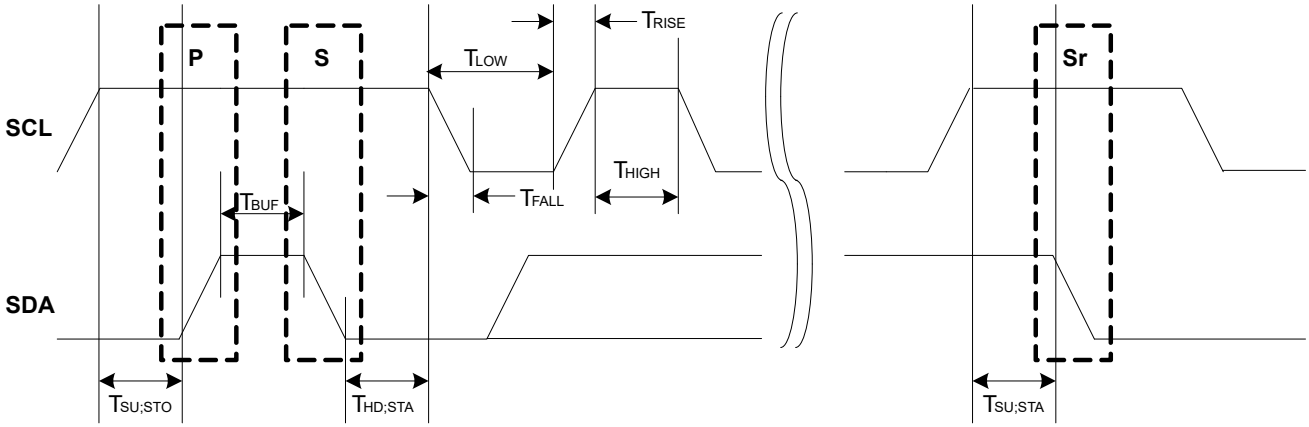
Clock Generation (Standard-Mode, Fast-Mode, and Fast-Mode Plus)

In I²C *Sm, Fm*, and *Fm+* mode, the Master clock (SCL) frequency is determined as described in this section:

The low (T_{LOW}) and high (T_{HIGH}) times are determined by the Baud Rate register (BAUD), while the rise (T_{RISE}) and fall (T_{FALL}) times are determined by the bus topology. Because of the wired-AND logic of the bus, T_{FALL} will be considered as part of T_{LOW}. Likewise, T_{RISE} will be in a state between T_{LOW} and T_{HIGH} until a high state has been detected.

Downloaded from Arrow.com.

Figure 37-7. SCL Timing



The following parameters are timed using the SCL low time period T_{LOW}. This comes from the Master Baud Rate Low bit group in the Baud Rate register (BAUD.BAUDLOW). When BAUD.BAUDLOW=0, or the Master Baud Rate bit group in the Baud Rate register (BAUD.BAUD) determines it.

- T_{LOW} – Low period of SCL clock
- $T_{SU;STO}$ – Set-up time for stop condition
- T_{BUF} – Bus free time between stop and start conditions
- $T_{HD;STA}$ – Hold time (repeated) start condition
- $T_{SU;STA}$ – Set-up time for repeated start condition
- T_{HIGH} is timed using the SCL high time count from BAUD.BAUD
- T_{RISE} is determined by the bus impedance; for internal pull-ups.
- T_{FALL} is determined by the open-drain current limit and bus impedance; can typically be regarded as zero.

The SCL frequency is given by:

$$f_{\mathrm{SCL}} = \frac{1}{T_{\mathrm{LOW}} + T_{\mathrm{HIGH}} + T_{\mathrm{RISE}}}$$

When BAUD.BAUDLOW is zero, the BAUD.BAUD value is used to time both SCL high and SCL low. In this case the following formula will give the SCL frequency:

$$f_{\mathrm{SCL}} = \frac{f_{\mathrm{GCLK}}}{10 + 2BAUD + f_{\mathrm{GCLK}} \cdot T_{\mathrm{RISE}}}$$

When BAUD.BAUDLOW is non-zero, the following formula determines the SCL frequency:

$$f_{\mathrm{SCL}} = \frac{f_{\mathrm{GCLK}}}{10 + BAUD + BAUDLOW + f_{\mathrm{GCLK}} \cdot T_{\mathrm{RISE}}}$$

The following formulas can determine the SCL T_{LOW} and T_{HIGH} times:

$$T_{\mathrm{LOW}} = \frac{BAUDLOW + 5}{f_{\mathrm{GCLK}}}$$

$$T_{\mathrm{HIGH}} = \frac{BAUD + 5}{f_{\mathrm{GCLK}}}$$

Downloaded from Arrow.com.

Note: The I²C standard *Fm+* (Fast-mode plus) requires a nominal high to low SCL ratio of 1:2, and BAUD should be set accordingly. At a minimum, BAUD.BAUD and/or BAUD.BAUDLOW must be non-zero.

Startup Timing The minimum time between SDA transition and SCL rising edge is 6 APB cycles when the DATA register is written in smart mode. If a greater startup time is required due to long rise times, the time between DATA write and IF clear must be controlled by software.
Note: When timing is controlled by user, the Smart Mode cannot be enabled.

Master Clock Generation (High-Speed Mode)
For I²C *Hs* transfers, there is no SCL synchronization. Instead, the SCL frequency is determined by the GCLK_SERCOMx_CORE frequency (f_{GCLK}) and the High-Speed Baud setting in the Baud register (BAUD.HSBAUD). When BAUD.HSBAUDLOW=0, the HSBAUD value will determine both SCL high and SCL low. In this case the following formula determines the SCL frequency.

$$f_{SCL} = \frac{f_{GCLK}}{2 + 2 \cdot HS\ BAUD}$$

When HSBAUDLOW is non-zero, the following formula determines the SCL frequency.

$$f_{SCL} = \frac{f_{GCLK}}{2 + HS\ BAUD + HSBAUDLOW}$$

Note: The I²C standard *Hs* (High-speed) requires a nominal high to low SCL ratio of 1:2, and HSBAUD should be set accordingly. At a minimum, BAUD.HSBAUD and/or BAUD.HSBAUDLOW must be non-zero.

37.6.2.4.2 Transmitting Address Packets

The I²C master starts a bus transaction by writing the I²C slave address to ADDR.ADDR and the direction bit, as described in 37.6.1 Principle of Operation. If the bus is busy, the I²C master will wait until the bus becomes idle before continuing the operation. When the bus is idle, the I²C master will issue a start condition on the bus. The I²C master will then transmit an address packet using the address written to ADDR.ADDR. After the address packet has been transmitted by the I²C master, one of four cases will arise according to arbitration and transfer direction.

Case 1: Arbitration lost or bus error during address packet transmission

If arbitration was lost during transmission of the address packet, the Master on Bus bit in the Interrupt Flag Status and Clear register (INTFLAG.MB) and the Arbitration Lost bit in the Status register (STATUS.ARBLOST) are both set. Serial data output to SDA is disabled, and the SCL is released, which disables clock stretching. In effect the I²C master is no longer allowed to execute any operation on the bus until the bus is idle again. A bus error will behave similarly to the arbitration lost condition. In this case, the MB interrupt flag and Master Bus Error bit in the Status register (STATUS.BUSERR) are both set in addition to STATUS.ARBLOST.

The Master Received Not Acknowledge bit in the Status register (STATUS.RXNACK) will always contain the last successfully received acknowledge or not acknowledge indication.

In this case, software will typically inform the application code of the condition and then clear the interrupt flag before exiting the interrupt routine. No other flags have to be cleared at this moment, because all flags will be cleared automatically the next time the ADDR.ADDR register is written.

Case 2: Address packet transmit complete – No ACK received

If there is no I²C slave device responding to the address packet, then the INTFLAG.MB interrupt flag and STATUS.RXNACK will be set. The clock hold is active at this point, preventing further activity on the bus.

Downloaded from Arrow.com.

The missing ACK response can indicate that the I²C slave is busy with other tasks or sleeping. Therefore, it is not able to respond. In this event, the next step can be either issuing a stop condition (recommended) or resending the address packet by a repeated start condition. When using SMBus logic, the slave must ACK the address. If there is no response, it means that the slave is not available on the bus.

Case 3: Address packet transmit complete – Write packet, Master on Bus set

If the I²C master receives an acknowledge response from the I²C slave, INTFLAG.MB will be set and STATUS.RXNACK will be cleared. The clock hold is active at this point, preventing further activity on the bus.

In this case, the software implementation becomes highly protocol dependent. Three possible actions can enable the I²C operation to continue:

- Initiate a data transmit operation by writing the data byte to be transmitted into DATA.DATA.
- Transmit a new address packet by writing ADDR.ADDR. A repeated start condition will automatically be inserted before the address packet.
- Issue a stop condition, consequently terminating the transaction.

Case 4: Address packet transmit complete – Read packet, Slave on Bus set

If the I²C master receives an ACK from the I²C slave, the I²C master proceeds to receive the next byte of data from the I²C slave. When the first data byte is received, the Slave on Bus bit in the Interrupt Flag register (INTFLAG.SB) will be set and STATUS.RXNACK will be cleared. The clock hold is active at this point, preventing further activity on the bus.

In this case, the software implementation becomes highly protocol dependent. Three possible actions can enable the I²C operation to continue:

- Let the I²C master continue to read data by acknowledging the data received. ACK can be sent by software, or automatically in smart mode.
- Transmit a new address packet.
- Terminate the transaction by issuing a stop condition.

Note: An ACK or NACK will be automatically transmitted if smart mode is enabled. The Acknowledge Action bit in the Control B register (CTRLB.ACKACT) determines whether ACK or NACK should be sent.

37.6.2.4.3 Transmitting Data Packets

When an address packet with direction Master Write (see Figure 37-3) was transmitted successfully , INTFLAG.MB will be set. The I²C master will start transmitting data via the I²C bus by writing to DATA.DATA, and monitor continuously for packet collisions. I

If a collision is detected, the I²C master will lose arbitration and STATUS.ARBLOST will be set. If the transmit was successful, the I²C master will receive an ACK bit from the I²C slave, and STATUS.RXNACK will be cleared. INTFLAG.MB will be set in both cases, regardless of arbitration outcome.

It is recommended to read STATUS.ARBLOST and handle the arbitration lost condition in the beginning of the I²C Master on Bus interrupt. This can be done as there is no difference between handling address and data packet arbitration.

STATUS.RXNACK must be checked for each data packet transmitted before the next data packet transmission can commence. The I²C master is not allowed to continue transmitting data packets if a NACK is received from the I²C slave.

Downloaded from Arrow.com.

37.6.2.4.4 Receiving Data Packets (SCLSM=0)

When INTFLAG.SB is set, the I²C master will already have received one data packet. The I²C master must respond by sending either an ACK or NACK. Sending a NACK may be unsuccessful when arbitration is lost during the transmission. In this case, a lost arbitration will prevent setting INTFLAG.SB. Instead, INTFLAG.MB will indicate a change in arbitration. Handling of lost arbitration is the same as for data bit transmission.

37.6.2.4.5 Receiving Data Packets (SCLSM=1)

When INTFLAG.SB is set, the I²C master will already have received one data packet and transmitted an ACK or NACK, depending on CTRLB.ACKACT. At this point, CTRLB.ACKACT must be set to the correct value for the next ACK bit, and the transaction can continue by reading DATA and issuing a command if not in the smart mode.

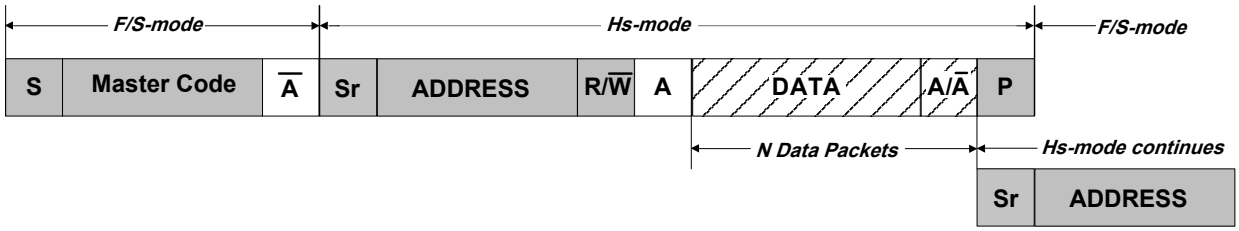
37.6.2.4.6 High-Speed Mode

High-speed transfers are a multi-step process, see High Speed Transfer.

First, a master code (0b00001nnn, where 'nnn' is a unique master code) is transmitted in Full-speed mode, followed by a NACK since no slaveshould acknowledge. Arbitration is performed only during the Full-speed Master Code phase. The master code is transmitted by writing the master code to the address register (ADDR.ADDR) and writing the high-speed bit (ADDR.HS) to '0'.

After the master code and NACK have been transmitted, the master write interrupt will be asserted. In the meanwhile, the slave address can be written to the ADDR.ADDR register together with ADDR.HS=1. Now in High-speed mode, the master will generate a repeated start, followed by the slave address with RW-direction. The bus will remain in High-speed mode until a stop is generated. If a repeated start is desired, the ADDR.HS bit must again be written to '1', along with the new address ADDR.ADDR to be transmitted.

Figure 37-8. High Speed Transfer



Transmitting in High-speed mode requires the I²C master to be configured in High-speed mode (CTRLA.SPEED=0x2) and the SCL clock stretch mode (CTRLA.SCLSM) bit set to '1'.

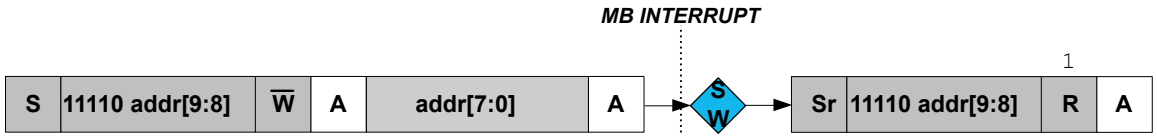
37.6.2.4.7 10-Bit Addressing

When 10-bit addressing is enabled by the Ten Bit Addressing Enable bit in the Address register (ADDR.TENBITEN=1) and the Address bit field ADDR.ADDR is written, the two address bytes will be transmitted, see 10-bit Address Transmission for a Read Transaction. The addressed slave acknowledges the two address bytes, and the transaction continues. Regardless of whether the transaction is a read or write, the master must start by sending the 10-bit address with the direction bit (ADDR.ADDR[0]) being zero.

If the master receives a NACK after the first byte, the write interrupt flag will be raised and the STATUS.RXNACK bit will be set. If the first byte is acknowledged by one or more slaves, then the master will proceed to transmit the second address byte and the master will first see the write interrupt flag after the second byte is transmitted. If the transaction direction is read-from-slave, the 10-bit address transmission must be followed by a repeated start and the first 7 bits of the address with the read/write bit equal to '1'.

Downloaded from Arrow.com.

Figure 37-9. 10-bit Address Transmission for a Read Transaction



This implies the following procedure for a 10-bit read operation:

1. Write the 10-bit address to ADDR.ADDR[10:1]. ADDR.TENBITEN must be '1', the direction bit (ADDR.ADDR[0]) must be '0' (can be written simultaneously with ADDR).
2. Once the Master on Bus interrupt is asserted, Write ADDR[7:0] register to '11110 `address[9:8]` 1'. ADDR.TENBITEN must be cleared (can be written simultaneously with ADDR).
3. Proceed to transmit data.

37.6.2.5 I²C Slave Operation

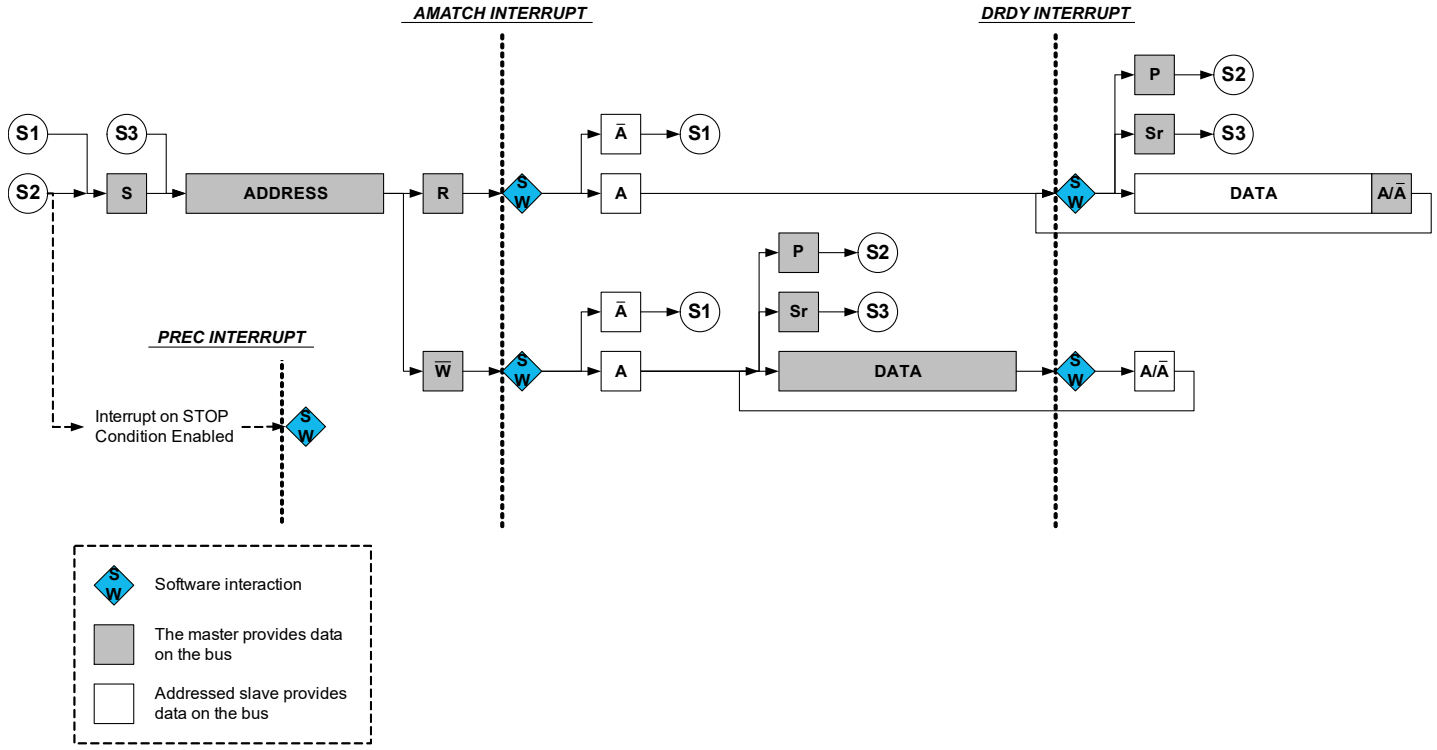
The I²C slave is byte-oriented and interrupt-based. The number of interrupts generated is kept at a minimum by automatic handling of most events. The software driver complexity and code size are reduced by auto-triggering of operations, and a special smart mode, which can be enabled by the Smart Mode Enable bit in the Control A register (CTRLA.SMEN).

The I²C slave has two interrupt strategies.

When SCL Stretch Mode bit (CTRLA.SCLSM) is '0', SCL is stretched before or after the acknowledge bit. In this mode, the I²C slave operates according to I²C Slave Behavioral Diagram (SCLSM=0). The circles labelled "S*n*" (S1, S2..) indicate the nodes the bus logic can jump to, based on software or hardware interaction.

This diagram is used as reference for the description of the I²C slave operation throughout the document.

Figure 37-10. I²C Slave Behavioral Diagram (SCLSM=0)

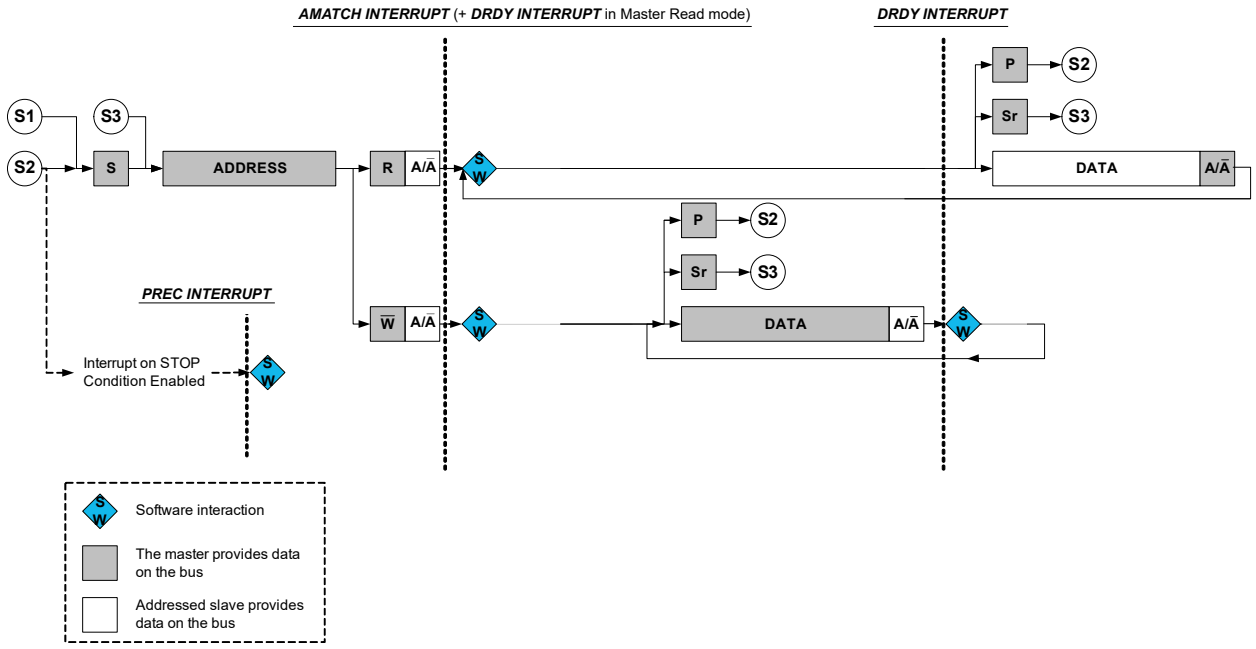


Downloaded from Arrow.com.

In the second strategy (CTRLA.SCLSM=1), interrupts only occur after the ACK bit is sent as shown in Slave Behavioral Diagram (SCLSM=1). This strategy can be used when it is not necessary to check DATA before acknowledging. For master reads, an address and data interrupt will be issued simultaneously after the address acknowledge. However, for master writes, the first data interrupt will be seen after the first data byte has been received by the slave and the acknowledge bit has been sent to the master.

Note: For I²C High-speed mode (*Hs*), SCLSM=1 is required.

Figure 37-11. I²C Slave Behavioral Diagram (SCLSM=1)



37.6.2.5.1 Receiving Address Packets (SCLSM=0)

When CTRLA.SCLSM=0, the I2C slave stretches the SCL line according to Figure 37-10. When the I²C slave is properly configured, it will wait for a start condition.

When a start condition is detected, the successive address packet will be received and checked by the address match logic. If the received address is not a match, the packet will be rejected, and the I²C slave will wait for a new start condition. If the received address is a match, the Address Match bit in the Interrupt Flag register (INTFLAG.AMATCH) will be set.

SCL will be stretched until the I²C slave clears INTFLAG.AMATCH. As the I²C slave holds the clock by forcing SCL low, the software has unlimited time to respond.

The direction of a transaction is determined by reading the Read / Write Direction bit in the Status register (STATUS.DIR). This bit will be updated only when a valid address packet is received.

If the Transmit Collision bit in the Status register (STATUS.COLL) is set, this indicates that the last packet addressed to the I²C slave had a packet collision. A collision causes the SDA and SCL lines to be released without any notification to software. Therefore, the next AMATCH interrupt is the first indication of the previous packet's collision. Collisions are intended to follow the SMBus Address Resolution Protocol (ARP).

After the address packet has been received from the I²C master, one of two cases will arise based on transfer direction.

Downloaded from Arrow.com.

Case 1: Address packet accepted – Read flag set

The STATUS.DIR bit is ‘1’, indicating an I2C master read operation. The SCL line is forced low, stretching the bus clock. If an ACK is sent, I2C slave hardware will set the Data Ready bit in the Interrupt Flag register (INTFLAG.DRDY), indicating data are needed for transmit. If a NACK is sent, the I2C slave will wait for a new start condition and address match.

Typically, software will immediately acknowledge the address packet by sending an ACK/NACK bit. The I2C slave Command bit field in the Control B register (CTRLB.CMD) can be written to '0x3' for both read and write operations as the command execution is dependent on the STATUS.DIR bit. Writing ‘1’ to INTFLAG.AMATCH will also cause an ACK/NACK to be sent corresponding to the CTRLB.ACKACT bit.

Case 2: Address packet accepted – Write flag set

The STATUS.DIR bit is cleared, indicating an I2C master write operation. The SCL line is forced low, stretching the bus clock. If an ACK is sent, the I2C slave will wait for data to be received. Data, repeated start or stop can be received.

If a NACK is sent, the I2C slave will wait for a new start condition and address match. Typically, software will immediately acknowledge the address packet by sending an ACK/NACK. The I2C slave command CTRLB.CMD = 3 can be used for both read and write operation as the command execution is dependent on STATUS.DIR.

Writing ‘1’ to INTFLAG.AMATCH will also cause an ACK/NACK to be sent corresponding to the CTRLB.ACKACT bit.

37.6.2.5.2 Receiving Address Packets (SCLSM=1)

When SCLSM=1, the I2C slave will stretch the SCL line only after an ACK, see Slave Behavioral Diagram (SCLSM=1). When the I2C slave is properly configured, it will wait for a start condition to be detected.

When a start condition is detected, the successive address packet will be received and checked by the address match logic.

If the received address is not a match, the packet will be rejected and the I2C slave will wait for a new start condition.

If the address matches, the acknowledge action as configured by the Acknowledge Action bit Control B register (CTRLB.ACKACT) will be sent and the Address Match bit in the Interrupt Flag register (INTFLAG.AMATCH) is set. SCL will be stretched until the I2C slave clears INTFLAG.AMATCH. As the I2C slave holds the clock by forcing SCL low, the software is given unlimited time to respond to the address.

The direction of a transaction is determined by reading the Read/Write Direction bit in the Status register (STATUS.DIR). This bit will be updated only when a valid address packet is received.

If the Transmit Collision bit in the Status register (STATUS.COLL) is set, the last packet addressed to the I2C slave had a packet collision. A collision causes the SDA and SCL lines to be released without any notification to software. The next AMATCH interrupt is, therefore, the first indication of the previous packet’s collision. Collisions are intended to follow the SMBus Address Resolution Protocol (*ARP*).

After the address packet has been received from the I2C master, INTFLAG.AMATCH be set to ‘1’ to clear it.

37.6.2.5.3 Receiving and Transmitting Data Packets

After the I2C slave has received an address packet, it will respond according to the direction either by waiting for the data packet to be received or by starting to send a data packet by writing to DATA.DATA.

Downloaded from Arrow.com.

When a data packet is received or sent, INTFLAG.DRDY will be set. After receiving data, the I²C slave will send an acknowledge according to CTRLB.ACKACT.

Case 1: Data received

INTFLAG.DRDY is set, and SCL is held low, pending for SW interaction.

Case 2: Data sent

When a byte transmission is successfully completed, the INTFLAG.DRDY interrupt flag is set. If NACK is received, indicated by STATUS.RXNACK=1, the I²C slave must expect a stop or a repeated start to be received. The I²C slave must release the data line to allow the I²C master to generate a stop or repeated start. Upon detecting a stop condition, the Stop Received bit in the Interrupt Flag register (INTFLAG.PREC) will be set and the I²C slave will return to IDLE state.

37.6.2.5.4 High-Speed Mode

When the I²C slave is configured in High-speed mode (*Hs*, CTRLA.SPEED=0x2) and CTRLA.SCLSM=1, switching between Full-speed and High-speed modes is automatic. When the slave recognizes a START followed by a master code transmission and a NACK, it automatically switches to High-speed mode and sets the High-speed status bit (STATUS.HS). The slave will then remain in High-speed mode until a STOP is received.

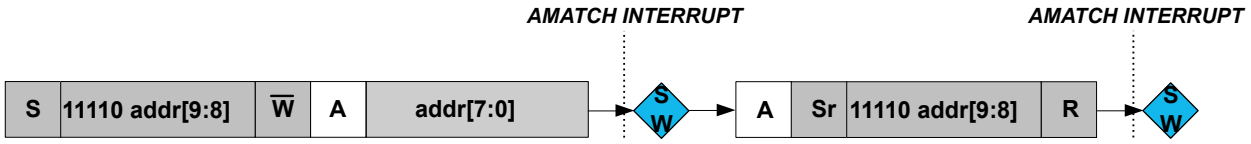
37.6.2.5.5 10-Bit Addressing

When 10-bit addressing is enabled (ADDR.TENBITEN=1), the two address bytes following a START will be checked against the 10-bit slave address recognition. The first byte of the address will always be acknowledged, and the second byte will raise the address interrupt flag, see 10-bit Addressing.

If the transaction is a write, then the 10-bit address will be followed by *N* data bytes.

If the operation is a read, the 10-bit address will be followed by a repeated START and reception of '11110 `ADDR[9:8]` 1', and the second address interrupt will be received with the DIR bit set. The slave matches on the second address as it it was addressed by the previous 10-bit address.

Figure 37-12. 10-bit Addressing



37.6.2.5.6 PMBus Group Command

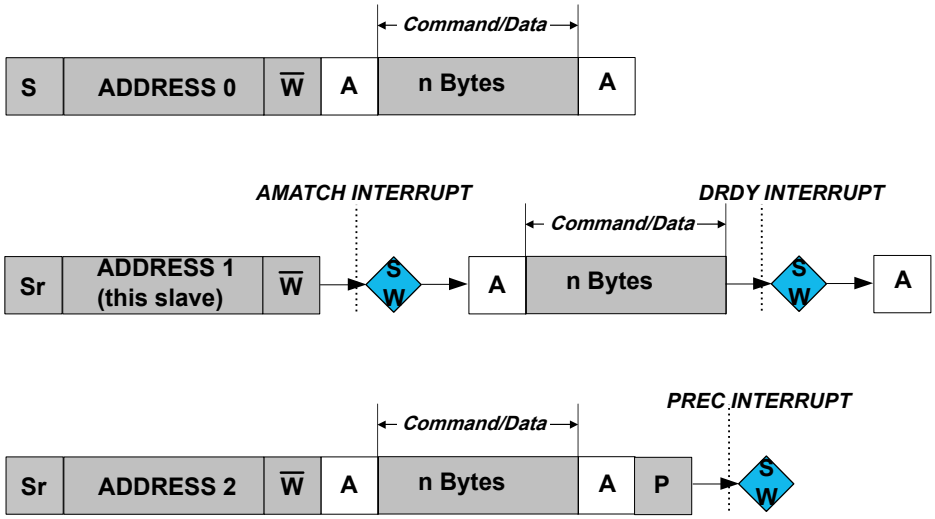
When the PMBus Group Command bit in the CTRLB register is set (CTRLB.GCMD=1) and 7-bit addressing is used, INTFLAG.PREC will be set if the slave has been addressed since the last STOP condition. When CTRLB.GCMD=0, a STOP condition without address match will not be set INTFLAG.PREC.

The group command protocol is used to send commands to more than one device. The commands are sent in one continuous transmission with a single STOP condition at the end. When the STOP condition is detected by the slaves addressed during the group command, they all begin executing the command they received.

PMBus Group Command Example shows an example where this slave, bearing ADDRESS 1, is addressed after a repeated START condition. There can be multiple slaves addressed before and after this slave. Eventually, at the end of the group command, a single STOP is generated by the master. At this point a STOP interrupt is asserted.

Downloaded from Arrow.com.

Figure 37-13. PMBus Group Command Example



37.6.3 Additional Features

37.6.3.1 SMBus

The I^{2}C includes three hardware SCL low time-outs which allow a time-out to occur for SMBus SCL low time-out, master extend time-out, and slave extend time-out. This allows for SMBus functionality These time-outs are driven by the GCLK_SERCOM_SLOW clock. The GCLK_SERCOM_SLOW clock is used to accurately time the time-out and must be configured to use a 32KHz oscillator. The I^{2}C interface also allows for a SMBus compatible SDA hold time.

- $T_{TIMEOUT}$: SCL low time of 25..35ms – Measured for a single SCL low period. It is enabled by CTRLA.LOWTOUTEN.
- $T_{LOW:SEXT}$: Cumulative clock low extend time of 25 ms – Measured as the cumulative SCL low extend time by a slave device in a single message from the initial START to the STOP. It is enabled by CTRLA.SEXTTOEN.
- $T_{LOW:MEXT}$: Cumulative clock low extend time of 10 ms – Measured as the cumulative SCL low extend time by the master device within a single byte from START-to-ACK, ACK-to-ACK, or ACK-to-STOP. It is enabled by CTRLA.MEXTTOEN.

37.6.3.2 Smart Mode

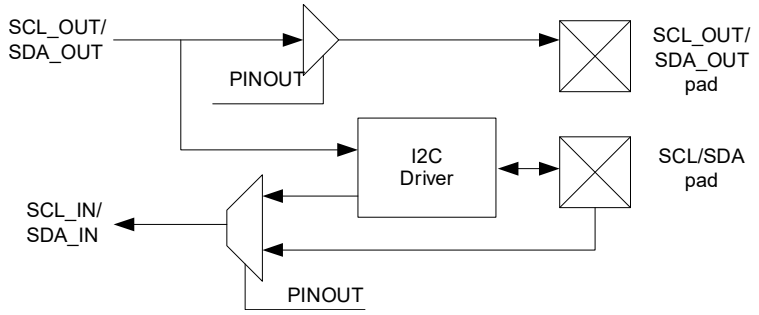
The I^{2}C interface has a smart mode that simplifies application code and minimizes the user interaction needed to adhere to the I^{2}C protocol. The smart mode accomplishes this by automatically issuing an ACK or NACK (based on the content of CTRLB.ACKACT) as soon as DATA.DATA is read.

37.6.3.3 4-Wire Mode

Writing a '1' to the Pin Usage bit in the Control A register (CTRLA.PINOUT) will enable 4-wire mode operation. In this mode, the internal I^{2}C tri-state drivers are bypassed, and an external I^{2}C compliant tri-state driver is needed when connecting to an I^{2}C bus.

Downloaded from Arrow.com.

Figure 37-14. I²C Pad Interface



37.6.3.4 Quick Command

Setting the Quick Command Enable bit in the Control B register (CTRLB.QCEN) enables quick command. When quick command is enabled, the corresponding interrupt flag (INTFLAG.SB or INTFLAG.MB) is set immediately after the slave acknowledges the address. At this point, the software can either issue a stop command or a repeated start by writing CTRLB.CMD or ADDR.ADDR.

37.6.4 DMA, Interrupts and Events

Each interrupt source has its own interrupt flag. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) will be set when the interrupt condition is meet. Each interrupt can be individually enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR). An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request is active until the interrupt flag is cleared, the interrupt is disabled or the I²C is reset. See the 37.8.5 INTFLAG (Slave) or 37.10.6 INTFLAG (Master) register for details on how to clear interrupt flags.

Table 37-1. Module Request for SERCOM I²C Slave

Condition	Request		
	DMA	Interrupt	Event
Data needed for transmit (TX) (Slave transmit mode)	Yes (request cleared when data is written)		NA
Data received (RX) (Slave receive mode)	Yes (request cleared when data is read)		
Data Ready (DRDY)		Yes	
Address Match (AMATCH)		Yes	
Stop received (PREC)		Yes	
Error (ERROR)		Yes	

Downloaded from Arrow.com.

Table 37-2. Module Request for SERCOM I²C Master

Condition	Request		
	DMA	Interrupt	Event
Data needed for transmit (TX) (Master transmit mode)	Yes (request cleared when data is written)		NA
Data needed for transmit (RX) (Master transmit mode)	Yes (request cleared when data is read)		
Master on Bus (MB)		Yes	
Stop received (SB)		Yes	
Error (ERROR)		Yes	

37.6.4.1 DMA Operation

Smart mode must be enabled for DMA operation in the Control B register by writing CTRLB.SMEN=1.

37.6.4.1.1 Slave DMA

When using the I²C slave with DMA, an address match will cause the address interrupt flag (INTFLAG.ADDRMATCH) to be raised. After the interrupt has been serviced, data transfer will be performed through DMA.

The I²C slave generates the following requests:

- Write data received (RX): The request is set when master write data is received. The request is cleared when DATA is read.
- Read data needed for transmit (TX): The request is set when data is needed for a master read operation. The request is cleared when DATA is written.

37.6.4.1.2 Master DMA

When using the I²C master with DMA, the ADDR register must be written with the desired address (ADDR.ADDR), transaction length (ADDR.LEN), and transaction length enable (ADDR.LENEN). When ADDR.LENEN is written to 1 along with ADDR.ADDR, ADDR.LEN determines the number of data bytes in the transaction from 0 to 255. DMA is then used to transfer ADDR.LEN bytes followed by an automatically generated NACK (for master reads) and a STOP.

If a NACK is received by the slave for a master write transaction before ADDR.LEN bytes, a STOP will be automatically generated and the length error (STATUS.LENERR) will be raised along with the INTFLAG.ERROR interrupt.

The I²C master generates the following requests:

- Read data received (RX): The request is set when master read data is received. The request is cleared when DATA is read.
- Write data needed for transmit (TX): The request is set when data is needed for a master write operation. The request is cleared when DATA is written.

Downloaded from Arrow.com.

37.6.4.2 Interrupts

The I²C slave has the following interrupt sources. These are asynchronous interrupts. They can wake-up the device from any sleep mode:

- Error (ERROR)
- Data Ready (DRDY)
- Address Match (AMATCH)
- Stop Received (PREC)

The I²C master has the following interrupt sources. These are asynchronous interrupts. They can wake-up the device from any sleep mode:

- Error (ERROR)
- Slave on Bus (SB)
- Master on Bus (MB)

Each interrupt source has its own interrupt flag. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) will be set when the interrupt condition is meet. Each interrupt can be individually enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR). An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request active until the interrupt flag is cleared, the interrupt is disabled or the I²C is reset. See the INTFLAG register for details on how to clear interrupt flags.

The value of INTFLAG indicates which interrupt is executed. Note that interrupts must be globally enabled for interrupt requests. Refer to *Nested Vector Interrupt Controller* for details.

37.6.4.3 Events

Not applicable.

37.6.5 Sleep Mode Operation

I²C Master Operation

The generic clock (GCLK_SERCOMx_CORE) will continue to run in Idle Sleep mode. If the Run In Standby bit in the Control A register (CTRLA.RUNSTDBY) is '1', the GLK_SERCOMx_CORE will also run in standby Sleep mode. Any interrupt can wake up the device.

If CTRLA.RUNSTDBY=0, the GLK_SERCOMx_CORE will be disabled after any ongoing transaction is finished. Any interrupt can wake up the device.

I²C Slave Operation

Writing CTRLA.RUNSTDBY=1 will allow the Address Match interrupt to wake up the device.

When CTRLA.RUNSTDBY=0, all receptions will be dropped.

37.6.6 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in the CTRLA register (CTRLA.SWRST)
- Enable bit in the CTRLA register (CTRLA.ENABLE)
- Command bits in CTRLB register (CTRLB.CMD)

Downloaded from Arrow.com.

- Write to Bus State bits in the Status register (STATUS.BUSSTATE)
- Address bits in the Address register (ADDR.ADDR) when in master operation.

The following registers are synchronized when written:

- Data (DATA) when in master operation

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Downloaded from Arrow.com.

37.7 Register Summary - I2C Slave

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
		15:8								
		23:16	SEXTTOEN		SDAHOLD[1:0]					PINOUT
		31:24		LOWTOUT			SCLSM		SPEED[1:0]	
0x04	CTRLB	7:0								
		15:8	AMODE[1:0]					AACKEN	GCMD	SMEN
		23:16						ACKACT	CMD[1:0]	
		31:24								
0x08 ... 0x13	Reserved									
0x14	INTENCLR	7:0	ERROR					DRDY	AMATCH	PREC
0x15	Reserved									
0x16	INTENSET	7:0	ERROR					DRDY	AMATCH	PREC
0x17	Reserved									
0x18	INTFLAG	7:0	ERROR					DRDY	AMATCH	PREC
0x19	Reserved									
0x1A	STATUS	7:0	CLKHOLD	LOWTOUT		SR	DIR	RXNACK	COLL	BUSERR
		15:8					LENERR	HS	SEXTTOUT	
0x1C	SYNCBUSY	7:0							ENABLE	SWRST
		15:8								
		23:16								
		31:24								
0x20 ... 0x23	Reserved									
0x24	ADDR	7:0	ADDR[6:0]							GENCEN
		15:8	TENBITEN					ADDR[9:7]		
		23:16	ADDRMASK[6:0]							
		31:24						ADDRMASK[9:7]		
0x28	DATA	7:0	DATA[7:0]							
		15:8								

37.8 Register Description - I2C Slave

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to 37.5.8 Register Access Protection.

Downloaded from Arrow.com.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to 37.6.6 Synchronization.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

Downloaded from Arrow.com.

37.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
		LOWTOUT			SCLSM		SPEED[1:0]	
Access		R/W			R/W		R/W	R/W
Reset		0			0		0	0

Bit	23	22	21	20	19	18	17	16
	SEXTTOEN		SDAHOLD[1:0]					PINOUT
Access	R/W		R/W	R/W				R/W
Reset	0		0	0				0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit 30 – LOWTOUT SCL Low Time-Out
This bit enables the SCL low time-out. If SCL is held low for 25ms-35ms, the slave will release its clock hold, if enabled, and reset the internal state machine. Any interrupt flags set at the time of time-out will remain set.

Value	Description
0	Time-out disabled.
1	Time-out enabled.

Bit 27 – SCLSM SCL Clock Stretch Mode
This bit controls when SCL will be stretched for software interaction.

This bit is not synchronized.

Value	Description
0	SCL stretch according to Figure 37-10
1	SCL stretch only after ACK bit according to Figure 37-11

Bits 25:24 – SPEED[1:0] Transfer Speed
These bits define bus speed.

These bits are not synchronized.

Downloaded from Arrow.com.

Value	Description
0x0	Standard-mode (Sm) up to 100 kHz and Fast-mode (Fm) up to 400 kHz
0x1	Fast-mode Plus (Fm+) up to 1 MHz
0x2	High-speed mode (Hs-mode) up to 3.4 MHz
0x3	Reserved

Bit 23 – SEXTTOEN Slave SCL Low Extend Time-Out
This bit enables the slave SCL low extend time-out. If SCL is cumulatively held low for greater than 25ms from the initial START to a STOP, the slave will release its clock hold if enabled and reset the internal state machine. Any interrupt flags set at the time of time-out will remain set. If the address was recognized, PREC will be set when a STOP is received.

This bit is not synchronized.

Value	Description
0	Time-out disabled
1	Time-out enabled

Bits 21:20 – SDAHOLD[1:0] SDA Hold Time
These bits define the SDA hold time with respect to the negative edge of SCL.

These bits are not synchronized.

Value	Name	Description
0x0	DIS	Disabled
0x1	75	50-100ns hold time
0x2	450	300-600ns hold time
0x3	600	400-800ns hold time

Bit 16 – PINOUT Pin Usage
This bit sets the pin usage to either two- or four-wire operation:

This bit is not synchronized.

Value	Description
0	4-wire operation disabled
1	4-wire operation enabled

Bit 7 – RUNSTDBY Run in Standby
This bit defines the functionality in standby sleep mode.

This bit is not synchronized.

Value	Description
0	Disabled – All reception is dropped.
1	Wake on address match, if enabled.

Bits 4:2 – MODE[2:0] Operating Mode
These bits must be written to 0x04 to select the I^2C slave serial communication interface of the SERCOM.

These bits are not synchronized.

Downloaded from Arrow.com.

Bit 1 – ENABLE Enable
Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRL.ENABLE will read back immediately and the Enable Synchronization Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

This bit is not enable-protected.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled.

Bit 0 – SWRST Software Reset
Writing '0' to this bit has no effect.

Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state, and the SERCOM will be disabled.

Writing '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded. Any register write access during the ongoing reset will result in an APB error. Reading any register will return the reset value of the register.

Due to synchronization, there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

This bit is not enable-protected.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

Downloaded from Arrow.com.

37.8.2 Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						ACKACT	CMD[1:0]	
Access						R/W	W	W
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
	AMODE[1:0]					AACKEN	GCMD	SMEN
Access	R/W	R/W				R/W	R/W	R/W
Reset	0	0				0	0	0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 18 – ACKACT Acknowledge Action
This bit defines the slave's acknowledge behavior after an address or data byte is received from the master. The acknowledge action is executed when a command is written to the CMD bits. If smart mode is enabled (CTRLB.SMEN=1), the acknowledge action is performed when the DATA register is read.

ACKACT shall not be updated more than once between each peripheral interrupts request.

This bit is not enable-protected.

Value	Description
0	Send ACK
1	Send NACK

Bits 17:16 – CMD[1:0] Command
This bit field triggers the slave operation as the below. The CMD bits are strobe bits, and always read as zero. The operation is dependent on the slave interrupt flags, INTFLAG.DRDY and INTFLAG.AMATCH, in addition to STATUS.DIR.

All interrupt flags (INTFLAG.DRDY, INTFLAG.AMATCH and INTFLAG.PREC) are automatically cleared when a command is given.

This bit is not enable-protected.

Downloaded from Arrow.com.

Table 37-3. Command Description

CMD[1:0]	DIR	Action
0x0	X	(No action)
0x1	X	(Reserved)
0x2	Used to complete a transaction in response to a data interrupt (DRDY)	
	0 (Master write)	Execute acknowledge action succeeded by waiting for any start (S/Sr) condition
	1 (Master read)	Wait for any start (S/Sr) condition
0x3	Used in response to an address interrupt (AMATCH)	
	0 (Master write)	Execute acknowledge action succeeded by reception of next byte
	1 (Master read)	Execute acknowledge action succeeded by slave data interrupt
	Used in response to a data interrupt (DRDY)	
	0 (Master write)	Execute acknowledge action succeeded by reception of next byte
	1 (Master read)	Execute a byte read operation followed by ACK/NACK reception

Bits 15:14 – AMODE[1:0] Address Mode
These bits set the addressing mode.

These bits are not write-synchronized.

Value	Name	Description
0x0	MASK	The slave responds to the address written in ADDR.ADDR masked by the value in ADDR.ADDRMASK. See *SERCOM – Serial Communication Interface* for additional information.
0x1	2_ADDRS	The slave responds to the two unique addresses in ADDR.ADDR and ADDR.ADDRMASK.
0x2	RANGE	The slave responds to the range of addresses between and including ADDR.ADDR and ADDR.ADDRMASK. ADDR.ADDR is the upper limit.
0x3	-	Reserved.

Bit 10 – AACKEN Automatic Acknowledge Enable
This bit enables the address to be automatically acknowledged if there is an address match.

This bit is not write-synchronized.

Value	Description
0	Automatic acknowledge is disabled.
1	Automatic acknowledge is enabled.

Bit 9 – GCMD PMBus Group Command
This bit enables PMBus group command support. When enabled, the Stop Recived interrupt flag (INTFLAG.PREC) will be set when a STOP condition is detected if the slave has been addressed since the last STOP condition on the bus.

This bit is not write-synchronized.

Downloaded from Arrow.com.

Value	Description
0	Group command is disabled.
1	Group command is enabled.

Bit 8 – SMEN Smart Mode Enable
When smart mode is enabled, data is acknowledged automatically when DATA.DATA is read.

This bit is not write-synchronized.

Value	Description
0	Smart mode is disabled.
1	Smart mode is enabled.

Related Links

34. SERCOM – Serial Communication Interface

Downloaded from Arrow.com.

37.8.3 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Bit	7	6	5	4	3	2	1	0
	ERROR					DRDY	AMATCH	PREC
Access	R/W					R/W	R/W	R/W
Reset	0					0	0	0

Bit 7 – ERROR Error Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 2 – DRDY Data Ready Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Data Ready bit, which disables the Data Ready interrupt.

Value	Description
0	The Data Ready interrupt is disabled.
1	The Data Ready interrupt is enabled.

Bit 1 – AMATCH Address Match Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Address Match Interrupt Enable bit, which disables the Address Match interrupt.

Value	Description
0	The Address Match interrupt is disabled.
1	The Address Match interrupt is enabled.

Bit 0 – PREC Stop Received Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Stop Received Interrupt Enable bit, which disables the Stop Received interrupt.

Value	Description
0	The Stop Received interrupt is disabled.
1	The Stop Received interrupt is enabled.

Downloaded from Arrow.com.

37.8.4 Interrupt Enable Set

Name: INTENSET
Offset: 0x16
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Bit	7	6	5	4	3	2	1	0
	ERROR					DRDY	AMATCH	PREC
Access	R/W					R/W	R/W	R/W
Reset	0					0	0	0

Bit 7 – ERROR Error Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 2 – DRDY Data Ready Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Data Ready bit, which enables the Data Ready interrupt.

Value	Description
0	The Data Ready interrupt is disabled.
1	The Data Ready interrupt is enabled.

Bit 1 – AMATCH Address Match Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Address Match Interrupt Enable bit, which enables the Address Match interrupt.

Value	Description
0	The Address Match interrupt is disabled.
1	The Address Match interrupt is enabled.

Bit 0 – PREC Stop Received Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Stop Received Interrupt Enable bit, which enables the Stop Received interrupt.

Value	Description
0	The Stop Received interrupt is disabled.
1	The Stop Received interrupt is enabled.

Downloaded from Arrow.com.

37.8.5 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	ERROR					DRDY	AMATCH	PREC
Access	R/W					R/W	R/W	R/W
Reset	0					0	0	0

Bit 7 – ERROR Error
This bit is set when any error is detected. Errors that will set this flag have corresponding status flags in the STATUS register. The corresponding bits in STATUS are SEXTTOUT, LOWTOUT, COLL, and BUSERR.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 2 – DRDY Data Ready
This flag is set when a I^2C slave byte transmission is successfully completed.

The flag is cleared by hardware when either:

- Writing to the DATA register.
- Reading the DATA register with smart mode enabled.
- Writing a valid command to the CMD register.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Data Ready interrupt flag.

Bit 1 – AMATCH Address Match
This flag is set when the I^2C slave address match logic detects that a valid address has been received.

The flag is cleared by hardware when CTRL.CMD is written.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Address Match interrupt flag. When cleared, an ACK/NACK will be sent according to CTRLB.ACKACT.

Bit 0 – PREC Stop Received
This flag is set when a stop condition is detected for a transaction being processed. A stop condition detected between a bus master and another slave will not set this flag, unless the PMBus Group Command is enabled in the Control B register (CTRLB.GCMD=1).

This flag is cleared by hardware after a command is issued on the next address match.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Stop Received interrupt flag.

Downloaded from Arrow.com.

37.8.6 Status

Name: STATUS
Offset: 0x1A
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
					LENERR	HS	SEXTTOUT	
Access					R/W	R/W	R/W	
Reset					0	0	0	

Bit	7	6	5	4	3	2	1	0
	CLKHOLD	LOWTOUT		SR	DIR	RXNACK	COLL	BUSERR
Access	R	R/W		R	R	R	R/W	R/W
Reset	0	0		0	0	0	0	0

Bit 11 – LENERR Transaction Length Error
This bit is set when the length counter is enabled (LENGTH.LENEN) and a STOP or repeated START is received before or after the length in LENGTH.LEN is reached.
This bit is cleared automatically when responding to a new start condition with ACK or NACK (CTRLB.CMD=0x3) or when INTFLAG.AMATCH is cleared.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the status.

Bit 10 – HS High-speed
This bit is set if the slave detects a START followed by a Master Code transmission.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the status. However, this flag is automatically cleared when a STOP is received.

Bit 9 – SEXTTOUT Slave SCL Low Extend Time-Out
This bit is set if a slave SCL low extend time-out occurs.

This bit is cleared automatically if responding to a new start condition with ACK or NACK (write 3 to CTRLB.CMD) or when INTFLAG.AMATCH is cleared.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the status.

Value	Description
0	No SCL low extend time-out has occurred.
1	SCL low extend time-out has occurred.

Downloaded from Arrow.com.

Bit 7 – CLKHOLD Clock Hold
The slave Clock Hold bit (STATUS.CLKHOLD) is set when the slave is holding the SCL line low, stretching the I2C clock. Software should consider this bit a read-only status flag that is set when INTFLAG.DRDY or INTFLAG.AMATCH is set.

This bit is automatically cleared when the corresponding interrupt is also cleared.

Bit 6 – LOWTOUT SCL Low Time-out
This bit is set if an SCL low time-out occurs.

This bit is cleared automatically if responding to a new start condition with ACK or NACK (write 3 to CTRLB.CMD) or when INTFLAG.AMATCH is cleared.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the status.

Value	Description
0	No SCL low time-out has occurred.
1	SCL low time-out has occurred.

Bit 4 – SR Repeated Start
When INTFLAG.AMATCH is raised due to an address match, SR indicates a repeated start or start condition.

This flag is only valid while the INTFLAG.AMATCH flag is one.

Value	Description
0	Start condition on last address match
1	Repeated start condition on last address match

Bit 3 – DIR Read / Write Direction
The Read/Write Direction (STATUS.DIR) bit stores the direction of the last address packet received from a master.

Value	Description
0	Master write operation is in progress.
1	Master read operation is in progress.

Bit 2 – RXNACK Received Not Acknowledge
This bit indicates whether the last data packet sent was acknowledged or not.

Value	Description
0	Master responded with ACK.
1	Master responded with NACK.

Bit 1 – COLL Transmit Collision
If set, the I2C slave was not able to transmit a high data or NACK bit, the I2C slave will immediately release the SDA and SCL lines and wait for the next packet addressed to it.

This flag is intended for the SMBus address resolution protocol (ARP). A detected collision in non-ARP situations indicates that there has been a protocol violation, and should be treated as a bus error.

Downloaded from Arrow.com.

Note that this status will not trigger any interrupt, and should be checked by software to verify that the data were sent correctly. This bit is cleared automatically if responding to an address match with an ACK or a NACK (writing 0x3 to CTRLB.CMD), or INTFLAG.AMATCH is cleared.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the status.

Value	Description
0	No collision detected on last data byte sent.
1	Collision detected on last data byte sent.

Bit 0 – BUSERR Bus Error
The Bus Error bit (STATUS.BUSERR) indicates that an illegal bus condition has occurred on the bus, regardless of bus ownership. An illegal bus condition is detected if a protocol violating start, repeated start or stop is detected on the I2C bus lines. A start condition directly followed by a stop condition is one example of a protocol violation. If a time-out occurs during a frame, this is also considered a protocol violation, and will set STATUS.BUSERR.

This bit is cleared automatically if responding to an address match with an ACK or a NACK (writing 0x3 to CTRLB.CMD) or INTFLAG.AMATCH is cleared.

Writing a '1' to this bit will clear the status.

Writing a '0' to this bit has no effect.

Value	Description
0	No bus error detected.
1	Bus error detected.

Downloaded from Arrow.com.

37.8.7 Synchronization Busy

Name: SYNCBUSY
Offset: 0x1C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							ENABLE	SWRST
Access							R	R
Reset							0	0

Bit 1 – ENABLE SERCOM Enable Synchronization Busy
Enabling and disabling the SERCOM (CTRLA.ENABLE) requires synchronization. When written, the SYNCBUSY.ENABLE bit will be set until synchronization is complete.

Value	Description
0	Enable synchronization is not busy.
1	Enable synchronization is busy.

Bit 0 – SWRST Software Reset Synchronization Busy
Resetting the SERCOM (CTRLA.SWRST) requires synchronization. When written, the SYNCBUSY.SWRST bit will be set until synchronization is complete.

Value	Description
0	SWRST synchronization is not busy.
1	SWRST synchronization is busy.

Downloaded from Arrow.com.

37.8.8 Address

Name: ADDR
Offset: 0x24
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
						ADDRMASK[9:7]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDRMASK[6:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	

Bit	15	14	13	12	11	10	9	8
	TENBITEN					ADDR[9:7]		
Access	R/W					R/W	R/W	R/W
Reset	0					0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[6:0]							GENCEN
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 26:17 – ADDRMASK[9:0] Address Mask
These bits act as a second address match register, an address mask register or the lower limit of an address range, depending on the CTRLB.AMODE setting.

Bit 15 – TENBITEN Ten Bit Addressing Enable

Value	Description
0	10-bit address recognition disabled.
1	10-bit address recognition enabled.

Bits 10:1 – ADDR[9:0] Address
These bits contain the I^2C slave address used by the slave address match logic to determine if a master has addressed the slave.

When using 7-bit addressing, the slave address is represented by ADDR[6:0].

When using 10-bit addressing (ADDR.TENBITEN=1), the slave address is represented by ADDR[9:0]

When the address match logic detects a match, INTFLAG.AMATCH is set and STATUS.DIR is updated to indicate whether it is a read or a write transaction.

Bit 0 – GENCEN General Call Address Enable
A general call address is an address consisting of all-zeroes, including the direction bit (master write).

Downloaded from Arrow.com.

Value	Description
0	General call address recognition disabled.
1	General call address recognition enabled.

Downloaded from Arrow.com.

37.8.9 Data

Name: DATA
Offset: 0x28
Reset: 0x0000
Property: Read/Write

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – DATA[7:0] Data
The slave data register I/O location (DATA.DATA) provides access to the master transmit and receive data buffers. Reading valid data or writing data to be transmitted can be successfully done only when SCL is held low by the slave (STATUS.CLKHOLD is set). An exception occurs when reading the last data byte after the stop condition has been received.

Accessing DATA.DATA auto-triggers I^2C bus operations. The operation performed depends on the state of CTRLB.ACKACT, CTRLB.SMEN and the type of access (read/write).

Downloaded from Arrow.com.

37.9 Register Summary - I2C Master

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
		15:8								
		23:16	SEXTTOEN	MEXTTOEN	SDAHOLD[1:0]					PINOUT
		31:24		LOWTOUT	INACTOUT[1:0]		SCLSM		SPEED[1:0]	
0x04	CTRLB	7:0								
		15:8							QCEN	SMEN
		23:16						ACKACT	CMD[1:0]	
		31:24								
0x08 ... 0x0B	Reserved									
0x0C	BAUD	7:0	BAUD[7:0]							
		15:8	BAUDLOW[7:0]							
		23:16	HSBAUD[7:0]							
		31:24	HSBAUDLOW[7:0]							
0x10 ... 0x13	Reserved									
0x14	INTENCLR	7:0	ERROR						SB	MB
0x15	Reserved									
0x16	INTENSET	7:0	ERROR						SB	MB
0x17	Reserved									
0x18	INTFLAG	7:0	ERROR						SB	MB
0x19	Reserved									
0x1A	STATUS	7:0	CLKHOLD	LOWTOUT	BUSSTATE[1:0]			RXNACK	ARBLOST	BUSERR
		15:8						LENERR	SEXTTOUT	MEXTTOUT
0x1C	SYNCBUSY	7:0						SYSOP	ENABLE	SWRST
		15:8								
		23:16								
		31:24								
0x20 ... 0x23	Reserved									
0x24	ADDR	7:0	ADDR[7:0]							
		15:8	TENBITEN	HS	LENEN			ADDR[10:8]		
		23:16	LEN[7:0]							
		31:24								
0x28	DATA	7:0	DATA[7:0]							
		15:8								
0x2A ... 0x2F	Reserved									
0x30	DBGCTRL	7:0								DBGSTOP

Downloaded from Arrow.com.

37.10 Register Description - I²C Master

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to 37.5.8 Register Access Protection.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to 37.6.6 Synchronization.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

Downloaded from Arrow.com.

37.10.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
		LOWTOUT	INACTOUT[1:0]		SCLSM		SPEED[1:0]	
Access		R/W	R/W	R/W	R/W		R/W	R/W
Reset		0	0	0	0		0	0

Bit	23	22	21	20	19	18	17	16
	SEXTTOEN	MEXTTOEN	SDAHOLD[1:0]					PINOUT
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit 30 – LOWTOUT SCL Low Time-Out
This bit enables the SCL low time-out. If SCL is held low for 25ms-35ms, the master will release its clock hold, if enabled, and complete the current transaction. A stop condition will automatically be transmitted.

INTFLAG.SB or INTFLAG.MB will be set as normal, but the clock hold will be released. The STATUS.LOWTOUT and STATUS.BUSERR status bits will be set.

This bit is not synchronized.

Value	Description
0	Time-out disabled.
1	Time-out enabled.

Bits 29:28 – INACTOUT[1:0] Inactive Time-Out
If the inactive bus time-out is enabled and the bus is inactive for longer than the time-out setting, the bus state logic will be set to idle. An inactive bus arise when either an I^2C master or slave is holding the SCL low.

Enabling this option is necessary for SMBus compatibility, but can also be used in a non-SMBus set-up.

Calculated time-out periods are based on a 100kHz baud rate.

These bits are not synchronized.

Downloaded from Arrow.com.

Value	Name	Description
0x0	DIS	Disabled
0x1	55US	5-6 SCL cycle time-out (50-60µs)
0x2	105US	10-11 SCL cycle time-out (100-110µs)
0x3	205US	20-21 SCL cycle time-out (200-210µs)

Bit 27 – SCLSM SCL Clock Stretch Mode
This bit controls when SCL will be stretched for software interaction.

This bit is not synchronized.

Value	Description
0	SCL stretch according to Figure 37-5.
1	SCL stretch only after ACK bit, Figure 37-6.

Bits 25:24 – SPEED[1:0] Transfer Speed
These bits define bus speed.

These bits are not synchronized.

Value	Description
0x0	Standard-mode (Sm) up to 100 kHz and Fast-mode (Fm) up to 400 kHz
0x1	Fast-mode Plus (Fm+) up to 1 MHz
0x2	High-speed mode (Hs-mode) up to 3.4 MHz
0x3	Reserved

Bit 23 – SEXTTOEN Slave SCL Low Extend Time-Out
This bit enables the slave SCL low extend time-out. If SCL is cumulatively held low for greater than 25ms from the initial START to a STOP, the master will release its clock hold if enabled, and complete the current transaction. A STOP will automatically be transmitted.

SB or MB will be set as normal, but CLKHOLD will be release. The MEXTTOUT and BUSERR status bits will be set.

This bit is not synchronized.

Value	Description
0	Time-out disabled
1	Time-out enabled

Bit 22 – MEXTTOEN Master SCL Low Extend Time-Out
This bit enables the master SCL low extend time-out. If SCL is cumulatively held low for greater than 10ms from START-to-ACK, ACK-to-ACK, or ACK-to-STOP the master will release its clock hold if enabled, and complete the current transaction. A STOP will automatically be transmitted.

SB or MB will be set as normal, but CLKHOLD will be released. The MEXTTOUT and BUSERR status bits will be set.

This bit is not synchronized.

Value	Description
0	Time-out disabled
1	Time-out enabled

Downloaded from Arrow.com.

Bits 21:20 – SDAHOLD[1:0] SDA Hold Time
These bits define the SDA hold time with respect to the negative edge of SCL.

These bits are not synchronized.

Value	Name	Description
0x0	DIS	Disabled
0x1	75NS	50-100ns hold time
0x2	450NS	300-600ns hold time
0x3	600NS	400-800ns hold time

Bit 16 – PINOUT Pin Usage
This bit set the pin usage to either two- or four-wire operation:

This bit is not synchronized.

Value	Description
0	4-wire operation disabled.
1	4-wire operation enabled.

Bit 7 – RUNSTDBY Run in Standby
This bit defines the functionality in standby sleep mode.

This bit is not synchronized.

Value	Description
0	GCLK_SERCOMx_CORE is disabled and the I^2C master will not operate in standby sleep mode.
1	GCLK_SERCOMx_CORE is enabled in all sleep modes.

Bits 4:2 – MODE[2:0] Operating Mode
These bits must be written to 0x5 to select the I^2C master serial communication interface of the SERCOM.

These bits are not synchronized.

Bit 1 – ENABLE Enable
Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRL.ENABLE will read back immediately and the Synchronization Enable Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

This bit is not enable-protected.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled.

Bit 0 – SWRST Software Reset
Writing '0' to this bit has no effect.

Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state, and the SERCOM will be disabled.

Downloaded from Arrow.com.

Writing '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded. Any register write access during the ongoing reset will result in an APB error. Reading any register will return the reset value of the register.

Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

This bit is not enable-protected.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

Downloaded from Arrow.com.

37.10.2 Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						ACKACT	CMD[1:0]	
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
							QCEN	SMEN
Access							R	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 18 – ACKACT Acknowledge Action
This bit defines the I^2C master's acknowledge behavior after a data byte is received from the I^2C slave. The acknowledge action is executed when a command is written to CTRLB.CMD, or if smart mode is enabled (CTRLB.SMEN is written to one), when DATA.DATA is read.

This bit is not enable-protected.

This bit is not write-synchronized.

Value	Description
0	Send ACK.
1	Send NACK.

Bits 17:16 – CMD[1:0] Command
Writing these bits triggers a master operation as described below. The CMD bits are strobe bits, and always read as zero. The acknowledge action is only valid in master read mode. In master write mode, a command will only result in a repeated start or stop condition. The CTRLB.ACKACT bit and the CMD bits can be written at the same time, and then the acknowledge action will be updated before the command is triggered.

Commands can only be issued when either the Slave on Bus interrupt flag (INTFLAG.SB) or Master on Bus interrupt flag (INTFLAG.MB) is '1'.

Downloaded from Arrow.com.

If CMD 0x1 is issued, a repeated start will be issued followed by the transmission of the current address in ADDR.ADDR. If another address is desired, ADDR.ADDR must be written instead of the CMD bits. This will trigger a repeated start followed by transmission of the new address.

Issuing a command will set the System Operation bit in the Synchronization Busy register (SYNCBUSY.SYSOP).

Table 37-4. Command Description

CMD[1:0]	Direction	Action
0x0	X	(No action)
0x1	X	Execute acknowledge action succeeded by repeated Start
0x2	0 (Write)	No operation
	1 (Read)	Execute acknowledge action succeeded by a byte read operation
0x3	X	Execute acknowledge action succeeded by issuing a stop condition

These bits are not enable-protected.

Bit 9 – QCEN Quick Command Enable
This bit is not write-synchronized.

Value	Description
0	Quick Command is disabled.
1	Quick Command is enabled.

Bit 8 – SMEN Smart Mode Enable
When smart mode is enabled, acknowledge action is sent when DATA.DATA is read.

This bit is not write-synchronized.

Value	Description
0	Smart mode is disabled.
1	Smart mode is enabled.

Downloaded from Arrow.com.

37.10.3 Baud Rate

Name: BAUD
Offset: 0x0C
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	HSBAUDLOW[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	HSBAUD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	BAUDLOW[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BAUD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – HSBAUDLOW[7:0] High Speed Master Baud Rate Low
HSBAUDLOW non-zero: HSBAUDLOW indicates the SCL low time in High-speed mode according to

$$\mathrm{HSBAUDLOW} = f_{\mathrm{GCLK}} \cdot T_{\mathrm{LOW}} - 1$$

HSBAUDLOW equal to zero: The HSBAUD register is used to time T_{LOW}, T_{HIGH}, $T_{SU;STO}$, $T_{HD;STA}$ and $T_{SU;STA}$. T_{BUF} is timed by the BAUD register.

Bits 23:16 – HSBAUD[7:0] High Speed Master Baud Rate
This bit field indicates the SCL high time in High-speed mode according to the following formula. When HSBAUDLOW is zero, T_{LOW}, T_{HIGH}, $T_{SU;STO}$, $T_{HD;STA}$ and $T_{SU;STA}$ are derived using this formula. T_{BUF} is timed by the BAUD register.

$$\mathrm{HSBAUD} = f_{\mathrm{GCLK}} \cdot T_{\mathrm{HIGH}} - 1$$

Bits 15:8 – BAUDLOW[7:0] Master Baud Rate Low
If this bit field is non-zero, the SCL low time will be described by the value written.

For more information on how to calculate the frequency, see SERCOM 34.6.2.3 Clock Generation – Baud-Rate Generator.

Downloaded from Arrow.com.

Bits 7:0 – BAUD[7:0] Master Baud Rate
This bit field is used to derive the SCL high time if BAUD.BAUDLOW is non-zero. If BAUD.BAUDLOW is zero, BAUD will be used to generate both high and low periods of the SCL.

For more information on how to calculate the frequency, see SERCOM 34.6.2.3 Clock Generation – Baud-Rate Generator.

Downloaded from Arrow.com.

37.10.4 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Bit	7	6	5	4	3	2	1	0
	ERROR						SB	MB
Access	R/W						R/W	R/W
Reset	0						0	0

Bit 7 – ERROR Error Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 1 – SB Slave on Bus Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Slave on Bus Interrupt Enable bit, which disables the Slave on Bus interrupt.

Value	Description
0	The Slave on Bus interrupt is disabled.
1	The Slave on Bus interrupt is enabled.

Bit 0 – MB Master on Bus Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Master on Bus Interrupt Enable bit, which disables the Master on Bus interrupt.

Value	Description
0	The Master on Bus interrupt is disabled.
1	The Master on Bus interrupt is enabled.

Downloaded from Arrow.com.

37.10.5 Interrupt Enable Set

Name: INTENSET
Offset: 0x16
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Bit	7	6	5	4	3	2	1	0
	ERROR						SB	MB
Access	R/W						R/W	R/W
Reset	0						0	0

Bit 7 – ERROR Error Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 1 – SB Slave on Bus Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Slave on Bus Interrupt Enable bit, which enables the Slave on Bus interrupt.

Value	Description
0	The Slave on Bus interrupt is disabled.
1	The Slave on Bus interrupt is enabled.

Bit 0 – MB Master on Bus Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Master on Bus Interrupt Enable bit, which enables the Master on Bus interrupt.

Value	Description
0	The Master on Bus interrupt is disabled.
1	The Master on Bus interrupt is enabled.

Downloaded from Arrow.com.

37.10.6 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	ERROR						SB	MB
Access	R/W						R/W	R/W
Reset	0						0	0

Bit 7 – ERROR Error
This flag is cleared by writing '1' to it.

This bit is set when any error is detected. Errors that will set this flag have corresponding status bits in the STATUS register. These status bits are LENERR, SEXTTOUT, MEXTTOUT, LOWTOUT, ARBLOST, and BUSERR.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 1 – SB Slave on Bus
The Slave on Bus flag (SB) is set when a byte is successfully received in master read mode, i.e., no arbitration lost or bus error occurred during the operation. When this flag is set, the master forces the SCL line low, stretching the I^2C clock period. The SCL line will be released and SB will be cleared on one of the following actions:

- Writing to ADDR.ADDR
- Writing to DATA.DATA
- Reading DATA.DATA when smart mode is enabled (CTRLB.SMEN)
- Writing a valid command to CTRLB.CMD

Writing '1' to this bit location will clear the SB flag. The transaction will not continue or be terminated until one of the above actions is performed.

Writing '0' to this bit has no effect.

Bit 0 – MB Master on Bus
This flag is set when a byte is transmitted in master write mode. The flag is set regardless of the occurrence of a bus error or an arbitration lost condition. MB is also set when arbitration is lost during sending of NACK in master read mode, or when issuing a start condition if the bus state is unknown. When this flag is set and arbitration is not lost, the master forces the SCL line low, stretching the I^2C clock period. The SCL line will be released and MB will be cleared on one of the following actions:

- Writing to ADDR.ADDR
- Writing to DATA.DATA
- Reading DATA.DATA when smart mode is enabled (CTRLB.SMEN)
- Writing a valid command to CTRLB.CMD

Downloaded from Arrow.com.

Writing '1' to this bit location will clear the MB flag. The transaction will not continue or be terminated until one of the above actions is performed.

Writing '0' to this bit has no effect.

Downloaded from Arrow.com.

37.10.7 Status

Name: STATUS
Offset: 0x1A
Reset: 0x0000
Property: Write-Synchronized

Bit	15	14	13	12	11	10	9	8
						LENERR	SEXTTOUT	MEXTTOUT
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	CLKHOLD	LOWTOUT	BUSSTATE[1:0]			RXNACK	ARBLOST	BUSERR
Access	R	R/W	R/W	R/W		R	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit 10 – LENERR Transaction Length Error
This bit is set when automatic length is used for a DMA transaction and the slave sends a NACK before ADDR.LEN bytes have been written by the master.

Writing '1' to this bit location will clear STATUS.LENERR. This flag is automatically cleared when writing to the ADDR register.

Writing '0' to this bit has no effect.

This bit is not write-synchronized.

Bit 9 – SEXTTOUT Slave SCL Low Extend Time-Out
This bit is set if a slave SCL low extend time-out occurs.

This bit is automatically cleared when writing to the ADDR register.

Writing '1' to this bit location will clear SEXTTOUT. Normal use of the I^2C interface does not require the SEXTTOUT flag to be cleared by this method.

Writing '0' to this bit has no effect.

This bit is not write-synchronized.

Bit 8 – MEXTTOUT Master SCL Low Extend Time-Out
This bit is set if a master SCL low time-out occurs.

Writing '1' to this bit location will clear STATUS.MEXTTOUT. This flag is automatically cleared when writing to the ADDR register.

Writing '0' to this bit has no effect.

This bit is not write-synchronized.

Bit 7 – CLKHOLD Clock Hold
This bit is set when the master is holding the SCL line low, stretching the I^2C clock. Software should consider this bit when INTFLAG.SB or INTFLAG.MB is set.

Downloaded from Arrow.com.

This bit is cleared when the corresponding interrupt flag is cleared and the next operation is given.

Writing '0' to this bit has no effect.

Writing '1' to this bit has no effect.

This bit is not write-synchronized.

Bit 6 – LOWTOUT SCL Low Time-Out
This bit is set if an SCL low time-out occurs.

Writing '1' to this bit location will clear this bit. This flag is automatically cleared when writing to the ADDR register.

Writing '0' to this bit has no effect.

This bit is not write-synchronized.

Bits 5:4 – BUSSTATE[1:0] Bus State
These bits indicate the current I^2C bus state.

When in UNKNOWN state, writing 0x1 to BUSSTATE forces the bus state into the IDLE state. The bus state cannot be forced into any other state.

Writing BUSSTATE to idle will set SYNCBUSY.SYSOP.

Value	Name	Description
0x0	UNKNOWN	The bus state is unknown to the I^2C master and will wait for a stop condition to be detected or wait to be forced into an idle state by software
0x1	IDLE	The bus state is waiting for a transaction to be initialized
0x2	OWNER	The I^2C master is the current owner of the bus
0x3	BUSY	Some other I^2C master owns the bus

Bit 2 – RXNACK Received Not Acknowledge
This bit indicates whether the last address or data packet sent was acknowledged or not.

Writing '0' to this bit has no effect.

Writing '1' to this bit has no effect.

This bit is not write-synchronized.

Value	Description
0	Slave responded with ACK.
1	Slave responded with NACK.

Bit 1 – ARBLOST Arbitration Lost
This bit is set if arbitration is lost while transmitting a high data bit or a NACK bit, or while issuing a start or repeated start condition on the bus. The Master on Bus interrupt flag (INTFLAG.MB) will be set when STATUS.ARBLOST is set.

Writing the ADDR.ADDR register will automatically clear STATUS.ARBLOST.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

This bit is not write-synchronized.

Downloaded from Arrow.com.

Bit 0 – BUSERR Bus Error

This bit indicates that an illegal bus condition has occurred on the bus, regardless of bus ownership. An illegal bus condition is detected if a protocol violating start, repeated start or stop is detected on the I^2C bus lines. A start condition directly followed by a stop condition is one example of a protocol violation. If a time-out occurs during a frame, this is also considered a protocol violation, and will set BUSERR.

If the I^2C master is the bus owner at the time a bus error occurs, STATUS.ARBLOST and INTFLAG.MB will be set in addition to BUSERR.

Writing the ADDR.ADDR register will automatically clear the BUSERR flag.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

This bit is not write-synchronized.

Downloaded from Arrow.com.

37.10.8 Synchronization Busy

Name: SYNCBUSY
Offset: 0x1C
Reset: 0x00000000

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						SYSOP	ENABLE	SWRST
Access						R	R	R
Reset						0	0	0

Bit 2 – SYSOP System Operation Synchronization Busy
Writing CTRLB.CMD, STATUS.BUSSTATE, ADDR, or DATA when the SERCOM is enabled requires synchronization. When written, the SYNCBUSY.SYSOP bit will be set until synchronization is complete.

Value	Description
0	System operation synchronization is not busy.
1	System operation synchronization is busy.

Bit 1 – ENABLE SERCOM Enable Synchronization Busy
Enabling and disabling the SERCOM (CTRLA.ENABLE) requires synchronization. When written, the SYNCBUSY.ENABLE bit will be set until synchronization is complete.

Value	Description
0	Enable synchronization is not busy.
1	Enable synchronization is busy.

Bit 0 – SWRST Software Reset Synchronization Busy
Resetting the SERCOM (CTRLA.SWRST) requires synchronization. When written, the SYNCBUSY.SWRST bit will be set until synchronization is complete.

Value	Description
0	SWRST synchronization is not busy.
1	SWRST synchronization is busy.

Downloaded from Arrow.com.

37.10.9 Address

Name: ADDR
Offset: 0x24
Reset: 0x0000
Property: Write-Synchronized

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	LEN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TENBITEN	HS	LENEN			ADDR[10:8]		
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – LEN[7:0] Transaction Length
These bits define the transaction length of a DMA transaction from 0 to 255 bytes. The Transfer Length Enable (LENEN) bit must be written to '1' in order to use DMA.

Bit 15 – TENBITEN Ten Bit Addressing Enable
This bit enables 10-bit addressing. This bit can be written simultaneously with ADDR to indicate a 10-bit or 7-bit address transmission.

Value	Description
0	10-bit addressing disabled.
1	10-bit addressing enabled.

Bit 14 – HS High Speed
This bit enables High-speed mode for the current transfer from repeated START to STOP. This bit can be written simultaneously with ADDR for a high speed transfer.

Value	Description
0	High-speed transfer disabled.
1	High-speed transfer enabled.

Bit 13 – LENEN Transfer Length Enable

Downloaded from Arrow.com.

Value	Description
0	Automatic transfer length disabled.
1	Automatic transfer length enabled.

Bits 10:0 – ADDR[10:0] Address
When ADDR is written, the consecutive operation will depend on the bus state:

UNKNOWN: INTFLAG.MB and STATUS.BUSERR are set, and the operation is terminated.

BUSY: The I^2C master will await further operation until the bus becomes IDLE.

IDLE: The I^2C master will issue a start condition followed by the address written in ADDR. If the address is acknowledged, SCL is forced and held low, and STATUS.CLKHOLD and INTFLAG.MB are set.

OWNER: A repeated start sequence will be performed. If the previous transaction was a read, the acknowledge action is sent before the repeated start bus condition is issued on the bus. Writing ADDR to issue a repeated start is performed while INTFLAG.MB or INTFLAG.SB is set.

STATUS.BUSERR, STATUS.ARBLOST, INTFLAG.MB and INTFLAG.SB will be cleared when ADDR is written.

The ADDR register can be read at any time without interfering with ongoing bus activity, as a read access does not trigger the master logic to perform any bus protocol related operations.

The I^2C master control logic uses bit 0 of ADDR as the bus protocol's read/write flag (R/W); 0 for write and 1 for read.

Downloaded from Arrow.com.

37.10.10 Data

Name: DATA
Offset: 0x28
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – DATA[7:0] Data
The master data register I/O location (DATA) provides access to the master transmit and receive data buffers. Reading valid data or writing data to be transmitted can be successfully done only when SCL is held low by the master (STATUS.CLKHOLD is set). An exception is reading the last data byte after the stop condition has been sent.

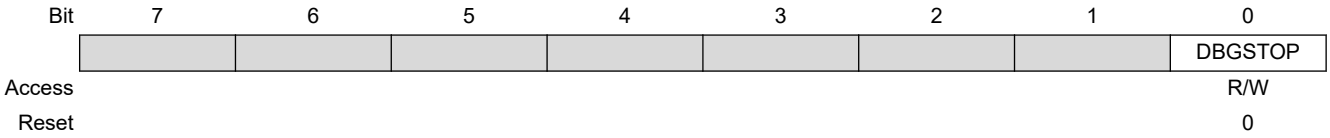
Accessing DATA.DATA auto-triggers I^2C bus operations. The operation performed depends on the state of CTRLB.ACKACT, CTRLB.SMEN and the type of access (read/write).

Writing or reading DATA.DATA when not in Smart mode does not require synchronization.

Downloaded from Arrow.com.

37.10.11 Debug Control

Name: DBGCTRL
Offset: 0x30
Reset: 0x00
Property: PAC Write-Protection



Bit 0 – DBGSTOP Debug Stop Mode
This bit controls functionality when the CPU is halted by an external debugger.

Value	Description
0	The baud-rate generator continues normal operation when the CPU is halted by an external debugger.
1	The baud-rate generator is halted when the CPU is halted by an external debugger.

Downloaded from Arrow.com.

38. TC – Timer/Counter

38.1 Overview

There are up to three TC peripheral instances.

Each TC consists of a counter, a prescaler, compare/capture channels and control logic. The counter can be set to count events, or clock pulses. The counter, together with the compare/capture channels, can be configured to timestamp input events or IO pin edges, allowing for capturing of frequency and/or pulse width.

A TC can also perform waveform generation, such as frequency generation and pulse-width modulation.

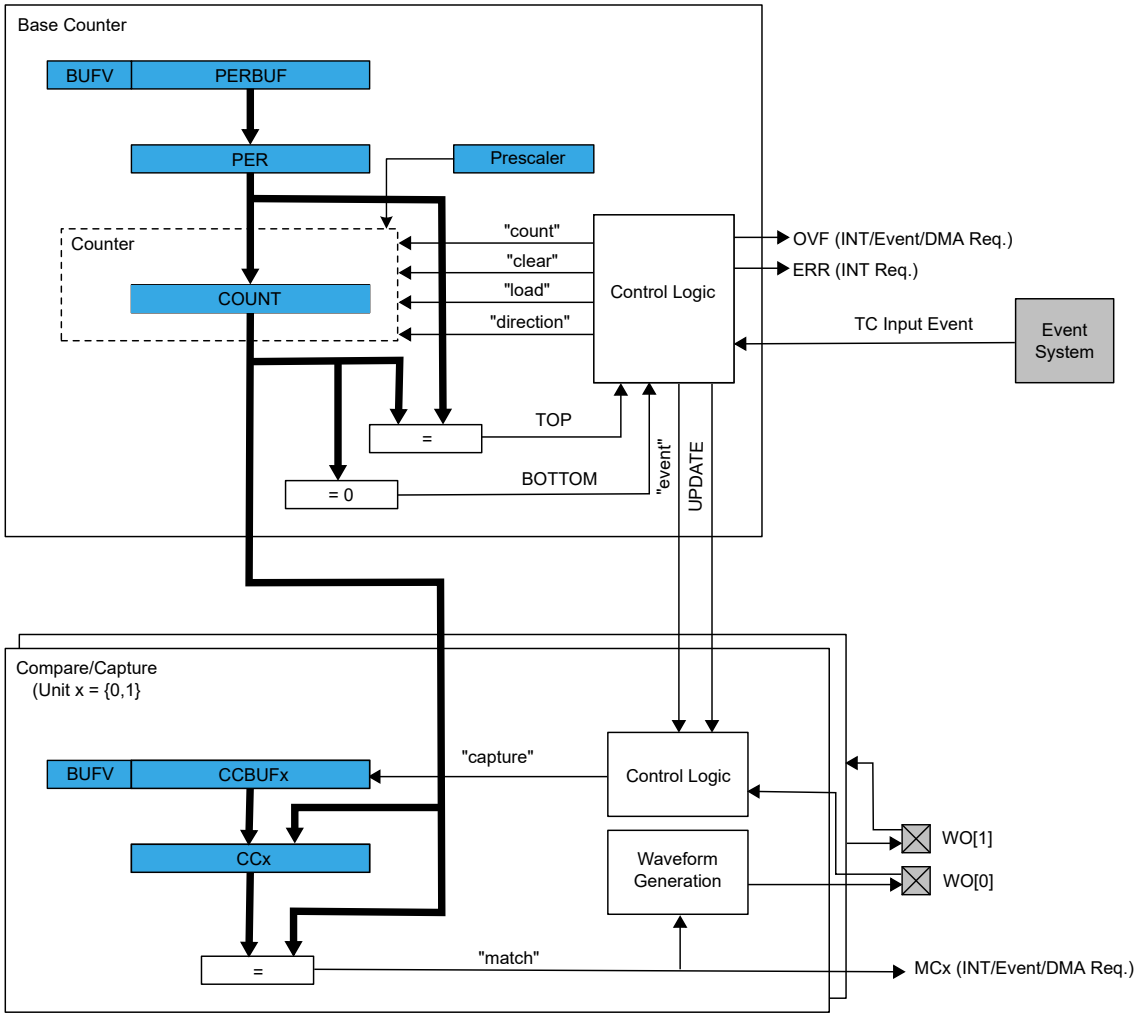
38.2 Features

- Selectable configuration
  - 8-, 16- or 32-bit TC operation, with compare/capture channels
- 2 compare/capture channels (CC) with:
  - Double buffered timer period setting (in 8-bit mode only)
  - Double buffered compare channel
- Waveform generation
  - Frequency generation
  - Single-slope pulse-width modulation
- Input capture
  - Event / IO pin edge capture
  - Frequency capture
  - Pulse-width capture
  - Time-stamp capture
- One input event
- Interrupts/output events on:
  - Counter overflow/underflow
  - Compare match or capture
- Internal prescaler
- DMA support

Downloaded from Arrow.com.

38.3 Block Diagram

Figure 38-1. Timer/Counter Block Diagram



38.4 Signal Description

Table 38-1. Signal Description for TC.

Signal Name	Type	Description
WO[1:0]	Digital output	Waveform output
	Digital input	Capture input

Refer to *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

Downloaded from Arrow.com.

38.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

38.5.1 I/O Lines

In order to use the I/O lines of this peripheral, the I/O pins must be configured using the I/O Pin Controller (PORT).

Related Links

32. PORT - I/O Pin Controller

38.5.2 Power Management

This peripheral can continue to operate in any sleep mode where its source clock is running. The interrupts can wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes.

Related Links

22. PM – Power Manager

38.5.3 Clocks

The TC bus clocks (CLK_TCx_APB) can be enabled and disabled in the Main Clock Module. The default state of CLK_TCx_APB can be found in the *Peripheral Clock Masking*.

The generic clocks (GCLK_TCx) are asynchronous to the user interface clock (CLK_TCx_APB). Due to this asynchronicity, accessing certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

Note: Two instances of the TC may share a peripheral clock channel. In this case, they cannot be set to different clock frequencies. Refer to the peripheral clock channel mapping of the Generic Clock Controller (GCLK.PCHTRLm) to identify shared peripheral clocks.

Related Links

18.8.4 PCHCTRLm

19.6.2.6 Peripheral Clock Masking

38.5.4 DMA

The DMA request lines are connected to the DMA Controller (DMAC). In order to use DMA requests with this peripheral the DMAC must be configured first. Refer to *DMAC – Direct Memory Access Controller* for details.

Related Links

28. DMAC – Direct Memory Access Controller

38.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. Refer to *Nested Vector Interrupt Controller* for details.

38.5.6 Events

The events of this peripheral are connected to the Event System.

Related Links

Downloaded from Arrow.com.

33. EVSYS – Event System

38.5.7 Debug Operation

When the CPU is halted in debug mode, this peripheral will halt normal operation. This peripheral can be forced to continue operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

Related Links

38.7.1.11 DBGCTRL

38.5.8 Register Access Protection

Registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC), except for the following:

- Interrupt Flag Status and Clear register (INTFLAG)
- Status register (STATUS)
- Count register (COUNT)
- Period and Period Buffer registers (PER, PERBUF)
- Compare/Capture Value registers and Compare/Capture Value Buffer registers (CCx, CCBUFx)

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

Write-protection does not apply for accesses through an external debugger.

38.5.9 Analog Connections

Not applicable.

38.5.10 SAM L11 TrustZone Specific Register Access Protection

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

38.6 Functional Description

38.6.1 Principle of Operation

The following definitions are used throughout the documentation:

Table 38-2. Timer/Counter Definitions

Name	Description
TOP	The counter reaches TOP when it becomes equal to the highest value in the count sequence. The TOP value can be the same as Period (PER)

Downloaded from Arrow.com.

Name	Description
	or the Compare Channel 0 (CC0) register value depending on the waveform generator mode in 38.6.2.6.1 Waveform Output Operations.
ZERO	The counter is ZERO when it contains all zeroes
MAX	The counter reaches MAX when it contains all ones
UPDATE	The timer/counter signals an update when it reaches ZERO or TOP, depending on the direction settings.
Timer	The timer/counter clock control is handled by an internal source
Counter	The clock control is handled externally (e.g. counting external events)
CC	For compare operations, the CC are referred to as “compare channels” For capture operations, the CC are referred to as “capture channels.”

Each TC instance has up to two compare/capture channels (CC0 and CC1).

The counter in the TC can either count events from the Event System, or clock ticks of the GCLK_TCx clock, which may be divided by the prescaler.

The counter value is passed to the CCx where it can be either compared to user-defined values or captured.

The CCx registers are using buffer registers (CCBUFx) for optimized timing. Each buffer register has a buffer valid (BUFV) flag that indicates when the buffer contains a new value.

The Counter register (COUNT) and the Compare and Capture registers with buffers (CCx and CCBUFx) can be configured as 8-, 16- or 32-bit registers, with according MAX values. Mode settings (CTRLA.MODE) determine the maximum range of the Counter register.

In 8-bit mode, a Period Value (PER) register and its Period Buffer Value (PERBUF) register are also available. The counter range and the operating frequency determine the maximum time resolution achievable with the TC peripheral.

The TC can be set to count up or down. Under normal operation, the counter value is continuously compared to the TOP or ZERO value to determine whether the counter has reached that value. On a comparison match the TC can request DMA transactions, or generate interrupts or events for the Event System.

In compare operation, the counter value is continuously compared to the values in the CCx registers. In case of a match the TC can request DMA transactions, or generate interrupts or events for the Event System. In waveform generator mode, these comparisons are used to set the waveform period or pulse width.

Capture operation can be enabled to perform input signal period and pulse width measurements, or to capture selectable edges from an IO pin or internal event from Event System.

38.6.2 Basic Operation

38.6.2.1 Initialization

The following registers are enable-protected, meaning that they can only be written when the TC is disabled (CTRLA.ENABLE =0):

Downloaded from Arrow.com.

- Control A register (CTRLA), except the Enable (ENABLE) and Software Reset (SWRST) bits
- Drive Control register (DRVCTRL)
- Wave register (WAVE)
- Event Control register (EVCTRL)

Writing to Enable-Protected bits and setting the CTRLA.ENABLE bit can be performed in a single 32-bit access of the CTRLA register. Writing to Enable-Protected bits and clearing the CTRLA.ENABLE bit cannot be performed in a single 32-bit access.

Before enabling the TC, the peripheral must be configured by the following steps:

1. Enable the TC bus clock (CLK_TCx_APB).
2. Select 8-, 16- or 32-bit counter mode via the TC Mode bit group in the Control A register (CTRLA.MODE). The default mode is 16-bit.
3. Select one wave generation operation in the Waveform Generation Operation bit group in the WAVE register (WAVE.WAVEGEN).
4. If desired, the GCLK_TCx clock can be prescaled via the Prescaler bit group in the Control A register (CTRLA.PRESCALER).
   – If the prescaler is used, select a prescaler synchronization operation via the Prescaler and Counter Synchronization bit group in the Control A register (CTRLA.PRESYNC).
5. If desired, select one-shot operation by writing a '1' to the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT).
6. If desired, configure the counting direction 'down' (starting from the TOP value) by writing a '1' to the Counter Direction bit in the Control B register (CTRLBSET.DIR).
7. For capture operation, enable the individual channels to capture in the Capture Channel x Enable bit group in the Control A register (CTRLA.CAPTEN).
8. If desired, enable inversion of the waveform output or IO pin input signal for individual channels via the Invert Enable bit group in the Drive Control register (DRVCTRL.INVEN).

38.6.2.2 Enabling, Disabling, and Resetting

The TC is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The TC is disbled by writing a zero to CTRLA.ENABLE.

The TC is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the TC, except DBGCTRL, will be reset to their initial state. Refer to the CTRLA register for details.

The TC should be disabled before the TC is reset in order to avoid undefined behavior.

38.6.2.3 Prescaler Selection

The GCLK_TCx is fed into the internal prescaler.

The prescaler consists of a counter that counts up to the selected prescaler value, whereupon the output of the prescaler toggles.

If the prescaler value is higher than one, the counter update condition can be optionally executed on the next GCLK_TCx clock pulse or the next prescaled clock pulse. For further details, refer to Prescaler (CTRLA.PRESCALER) and Counter Synchronization (CTRLA.PRESYNC) description.

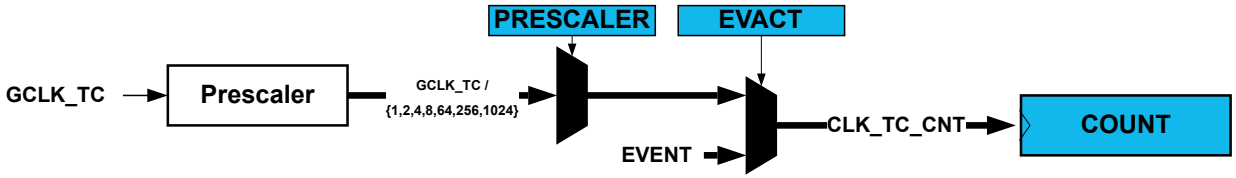
Prescaler outputs from 1 to 1/1024 are available. For a complete list of available prescaler outputs, see the register description for the Prescaler bit group in the Control A register (CTRLA.PRESCALER).

Note: When counting events, the prescaler is bypassed.

The joint stream of prescaler ticks and event action ticks is called CLK_TC_CNT.

Downloaded from Arrow.com.

Figure 38-2. Prescaler



38.6.2.4 Counter Mode

The counter mode is selected by the Mode bit group in the Control A register (CTRLA.MODE). By default, the counter is enabled in the 16-bit counter resolution. Three counter resolutions are available:

- COUNT8: The 8-bit TC has its own Period Value and Period Buffer Value registers (PER and PERBUF).
- COUNT16: 16-bit is the default counter mode. There is no dedicated period register in this mode.
- COUNT32: This mode is achieved by pairing two 16-bit TC peripherals. TCn is paired with TCn+1. TC2 does not support 32-bit resolution.
  When paired, the TC peripherals are configured using the registers of the even-numbered TC. The odd-numbered partner will act as a slave, and the Slave bit in the Status register (STATUS.SLAVE) will be set. The register values of a slave will not reflect the registers of the 32-bit counter. Writing to any of the slave registers will not affect the 32-bit counter. Normal access to the slave COUNT and CCx registers is not allowed.

38.6.2.5 Counter Operations

Depending on the mode of operation, the counter is cleared, reloaded, incremented, or decremented at each TC clock input (CLK_TC_CNT). A counter clear or reload marks the end of the current counter cycle and the start of a new one.

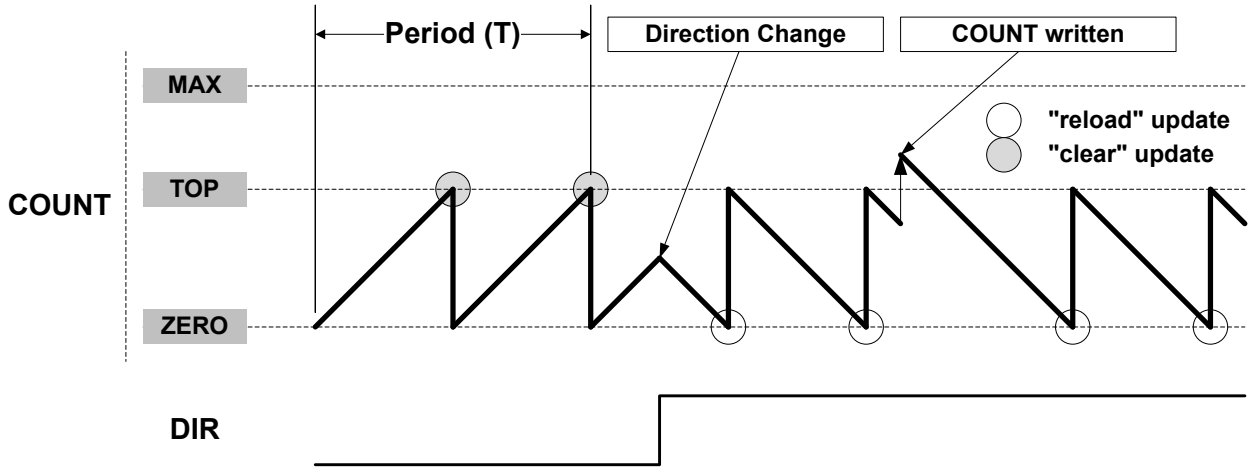
The counting direction is set by the Direction bit in the Control B register (CTRLB.DIR). If this bit is zero the counter is counting up, and counting down if CTRLB.DIR=1. The counter will count up or down for each tick (clock or event) until it reaches TOP or ZERO. When it is counting up and TOP is reached, the counter will be set to zero at the next tick (overflow) and the Overflow Interrupt Flag in the Interrupt Flag Status and Clear register (INTFLAG.OVF) will be set. When it is counting down, the counter is reloaded with the TOP value when ZERO is reached (underflow), and INTFLAG.OVF is set.

INTFLAG.OVF can be used to trigger an interrupt, a DMA request, or an event. An overflow/underflow occurrence (i.e., a compare match with TOP/ZERO) will stop counting if the One-Shot bit in the Control B register is set (CTRLBSET.ONESHOT).

It is possible to change the counter value (by writing directly in the COUNT register) even when the counter is running. When starting the TC, the COUNT value will be either ZERO or TOP (depending on the counting direction set by CTRLBSET.DIR or CTRLBCLR.DIR), unless a different value has been written to it, or the TC has been stopped at a value other than ZERO. The write access has higher priority than count, clear, or reload. The direction of the counter can also be changed when the counter is running. See also the following figure.

Downloaded from Arrow.com.

Figure 38-3. Counter Operation



Due to asynchronous clock domains, the internal counter settings are written when the synchronization is complete. Normal operation must be used when using the counter as timer base for the capture channels.

38.6.2.5.1 Stop Command and Event Action

A Stop command can be issued from software by using Command bits in the Control B Set register (CTRLBSET.CMD = 0x2, STOP). When a Stop is detected while the counter is running, the counter will be loaded with the starting value (ZERO or TOP, depending on direction set by CTRLBSET.DIR or CTRLBCLR.DIR). All waveforms are cleared and the Stop bit in the Status register is set (STATUS.STOP).

38.6.2.5.2 Re-Trigger Command and Event Action

A re-trigger command can be issued from software by writing the Command bits in the Control B Set register (CTRLBSET.CMD = 0x1, RETRIGGER), or from event when a re-trigger event action is configured in the Event Control register (EVCTRL.EVACT = 0x1, RETRIGGER).

When the command is detected during counting operation, the counter will be reloaded or cleared, depending on the counting direction (CTRLBSET.DIR or CTRLBCLR.DIR). When the re-trigger command is detected while the counter is stopped, the counter will resume counting from the current value in the COUNT register.

Note: When a re-trigger event action is configured in the Event Action bits in the Event Control register (EVCTRL.EVACT=0x1, RETRIGGER), enabling the counter will not start the counter. The counter will start on the next incoming event and restart on corresponding following event.

38.6.2.5.3 Count Event Action

The TC can count events. When an event is received, the counter increases or decreases the value, depending on direction settings (CTRLBSET.DIR or CTRLBCLR.DIR). The count event action can be selected by the Event Action bit group in the Event Control register (EVCTRL.EVACT=0x2, COUNT).

Note: If this operation mode is selected, PWM generation is not supported.

38.6.2.5.4 Start Event Action

The TC can start counting operation on an event when previously stopped. In this configuration, the event has no effect if the counter is already counting. When the peripheral is enabled, the counter operation starts when the event is received or when a re-trigger software command is applied.

The Start TC on Event action can be selected by the Event Action bit group in the Event Control register (EVCTRL.EVACT=0x3, START).

Downloaded from Arrow.com.

38.6.2.6 Compare Operations

By default, the Compare/Capture channel is configured for compare operations.

When using the TC and the Compare/Capture Value registers (CCx) for compare operations, the counter value is continuously compared to the values in the CCx registers. This can be used for timer or for waveform operation.

The Channel x Compare Buffer (CCBUFx) registers provide double buffer capability. The double buffering synchronizes the update of the CCx register with the buffer value at the UPDATE condition or a forced update command (CTRLBSET.CMD=UPDATE). For further details, refer to 38.6.2.7 Double Buffering. The synchronization prevents the occurrence of odd-length, non-symmetrical pulses and ensures glitch-free output.

38.6.2.6.1 Waveform Output Operations

The compare channels can be used for waveform generation on output port pins. To make the waveform available on the connected pin, the following requirements must be fulfilled:

1. Choose a waveform generation mode in the Waveform Generation Operation bit in Waveform register (WAVE.WAVEGEN).
2. Optionally invert the waveform output WO[x] by writing the corresponding Output Waveform x Invert Enable bit in the Driver Control register (DRVCTRL.INVENx).
3. Configure the pins with the I/O Pin Controller. Refer to *PORT - I/O Pin Controller* for details.

The counter value is continuously compared with each CCx value. On a comparison match, the Match or Capture Channel x bit in the Interrupt Flag Status and Clear register (INTFLAG.MCx) will be set on the next zero-to-one transition of CLK_TC_CNT (see Normal Frequency Operation). An interrupt/and or event can be generated on comparison match if enabled. The same condition generates a DMA request.

There are four waveform configurations for the Waveform Generation Operation bit group in the Waveform register (WAVE.WAVEGEN). This will influence how the waveform is generated and impose restrictions on the top value. The configurations are:

- Normal frequency (NFRQ)
- Match frequency (MFRQ)
- Normal pulse-width modulation (NPWM)
- Match pulse-width modulation (MPWM)

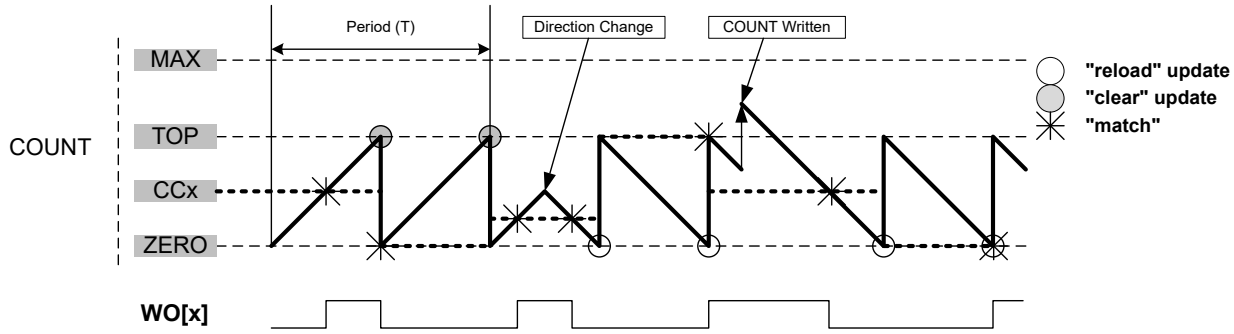
When using NPWM or NFRQ configuration, the TOP will be determined by the counter resolution. In 8-bit counter mode, the Period register (PER) is used as TOP, and the TOP can be changed by writing to the PER register. In 16- and 32-bit counter mode, TOP is fixed to the maximum (MAX) value of the counter.

Normal Frequency Generation (NFRQ)

For Normal Frequency Generation, the period time (T) is controlled by the period register (PER) for 8-bit counter mode and MAX for 16- and 32-bit mode. The waveform generation output (WO[x]) is toggled on each compare match between COUNT and CCx, and the corresponding Match or Capture Channel x Interrupt Flag (INTFLAG.MCx) will be set.

Downloaded from Arrow.com.

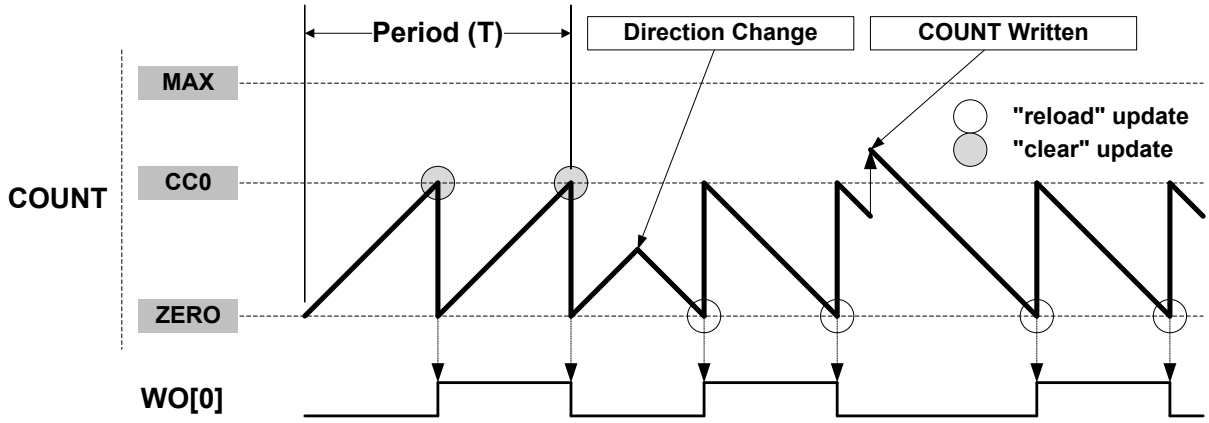
Figure 38-4. Normal Frequency Operation



Match Frequency Generation (MFRQ)

For Match Frequency Generation, the period time (T) is controlled by the CC0 register instead of PER or MAX. WO[0] toggles on each update condition.

Figure 38-5. Match Frequency Operation



Normal Pulse-Width Modulation Operation (NPWM)

NPWM uses single-slope PWM generation.

For single-slope PWM generation, the period time (T) is controlled by the TOP value, and CCx controls the duty cycle of the generated waveform output. When up-counting, the WO[x] is set at start or compare match between the COUNT and TOP values, and cleared on compare match between COUNT and CCx register values. When down-counting, the WO[x] is cleared at start or compare match between the COUNT and ZERO values, and set on compare match between COUNT and CCx register values.

The following equation calculates the exact resolution for a single-slope PWM (R_{PWM_SS}) waveform:

$$R_{PWM_SS} = \frac{\log(TOP+1)}{\log(2)}$$

The PWM frequency (f_{PWM_SS}) depends on TOP value and the peripheral clock frequency (f_{GCLK_TC}), and can be calculated by the following equation:

$$f_{PWM_SS} = \frac{f_{GCLK_TC}}{N(TOP+1)}$$

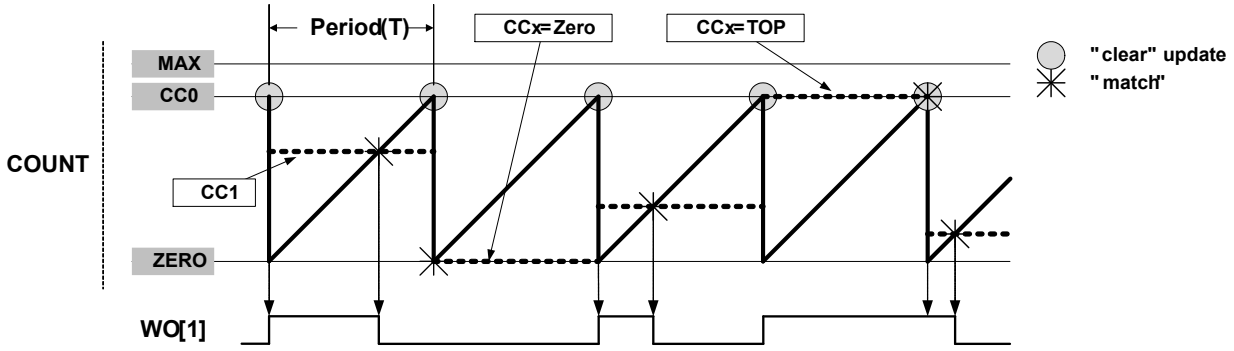
Where N represents the prescaler divider used (1, 2, 4, 8, 16, 64, 256, 1024).

Match Pulse-Width Modulation Operation (MPWM)

In MPWM, the output of WO[1] is depending on CC1 as shown in the figure below. On every overflow/ underflow, a one-TC-clock-cycle negative pulse is put out on WO[0] (not shown in the figure).

Downloaded from Arrow.com.

Figure 38-6. Match PWM Operation



The table below shows the update counter and overflow event/interrupt generation conditions in different operation modes.

Table 38-3. Counter Update and Overflow Event/interrupt Conditions in TC

Name	Operation	TOP	Update	Output Waveform		OVFIF/Event	
				On Match	On Update	Up	Down
NFRQ	Normal Frequency	PER	TOP/ ZERO	Toggle	Stable	TOP	ZERO
MFRQ	Match Frequency	CC0	TOP/ ZERO	Toggle	Stable	TOP	ZERO
NPWM	Single-slope PWM	PER	TOP/ ZERO	See description above.		TOP	ZERO
MPWM	Single-slope PWM	CC0	TOP/ ZERO	Toggle	Toggle	TOP	ZERO

Related Links

32. PORT - I/O Pin Controller

38.6.2.7 Double Buffering

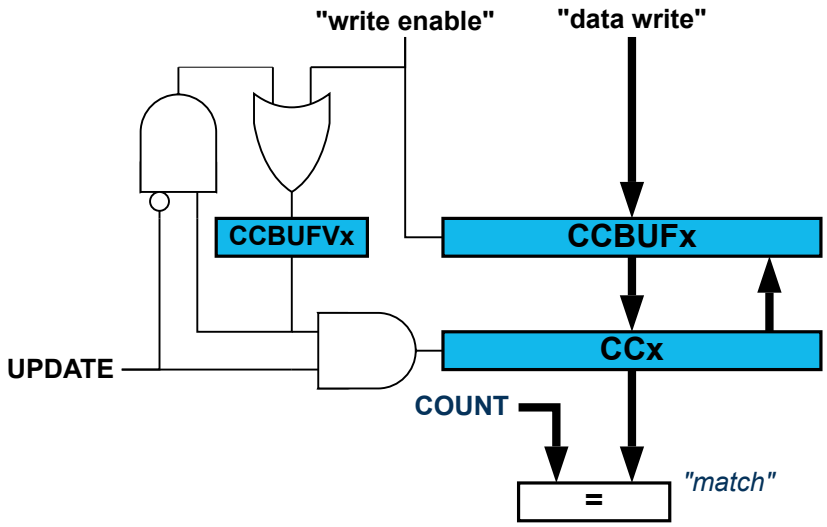
The Compare Channels (CCx) registers, and the Period (PER) register in 8-bit mode are double buffered. Each buffer register has a buffer valid bit (CCBUFVx or PERBUFV) in the STATUS register, which indicates that the buffer register contains a new valid value that can be copied into the corresponding register. As long as the respective buffer valid status flag (PERBUFV or CCBUFVx) are set to '1', related syncbusy bits are set (SYNCBUSY.PER or SYNCBUSY.CCx), a write to the respective PER/PERBUF or CCx/CCBUFx registers will generate a PAC error, and access to the respective PER or CCx register is invalid.

When the buffer valid flag bit in the STATUS register is '1' and the Lock Update bit in the CTRLB register is set to '0', (writing CTRLBCLR.LUPD to '1'), double buffering is enabled: the data from buffer registers will be copied into the corresponding register under hardware UPDATE conditions, then the buffer valid flags bit in the STATUS register are automatically cleared by hardware.
Note: The software update command (CTRLBSET.CMD=0x3) is acting independently of the LUPD value.

A compare register is double buffered as in the following figure.

Downloaded from Arrow.com.

Figure 38-7. Compare Channel Double Buffering



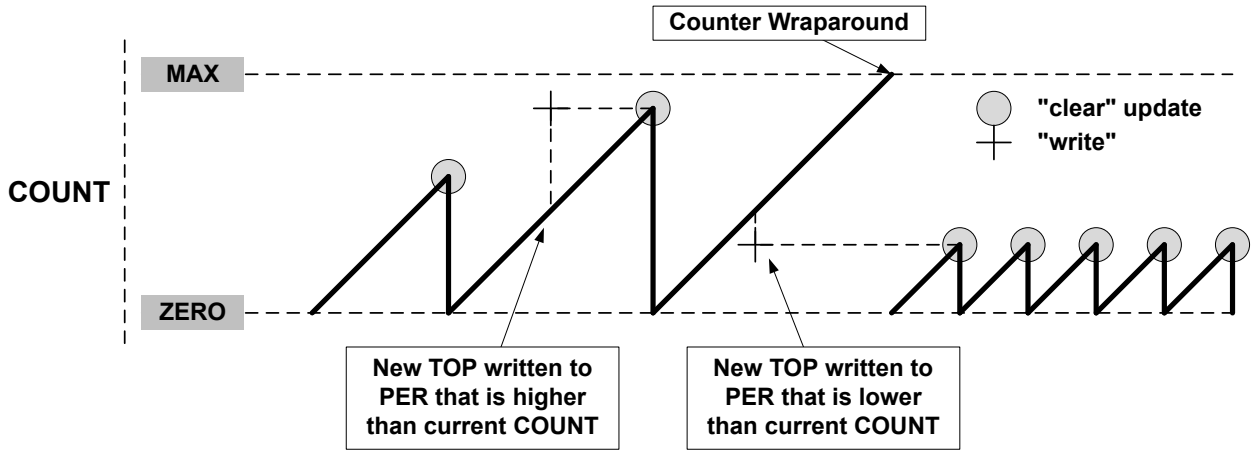
Both the registers (PER/CCx) and corresponding buffer registers (PERBUF/CCBUFx) are available in the I/O register map, and the double buffering feature is not mandatory. The double buffering is disabled by writing a '1' to CTRLBSET.LUPD.

Note: In NFRQ, MFRQ or PWM down-counting counter mode (CTRLBSET.DIR=1), when double buffering is enabled (CTRLBCLR.LUPD=1), PERBUF register is continously copied into the PER independently of update conditions.

Changing the Period

The counter period can be changed by writing a new TOP value to the Period register (PER or CC0, depending on the waveform generation mode), which is available in 8-bit mode. Any period update on registers (PER or CCx) is effective after the synchronization delay.

Figure 38-8. Unbuffered Single-Slope Up-Counting Operation

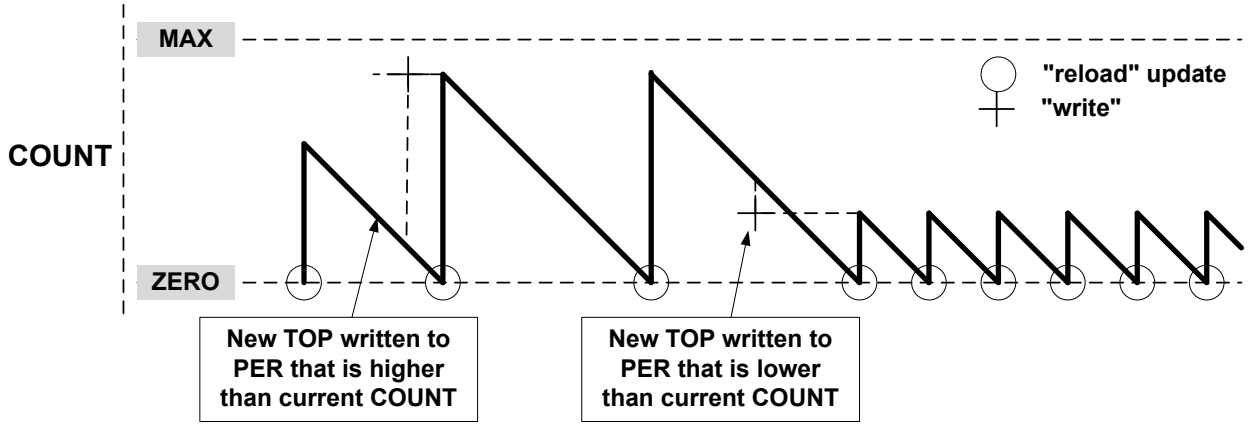


A counter wraparound can occur in any operation mode when up-counting without buffering, see Figure 38-8.

COUNT and TOP are continuously compared, so when a new TOP value that is lower than current COUNT is written to TOP, COUNT will wrap before a compare match.

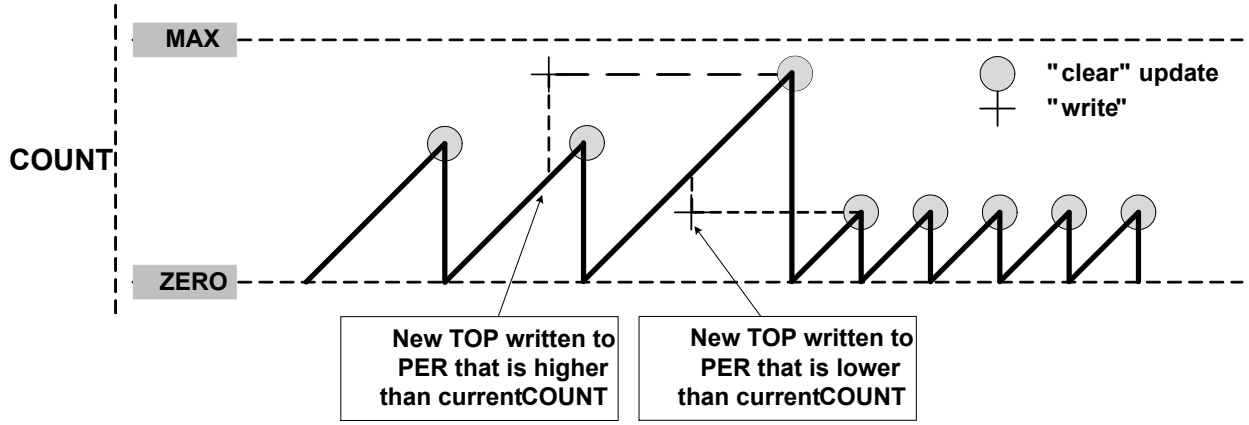
Downloaded from Arrow.com.

Figure 38-9. Unbuffered Single-Slope Down-Counting Operation



When double buffering is used, the buffer can be written at any time and the counter will still maintain correct operation. The period register is always updated on the update condition, as shown in Figure 38-10. This prevents wraparound and the generation of odd waveforms.

Figure 38-10. Changing the Period Using Buffering



38.6.2.8 Capture Operations

To enable and use capture operations, the corresponding Capture Channel x Enable bit in the Control A register (CTRLA.CAPTENx) must be written to '1'.

A capture trigger can be provided by input event line TC_EV or by asynchronous IO pin WO[x] for each capture channel or by a TC event. To enable the capture from input event line, Event Input Enable bit in the Event Control register (EVCTRL.TCEI) must be written to '1'. To enable the capture from the IO pin, the Capture On Pin x Enable bit in CTRLA register (CTRLA.COPENx) must be written to '1'.

Note:

1. The RETRIGGER, COUNT and START event actions are available only on an event from the Event System.
2. Event system channels must be configured to operate in asynchronous mode of operation when used for capture operations.

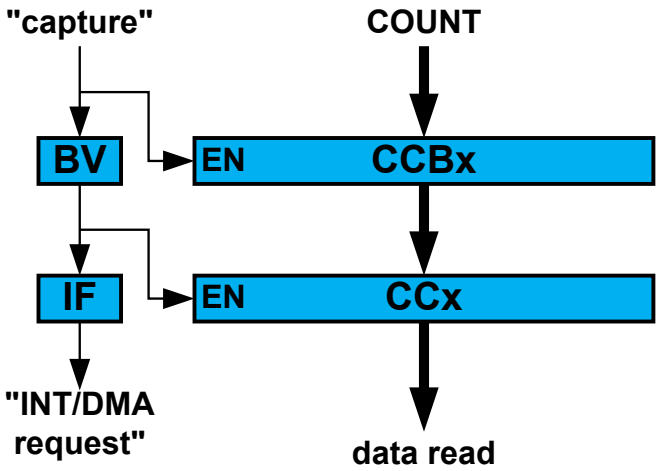
By default, a capture operation is done when a rising edge is detected on the input signal. Capture on falling edge is available, its activation is depending on the input source:

- When the channel is used with a IO pin, write a '1' to the corresponding Invert Enable bit in the Drive Control register (DRVCTRL.INVENx).

Downloaded from Arrow.com.

- When the channel is counting events from the Event System, write a '1' to the TC Event Input Invert Enable bit in Event Control register (EVCTRL.TCINV).

Figure 38-11. Capture Double Buffering

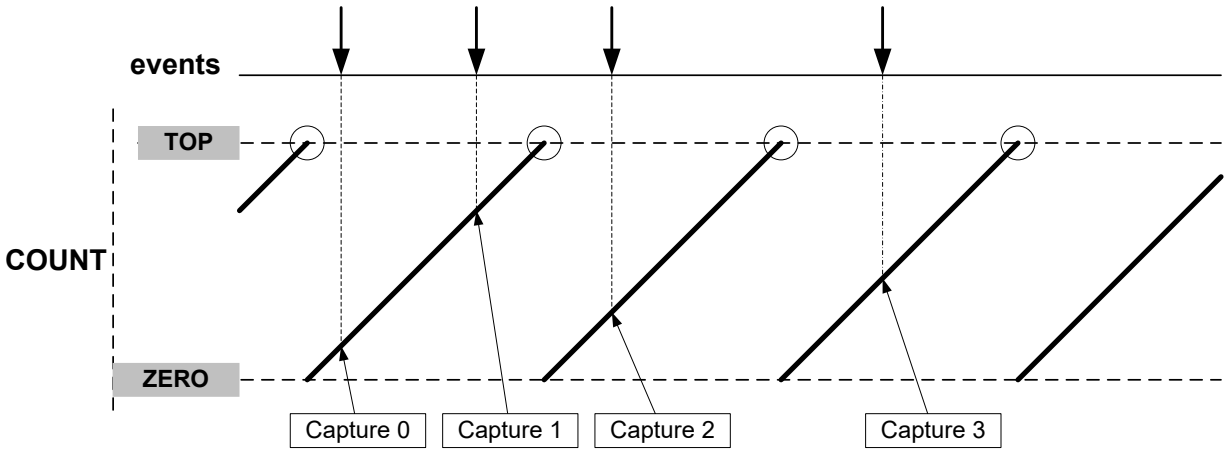


For input capture, the buffer register and the corresponding CCx act like a FIFO. When CCx is empty or read, any content in CCBUFx is transferred to CCx. The buffer valid flag is passed to set the CCx interrupt flag (IF) and generate the optional interrupt, event or DMA request. The CCBUFx register value can't be read, all captured data must be read from CCx register.

38.6.2.8.1 Event Capture Action

The compare/capture channels can be used as input capture channels to capture events from the Event System and give them a timestamp. The following figure shows four capture events for one capture channel.

Figure 38-12. Input Capture Timing



The TC can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Interrupt flag (INTFLAG.MCx) is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

38.6.2.8.2 Period and Pulse-Width (PPW) Capture Action

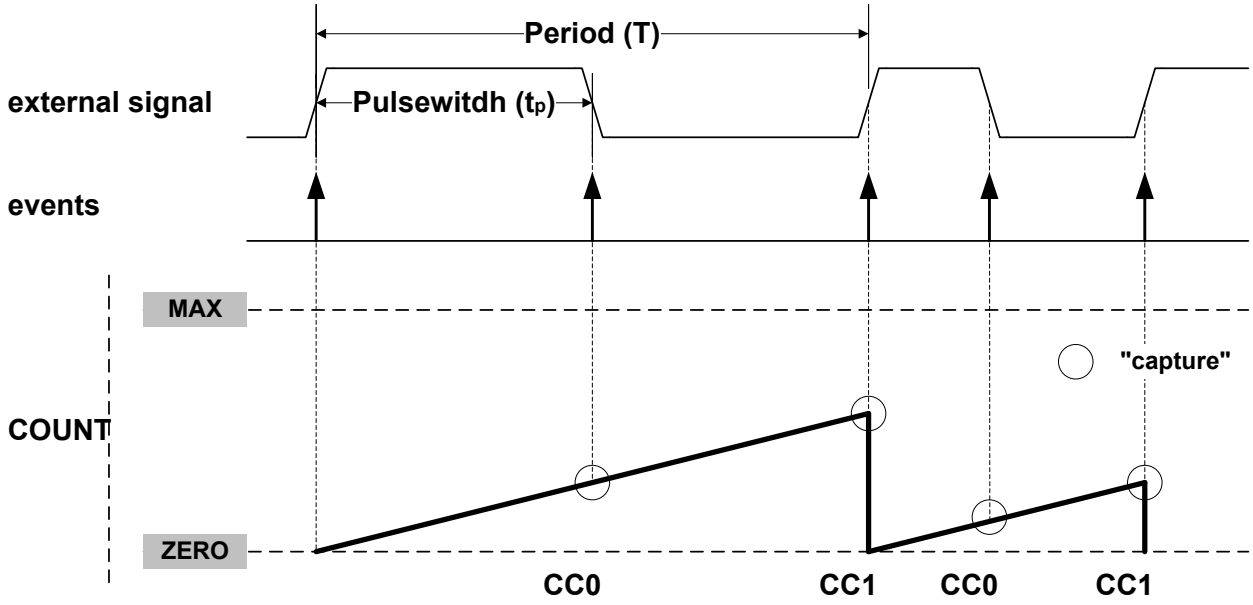
The TC can perform two input captures and restart the counter on one of the edges. This enables the TC to measure the pulse width and period and to characterize the frequency *f* and duty cycle of an input signal:

Downloaded from Arrow.com.

$$f = \frac{1}{T}$$

$$\text{dutyCycle} = \frac{t_p}{T}$$

Figure 38-13. PWP Capture



Selecting PWP in the Event Action bit group in the Event Control register (EVCTRL.EVACT) enables the TC to perform one capture action on the rising edge and the other one on the falling edge. The period *T* will be captured into CC1 and the pulse width t_p in CC0. EVCTRL.EVACT=PPW (period and pulse-width) offers identical functionality, but will capture *T* into CC0 and t_p into CC1.

The TC Event Input Invert Enable bit in the Event Control register (EVCTRL.TCINV) is used to select whether the wraparound should occur on the rising edge or the falling edge. If EVCTRL.TCINV=1, the wraparound will happen on the falling edge. In case pin capture is enabled, this can also be achieved by modifying the value of the DRVCTRL.INVENx bit.

The TC can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Interrupt flag (INTFLAG.MCx) is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

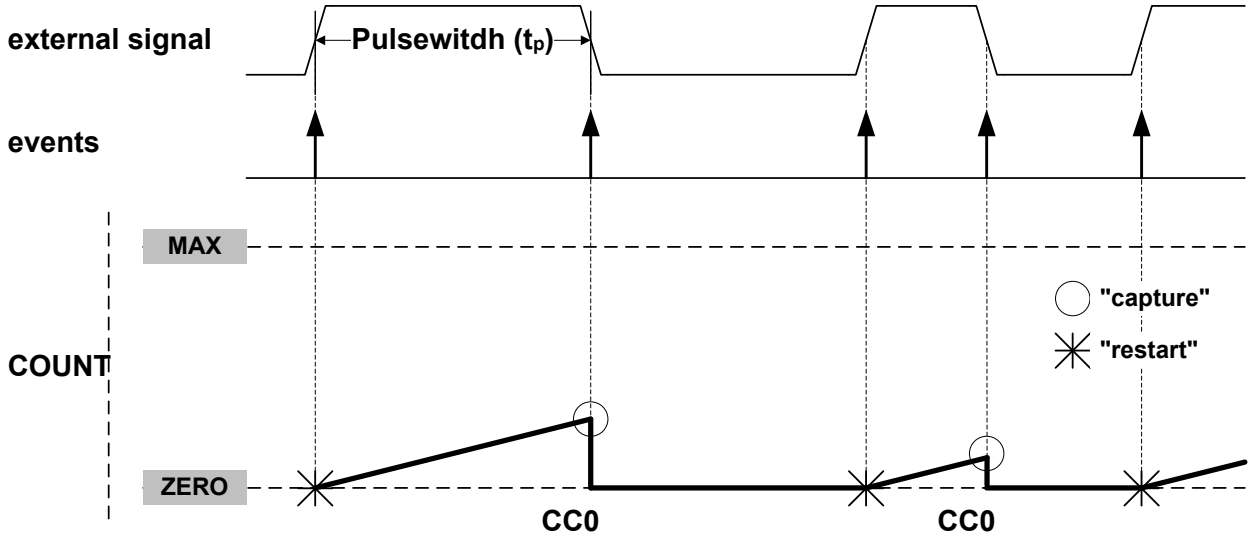
Note: The corresponding capture is working only if the channel is enabled in capture mode (CTRLA.CAPTENx=1). If not, the capture action is ignored and the channel is enabled in compare mode of operation. Consequently, both channels must be enabled in order to fully characterize the input.

38.6.2.8.3 Pulse-Width Capture Action

The TC performs the input capture on the falling edge of the input signal. When the edge is detected, the counter value is cleared and the TC stops counting. When a rising edge is detected on the input signal, the counter restarts the counting operation. To enable the operation on opposite edges, the input signal to capture must be inverted (refer to DRVCTRL.INVEN or EVCTRL.TCEINV).

Downloaded from Arrow.com.

Figure 38-14. Pulse-Width Capture on Channel 0



The TC can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Interrupt flag (INTFLAG.MCx) is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

38.6.3 Additional Features

38.6.3.1 One-Shot Operation

When one-shot is enabled, the counter automatically stops on the next counter overflow or underflow condition. When the counter is stopped, the Stop bit in the Status register (STATUS.STOP) is automatically set and the waveform outputs are set to zero.

One-shot operation is enabled by writing a '1' to the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT), and disabled by writing a '1' to CTRLBCLR.ONESHOT. When enabled, the TC will count until an overflow or underflow occurs and stops counting operation. The one-shot operation can be restarted by a re-trigger software command, a re-trigger event, or a start event. When the counter restarts its operation, STATUS.STOP is automatically cleared.

38.6.3.2 Time-Stamp Capture

This feature is enabled when the Capture Time Stamp (STAMP) Event Action in Event Control register (EVCTRL.EVACT) is selected. The counter TOP value must be smaller than MAX.

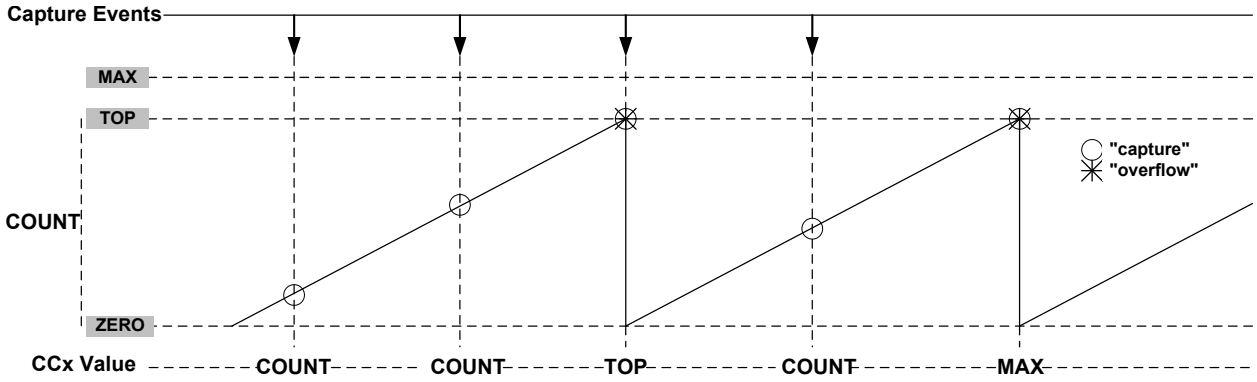
When a capture event is detected, the COUNT value is copied into the corresponding Channel x Compare/Capture Value (CCx) register. In case of an overflow, the MAX value is copied into the corresponding CCx register.

When a valid captured value is present in the capture channel register, the corresponding Capture Channel x Interrupt Flag (INTFLAG.MCx) is set.

The timer/counter can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Channel interrupt flag (INTFLAG.MCx) is still set, the new time-stamp will not be stored and INTFLAG.ERR will be set.

Downloaded from Arrow.com.

Figure 38-15. Time-Stamp



38.6.4 DMA Operation

The TC can generate the following DMA requests:

- Overflow (OVF): the request is set when an update condition (overflow, underflow or re-trigger) is detected, the request is cleared by hardware on DMA acknowledge.
- Match or Capture Channel x (MCx): for a compare channel, the request is set on each compare match detection, the request is cleared by hardware on DMA acknowledge. For a capture channel, the request is set when valid data is present in the CCx register, and cleared when CCx register is read.

38.6.5 Interrupts

The TC has the following interrupt sources:

- Overflow/Underflow (OVF)
- Match or Capture Channel x (MCx)
- Capture Overflow Error (ERR)

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition occurs.

Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is disabled, or the TC is reset. See INTFLAG for details on how to clear interrupt flags.

The TC has one common interrupt request line for all the interrupt sources. The user must read the INTFLAG register to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated. Refer to *Nested Vector Interrupt Controller* for details.

38.6.6 Events

The TC can generate the following output events:

- Overflow/Underflow (OVF)
- Match or Capture Channel x (MCx)

Downloaded from Arrow.com.

Writing a '1' to an Event Output bit in the Event Control register (EVCTRL.MCEOx) enables the corresponding output event. The output event is disabled by writing EVCTRL.MCEOx=0.

One of the following event actions can be selected by the Event Action bit group in the Event Control register (EVCTRL.EVACT):

- Disable event action (OFF)
- Start TC (START)
- Re-trigger TC (RETRIGGER)
- Count on event (COUNT)
- Capture time stamp (STAMP)
- Capture Period (PPW and PWP)
- Capture Pulse Width (PW)

Writing a '1' to the TC Event Input bit in the Event Control register (EVCTRL.TCEI) enables input events to the TC. Writing a '0' to this bit disables input events to the TC. The TC requires only asynchronous event inputs. For further details on how configuring the asynchronous events, refer to *EVSYS - Event System*.

Related Links

33. EVSYS – Event System

38.6.7 Sleep Mode Operation

The TC can be configured to operate in any sleep mode. To be able to run in standby, the RUNSTDBY bit in the Control A register (CTRLA.RUNSTDBY) must be '1'. This peripheral can wake up the device from any sleep mode using interrupts or perform actions through the Event System.

If the On Demand bit in the Control A register (CTRLA.ONDEMAND) is written to '1', the module stops requesting its peripheral clock when the STOP bit in STATUS register (STATUS.STOP) is set to '1'. When a re-trigger or start condition is detected, the TC requests the clock before the operation starts.

38.6.8 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset and Enable bits in Control A register (CTRLA.SWRST and CTRLA.ENABLE)
- Capture Channel Buffer Valid bit in STATUS register (STATUS.CCBUFVx)

The following registers are synchronized when written:

- Control B Clear and Control B Set registers (CTRLBCLR and CTRLBSET)
- Count Value register (COUNT)
- Period Value and Period Buffer Value registers (PER and PERBUF)
- Channel x Compare/Capture Value and Channel x Compare/Capture Buffer Value registers (CCx and CCBUFx)

The following registers are synchronized when read:

- Count Value register (COUNT): synchronization is done on demand through READSYNC command (CTRLBSET.CMD).

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Downloaded from Arrow.com.

Required read-synchronization is denoted by the "Read-Synchronized" property in the register description.

38.7 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to Synchronization.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

Downloaded from Arrow.com.

38.7.1 Register Summary - 8-bit Mode

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST
		15:8					ALOCK	PRESCALER[2:0]		
		23:16			COPEN1	COPEN0			CAPTEN1	CAPTEN0
		31:24								
0x04	CTRLBCLR	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x05	CTRLBSET	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x06	EVCTRL	7:0			TCEI	TCINV		EVACT[2:0]		
		15:8			MCEOx	MCEOx				OVFEO
0x08	INTENCLR	7:0				MCx			ERR	OVF
0x09	INTENSET	7:0				MCx			ERR	OVF
0x0A	INTFLAG	7:0				MCx			ERR	OVF
0x0B	STATUS	7:0				CCBUFVx	PERBUFV		SLAVE	STOP
0x0C	WAVE	7:0							WAVEGEN[1:0]	
0x0D	DRVCTRL	7:0								INVENx
0x0E	Reserved									
0x0F	DBGCTRL	7:0								DBGRUN
0x10	SYNCBUSY	7:0		CCx	PER	COUNT	STATUS	CTRLB	ENABLE	SWRST
		15:8								
		23:16								
		31:24								
0x14	COUNT	7:0	COUNT[7:0]							
0x15 ... 0x1A	Reserved									
0x1B	PER	7:0	PER[7:0]							
0x1C	CC0	7:0	CC[7:0]							
0x1D	CC1	7:0	CC[7:0]							
0x1E ... 0x2E	Reserved									
0x2F	PERBUF	7:0	PERBUF[7:0]							
0x30	CCBUF0	7:0	CCBUF[7:0]							
0x31	CCBUF1	7:0	CCBUF[7:0]							

Downloaded from Arrow.com.

38.7.1.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			COPEN1	COPEN0			CAPTEN1	CAPTEN0
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	15	14	13	12	11	10	9	8
					ALOCK	PRESCALER[2:0]		
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
Reset	0	0	0	0	0	0	0	0

Bits 20, 21 – COPENx Capture On Pin x Enable
Bit x of COPEN[1:0] selects the trigger source for capture operation, either events or I/O pin input.

Value	Description
0	Event from Event System is selected as trigger source for capture operation on channel x.
1	I/O pin is selected as trigger source for capture operation on channel x.

Bits 16, 17 – CAPTENx Capture Channel x Enable
Bit x of CAPTEN[1:0] selects whether channel x is a capture or a compare channel.

These bits are not synchronized.

Value	Description
0	CAPTEN disables capture on channel x.
1	CAPTEN enables capture on channel x.

Bit 11 – ALOCK Auto Lock
When this bit is set, Lock bit update (LUPD) is set to '1' on each overflow/underflow or re-trigger event.

This bit is not synchronized.

Downloaded from Arrow.com.

Value	Description
0	The LUPD bit is not affected on overflow/underflow, and re-trigger event.
1	The LUPD bit is set on each overflow/underflow or re-trigger event.

Bits 10:8 – PRESCALER[2:0] Prescaler
These bits select the counter prescaler factor.

These bits are not synchronized.

Value	Name	Description
0x0	DIV1	Prescaler: GCLK_TC
0x1	DIV2	Prescaler: GCLK_TC/2
0x2	DIV4	Prescaler: GCLK_TC/4
0x3	DIV8	Prescaler: GCLK_TC/8
0x4	DIV16	Prescaler: GCLK_TC/16
0x5	DIV64	Prescaler: GCLK_TC/64
0x6	DIV256	Prescaler: GCLK_TC/256
0x7	DIV1024	Prescaler: GCLK_TC/1024

Bit 7 – ONDEMAND Clock On Demand
This bit selects the clock requirements when the TC is stopped.

In standby mode, if the Run in Standby bit (CTRLA.RUNSTDBY) is '0', ONDEMAND is forced to '0'.

This bit is not synchronized.

Value	Description
0	The On Demand is disabled. If On Demand is disabled, the TC will continue to request the clock when its operation is stopped (STATUS.STOP=1).
1	The On Demand is enabled. When On Demand is enabled, the stopped TC will not request the clock. The clock is requested when a software re-trigger command is applied or when an event with start/re-trigger action is detected.

Bit 6 – RUNSTDBY Run in Standby
This bit is used to keep the TC running in standby mode.

This bit is not synchronized.

Value	Description
0	The TC is halted in standby.
1	The TC continues to run in standby.

Bits 5:4 – PRESCSYNC[1:0] Prescaler and Counter Synchronization
These bits select whether the counter should wrap around on the next GCLK_TCx clock or the next prescaled GCLK_TCx clock. It also makes it possible to reset the prescaler.

These bits are not synchronized.

Value	Name	Description
0x0	GCLK	Reload or reset the counter on next generic clock
0x1	PRESC	Reload or reset the counter on next prescaler clock
0x2	RESYNC	Reload or reset the counter on next generic clock. Reset the prescaler counter
0x3	-	Reserved

Downloaded from Arrow.com.

Bits 3:2 – MODE[1:0] Timer Counter Mode
These bits select the counter mode.

These bits are not synchronized.

Value	Name	Description
0x0	COUNT16	Counter in 16-bit mode
0x1	COUNT8	Counter in 8-bit mode
0x2	COUNT32	Counter in 32-bit mode
0x3	-	Reserved

Bit 1 – ENABLE Enable
Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately, and the ENABLE Synchronization Busy bit in the SYNCBUSY register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

This bit is not enable protected.

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled.

Bit 0 – SWRST Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the TC, except DBGCTRL, to their initial state, and the TC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence; all other writes in the same write-operation will be discarded.

Downloaded from Arrow.com.

38.7.1.2 Control B Clear

Name: CTRLBCLR
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection, Read-Synchronized, Write-Synchronized

This register allows the user to clear bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Set register (CTRLBSET).

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]					ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bits 7:5 – CMD[2:0] Command
These bits are used for software control of the TC. The commands are executed on the next prescaled GCLK_TC clock cycle. When a command has been executed, the CMD bit group will be read back as zero.

Writing 0x0 to these bits has no effect.

Writing a '1' to any of these bits will clear the pending command.

Bit 2 – ONESHOT One-Shot on Counter
This bit controls one-shot operation of the TC.

Writing a '0' to this bit has no effect

Writing a '1' to this bit will disable one-shot operation.

Value	Description
0	The TC will wrap around and continue counting on an overflow/underflow condition.
1	The TC will wrap around and stop on the next underflow/overflow condition.

Bit 1 – LUPD Lock Update
This bit controls the update operation of the TC buffered registers.

When CTRLB.LUPD is set, no any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.

This bit has no effect when input capture operation is enabled.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the LUPD bit.

Value	Description
0	The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on hardware update condition.
1	The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers on hardware update condition.

Downloaded from Arrow.com.

Bit 0 – DIR Counter Direction
This bit is used to change the direction of the counter.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the bit and make the counter count up.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

Downloaded from Arrow.com.

38.7.1.3 Control B Set

Name: CTRLBSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection, Read-synchronized, Write-Synchronized

This register allows the user to set bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Clear register (CTRLBCLR).

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]					ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bits 7:5 – CMD[2:0] Command
These bits are used for software control of the TC. The commands are executed on the next prescaled GCLK_TC clock cycle. When a command has been executed, the CMD bit group will be read back as zero.

Writing 0x0 to these bits has no effect.

Writing a value different from 0x0 to these bits will issue a command for execution.

Value	Name	Description
0x0	NONE	No action
0x1	RETRIGGER	Force a start, restart or retrigger
0x2	STOP	Force a stop
0x3	UPDATE	Force update of double buffered registers
0x4	READSYNC	Force a read synchronization of COUNT

Bit 2 – ONESHOT One-Shot on Counter
This bit controls one-shot operation of the TC.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will enable one-shot operation.

Value	Description
0	The TC will wrap around and continue counting on an overflow/underflow condition.
1	The TC will wrap around and stop on the next underflow/overflow condition.

Bit 1 – LUPD Lock Update
This bit controls the update operation of the TC buffered registers.

When CTRLB.LUPD is set, no any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the LUPD bit.

Downloaded from Arrow.com.

This bit has no effect when input capture operation is enabled.

Value	Description
0	The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on hardware update condition.
1	The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers on hardware update condition.

Bit 0 – DIR Counter Direction
This bit is used to change the direction of the counter.

Writing a '0' to this bit has no effect

Writing a '1' to this bit will clear the bit and make the counter count up.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

Downloaded from Arrow.com.

38.7.1.4 Event Control

Name: EVCTRL
Offset: 0x06
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
			MCEOx	MCEOx				OVFEO
Access			R/W	R/W				R/W
Reset			0	0				0

Bit	7	6	5	4	3	2	1	0
			TCEI	TCINV		EVACT[2:0]		
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bits 13,12 – MCEOx Match or Capture Channel x Event Output Enable [x = 1..0]
These bits enable the generation of an event for every match or capture on channel x.

Value	Description
0	Match/Capture event on channel x is disabled and will not be generated.
1	Match/Capture event on channel x is enabled and will be generated for every compare/capture.

Bit 8 – OVFEO Overflow/Underflow Event Output Enable
This bit enables the Overflow/Underflow event. When enabled, an event will be generated when the counter overflows/underflows.

Value	Description
0	Overflow/Underflow event is disabled and will not be generated.
1	Overflow/Underflow event is enabled and will be generated for every counter overflow/underflow.

Bit 5 – TCEI TC Event Enable
This bit is used to enable asynchronous input events to the TC.

Value	Description
0	Incoming events are disabled.
1	Incoming events are enabled.

Bit 4 – TCINV TC Inverted Event Input Polarity
This bit inverts the asynchronous input event source.

Value	Description
0	Input event source is not inverted.
1	Input event source is inverted.

Bits 2:0 – EVACT[2:0] Event Action
These bits define the event action the TC will perform on an event.

Downloaded from Arrow.com.

Value	Name	Description
0x0	OFF	Event action disabled
0x1	RETRIGGER	Start, restart or retrigger TC on event
0x2	COUNT	Count on event
0x3	START	Start TC on event
0x4	STAMP	Time stamp capture
0x5	PPW	Period captured in CC0, pulse width in CC1
0x6	PWP	Period captured in CC1, pulse width in CC0
0x7	PW	Pulse width capture

Downloaded from Arrow.com.

38.7.1.5 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x08
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Bit	7	6	5	4	3	2	1	0
				MCx			ERR	OVF
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – MCx Match or Capture Channel x Interrupt Enable
Writing a '0' to these bits has no effect.

Writing a '1' to MCx will clear the corresponding Match or Capture Channel x Interrupt Enable bit, which disables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bit 1 – ERR Error Interrupt Disable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 0 – OVF Overflow Interrupt Disable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Downloaded from Arrow.com.

38.7.1.6 Interrupt Enable Set

Name: INTENSET
Offset: 0x09
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Bit	7	6	5	4	3	2	1	0
				MCx			ERR	OVF
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – MCx Match or Capture Channel x Interrupt Enable
Writing a '0' to these bits has no effect.

Writing a '1' to MCx will set the corresponding Match or Capture Channel x Interrupt Enable bit, which enables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bit 1 – ERR Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 0 – OVF Overflow Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Downloaded from Arrow.com.

38.7.1.7 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x0A
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
				MCx			ERR	OVF
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – MCx Match or Capture Channel x
This flag is set on a comparison match, or when the corresponding CCx register contains a valid capture value. This flag is set on the next CLK_TC_CNT cycle, and will generate an interrupt request if the corresponding Match or Capture Channel x Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.MCx) is '1'.

Writing a '0' to one of these bits has no effect.

Writing a '1' to one of these bits will clear the corresponding Match or Capture Channel x interrupt flag

In capture operation, this flag is automatically cleared when CCx register is read.

Bit 1 – ERR Error Interrupt Flag
This flag is set when a new capture occurs on a channel while the corresponding Match or Capture Channel x interrupt flag is set, in which case there is nowhere to store the new capture.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Error interrupt flag.

Bit 0 – OVF Overflow Interrupt Flag
This flag is set on the next CLK_TC_CNT cycle after an overflow condition occurs, and will generate an interrupt request if INTENCLR.OVF or INTENSET.OVF is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Overflow interrupt flag.

Downloaded from Arrow.com.

38.7.1.8 Status

Name: STATUS
Offset: 0x0B
Reset: 0x01
Property: Read-Synchronized

Bit	7	6	5	4	3	2	1	0
				CCBUFVx	PERBUFV		SLAVE	STOP
Access				R/W	R/W		R	R
Reset				0	0		0	1

Bit 4 – CCBUFVx Channel x Compare or Capture Buffer Valid
For a compare channel x, the bit x is set when a new value is written to the corresponding CCBUFx register.

The bit x is cleared by writing a '1' to it when CTRLB.LUPD is set, or it is cleared automatically by hardware on UPDATE condition.

For a capture channel x, the bit x is set when a valid capture value is stored in the CCBUFx register. The bit x is cleared automatically when the CCx register is read.

Bit 3 – PERBUFV Period Buffer Valid
This bit is set when a new value is written to the PERBUF register. The bit is cleared by writing '1' to the corresponding location when CTRLB.LUPD is set, or automatically cleared by hardware on UPDATE condition. This bit is available only in 8-bit mode and will always read zero in 16- and 32-bit modes.

Bit 1 – SLAVE Slave Status Flag
This bit is only available in 32-bit mode on the slave TC (i.e., TC1 and/or TC3). The bit is set when the associated master TC (TC0 and TC2, respectively) is set to run in 32-bit mode.

Bit 0 – STOP Stop Status Flag
This bit is set when the TC is disabled, on a Stop command, or on an overflow/underflow condition when the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT) is '1'.

Value	Description
0	Counter is running.
1	Counter is stopped.

Downloaded from Arrow.com.

38.7.1.9 Waveform Generation Control

Name: WAVE
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
							WAVEGEN[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 1:0 – WAVEGEN[1:0] Waveform Generation Mode
These bits select the waveform generation operation. They affect the top value, as shown in 38.6.2.6.1 Waveform Output Operations. They also control whether frequency or PWM waveform generation should be used. The waveform generation operations are explained in 38.6.2.6.1 Waveform Output Operations.

These bits are not synchronized.

Value	Name	Operation	Top Value	Output Waveform on Match	Output Waveform on Wraparound
0x0	NFRQ	Normal frequency	PER[1] / Max	Toggle	No action
0x1	MFRQ	Match frequency	CC0	Toggle	No action
0x2	NPWM	Normal PWM	PER[1] / Max	Set	Clear
0x3	MPWM	Match PWM	CC0	Set	Clear

1) This depends on the TC mode: In 8-bit mode, the top value is the Period Value register (PER). In 16- and 32-bit mode it is the respective MAX value.

Downloaded from Arrow.com.

38.7.1.10 Driver Control

Name: DRVCTRL
Offset: 0x0D
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
								INVENx
Access								R/W
Reset								0

Bit 0 – INVENx Output Waveform x Invert Enable
Bit x of INVEN[1:0] selects inversion of the output or capture trigger input of channel x.

Value	Description
0	Disable inversion of the WO[x] output and IO input pin.
1	Enable inversion of the WO[x] output and IO input pin.

Downloaded from Arrow.com.

38.7.1.11 Debug Control

Name: DBGCTRL
Offset: 0x0F
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN Run in Debug Mode
This bit is not affected by a software Reset, and should not be changed by software while the TC is enabled.

Value	Description
0	The TC is halted when the device is halted in debug mode.
1	The TC continues normal operation when the device is halted in debug mode.

Downloaded from Arrow.com.

38.7.1.12 Synchronization Busy

Name: SYNCBUSY
Offset: 0x10
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		CCx	PER	COUNT	STATUS	CTRLB	ENABLE	SWRST
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit 6 – CCx Compare/Capture Channel x Synchronization Busy
For details on CC channels number, refer to each TC feature list.

This bit is set when the synchronization of CCx between clock domains is started.

This bit is also set when the CCBUFx is written, and cleared on update condition. The bit is automatically cleared when the STATUS.CCBUFx bit is cleared.

Bit 5 – PER PER Synchronization Busy
This bit is cleared when the synchronization of PER between the clock domains is complete.

This bit is set when the synchronization of PER between clock domains is started.

This bit is also set when the PER is written, and cleared on update condition. The bit is automatically cleared when the STATUS.PERBUF bit is cleared.

Bit 4 – COUNT COUNT Synchronization Busy
This bit is cleared when the synchronization of COUNT between the clock domains is complete.

This bit is set when the synchronization of COUNT between clock domains is started.

Bit 3 – STATUS STATUS Synchronization Busy
This bit is cleared when the synchronization of STATUS between the clock domains is complete.

Downloaded from Arrow.com.

This bit is set when a '1' is written to the Capture Channel Buffer Valid status flags (STATUS.CCBUFVx) and the synchronization of STATUS between clock domains is started.

Bit 2 – CTRLB CTRLB Synchronization Busy
This bit is cleared when the synchronization of CTRLB between the clock domains is complete.

This bit is set when the synchronization of CTRLB between clock domains is started.

Bit 1 – ENABLE ENABLE Synchronization Busy
This bit is cleared when the synchronization of ENABLE bit between the clock domains is complete.

This bit is set when the synchronization of ENABLE bit between clock domains is started.

Bit 0 – SWRST SWRST Synchronization Busy
This bit is cleared when the synchronization of SWRST bit between the clock domains is complete.

This bit is set when the synchronization of SWRST bit between clock domains is started.

Downloaded from Arrow.com.

38.7.1.13 Counter Value, 8-bit Mode

Name: COUNT
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Note: Prior to any read access, this register must be synchronized by user by writing the according TC Command value to the Control B Set register (CTRLBSET.CMD=READSYNC).

Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – COUNT[7:0] Counter Value
These bits contain the current counter value.

Downloaded from Arrow.com.

38.7.1.14 Period Value, 8-bit Mode

Name: PER
Offset: 0x1B
Reset: 0xFF
Property: Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	PER[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bits 7:0 – PER[7:0] Period Value
These bits hold the value of the Period Buffer register PERBUF. The value is copied to PER register on UPDATE condition.

Downloaded from Arrow.com.

38.7.1.15 Channel x Compare/Capture Value, 8-bit Mode

Name: CCx
Offset: 0x1C + x*0x01 [x=0..1]
Reset: 0x00
Property: Write-Synchronized, Read-Synchronized

Bit	7	6	5	4	3	2	1	0
	CC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – CC[7:0] Channel x Compare/Capture Value
These bits contain the compare/capture value in 8-bit TC mode. In Match frequency (MFRQ) or Match PWM (MPWM) waveform operation (WAVE.WAVEGEN), the CC0 register is used as a period register.

Downloaded from Arrow.com.

38.7.1.16 Period Buffer Value, 8-bit Mode

Name: PERBUF
Offset: 0x2F
Reset: 0xFF
Property: Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	PERBUF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bits 7:0 – PERBUF[7:0] Period Buffer Value
These bits hold the value of the period buffer register. The value is copied to PER register on UPDATE condition.

Downloaded from Arrow.com.

38.7.1.17 Channel x Compare Buffer Value, 8-bit Mode

Name: CCBUFx
Offset: 0x30 + x*0x01 [x=0..1]
Reset: 0x00
Property: Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	CCBUF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – CCBUF[7:0] Channel x Compare Buffer Value
These bits hold the value of the Channel x Compare Buffer Value. When the buffer valid flag is '1' and double buffering is enabled (CTRLBCLR.LUPD=1), the data from buffer registers will be copied into the corresponding CCx register under UPDATE condition (CTRLBSET.CMD=0x3), including the software update command.

Downloaded from Arrow.com.

38.7.2 Register Summary - 16-bit Mode

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST
		15:8					ALOCK	PRESCALER[2:0]		
		23:16			COPEN1	COPEN0			CAPTEN1	CAPTEN0
		31:24								
0x04	CTRLBCLR	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x05	CTRLBSET	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x06	EVCTRL	7:0			TCEI	TCINV		EVACT[2:0]		
		15:8			MCEOx	MCEOx				OVFEO
0x08	INTENCLR	7:0				MCx			ERR	OVF
0x09	INTENSET	7:0				MCx			ERR	OVF
0x0A	INTFLAG	7:0				MCx			ERR	OVF
0x0B	STATUS	7:0				CCBUFVx	PERBUFV		SLAVE	STOP
0x0C	WAVE	7:0							WAVEGEN[1:0]	
0x0D	DRVCTRL	7:0								INVENx
0x0E	Reserved									
0x0F	DBGCTRL	7:0								DBGRUN
0x10	SYNCBUSY	7:0		CCx	PER	COUNT	STATUS	CTRLB	ENABLE	SWRST
		15:8								
		23:16								
		31:24								
0x14	COUNT	7:0	COUNT[7:0]							
		15:8	COUNT[15:8]							
0x16 ... 0x19	Reserved									
0x1A	PER	7:0	PER[7:0]							
		15:8	PER[15:8]							
0x1C	CC0	7:0	CC[7:0]							
		15:8	CC[15:8]							
0x1E	CC1	7:0	CC[7:0]							
		15:8	CC[15:8]							
0x20 ... 0x2D	Reserved									
0x2E	PERBUF	7:0	PERBUF[7:0]							
		15:8	PERBUF[15:8]							
0x30	CCBUF0	7:0	CCBUF[7:0]							
		15:8	CCBUF[15:8]							
0x32	CCBUF1	7:0	CCBUF[7:0]							
		15:8	CCBUF[15:8]							

Downloaded from Arrow.com.

38.7.2.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			COPEN1	COPEN0			CAPTEN1	CAPTEN0
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	15	14	13	12	11	10	9	8
					ALOCK	PRESCALER[2:0]		
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
Reset	0	0	0	0	0	0	0	0

Bits 20, 21 – COPENx Capture On Pin x Enable
Bit x of COPEN[1:0] selects the trigger source for capture operation, either events or I/O pin input.

Value	Description
0	Event from Event System is selected as trigger source for capture operation on channel x.
1	I/O pin is selected as trigger source for capture operation on channel x.

Bits 16, 17 – CAPTENx Capture Channel x Enable
Bit x of CAPTEN[1:0] selects whether channel x is a capture or a compare channel.

These bits are not synchronized.

Value	Description
0	CAPTEN disables capture on channel x.
1	CAPTEN enables capture on channel x.

Bit 11 – ALOCK Auto Lock
When this bit is set, Lock bit update (LUPD) is set to '1' on each overflow/underflow or re-trigger event.

This bit is not synchronized.

Downloaded from Arrow.com.

Value	Description
0	The LUPD bit is not affected on overflow/underflow, and re-trigger event.
1	The LUPD bit is set on each overflow/underflow or re-trigger event.

Bits 10:8 – PRESCALER[2:0] Prescaler
These bits select the counter prescaler factor.

These bits are not synchronized.

Value	Name	Description
0x0	DIV1	Prescaler: GCLK_TC
0x1	DIV2	Prescaler: GCLK_TC/2
0x2	DIV4	Prescaler: GCLK_TC/4
0x3	DIV8	Prescaler: GCLK_TC/8
0x4	DIV16	Prescaler: GCLK_TC/16
0x5	DIV64	Prescaler: GCLK_TC/64
0x6	DIV256	Prescaler: GCLK_TC/256
0x7	DIV1024	Prescaler: GCLK_TC/1024

Bit 7 – ONDEMAND Clock On Demand
This bit selects the clock requirements when the TC is stopped.

In standby mode, if the Run in Standby bit (CTRLA.RUNSTDBY) is '0', ONDEMAND is forced to '0'.

This bit is not synchronized.

Value	Description
0	The On Demand is disabled. If On Demand is disabled, the TC will continue to request the clock when its operation is stopped (STATUS.STOP=1).
1	The On Demand is enabled. When On Demand is enabled, the stopped TC will not request the clock. The clock is requested when a software re-trigger command is applied or when an event with start/re-trigger action is detected.

Bit 6 – RUNSTDBY Run in Standby
This bit is used to keep the TC running in standby mode.

This bit is not synchronized.

Value	Description
0	The TC is halted in standby.
1	The TC continues to run in standby.

Bits 5:4 – PRESCSYNC[1:0] Prescaler and Counter Synchronization
These bits select whether the counter should wrap around on the next GCLK_TCx clock or the next prescaled GCLK_TCx clock. It also makes it possible to reset the prescaler.

These bits are not synchronized.

Value	Name	Description
0x0	GCLK	Reload or reset the counter on next generic clock
0x1	PRESC	Reload or reset the counter on next prescaler clock
0x2	RESYNC	Reload or reset the counter on next generic clock. Reset the prescaler counter
0x3	-	Reserved

Downloaded from Arrow.com.

Bits 3:2 – MODE[1:0] Timer Counter Mode
These bits select the counter mode.

These bits are not synchronized.

Value	Name	Description
0x0	COUNT16	Counter in 16-bit mode
0x1	COUNT8	Counter in 8-bit mode
0x2	COUNT32	Counter in 32-bit mode
0x3	-	Reserved

Bit 1 – ENABLE Enable
Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately, and the ENABLE Synchronization Busy bit in the SYNCBUSY register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

This bit is not enable protected.

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled.

Bit 0 – SWRST Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the TC, except DBGCTRL, to their initial state, and the TC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence; all other writes in the same write-operation will be discarded.

Downloaded from Arrow.com.

38.7.2.2 Control B Clear

Name: CTRLBCLR
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection, Read-Synchronized, Write-Synchronized

This register allows the user to clear bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Set register (CTRLBSET).

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]					ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bits 7:5 – CMD[2:0] Command
These bits are used for software control of the TC. The commands are executed on the next prescaled GCLK_TC clock cycle. When a command has been executed, the CMD bit group will be read back as zero.

Writing 0x0 to these bits has no effect.

Writing a '1' to any of these bits will clear the pending command.

Bit 2 – ONESHOT One-Shot on Counter
This bit controls one-shot operation of the TC.

Writing a '0' to this bit has no effect

Writing a '1' to this bit will disable one-shot operation.

Value	Description
0	The TC will wrap around and continue counting on an overflow/underflow condition.
1	The TC will wrap around and stop on the next underflow/overflow condition.

Bit 1 – LUPD Lock Update
This bit controls the update operation of the TC buffered registers.

When CTRLB.LUPD is set, no any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.

This bit has no effect when input capture operation is enabled.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the LUPD bit.

Value	Description
0	The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on hardware update condition.
1	The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers on hardware update condition.

Downloaded from Arrow.com.

Bit 0 – DIR Counter Direction
This bit is used to change the direction of the counter.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the bit and make the counter count up.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

Downloaded from Arrow.com.

38.7.2.3 Control B Set

Name: CTRLBSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection, Read-synchronized, Write-Synchronized

This register allows the user to set bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Clear register (CTRLBCLR).

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]					ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bits 7:5 – CMD[2:0] Command
These bits are used for software control of the TC. The commands are executed on the next prescaled GCLK_TC clock cycle. When a command has been executed, the CMD bit group will be read back as zero.

Writing 0x0 to these bits has no effect.

Writing a value different from 0x0 to these bits will issue a command for execution.

Value	Name	Description
0x0	NONE	No action
0x1	RETRIGGER	Force a start, restart or retrigger
0x2	STOP	Force a stop
0x3	UPDATE	Force update of double buffered registers
0x4	READSYNC	Force a read synchronization of COUNT

Bit 2 – ONESHOT One-Shot on Counter
This bit controls one-shot operation of the TC.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will enable one-shot operation.

Value	Description
0	The TC will wrap around and continue counting on an overflow/underflow condition.
1	The TC will wrap around and stop on the next underflow/overflow condition.

Bit 1 – LUPD Lock Update
This bit controls the update operation of the TC buffered registers.

When CTRLB.LUPD is set, no any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the LUPD bit.

Downloaded from Arrow.com.

This bit has no effect when input capture operation is enabled.

Value	Description
0	The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on hardware update condition.
1	The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers on hardware update condition.

Bit 0 – DIR Counter Direction
This bit is used to change the direction of the counter.

Writing a '0' to this bit has no effect

Writing a '1' to this bit will clear the bit and make the counter count up.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

Downloaded from Arrow.com.

38.7.2.4 Event Control

Name: EVCTRL
Offset: 0x06
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
			MCEOx	MCEOx				OVFEO
Access			R/W	R/W				R/W
Reset			0	0				0

Bit	7	6	5	4	3	2	1	0
			TCEI	TCINV		EVACT[2:0]		
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bits 13,12 – MCEOx Match or Capture Channel x Event Output Enable [x = 1..0]
These bits enable the generation of an event for every match or capture on channel x.

Value	Description
0	Match/Capture event on channel x is disabled and will not be generated.
1	Match/Capture event on channel x is enabled and will be generated for every compare/capture.

Bit 8 – OVFEO Overflow/Underflow Event Output Enable
This bit enables the Overflow/Underflow event. When enabled, an event will be generated when the counter overflows/underflows.

Value	Description
0	Overflow/Underflow event is disabled and will not be generated.
1	Overflow/Underflow event is enabled and will be generated for every counter overflow/underflow.

Bit 5 – TCEI TC Event Enable
This bit is used to enable asynchronous input events to the TC.

Value	Description
0	Incoming events are disabled.
1	Incoming events are enabled.

Bit 4 – TCINV TC Inverted Event Input Polarity
This bit inverts the asynchronous input event source.

Value	Description
0	Input event source is not inverted.
1	Input event source is inverted.

Bits 2:0 – EVACT[2:0] Event Action
These bits define the event action the TC will perform on an event.

Downloaded from Arrow.com.

Value	Name	Description
0x0	OFF	Event action disabled
0x1	RETRIGGER	Start, restart or retrigger TC on event
0x2	COUNT	Count on event
0x3	START	Start TC on event
0x4	STAMP	Time stamp capture
0x5	PPW	Period captured in CC0, pulse width in CC1
0x6	PWP	Period captured in CC1, pulse width in CC0
0x7	PW	Pulse width capture

Downloaded from Arrow.com.

38.7.2.5 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x08
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Bit	7	6	5	4	3	2	1	0
				MCx			ERR	OVF
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – MCx Match or Capture Channel x Interrupt Enable
Writing a '0' to these bits has no effect.

Writing a '1' to MCx will clear the corresponding Match or Capture Channel x Interrupt Enable bit, which disables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bit 1 – ERR Error Interrupt Disable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 0 – OVF Overflow Interrupt Disable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Downloaded from Arrow.com.

38.7.2.6 Interrupt Enable Set

Name: INTENSET
Offset: 0x09
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Bit	7	6	5	4	3	2	1	0
				MCx			ERR	OVF
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – MCx Match or Capture Channel x Interrupt Enable
Writing a '0' to these bits has no effect.

Writing a '1' to MCx will set the corresponding Match or Capture Channel x Interrupt Enable bit, which enables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bit 1 – ERR Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 0 – OVF Overflow Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Downloaded from Arrow.com.

38.7.2.7 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x0A
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
				MCx			ERR	OVF
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – MCx Match or Capture Channel x
This flag is set on a comparison match, or when the corresponding CCx register contains a valid capture value. This flag is set on the next CLK_TC_CNT cycle, and will generate an interrupt request if the corresponding Match or Capture Channel x Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.MCx) is '1'.

Writing a '0' to one of these bits has no effect.

Writing a '1' to one of these bits will clear the corresponding Match or Capture Channel x interrupt flag

In capture operation, this flag is automatically cleared when CCx register is read.

Bit 1 – ERR Error Interrupt Flag
This flag is set when a new capture occurs on a channel while the corresponding Match or Capture Channel x interrupt flag is set, in which case there is nowhere to store the new capture.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Error interrupt flag.

Bit 0 – OVF Overflow Interrupt Flag
This flag is set on the next CLK_TC_CNT cycle after an overflow condition occurs, and will generate an interrupt request if INTENCLR.OVF or INTENSET.OVF is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Overflow interrupt flag.

Downloaded from Arrow.com.

38.7.2.8 Status

Name: STATUS
Offset: 0x0B
Reset: 0x01
Property: Read-Synchronized

Bit	7	6	5	4	3	2	1	0
				CCBUFVx	PERBUFV		SLAVE	STOP
Access				R/W	R/W		R	R
Reset				0	0		0	1

Bit 4 – CCBUFVx Channel x Compare or Capture Buffer Valid
For a compare channel x, the bit x is set when a new value is written to the corresponding CCBUFx register.

The bit x is cleared by writing a '1' to it when CTRLB.LUPD is set, or it is cleared automatically by hardware on UPDATE condition.

For a capture channel x, the bit x is set when a valid capture value is stored in the CCBUFx register. The bit x is cleared automatically when the CCx register is read.

Bit 3 – PERBUFV Period Buffer Valid
This bit is set when a new value is written to the PERBUF register. The bit is cleared by writing '1' to the corresponding location when CTRLB.LUPD is set, or automatically cleared by hardware on UPDATE condition. This bit is available only in 8-bit mode and will always read zero in 16- and 32-bit modes.

Bit 1 – SLAVE Slave Status Flag
This bit is only available in 32-bit mode on the slave TC (i.e., TC1 and/or TC3). The bit is set when the associated master TC (TC0 and TC2, respectively) is set to run in 32-bit mode.

Bit 0 – STOP Stop Status Flag
This bit is set when the TC is disabled, on a Stop command, or on an overflow/underflow condition when the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT) is '1'.

Value	Description
0	Counter is running.
1	Counter is stopped.

Downloaded from Arrow.com.

38.7.2.9 Waveform Generation Control

Name: WAVE
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
							WAVEGEN[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 1:0 – WAVEGEN[1:0] Waveform Generation Mode
These bits select the waveform generation operation. They affect the top value, as shown in 38.6.2.6.1 Waveform Output Operations. They also control whether frequency or PWM waveform generation should be used. The waveform generation operations are explained in 38.6.2.6.1 Waveform Output Operations.

These bits are not synchronized.

Value	Name	Operation	Top Value	Output Waveform on Match	Output Waveform on Wraparound
0x0	NFRQ	Normal frequency	PER[1] / Max	Toggle	No action
0x1	MFRQ	Match frequency	CC0	Toggle	No action
0x2	NPWM	Normal PWM	PER[1] / Max	Set	Clear
0x3	MPWM	Match PWM	CC0	Set	Clear

1) This depends on the TC mode: In 8-bit mode, the top value is the Period Value register (PER). In 16- and 32-bit mode it is the respective MAX value.

Downloaded from Arrow.com.

38.7.2.10 Driver Control

Name: DRVCTRL
Offset: 0x0D
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
								INVENx
Access								R/W
Reset								0

Bit 0 – INVENx Output Waveform x Invert Enable
Bit x of INVEN[1:0] selects inversion of the output or capture trigger input of channel x.

Value	Description
0	Disable inversion of the WO[x] output and IO input pin.
1	Enable inversion of the WO[x] output and IO input pin.

Downloaded from Arrow.com.

38.7.2.11 Debug Control

Name: DBGCTRL
Offset: 0x0F
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN Run in Debug Mode
This bit is not affected by a software Reset, and should not be changed by software while the TC is enabled.

Value	Description
0	The TC is halted when the device is halted in debug mode.
1	The TC continues normal operation when the device is halted in debug mode.

Downloaded from Arrow.com.

38.7.2.12 Synchronization Busy

Name: SYNCBUSY
Offset: 0x10
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		CCx	PER	COUNT	STATUS	CTRLB	ENABLE	SWRST
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit 6 – CCx Compare/Capture Channel x Synchronization Busy
For details on CC channels number, refer to each TC feature list.

This bit is set when the synchronization of CCx between clock domains is started.

This bit is also set when the CCBUFx is written, and cleared on update condition. The bit is automatically cleared when the STATUS.CCBUFx bit is cleared.

Bit 5 – PER PER Synchronization Busy
This bit is cleared when the synchronization of PER between the clock domains is complete.

This bit is set when the synchronization of PER between clock domains is started.

This bit is also set when the PER is written, and cleared on update condition. The bit is automatically cleared when the STATUS.PERBUF bit is cleared.

Bit 4 – COUNT COUNT Synchronization Busy
This bit is cleared when the synchronization of COUNT between the clock domains is complete.

This bit is set when the synchronization of COUNT between clock domains is started.

Bit 3 – STATUS STATUS Synchronization Busy
This bit is cleared when the synchronization of STATUS between the clock domains is complete.

Downloaded from Arrow.com.

This bit is set when a '1' is written to the Capture Channel Buffer Valid status flags (STATUS.CCBUFVx) and the synchronization of STATUS between clock domains is started.

Bit 2 – CTRLB CTRLB Synchronization Busy
This bit is cleared when the synchronization of CTRLB between the clock domains is complete.

This bit is set when the synchronization of CTRLB between clock domains is started.

Bit 1 – ENABLE ENABLE Synchronization Busy
This bit is cleared when the synchronization of ENABLE bit between the clock domains is complete.

This bit is set when the synchronization of ENABLE bit between clock domains is started.

Bit 0 – SWRST SWRST Synchronization Busy
This bit is cleared when the synchronization of SWRST bit between the clock domains is complete.

This bit is set when the synchronization of SWRST bit between clock domains is started.

Downloaded from Arrow.com.

38.7.2.13 Counter Value, 16-bit Mode

Name: COUNT
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Note: Prior to any read access, this register must be synchronized by user by writing the according TC Command value to the Control B Set register (CTRLBSET.CMD=READSYNC).

Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – COUNT[15:0] Counter Value
These bits contain the current counter value.

Downloaded from Arrow.com.

38.7.2.14 Period Value, 16-bit Mode

Name: PER
Offset: 0x1A
Reset: 0xFFFF
Property: Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	PER[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PER[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bits 15:0 – PER[15:0] Period Value
These bits hold the value of the Period Buffer register PERBUF. The value is copied to PER register on UPDATE condition.

Downloaded from Arrow.com.

38.7.2.15 Channel x Compare/Capture Value, 16-bit Mode

Name: CCx
Offset: 0x1C + x*0x02 [x=0..1]
Reset: 0x0000
Property: Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	CC[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – CC[15:0] Channel x Compare/Capture Value
These bits contain the compare/capture value in 16-bit TC mode. In Match frequency (MFRQ) or Match PWM (MPWM) waveform operation (WAVE.WAVEGEN), the CC0 register is used as a period register.

Downloaded from Arrow.com.

38.7.2.16 Period Buffer Value, 16-bit Mode

Name: PERBUF
Offset: 0x2E
Reset: 0xFFFF
Property: Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	PERBUF[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PERBUF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bits 15:0 – PERBUF[15:0] Period Buffer Value
These bits hold the value of the period buffer register. The value is copied to PER register on UPDATE condition.

Downloaded from Arrow.com.

38.7.2.17 Channel x Compare Buffer Value, 16-bit Mode

Name: CCBUFx
Offset: 0x30 + x*0x02 [x=0..1]
Reset: 0x0000
Property: Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	CCBUF[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CCBUF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – CCBUF[15:0] Channel x Compare Buffer Value
These bits hold the value of the Channel x Compare Buffer Value. When the buffer valid flag is '1' and double buffering is enabled (CTRLBCLR.LUPD=1), the data from buffer registers will be copied into the corresponding CCx register under UPDATE condition (CTRLBSET.CMD=0x3), including the software update command.

Downloaded from Arrow.com.

38.7.3 Register Summary - 32-bit Mode

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST
		15:8					ALOCK	PRESCALER[2:0]		
		23:16			COPEN1	COPEN0			CAPTEN1	CAPTEN0
		31:24								
0x04	CTRLBCLR	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x05	CTRLBSET	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x06	EVCTRL	7:0			TCEI	TCINV		EVACT[2:0]		
		15:8			MCEOx	MCEOx				OVFEO
0x08	INTENCLR	7:0				MCx			ERR	OVF
0x09	INTENSET	7:0				MCx			ERR	OVF
0x0A	INTFLAG	7:0				MCx			ERR	OVF
0x0B	STATUS	7:0				CCBUFVx	PERBUFV		SLAVE	STOP
0x0C	WAVE	7:0							WAVEGEN[1:0]	
0x0D	DRVCTRL	7:0								INVENx
0x0E	Reserved									
0x0F	DBGCTRL	7:0								DBGRUN
0x10	SYNCBUSY	7:0		CCx	PER	COUNT	STATUS	CTRLB	ENABLE	SWRST
		15:8								
		23:16								
		31:24								
0x14	COUNT	7:0	COUNT[7:0]							
		15:8	COUNT[15:8]							
		23:16	COUNT[23:16]							
		31:24	COUNT[31:24]							
0x18 ... 0x19	Reserved									
0x1A	PER	7:0	PER[7:0]							
		15:8	PER[15:8]							
		23:16	PER[23:16]							
		31:24	PER[31:24]							
0x1C	CC0	7:0	CC[7:0]							
		15:8	CC[15:8]							
		23:16	CC[23:16]							
		31:24	CC[31:24]							
0x20	CC1	7:0	CC[7:0]							
		15:8	CC[15:8]							
		23:16	CC[23:16]							
		31:24	CC[31:24]							
0x24 ... 0x2B	Reserved									
0x2C	PERBUF	7:0	PERBUF[7:0]							
		15:8	PERBUF[15:8]							

Downloaded from Arrow.com.

Offset	Name	Bit Pos.								
		23:16	PERBUF[23:16]							
		31:24	PERBUF[31:24]							
0x30	CCBUF0	7:0	CCBUF[7:0]							
		15:8	CCBUF[15:8]							
		23:16	CCBUF[23:16]							
		31:24	CCBUF[31:24]							
0x34	CCBUF1	7:0	CCBUF[7:0]							
		15:8	CCBUF[15:8]							
		23:16	CCBUF[23:16]							
		31:24	CCBUF[31:24]							

Downloaded from Arrow.com.

38.7.3.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			COPEN1	COPEN0			CAPTEN1	CAPTEN0
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	15	14	13	12	11	10	9	8
					ALOCK	PRESCALER[2:0]		
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
Reset	0	0	0	0	0	0	0	0

Bits 20, 21 – COPENx Capture On Pin x Enable
Bit x of COPEN[1:0] selects the trigger source for capture operation, either events or I/O pin input.

Value	Description
0	Event from Event System is selected as trigger source for capture operation on channel x.
1	I/O pin is selected as trigger source for capture operation on channel x.

Bits 16, 17 – CAPTENx Capture Channel x Enable
Bit x of CAPTEN[1:0] selects whether channel x is a capture or a compare channel.

These bits are not synchronized.

Value	Description
0	CAPTEN disables capture on channel x.
1	CAPTEN enables capture on channel x.

Bit 11 – ALOCK Auto Lock
When this bit is set, Lock bit update (LUPD) is set to '1' on each overflow/underflow or re-trigger event.

This bit is not synchronized.

Downloaded from Arrow.com.

Value	Description
0	The LUPD bit is not affected on overflow/underflow, and re-trigger event.
1	The LUPD bit is set on each overflow/underflow or re-trigger event.

Bits 10:8 – PRESCALER[2:0] Prescaler
These bits select the counter prescaler factor.

These bits are not synchronized.

Value	Name	Description
0x0	DIV1	Prescaler: GCLK_TC
0x1	DIV2	Prescaler: GCLK_TC/2
0x2	DIV4	Prescaler: GCLK_TC/4
0x3	DIV8	Prescaler: GCLK_TC/8
0x4	DIV16	Prescaler: GCLK_TC/16
0x5	DIV64	Prescaler: GCLK_TC/64
0x6	DIV256	Prescaler: GCLK_TC/256
0x7	DIV1024	Prescaler: GCLK_TC/1024

Bit 7 – ONDEMAND Clock On Demand
This bit selects the clock requirements when the TC is stopped.

In standby mode, if the Run in Standby bit (CTRLA.RUNSTDBY) is '0', ONDEMAND is forced to '0'.

This bit is not synchronized.

Value	Description
0	The On Demand is disabled. If On Demand is disabled, the TC will continue to request the clock when its operation is stopped (STATUS.STOP=1).
1	The On Demand is enabled. When On Demand is enabled, the stopped TC will not request the clock. The clock is requested when a software re-trigger command is applied or when an event with start/re-trigger action is detected.

Bit 6 – RUNSTDBY Run in Standby
This bit is used to keep the TC running in standby mode.

This bit is not synchronized.

Value	Description
0	The TC is halted in standby.
1	The TC continues to run in standby.

Bits 5:4 – PRESCSYNC[1:0] Prescaler and Counter Synchronization
These bits select whether the counter should wrap around on the next GCLK_TCx clock or the next prescaled GCLK_TCx clock. It also makes it possible to reset the prescaler.

These bits are not synchronized.

Value	Name	Description
0x0	GCLK	Reload or reset the counter on next generic clock
0x1	PRESC	Reload or reset the counter on next prescaler clock
0x2	RESYNC	Reload or reset the counter on next generic clock. Reset the prescaler counter
0x3	-	Reserved

Downloaded from Arrow.com.

Bits 3:2 – MODE[1:0] Timer Counter Mode
These bits select the counter mode.

These bits are not synchronized.

Value	Name	Description
0x0	COUNT16	Counter in 16-bit mode
0x1	COUNT8	Counter in 8-bit mode
0x2	COUNT32	Counter in 32-bit mode
0x3	-	Reserved

Bit 1 – ENABLE Enable
Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately, and the ENABLE Synchronization Busy bit in the SYNCBUSY register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

This bit is not enable protected.

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled.

Bit 0 – SWRST Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the TC, except DBGCTRL, to their initial state, and the TC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence; all other writes in the same write-operation will be discarded.

Downloaded from Arrow.com.

38.7.3.2 Control B Clear

Name: CTRLBCLR
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection, Read-Synchronized, Write-Synchronized

This register allows the user to clear bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Set register (CTRLBSET).

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]					ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bits 7:5 – CMD[2:0] Command
These bits are used for software control of the TC. The commands are executed on the next prescaled GCLK_TC clock cycle. When a command has been executed, the CMD bit group will be read back as zero.

Writing 0x0 to these bits has no effect.

Writing a '1' to any of these bits will clear the pending command.

Bit 2 – ONESHOT One-Shot on Counter
This bit controls one-shot operation of the TC.

Writing a '0' to this bit has no effect

Writing a '1' to this bit will disable one-shot operation.

Value	Description
0	The TC will wrap around and continue counting on an overflow/underflow condition.
1	The TC will wrap around and stop on the next underflow/overflow condition.

Bit 1 – LUPD Lock Update
This bit controls the update operation of the TC buffered registers.

When CTRLB.LUPD is set, no any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.

This bit has no effect when input capture operation is enabled.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the LUPD bit.

Value	Description
0	The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on hardware update condition.
1	The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers on hardware update condition.

Downloaded from Arrow.com.

Bit 0 – DIR Counter Direction
This bit is used to change the direction of the counter.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the bit and make the counter count up.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

Downloaded from Arrow.com.

38.7.3.3 Control B Set

Name: CTRLBSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection, Read-synchronized, Write-Synchronized

This register allows the user to set bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Clear register (CTRLBCLR).

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]					ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bits 7:5 – CMD[2:0] Command
These bits are used for software control of the TC. The commands are executed on the next prescaled GCLK_TC clock cycle. When a command has been executed, the CMD bit group will be read back as zero.

Writing 0x0 to these bits has no effect.

Writing a value different from 0x0 to these bits will issue a command for execution.

Value	Name	Description
0x0	NONE	No action
0x1	RETRIGGER	Force a start, restart or retrigger
0x2	STOP	Force a stop
0x3	UPDATE	Force update of double buffered registers
0x4	READSYNC	Force a read synchronization of COUNT

Bit 2 – ONESHOT One-Shot on Counter
This bit controls one-shot operation of the TC.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will enable one-shot operation.

Value	Description
0	The TC will wrap around and continue counting on an overflow/underflow condition.
1	The TC will wrap around and stop on the next underflow/overflow condition.

Bit 1 – LUPD Lock Update
This bit controls the update operation of the TC buffered registers.

When CTRLB.LUPD is set, no any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the LUPD bit.

Downloaded from Arrow.com.

This bit has no effect when input capture operation is enabled.

Value	Description
0	The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on hardware update condition.
1	The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers on hardware update condition.

Bit 0 – DIR Counter Direction
This bit is used to change the direction of the counter.

Writing a '0' to this bit has no effect

Writing a '1' to this bit will clear the bit and make the counter count up.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

Downloaded from Arrow.com.

38.7.3.4 Event Control

Name: EVCTRL
Offset: 0x06
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
			MCEOx	MCEOx				OVFEO
Access			R/W	R/W				R/W
Reset			0	0				0

Bit	7	6	5	4	3	2	1	0
			TCEI	TCINV		EVACT[2:0]		
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bits 13,12 – MCEOx Match or Capture Channel x Event Output Enable [x = 1..0]
These bits enable the generation of an event for every match or capture on channel x.

Value	Description
0	Match/Capture event on channel x is disabled and will not be generated.
1	Match/Capture event on channel x is enabled and will be generated for every compare/capture.

Bit 8 – OVFEO Overflow/Underflow Event Output Enable
This bit enables the Overflow/Underflow event. When enabled, an event will be generated when the counter overflows/underflows.

Value	Description
0	Overflow/Underflow event is disabled and will not be generated.
1	Overflow/Underflow event is enabled and will be generated for every counter overflow/underflow.

Bit 5 – TCEI TC Event Enable
This bit is used to enable asynchronous input events to the TC.

Value	Description
0	Incoming events are disabled.
1	Incoming events are enabled.

Bit 4 – TCINV TC Inverted Event Input Polarity
This bit inverts the asynchronous input event source.

Value	Description
0	Input event source is not inverted.
1	Input event source is inverted.

Bits 2:0 – EVACT[2:0] Event Action
These bits define the event action the TC will perform on an event.

Downloaded from Arrow.com.

Value	Name	Description
0x0	OFF	Event action disabled
0x1	RETRIGGER	Start, restart or retrigger TC on event
0x2	COUNT	Count on event
0x3	START	Start TC on event
0x4	STAMP	Time stamp capture
0x5	PPW	Period captured in CC0, pulse width in CC1
0x6	PWP	Period captured in CC1, pulse width in CC0
0x7	PW	Pulse width capture

Downloaded from Arrow.com.

38.7.3.5 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x08
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Bit	7	6	5	4	3	2	1	0
				MCx			ERR	OVF
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – MCx Match or Capture Channel x Interrupt Enable
Writing a '0' to these bits has no effect.

Writing a '1' to MCx will clear the corresponding Match or Capture Channel x Interrupt Enable bit, which disables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bit 1 – ERR Error Interrupt Disable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 0 – OVF Overflow Interrupt Disable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Downloaded from Arrow.com.

38.7.3.6 Interrupt Enable Set

Name: INTENSET
Offset: 0x09
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Bit	7	6	5	4	3	2	1	0
				MCx			ERR	OVF
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – MCx Match or Capture Channel x Interrupt Enable
Writing a '0' to these bits has no effect.

Writing a '1' to MCx will set the corresponding Match or Capture Channel x Interrupt Enable bit, which enables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bit 1 – ERR Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 0 – OVF Overflow Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Downloaded from Arrow.com.

38.7.3.7 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x0A
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
				MCx			ERR	OVF
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – MCx Match or Capture Channel x
This flag is set on a comparison match, or when the corresponding CCx register contains a valid capture value. This flag is set on the next CLK_TC_CNT cycle, and will generate an interrupt request if the corresponding Match or Capture Channel x Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.MCx) is '1'.

Writing a '0' to one of these bits has no effect.

Writing a '1' to one of these bits will clear the corresponding Match or Capture Channel x interrupt flag

In capture operation, this flag is automatically cleared when CCx register is read.

Bit 1 – ERR Error Interrupt Flag
This flag is set when a new capture occurs on a channel while the corresponding Match or Capture Channel x interrupt flag is set, in which case there is nowhere to store the new capture.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Error interrupt flag.

Bit 0 – OVF Overflow Interrupt Flag
This flag is set on the next CLK_TC_CNT cycle after an overflow condition occurs, and will generate an interrupt request if INTENCLR.OVF or INTENSET.OVF is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Overflow interrupt flag.

Downloaded from Arrow.com.

38.7.3.8 Status

Name: STATUS
Offset: 0x0B
Reset: 0x01
Property: Read-Synchronized

Bit	7	6	5	4	3	2	1	0
				CCBUFVx	PERBUFV		SLAVE	STOP
Access				R/W	R/W		R	R
Reset				0	0		0	1

Bit 4 – CCBUFVx Channel x Compare or Capture Buffer Valid
For a compare channel x, the bit x is set when a new value is written to the corresponding CCBUFx register.

The bit x is cleared by writing a '1' to it when CTRLB.LUPD is set, or it is cleared automatically by hardware on UPDATE condition.

For a capture channel x, the bit x is set when a valid capture value is stored in the CCBUFx register. The bit x is cleared automatically when the CCx register is read.

Bit 3 – PERBUFV Period Buffer Valid
This bit is set when a new value is written to the PERBUF register. The bit is cleared by writing '1' to the corresponding location when CTRLB.LUPD is set, or automatically cleared by hardware on UPDATE condition. This bit is available only in 8-bit mode and will always read zero in 16- and 32-bit modes.

Bit 1 – SLAVE Slave Status Flag
This bit is only available in 32-bit mode on the slave TC (i.e., TC1 and/or TC3). The bit is set when the associated master TC (TC0 and TC2, respectively) is set to run in 32-bit mode.

Bit 0 – STOP Stop Status Flag
This bit is set when the TC is disabled, on a Stop command, or on an overflow/underflow condition when the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT) is '1'.

Value	Description
0	Counter is running.
1	Counter is stopped.

Downloaded from Arrow.com.

38.7.3.9 Waveform Generation Control

Name: WAVE
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
							WAVEGEN[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 1:0 – WAVEGEN[1:0] Waveform Generation Mode
These bits select the waveform generation operation. They affect the top value, as shown in 38.6.2.6.1 Waveform Output Operations. They also control whether frequency or PWM waveform generation should be used. The waveform generation operations are explained in 38.6.2.6.1 Waveform Output Operations.

These bits are not synchronized.

Value	Name	Operation	Top Value	Output Waveform on Match	Output Waveform on Wraparound
0x0	NFRQ	Normal frequency	PER[1] / Max	Toggle	No action
0x1	MFRQ	Match frequency	CC0	Toggle	No action
0x2	NPWM	Normal PWM	PER[1] / Max	Set	Clear
0x3	MPWM	Match PWM	CC0	Set	Clear

1) This depends on the TC mode: In 8-bit mode, the top value is the Period Value register (PER). In 16- and 32-bit mode it is the respective MAX value.

Downloaded from Arrow.com.

38.7.3.10 Driver Control

Name: DRVCTRL
Offset: 0x0D
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
								INVENx
Access								R/W
Reset								0

Bit 0 – INVENx Output Waveform x Invert Enable
Bit x of INVEN[1:0] selects inversion of the output or capture trigger input of channel x.

Value	Description
0	Disable inversion of the WO[x] output and IO input pin.
1	Enable inversion of the WO[x] output and IO input pin.

Downloaded from Arrow.com.

38.7.3.11 Debug Control

Name: DBGCTRL
Offset: 0x0F
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN Run in Debug Mode
This bit is not affected by a software Reset, and should not be changed by software while the TC is enabled.

Value	Description
0	The TC is halted when the device is halted in debug mode.
1	The TC continues normal operation when the device is halted in debug mode.

Downloaded from Arrow.com.

38.7.3.12 Synchronization Busy

Name: SYNCBUSY
Offset: 0x10
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		CCx	PER	COUNT	STATUS	CTRLB	ENABLE	SWRST
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit 6 – CCx Compare/Capture Channel x Synchronization Busy
For details on CC channels number, refer to each TC feature list.

This bit is set when the synchronization of CCx between clock domains is started.

This bit is also set when the CCBUFx is written, and cleared on update condition. The bit is automatically cleared when the STATUS.CCBUFx bit is cleared.

Bit 5 – PER PER Synchronization Busy
This bit is cleared when the synchronization of PER between the clock domains is complete.

This bit is set when the synchronization of PER between clock domains is started.

This bit is also set when the PER is written, and cleared on update condition. The bit is automatically cleared when the STATUS.PERBUF bit is cleared.

Bit 4 – COUNT COUNT Synchronization Busy
This bit is cleared when the synchronization of COUNT between the clock domains is complete.

This bit is set when the synchronization of COUNT between clock domains is started.

Bit 3 – STATUS STATUS Synchronization Busy
This bit is cleared when the synchronization of STATUS between the clock domains is complete.

Downloaded from Arrow.com.

This bit is set when a '1' is written to the Capture Channel Buffer Valid status flags (STATUS.CCBUFVx) and the synchronization of STATUS between clock domains is started.

Bit 2 – CTRLB CTRLB Synchronization Busy
This bit is cleared when the synchronization of CTRLB between the clock domains is complete.

This bit is set when the synchronization of CTRLB between clock domains is started.

Bit 1 – ENABLE ENABLE Synchronization Busy
This bit is cleared when the synchronization of ENABLE bit between the clock domains is complete.

This bit is set when the synchronization of ENABLE bit between clock domains is started.

Bit 0 – SWRST SWRST Synchronization Busy
This bit is cleared when the synchronization of SWRST bit between the clock domains is complete.

This bit is set when the synchronization of SWRST bit between clock domains is started.

Downloaded from Arrow.com.

38.7.3.13 Counter Value, 32-bit Mode

Name: COUNT
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Note: Prior to any read access, this register must be synchronized by user by writing the according TC Command value to the Control B Set register (CTRLBSET.CMD=READSYNC).

Bit	31	30	29	28	27	26	25	24
	COUNT[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	COUNT[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – COUNT[31:0] Counter Value
These bits contain the current counter value.

Downloaded from Arrow.com.

38.7.3.14 Period Value, 32-bit Mode

Name: PER
Offset: 0x1A
Reset: 0xFFFFFFFF
Property: Write-Synchronized

Bit	31	30	29	28	27	26	25	24
	PER[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PER[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PER[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PER[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bits 31:0 – PER[31:0] Period Value
These bits hold the value of the Period Buffer register PERBUF. The value is copied to PER register on UPDATE condition.

Downloaded from Arrow.com.

38.7.3.15 Channel x Compare/Capture Value, 32-bit Mode

Name: CCx
Offset: 0x1C + x*0x04 [x=0..1]
Reset: 0x00000000
Property: Write-Synchronized

Bit	31	30	29	28	27	26	25	24
	CC[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CC[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CC[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CC[31:0] Channel x Compare/Capture Value
These bits contain the compare/capture value in 32-bit TC mode. In Match frequency (MFRQ) or Match PWM (MPWM) waveform operation (WAVE.WAVEGEN), the CC0 register is used as a period register.

Downloaded from Arrow.com.

38.7.3.16 Period Buffer Value, 32-bit Mode

Name: PERBUF
Offset: 0x2C
Reset: 0xFFFFFFFF
Property: Write-Synchronized

Bit	31	30	29	28	27	26	25	24
	PERBUF[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PERBUF[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PERBUF[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PERBUF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bits 31:0 – PERBUF[31:0] Period Buffer Value
These bits hold the value of the period buffer register. The value is copied to PER register on UPDATE condition.

Downloaded from Arrow.com.

38.7.3.17 Channel x Compare Buffer Value, 32-bit Mode

Name: CCBUFx
Offset: 0x30 + x*0x04 [x=0..1]
Reset: 0x00000000
Property: Write-Synchronized

Bit	31	30	29	28	27	26	25	24
	CCBUF[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CCBUF[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CCBUF[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CCBUF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CCBUF[31:0] Channel x Compare Buffer Value
These bits hold the value of the Channel x Compare Buffer Value. When the buffer valid flag is '1' and double buffering is enabled (CTRLBCLR.LUPD=1), the data from buffer registers will be copied into the corresponding CCx register under UPDATE condition (CTRLBSET.CMD=0x3), including the software update command.

Downloaded from Arrow.com.

39. TRNG – True Random Number Generator

39.1 Overview

The True Random Number Generator (TRNG) generates unpredictable random numbers that are not generated by an algorithm. It passes the American NIST Special Publication 800-22 and Diehard Random Tests Suites.

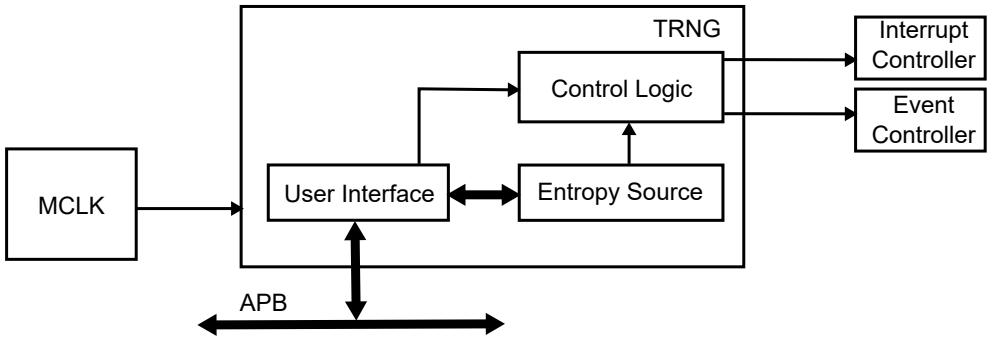
The TRNG may be used as an entropy source for seeding an NIST approved DRNG (Deterministic RNG) as required by FIPS PUB 140-2 and 140-3.

39.2 Features

- Passed NIST Special Publication 800-22 Tests Suite
- Passed Diehard Random Tests Suite
- May be used as Entropy Source for seeding an NIST approved DRNG (Deterministic RNG) as required by FIPS PUB 140-2 and 140-3
- Provides a 32-bit random number every 84 clock cycles

39.3 Block Diagram

Figure 39-1. TRNG Block Diagram.



39.4 Signal Description

Not applicable.

39.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

39.5.1 I/O Lines

Not applicable.

39.5.2 Power Management

The functioning of TRNG depends on the sleep mode of device.

Downloaded from Arrow.com.

The TRNG interrupts can be used to wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes.

Related Links

22. PM – Power Manager

39.6.5 Sleep Mode Operation

39.5.3 Clocks

The TRNG bus clock (CLK_TRNG_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_TRNG_APB can be found in *Peripheral Clock Masking*.

Related Links

19.6.2.6 Peripheral Clock Masking

39.5.4 DMA

Not applicable.

39.5.5 Interrupts

The interrupt request line is connected to the interrupt controller. Using the TRNG interrupt(s) requires the interrupt controller to be configured first. Refer to NVIC - Nested Interrupt *Nested Vector Interrupt Controller* for details.

39.5.6 Events

TRNG can generate Events that are used by the Event System (EVSYS) and EVSYS users.

TRNG cannot use any Events from other peripherals, as it is not an Event User.

Related Links

33. EVSYS – Event System

39.5.7 Debug Operation

When the CPU is halted in debug mode the TRNG continues normal operation. If the TRNG is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

39.5.8 Register Access Protection

All registers with write-access are optionally write-protected by the Peripheral Access Controller (PAC), except the following register:

Interrupt Flag Status and Clear (INTFLAG) register

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

39.5.9 SAM L11 TrustZone Specific Register Access Protection

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted

Downloaded from Arrow.com.

– Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

39.5.10 Analog Connections

Not applicable.

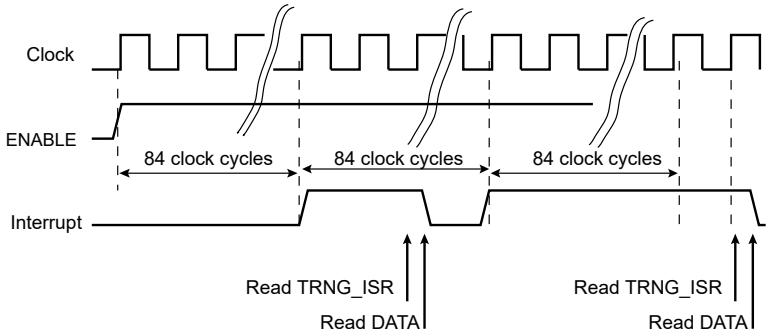
39.6 Functional Description

39.6.1 Principle of Operation

When the TRNG is enabled, the peripheral starts providing new 32-bit random numbers every 84 CLK_TRNG_APB clock cycles.

The TRNG can be configured to generate an interrupt or event when a new random number is available.

Figure 39-2. TRNG Data Generation Sequence



39.6.2 Basic Operation

39.6.2.1 Initialization

To operate the TRNG, do the following:

- Configure the clock source for CLK_TRNG_APB in the Main Clock Controller (MCLK) and enable the clock by writing a ‘1’ to the TRNG bit in the APB Mask register of the MCLK.
- Optional: Enable the output event by writing a ‘1’ to the EVCTRL.DATARDYEO bit.
- Optional: Enable the TRNG to Run in Standby sleep mode by writing a ‘1’ to CTRLA.RUNSTDBY.
- Enable the TRNG operation by writing a ‘1’ to CTRLA.ENABLE.

The following register is enable-protected, meaning that it can only be written when the TRNG is disabled (CTRLA.ENABLE is zero):

- Event Control register (EVCTRL)

Enable-protection is denoted by the Enable-Protected property in the register description.

39.6.2.2 Enabling, Disabling and Resetting

The TRNG is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The TRNG is disabled by writing a zero to CTRLA.ENABLE.

39.6.3 Interrupts

The TRNG has the following interrupt source:

- Data Ready (DATARDY): Indicates that a new random number is available in the DATA register and ready to be read.

Downloaded from Arrow.com.

This interrupt is a synchronous wake-up source. See *Sleep Mode Controller* for details.

The interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.DATARDY) is set to ‘1’ when the interrupt condition occurs. The interrupt can be enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET.DATARDY), and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, or the interrupt is disabled. See 39.8.5 INTFLAG for details on how to clear interrupt flags.

Note that interrupts must be globally enabled for interrupt requests to be generated.

Related Links

22.6.3.3 Sleep Mode Controller

39.6.4 Events

The TRNG can generate the following output event:

- Data Ready (DATARDY): Generated when a new random number is available in the DATA register.

Writing '1' to the Data Ready Event Output bit in the Event Control Register (EVCTRL.DATARDYEO) enables the DTARDY event. Writing a '0' to this bit disables the corresponding output event. Refer to *EVSYS – Event System* for details on configuring the Event System.

Related Links

33. EVSYS – Event System

39.6.5 Sleep Mode Operation

The Run in Standby bit in Control A register (CTRLA.RUNSTDBY) controls the behavior of the TRNG during standby sleep mode:

When this bit is '0', the TRNG is disabled during sleep, but maintains its current configuration.

When this bit is '1', the TRNG continues to operate during sleep and any enabled TRNG interrupt source can wake up the CPU.

39.6.6 Synchronization

Not applicable.

Downloaded from Arrow.com.

39.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0		RUNSTDBY					ENABLE	
0x01 ... 0x03	Reserved									
0x04	EVCTRL	7:0								DATARDYEO
0x05 ... 0x07	Reserved									
0x08	INTENCLR	7:0								DATARDY
0x09	INTENSET	7:0								DATARDY
0x0A	INTFLAG	7:0								DATARDY
0x0B ... 0x1F	Reserved									
0x20	DATA	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							

39.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

Refer to *PAC - Peripheral Access Controller* and 39.6.6 Synchronizationfor details.

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

Related Links

Downloaded from Arrow.com.

15. PAC - Peripheral Access Controller

Downloaded from Arrow.com.

39.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY					ENABLE	
Access		R/W					R/W	
Reset		0					0	

Bit 6 – RUNSTDBY Run in Standby
This bit controls how the ADC behaves during standby sleep mode:

Value	Description
0	The TRNG is halted during standby sleep mode.
1	The TRNG is not stopped in standby sleep mode.

Bit 1 – ENABLE Enable

Value	Description
0	The TRNG is disabled.
1	The TRNG is enabled.

Downloaded from Arrow.com.

39.8.2 Event Control

Name: EVCTRL
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
								DATARDYEO
Access								R/W
Reset								0

Bit 0 – DATARDYEO Data Ready Event Output
This bit indicates whether the Data Ready event output is enabled and whether an output event will be generated when a new random value is ready.

Value	Description
0	Data Ready event output is disabled and an event will not be generated.
1	Data Ready event output is enabled and an event will be generated.

Downloaded from Arrow.com.

39.8.3 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x08
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Bit	7	6	5	4	3	2	1	0
								DATARDY
Access								R/W
Reset								0

Bit 0 – DATARDY Data Ready Interrupt Enable
Writing a '1' to this bit will clear the Data Ready Interrupt Enable bit, which disables the corresponding interrupt request.

Value	Description
0	The DATARDY interrupt is disabled.
1	The DATARDY interrupt is enabled.

Downloaded from Arrow.com.

39.8.4 Interrupt Enable Set

Name: INTENSET
Offset: 0x09
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Bit	7	6	5	4	3	2	1	0
								DATARDY
Access								R/W
Reset								0

Bit 0 – DATARDY Data Ready Interrupt Enable
Writing a '1' to this bit will set the Data Ready Interrupt Enable bit, which enables the corresponding interrupt request.

Value	Description
0	The DATARDY interrupt is disabled.
1	The DATARDY interrupt is enabled.

Downloaded from Arrow.com.

39.8.5 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x0A
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
								DATARDY
Access								R/W
Reset								0

Bit 0 – DATARDY Data Ready
This flag is set when a new random value is generated, and an interrupt will be generated if INTENCLR/SET.DATARDY=1.

This flag is cleared by writing a '1' to the flag or by reading the DATA register.

Writing a '0' to this bit has no effect.

Downloaded from Arrow.com.

39.8.6 Output Data

Name: DATA
Offset: 0x20
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0] Output Data
These bits hold the 32-bit randomly generated output data.

Downloaded from Arrow.com.

40. CCL – Configurable Custom Logic

40.1 Overview

The Configurable Custom Logic (CCL) is a programmable logic peripheral which can be connected to the device pins, to events, or to other internal peripherals. This allows the user to eliminate logic gates for simple glue logic functions on the PCB.

Each LookUp Table (LUT) consists of three inputs, a truth table, an optional synchronizer/filter, and an optional edge detector. Each LUT can generate an output as a user programmable logic expression with three inputs. Inputs can be individually masked.

The output can be combinatorially generated from the inputs, and can be filtered to remove spikes. Optional sequential logic can be used. The inputs of the sequential module are individually controlled by two independent, adjacent LUT (LUT0/LUT1, LUT2/LUT3 etc.) outputs, enabling complex waveform generation.

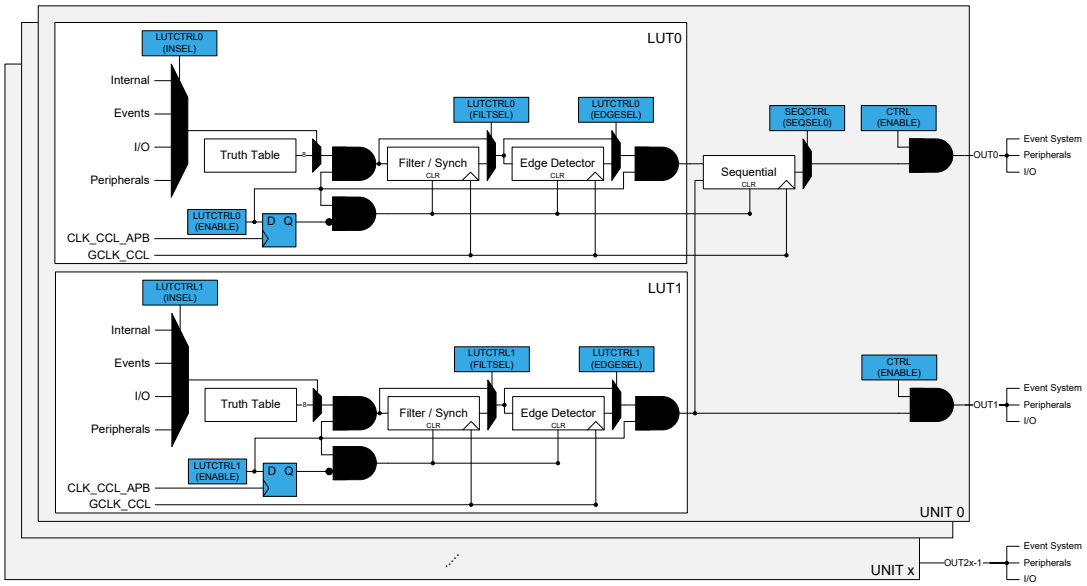
40.2 Features

- Glue logic for general purpose PCB design
- Up to 2 programmable LookUp Tables (LUTs)
- Combinatorial logic functions:
  AND, NAND, OR, NOR, XOR, XNOR, NOT
- Sequential logic functions:
  Gated D Flip-Flop, JK Flip-Flop, gated D Latch, RS Latch
- Flexible LUT inputs selection:
  - I/Os
  - Events
  - Internal peripherals
  - Subsequent LUT output
- Output can be connected to the I/O pins or the Event System
- Optional synchronizer, filter, or edge detector available on each LUT output

Downloaded from Arrow.com.

40.3 Block Diagram

Figure 40-1. Configurable Custom Logic



40.4 Signal Description

Pin Name	Type	Description
OUT[n:0]	Digital output	Output from lookup table
IN[3n+2:0]	Digital input	Input to lookup table

1. n is the number of CCL groups.

Refer to *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

40.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

40.5.1 I/O Lines

The CCL can take inputs and generate output through I/O pins. For this to function properly, the I/O pins must be configured to be used by a Look-up Table (LUT).

Related Links

32. PORT - I/O Pin Controller

40.5.2 Power Management

This peripheral can continue to operate in any sleep mode where its source clock is running. Events connected to the event system can trigger other operations in the system without exiting sleep modes.

Related Links

22. PM – Power Manager

Downloaded from Arrow.com.

40.5.3 Clocks

The CCL bus clock (CLK_CCL_APB) can be enabled and disabled in the Main Clock module, MCLK (see *MCLK - Main Clock*), and the default state of CLK_CCL_APB can be found in *Peripheral Clock Masking*.

A generic clock (GCLK_CCL) is optionally required to clock the CCL. This clock must be configured and enabled in the Generic Clock Controller (GCLK) before using input events, filter, edge detection or sequential logic. GCLK_CCL is required when input events, a filter, an edge detector, or a sequential sub-module is enabled. Refer to *GCLK - Generic Clock Controller* for details.

This generic clock is asynchronous to the user interface clock (CLK_CCL_APB).

Related Links

19. MCLK – Main Clock

19.6.2.6 Peripheral Clock Masking

18. GCLK - Generic Clock Controller

40.5.4 DMA

Not applicable.

40.5.5 Interrupts

Not applicable.

40.5.6 Events

The CCL can use events from other peripherals and generate events that can be used by other peripherals. For this feature to function, the Events have to be configured properly. Refer to the Related Links below for more information about the Event Users and Event Generators.

Related Links

33. EVSYS – Event System

40.5.7 Debug Operation

When the CPU is halted in Debug mode the CCL continues normal operation. However, the CCL cannot be halted when the CPU is halted in Debug mode. If the CCL is configured in a way that requires it to be periodically serviced by the CPU, improper operation or data loss may result during debugging.

40.5.8 Register Access Protection

All registers with write-access can be write-protected optionally by the peripheral access controller (PAC). Refer to *PAC - Peripheral Access Controller* for details.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

Related Links

15. PAC - Peripheral Access Controller

40.5.9 SAM L11 TrustZone Specific Register Access Protection

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted

Downloaded from Arrow.com.

- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

40.5.10 Analog Connections

Not applicable.

40.6 Functional Description

40.6.1 Principle of Operation

Configurable Custom Logic (CCL) is a programmable logic block that can use the device port pins, internal peripherals, and the internal Event System as both input and output channels. The CCL can serve as glue logic between the device and external devices. The CCL can eliminate the need for external logic component and can also help the designer overcome challenging real-time constrains by combining core independent peripherals in clever ways to handle the most time critical parts of the application independent of the CPU.

40.6.2 Operation

40.6.2.1 Initialization

The following bits are enable-protected, meaning that they can only be written when the corresponding even LUT is disabled (LUTCTRLx.ENABLE=0):

- Sequential Selection bits in the Sequential Control x (SEQCTRLx.SEQSEL) register

The following registers are enable-protected, meaning that they can only be written when the corresponding LUT is disabled (LUTCTRLx.ENABLE=0):

- LUT Control x (LUTCTRLx) register, except the ENABLE bit

Enable-protected bits in the LUTCTRLx registers can be written at the same time as LUTCTRLx.ENABLE is written to '1', but not at the same time as LUTCTRLx.ENABLE is written to '0'.

Enable-protection is denoted by the Enable-Protected property in the register description.

40.6.2.2 Enabling, Disabling, and Resetting

The CCL is enabled by writing a '1' to the Enable bit in the Control register (CTRL.ENABLE). The CCL is disabled by writing a '0' to CTRL.ENABLE.

Each LUT is enabled by writing a '1' to the Enable bit in the LUT Control x register (LUTCTRLx.ENABLE). Each LUT is disabled by writing a '0' to LUTCTRLx.ENABLE.

The CCL is reset by writing a '1' to the Software Reset bit in the Control register (CTRL.SWRST). All registers in the CCL will be reset to their initial state, and the CCL will be disabled. Refer to 40.8.1 CTRL for details.

40.6.2.3 Lookup Table Logic

The lookup table in each LUT unit can generate any logic expression OUT as a function of three inputs (IN[2:0]), as shown in Figure 40-2. One or more inputs can be masked. The truth table for the expression is defined by TRUTH bits in LUT Control x register (LUTCTRLx.TRUTH).

Downloaded from Arrow.com.

Figure 40-2. Truth Table Output Value Selection

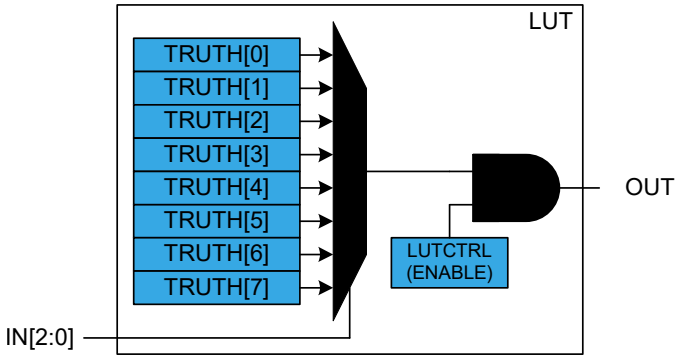


Table 40-1. Truth Table of LUT

IN[2]	IN[1]	IN[0]	OUT
0	0	0	TRUTH[0]
0	0	1	TRUTH[1]
0	1	0	TRUTH[2]
0	1	1	TRUTH[3]
1	0	0	TRUTH[4]
1	0	1	TRUTH[5]
1	1	0	TRUTH[6]
1	1	1	TRUTH[7]

40.6.2.4 Truth Table Inputs Selection

Input Overview

The inputs can be individually:

- Masked
- Driven by peripherals:
  - Analog comparator output (AC)
  - Timer/Counters waveform outputs (TC)
  - Serial Communication output transmit interface (SERCOM)
- Driven by internal events from Event System
- Driven by other CCL sub-modules

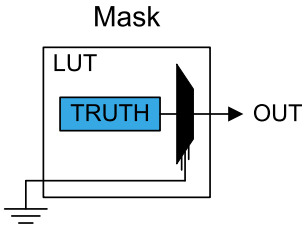
The Input Selection for each input y of LUT x is configured by writing the Input y Source Selection bit in the LUT x Control register (LUTCTRLx.INSELy).

Masked Inputs (MASK)

When a LUT input is masked (LUTCTRLx.INSELy=MASK), the corresponding TRUTH input (IN) is internally tied to zero, as shown in this figure:

Downloaded from Arrow.com.

Figure 40-3. Masked Input Selection



Internal Feedback Inputs (FEEDBACK)

When selected (LUTCTRLx.INSELy=FEEDBACK), the Sequential (SEQ) output is used as input for the corresponding LUT.

The output from an internal sequential sub-module can be used as input source for the LUT, see figure below for an example for LUT0 and LUT1. The sequential selection for each LUT follows the formula:

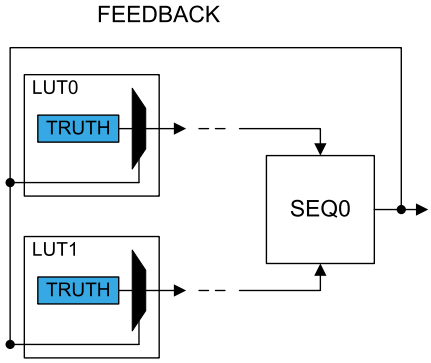
$\mathrm{IN}[2\mathrm{N}][i] = \mathrm{SEQ}[N]$

$\mathrm{IN}[2\mathrm{N}+1][i] = \mathrm{SEQ}[N]$

With *N* representing the sequencer number and *i*=0,1,2 representing the LUT input index.

For details, refer to 40.6.2.7 Sequential Logic.

Figure 40-4. Feedback Input Selection

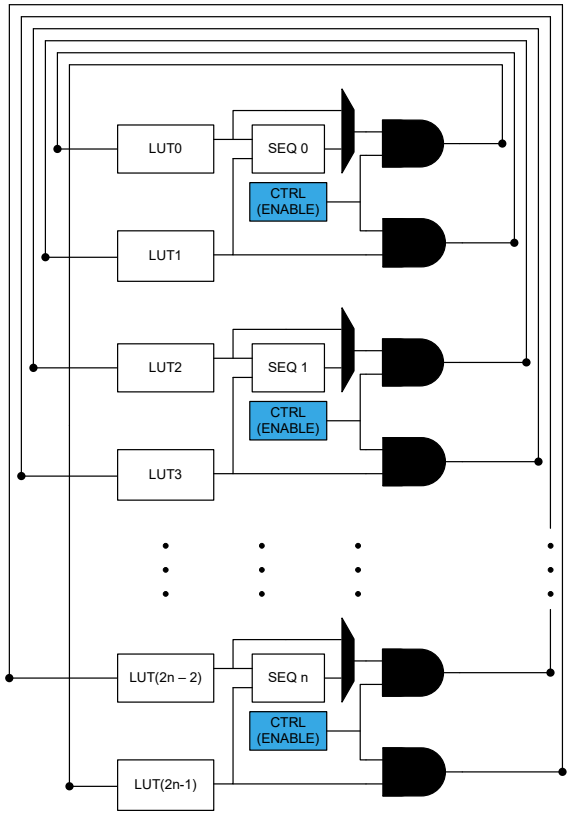


Linked LUT (LINK)

When selected (LUTCTRLx.INSELy=LINK), the subsequent LUT output is used as the LUT input (e.g., LUT2 is the input for LUT1), as shown in this figure:

Downloaded from Arrow.com.

Figure 40-5. Linked LUT Input Selection



Internal Events Inputs Selection (EVENT)

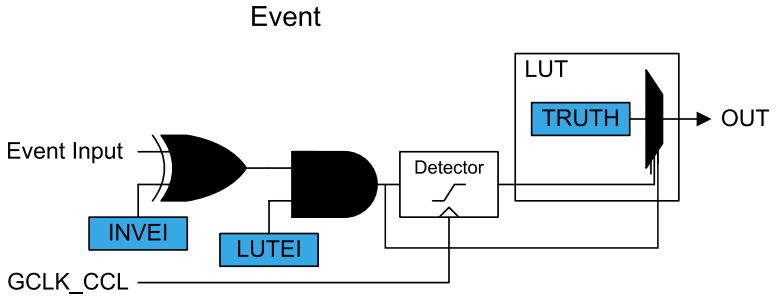
Asynchronous events from the Event System can be used as input selection, as shown in Figure 40-6. For each LUT, one event input line is available and can be selected on each LUT input. Before enabling the event selection by writing LUTCTRLx.INSELy=EVENT, the Event System must be configured first.

By default CCL includes an edge detector. When the event is received, an internal strobe is generated when a rising edge is detected. The pulse duration is one GCLK_CCL clock cycle. Writing the LUTCTRLx.INSELy=ASYNCEVENT will disable the edge detector. In this case, it is possible to combine an asynchronous event input with any other input source. This is typically useful with event levels inputs (external IO pin events, as example). The following steps ensure proper operation:

1. Enable the GCLK_CCL clock.
2. Configure the Event System to route the event asynchronously.
3. Select the event input type (LUTCTRLx.INSEL).
4. If a strobe must be generated on the event input falling edge, write a '1' to the Inverted Event Input Enable bit in LUT Control register (LUTCTRLx.INVEI) .
5. Enable the event input by writing the Event Input Enable bit in LUT Control register (LUTCTRLx.LUTEI) to '1'.

Downloaded from Arrow.com.

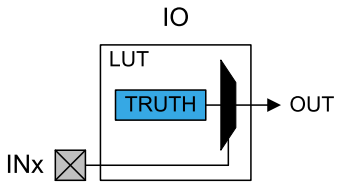
Figure 40-6. Event Input Selection



I/O Pin Inputs (IO)

When the IO pin is selected as LUT input (LUTCTRLx.INSELy=IO), the corresponding LUT input will be connected to the pin, as shown in the figure below.

Figure 40-7. I/O Pin Input Selection



Analog Comparator Inputs (AC)

The AC outputs can be used as input source for the LUT (LUTCTRLx.INSELy=AC).

The analog comparator outputs are distributed following the formula:

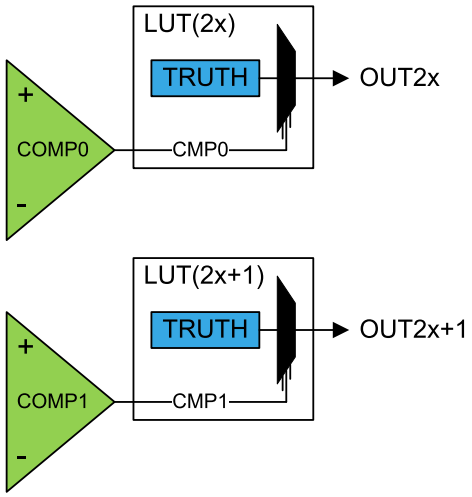
```
IN[N][i]=AC[N % ComparatorOutput_Number]
```

With *N* representing the LUT number and *i*=[0,1,2] representing the LUT input index.

Before selecting the comparator output, the AC must be configured first.

The output of comparator 0 is available on even LUTs ("LUT(2x)": LUT0, LUT2) and the comparator 1 output is available on odd LUTs ("LUT(2x+1)": LUT1, LUT3), as shown in the figure below.

Figure 40-8. AC Input Selection



Downloaded from Arrow.com.

Timer/Counter Inputs (TC)

The TC waveform output WO[0] can be used as input source for the LUT (LUTCTRLx.INSELy=TC). Only consecutive instances of the TC, i.e. TCx and the subsequent TC(x+1), are available as default and alternative TC selections (e.g., TC0 and TC1 are sources for LUT0, TC1 and TC2 are sources for LUT1, etc). See the figure below for an example for LUT0. More general, the Timer/Counter selection for each LUT follows the formula:

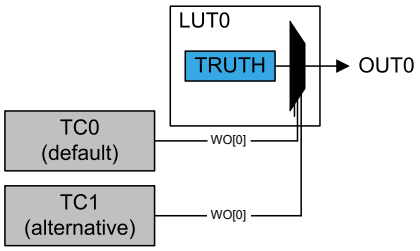
$$\mathrm{IN}[N][i] = DefaultTC[N \,\%\, \mathrm{TC_Instance_Number}]$$

$$\mathrm{IN}[N][i] = AlternativeTC[(N+1) \,\%\, \mathrm{TC_Instance_Number}]$$

Where N represents the LUT number and i represents the LUT input index (i=0,1,2).

Before selecting the waveform outputs, the TC must be configured first.

Figure 40-9. TC Input Selection

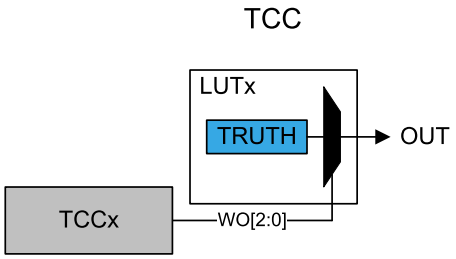


Timer/Counter for Control Application Inputs (TCC)

The TCC waveform outputs can be used as input source for the LUT. Only WO[2:0] outputs can be selected and routed to the respective LUT input (i.e., IN0 is connected to WO0, IN1 to WO1, and IN2 to WO2), as shown in the figure below.

Before selecting the waveform outputs, the TCC must be configured first.

Figure 40-10. TCC Input Selection



Serial Communication Output Transmit Inputs (SERCOM)

The serial engine transmitter output from Serial Communication Interface (SERCOM TX, TXd for USART, MOSI for SPI) can be used as input source for the LUT. The figure below shows an example for LUT0 and LUT1. The SERCOM selection for each LUT follows the formula:

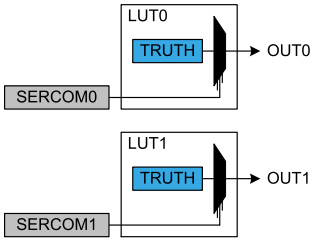
$$\mathrm{IN}[N][i] = SERCOM[N \,\%\, \mathrm{SERCOM_Instance_Number}]$$

With *N* representing the LUT number and *i*=0,1,2 representing the LUT input index.

Before selecting the SERCOM as input source, the SERCOM must be configured first: the SERCOM TX signal must be output on SERCOMn/pad[0], which serves as input pad to the CCL.

Downloaded from Arrow.com.

Figure 40-11. SERCOM Input Selection



Related Links

32. PORT - I/O Pin Controller
18. GCLK - Generic Clock Controller
42. AC – Analog Comparators
38. TC – Timer/Counter
34. SERCOM – Serial Communication Interface

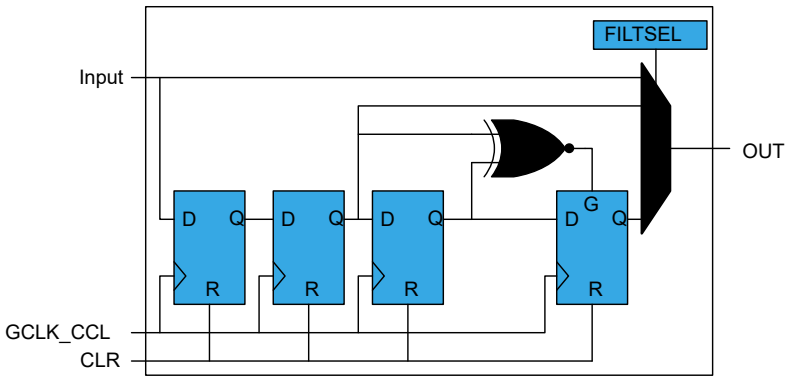
40.6.2.5 Filter

By default, the LUT output is a combinatorial function of the LUT inputs. This may cause some short glitches when the inputs change value. These glitches can be removed by clocking through filters, if demanded by application needs.

The Filter Selection bits in LUT Control register (LUTCTRLx.FILTSEL) define the synchronizer or digital filter options. When a filter is enabled, the OUT output will be delayed by two to five GCLK cycles. One APB clock after the corresponding LUT is disabled, all internal filter logic is cleared.

Note: Events used as LUT input will also be filtered, if the filter is enabled.

Figure 40-12. Filter



40.6.2.6 Edge Detector

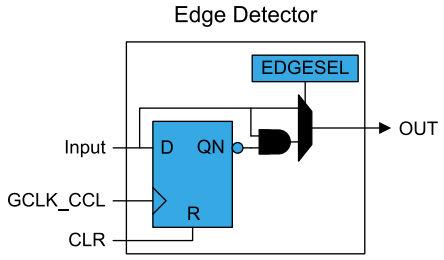
The edge detector can be used to generate a pulse when detecting a rising edge on its input. To detect a falling edge, the TRUTH table should be inverted.

The edge detector is enabled by writing '1' to the Edge Selection bit in LUT Control register (LUTCTRLx.EDGESEL). In order to avoid unpredictable behavior, either the filter or synchronizer must be enabled.

Edge detection is disabled by writing a '0' to LUTCTRLx.EDGESEL. After disabling a LUT, the corresponding internal Edge Detector logic is cleared one APB clock cycle later.

Downloaded from Arrow.com.

Figure 40-13. Edge Detector



40.6.2.7 Sequential Logic

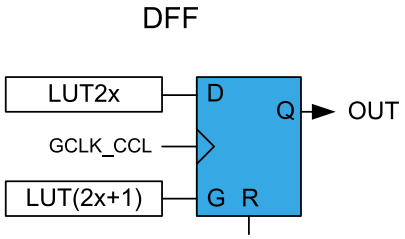
Each LUT pair can be connected to the internal sequential logic which can be configured to work as D flip flop, JK flip flop, gated D-latch or RS-latch by writing the Sequential Selection bits on the corresponding Sequential Control x register (SEQCTRLx.SEQSEL). Before using sequential logic, the GCLK_CCL clock and optionally each LUT filter or edge detector must be enabled.

Note: While configuring the sequential logic, the even LUT must be disabled. When configured the even LUT must be enabled.

Gated D Flip-Flop (DFF)

When the DFF is selected, the D-input is driven by the even LUT output (), and the G-input is driven by the odd LUT output (), as shown in Figure 40-14.

Figure 40-14. D Flip Flop



When the even LUT is disabled (), the flip-flop is asynchronously cleared. The reset command (R) is kept enabled for one APB clock cycle. In all other cases, the flip-flop output (OUT) is refreshed on rising edge of the GCLK_CCL, as shown in Table 40-2.

Table 40-2. DFF Characteristics

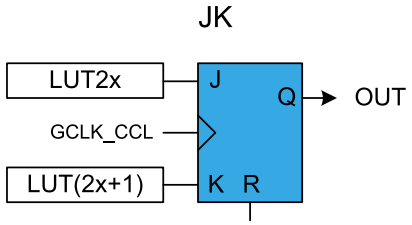
R	G	D	OUT
1	X	X	Clear
0	1	1	Set
		0	Clear
	0	X	Hold state (no change)

JK Flip-Flop (JK)

When this configuration is selected, the J-input is driven by the even LUT output (), and the K-input is driven by the odd LUT output (), as shown in Figure 40-15.

Downloaded from Arrow.com.

Figure 40-15. JK Flip Flop



When the even LUT is disabled (), the flip-flop is asynchronously cleared. The reset command (R) is kept enabled for one APB clock cycle. In all other cases, the flip-flop output (OUT) is refreshed on rising edge of the GCLK_CCL, as shown in Table 40-3.

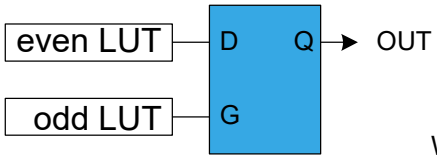
Table 40-3. JK Characteristics

R	J	K	OUT
1	X	X	Clear
0	0	0	Hold state (no change)
0	0	1	Clear
0	1	0	Set
0	1	1	Toggle

Gated D-Latch (DLATCH)

When the DLATCH is selected, the D-input is driven by the even LUT output (), and the G-input is driven by the odd LUT output (), as shown in Figure 40-14.

Figure 40-16. D-Latch



When the even LUT is disabled (), the latch output will be cleared. The G-input is forced enabled for one more APB clock cycle, and the D-input to zero. In all other cases, the latch output (OUT) is refreshed as shown in Table 40-4.

Table 40-4. D-Latch Characteristics

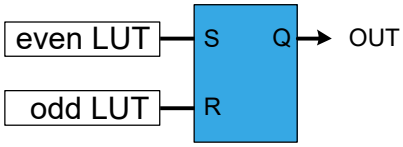
G	D	OUT
0	X	Hold state (no change)
1	0	Clear
1	1	Set

RS Latch (RS)

When this configuration is selected, the S-input is driven by the even LUT output (), and the R-input is driven by the odd LUT output (), as shown in Figure 40-17.

Downloaded from Arrow.com.

Figure 40-17. RS-Latch



When the even LUT is disabled), the latch output will be cleared. The R-input is forced enabled for one more APB clock cycle and S-input to zero. In all other cases, the latch output (OUT) is refreshed as shown in Table 40-5.

Table 40-5. RS-Latch Characteristics

S	R	OUT
0	0	Hold state (no change)
0	1	Clear
1	0	Set
1	1	Forbidden state

40.6.3 Events

The CCL can generate the following output events:

- OUTx: Lookup Table Output Value

Writing a '1' to the LUT Control Event Output Enable bit (LUTCTRL.LUTEO) enables the corresponding output event. Writing a '0' to this bit disables the corresponding output event.

The CCL can take the following actions on an input event:

- INSELx: The event is used as input for the TRUTH table. For further details refer to 40.5.6 Events.

Writing a '1' to the LUT Control Event Input Enable bit (LUTCTRL.LUTEI) enables the corresponding action on input event. Writing a '0' to this bit disables the corresponding action on input event.

Related Links

33. EVSYS – Event System

40.6.4 Sleep Mode Operation

When using the GCLK_CCL internal clocking, writing the Run In Standby bit in the Control register (CTRL.RUNSTDBY) to '1' will allow GCLK_CCL to be enabled in Standby Sleep mode.

If CTRL.RUNSTDBY=0, the GCLK_CCL will be disabled in Standby Sleep mode. If the Filter, Edge Detector or Sequential logic are enabled, the LUT output will be forced to zero in STANDBY mode. In all other cases, the TRUTH table decoder will continue operation and the LUT output will be refreshed accordingly.

Related Links

22. PM – Power Manager

Downloaded from Arrow.com.

40.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRL	7:0		RUNSTDBY					ENABLE	SWRST
0x01 ... 0x03	Reserved									
0x04	SEQCTRL0	7:0					SEQSEL[3:0]			
0x05 ... 0x07	Reserved									
0x08	LUTCTRL0	7:0	EDGESEL		FILTSEL[1:0]				ENABLE	
		15:8	INSELx[3:0]				INSELx[3:0]			
		23:16		LUTEO	LUTEI	INVEI	INSELx[3:0]			
		31:24	TRUTH[7:0]							
0x0C	LUTCTRL1	7:0	EDGESEL		FILTSEL[1:0]				ENABLE	
		15:8	INSELx[3:0]				INSELx[3:0]			
		23:16		LUTEO	LUTEI	INVEI	INSELx[3:0]			
		31:24	TRUTH[7:0]							

40.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to 40.5.8 Register Access Protection.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

Downloaded from Arrow.com.

40.8.1 Control

Name: CTRL
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY					ENABLE	SWRST
Access		R/W					R/W	W
Reset		0					0	0

Bit 6 – RUNSTDBY Run in Standby
This bit indicates if the GCLK_CCL clock must be kept running in standby mode. The setting is ignored for configurations where the generic clock is not required. For details refer to 40.6.4 Sleep Mode Operation.

Important: This bit must be written before enabling the CCL.

Value	Description
0	Generic clock is not required in standby sleep mode.
1	Generic clock is required in standby sleep mode.

Bit 1 – ENABLE Enable

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled.

Bit 0 – SWRST Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the CCL to their initial state.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

Downloaded from Arrow.com.

40.8.2 Sequential Control x

Name: SEQCTRL
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
					SEQSEL[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 3:0 – SEQSEL[3:0] Sequential Selection
These bits select the sequential configuration:

Sequential Selection

Value	Name	Description
0x0	DISABLE	Sequential logic is disabled
0x1	DFF	D flip flop
0x2	JK	JK flip flop
0x3	LATCH	D latch
0x4	RS	RS latch
0x5 – 0xF		Reserved

Downloaded from Arrow.com.

40.8.3 LUT Control x

Name: LUTCTRL
Offset: 0x08 + n*0x04 [n=0..1]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-protected

Bit	31	30	29	28	27	26	25	24
	TRUTH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
		LUTEO	LUTEI	INVEI	INSELx[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	INSELx[3:0]				INSELx[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EDGESEL		FILTSEL[1:0]				ENABLE	
Access	R/W		R/W	R/W			R/W	
Reset	0		0	0			0	

Bits 31:24 – TRUTH[7:0] Truth Table
These bits define the value of truth logic as a function of inputs IN[2:0].

Bit 22 – LUTEO LUT Event Output Enable

Value	Description
0	LUT event output is disabled.
1	LUT event output is enabled.

Bit 21 – LUTEI LUT Event Input Enable

Value	Description
0	LUT incoming event is disabled.
1	LUT incoming event is enabled.

Bit 20 – INVEI Inverted Event Input Enable

Value	Description
0	Incoming event is not inverted.
1	Incoming event is inverted.

Bit 7 – EDGESEL Edge Selection

Downloaded from Arrow.com.

Value	Description
0	Edge detector is disabled.
1	Edge detector is enabled.

Bits 5:4 – FILTSEL[1:0] Filter Selection
These bits select the LUT output filter options:

Filter Selection

Value	Name	Description
0x0	DISABLE	Filter disabled
0x1	SYNCH	Synchronizer enabled
0x2	FILTER	Filter enabled
0x3	-	Reserved

Bit 1 – ENABLE LUT Enable

Value	Description
0	The LUT is disabled.
1	The LUT is enabled.

Bits 19:16,15:12,11:8 – INSELx LUT Input x Source Selection
These bits select the LUT input x source:

Value	Name	Description
0x0	MASK	Masked input
0x1	FEEDBACK	Feedback input source
0x2	LINK	Linked LUT input source
0x3	EVENT	Event input source
0x4	IO	I/O pin input source
0x5	AC	AC input source
0x6	TC	TC input source
0x7	ALTTC	Alternative TC input source
0x8	TCC	TCC input source
0x9	SERCOM	SERCOM input source
0xB	ASYNCEVENT	Asynchronous event input source

Downloaded from Arrow.com.

41. ADC – Analog-to-Digital Converter

41.1 Overview

The Analog-to-Digital Converter (ADC) converts analog signals to digital values. The ADC has up to 12-bit resolution, and is capable of a sampling rate of up to 1MSPS. The input selection is flexible, and both differential and single-ended measurements can be performed. In addition, several internal signal inputs are available. The ADC can provide both signed and unsigned results.

ADC measurements can be started by either application software or an incoming event from another peripheral in the device. ADC measurements can be started with predictable timing, and without software intervention.

Both internal and external reference voltages can be used.

An integrated temperature sensor is available for use with the ADC. The INTREF voltage reference, as well as the scaled I/O and core voltages, can also be measured by the ADC.

The ADC has a compare function for accurate monitoring of user-defined thresholds, with minimum software intervention required.

The ADC can be configured for 8-, 10- or 12-bit results. ADC conversion results are provided left- or right-adjusted, which eases calculation when the result is represented as a signed value. It is possible to use DMA to move ADC results directly to memory or peripherals when conversions are done.

41.2 Features

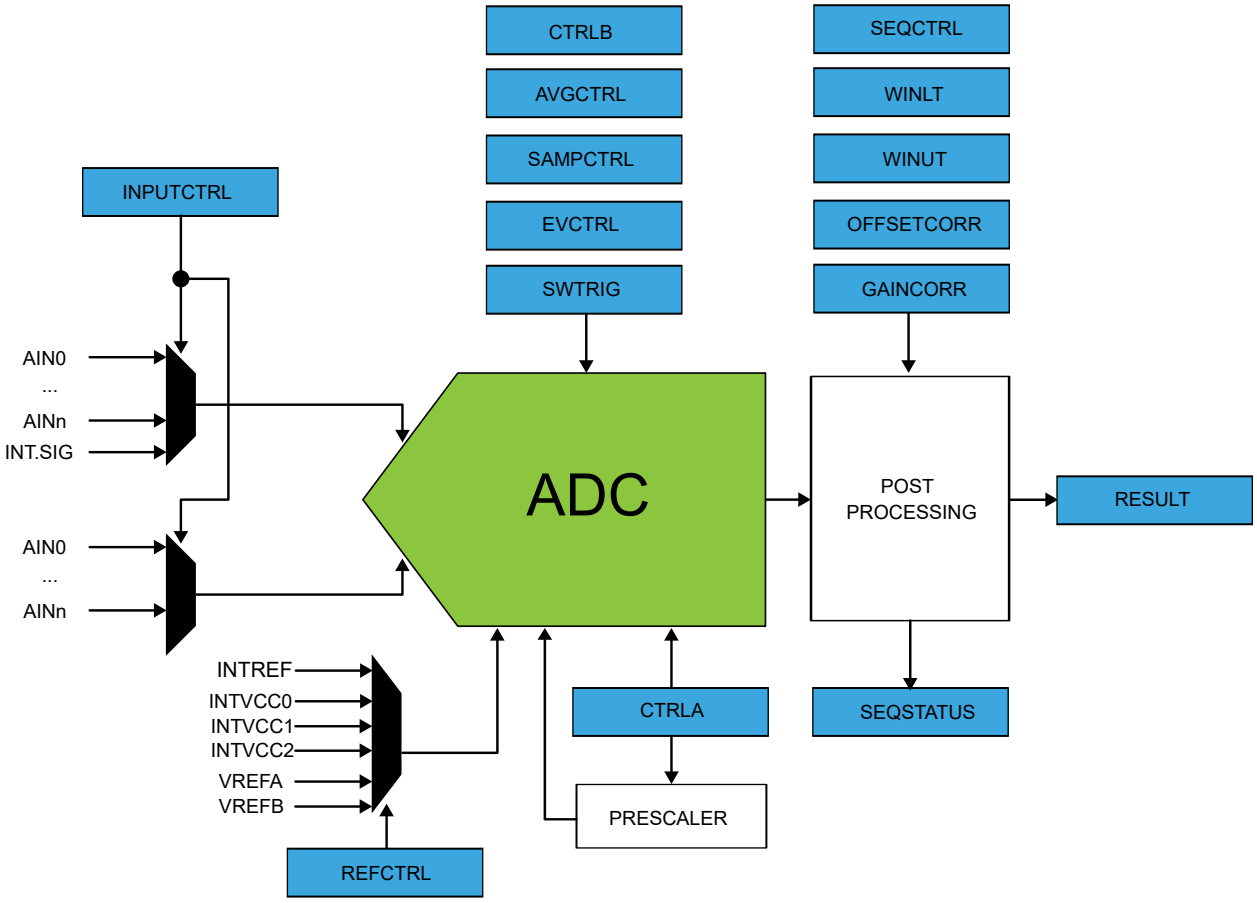
- 8-, 10- or 12-bit resolution
- Up to 1,000,000 samples per second (1MSPS)
- Differential and single-ended inputs
  - Up to 10 analog inputs
    10 positive and 8 negative, including internal and external
- Internal inputs:
  - Internal temperature sensor
  - INTREF voltage reference
  - Scaled core supply
  - Scaled I/O supply
  - DAC
- Single, continuous and sequencing options
- Windowing monitor with selectable channel
- Conversion range: V_{ref} = [1.0V to VDD_{ANA}]
- Built-in internal reference and external reference options
- Event-triggered conversion for accurate timing (one event input)
- Optional DMA transfer of conversion settings or result
- Hardware gain and offset compensation
- Averaging and oversampling with decimation to support up to 16-bit result
- Selectable sampling time

Downloaded from Arrow.com.

- Flexible Power / Throughput rate management

41.3 Block Diagram

Figure 41-1. ADC Block Diagram



41.4 Signal Description

Signal	Description	Type
VREFA/B	Analog input	External reference voltage
AIN[9..0]	Analog input	Analog input channels

Note: One signal can be mapped on several pins.

Related Links

1. Configuration Summary

41.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

Downloaded from Arrow.com.

41.5.1 I/O Lines

Using the ADC's I/O lines requires the I/O pins to be configured using the port configuration (PORT).

Related Links

32. PORT - I/O Pin Controller

41.5.2 Power Management

The ADC will continue to operate in any Sleep mode where the selected source clock is running. The ADC's interrupts, except the OVERRUN interrupt, can be used to wake up the device from sleep modes, except the OVERRUN interrupt. Events connected to the event system can trigger other operations in the system without exiting sleep modes.

Related Links

22. PM – Power Manager

41.5.3 Clocks

The ADC bus clock (CLK_APB_ADCx) can be enabled in the Main Clock, which also defines the default state.

The ADC requires a generic clock (GCLK_ADC). This clock must be configured and enabled in the Generic Clock Controller (GCLK) before using the ADC.

A generic clock is asynchronous to the bus clock. Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to *Synchronization* for further details.

Related Links

41.6.8 Synchronization

19.6.2.6 Peripheral Clock Masking

18. GCLK - Generic Clock Controller

41.5.4 DMA

The DMA request line is connected to the DMA Controller (DMAC). Using the ADC DMA requests requires the DMA Controller to be configured first.

Related Links

28. DMAC – Direct Memory Access Controller

41.5.5 Interrupts

The interrupt request line is connected to the interrupt controller. Using the ADC interrupt requires the interrupt controller to be configured first.

41.5.6 Events

The events are connected to the Event System.

Related Links

33. EVSYS – Event System

41.5.7 Debug Operation

When the CPU is halted in debug mode the ADC will halt normal operation. The ADC can be forced to continue operation during debugging. Refer to DBGCTRL register for details.

Downloaded from Arrow.com.

41.5.8 Register Access Protection

All registers with write-access are optionally write-protected by the peripheral access controller (PAC), except the following register:

- Interrupt Flag Status and Clear (INTFLAG) register

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

Related Links

15. PAC - Peripheral Access Controller

41.5.9 SAM L11 TrustZone Specific Register Access Protection

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

41.5.10 Analog Connections

I/O-pins (AINx), as well as the VREFA/VREFB reference voltage pins are analog inputs to the ADC. Any internal reference source, such as a bandgap voltage reference, or DAC must be configured and enabled prior to its use with the ADC.

The analog signals of AC, ADC, DAC and OPAMP can be interconnected. The AC and ADC peripheral can request the OPAMP using an analog ONDEMAND functionality.

See *Analog Connections of Peripherals* for details.

41.5.11 Calibration

The BIAS and LINEARITY calibration values from the production test must be loaded from the NVM Software Calibration Area into the ADC Calibration register (CALIB) by software to achieve specified accuracy.

41.6 Functional Description

41.6.1 Principle of Operation

By default, the ADC provides results with 12-bit resolution. 8-bit or 10-bit results can be selected in order to reduce the conversion time, see 41.6.2.8 Conversion Timing and Sampling Rate.

The ADC has an oversampling with decimation option that can extend the resolution to 16 bits. The input values can be either internal (e.g., an internal temperature sensor) or external (connected I/O pins). The user can also configure whether the conversion should be single-ended or differential.

Downloaded from Arrow.com.

41.6.2 Basic Operation

41.6.2.1 Initialization

The following registers are enable-protected, meaning that they can only be written when the ADC is disabled (CTRLA.ENABLE=0):

- Control B register (CTRLB)
- Reference Control register (REFCTRL)
- Event Control register (EVCTRL)
- Calibration register (CALIB)

Enable-protection is denoted by the "Enable-Protected" property in the register description.

41.6.2.2 Enabling, Disabling and Resetting

The ADC is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The ADC is disabled by writing CTRLA.ENABLE=0.

The ADC is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the ADC, except DBGCTRL, will be reset to their initial state, and the ADC will be disabled. Refer to 41.8.1 CTRLA for details.

41.6.2.3 Operation

In the most basic configuration, the ADC samples values from the configured internal or external sources (INPUTCTRL register). The rate of the conversion depends on the combination of the GCLK_ADC frequency and the clock prescaler.

To convert analog values to digital values, the ADC needs to be initialized first, as described in the Initialization section. Data conversion can be started either manually by setting the Start bit in the Software Trigger register (SWTRIG.START=1), or automatically by configuring an automatic trigger to initiate the conversions. A free-running mode can be used to continuously convert an input channel. When using free-running mode the first conversion must be started, while subsequent conversions will start automatically at the end of previous conversions.

The result of the conversion is stored in the Result register (RESULT) overwriting the result from the previous conversion.

To avoid data loss if more than one channel is enabled, the conversion result must be read as soon as it is available (INTFLAG.RESRDY). Failing to do so will result in an overrun error condition, indicated by the OVERRUN bit in the Interrupt Flag Status and Clear register (INTFLAG.OVERRUN).

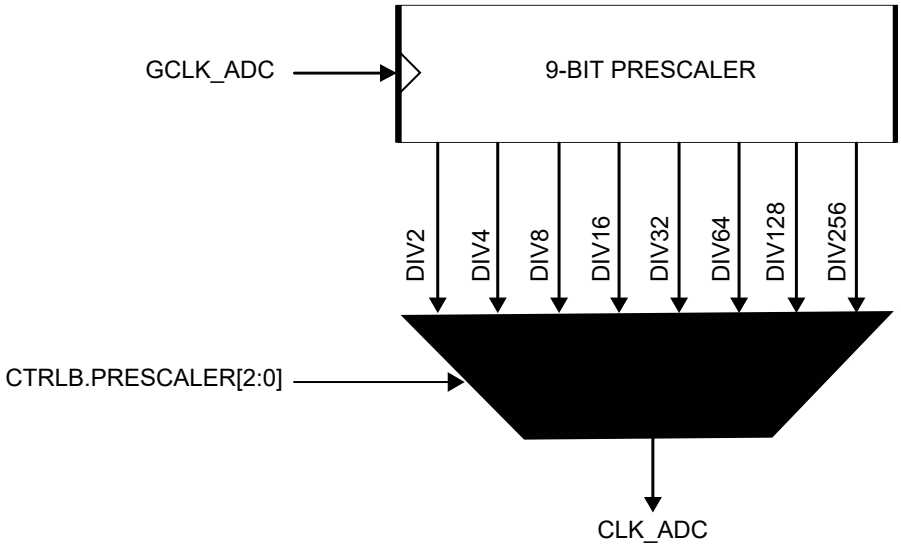
To enable one of the available interrupts sources, the corresponding bit in the Interrupt Enable Set register (INTENSET) must be written to '1'.

41.6.2.4 Prescaler Selection

The ADC is clocked by GCLK_ADC. There is also a prescaler in the ADC to enable conversion at lower clock rates. Refer to CTRLB for details on prescaler settings. Refer to 41.6.2.8 Conversion Timing and Sampling Rate for details on timing and sampling rate.

Downloaded from Arrow.com.

Figure 41-2. ADC Prescaler



Note: The minimum prescaling factor is DIV2.

41.6.2.5 Reference Configuration

The ADC has various sources for its reference voltage V_{REF}. The Reference Voltage Selection bit field in the Reference Control register (REFCTRL.REFSEL) determines which reference is selected. By default, the internal voltage reference INTREF is selected. Based on customer application requirements, the external or internal reference can be selected. Refer to REFCTRL.REFSEL for further details on available selections.

Related Links

41.8.3 REFCTRL

41.6.2.6 ADC Resolution

The ADC supports 8-bit, 10-bit or 12-bit resolution. Resolution can be changed by writing the Resolution bit group in the Control C register (CTRLC.RESSEL). By default, the ADC resolution is set to 12 bits. The resolution affects the propagation delay, see also 41.6.2.8 Conversion Timing and Sampling Rate.

41.6.2.7 Differential and Single-Ended Conversions

The ADC has two conversion options: differential and single-ended:

If the positive input is always positive, the single-ended conversion should be used in order to have full 12-bit resolution in the conversion.

If the positive input may go below the negative input, the differential mode should be used in order to get correct results.

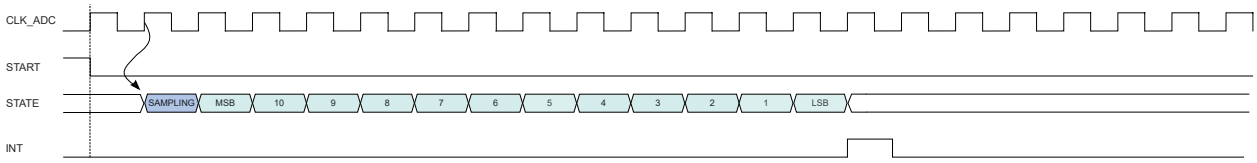
The differential mode is enabled by setting DIFFMODE bit in the Control C register (CTRLC.DIFFMODE). Both conversion types could be run in single mode or in free-running mode. When the free-running mode is selected, an ADC input will continuously sample the input and performs a new conversion. The INTFLAG.RESRDY bit will be set at the end of each conversion.

41.6.2.8 Conversion Timing and Sampling Rate

The following figure shows the ADC timing for one single conversion. A conversion starts after the software or event start are synchronized with the GCLK_ADC clock. The input channel is sampled in the first half CLK_ADC period.

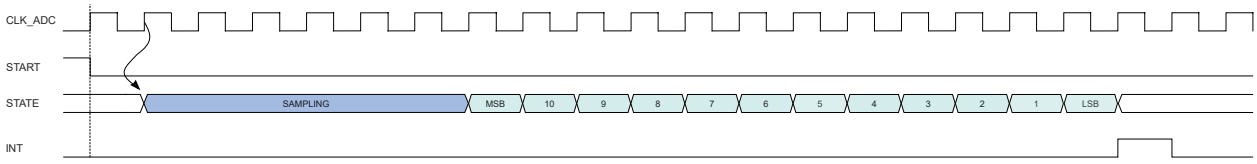
Downloaded from Arrow.com.

Figure 41-3. ADC Timing for One Conversion in 12-bit Resolution



The sampling time can be increased by using the Sampling Time Length bit group in the Sampling Time Control register (SAMPCTRL.SAMPLEN). As example, the next figure is showing the timing conversion with sampling time increased to six CLK_ADC cycles.

Figure 41-4. ADC Timing for One Conversion with Increased Sampling Time, 12-bit



The ADC provides also offset compensation, see the following figure. The offset compensation is enabled by the Offset Compensation bit in the Sampling Control register (SAMPCTRL.OFFCOMP).

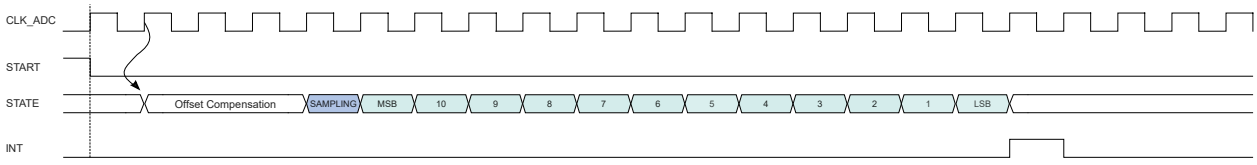
Note: If offset compensation is used, the sampling time must be set to one cycle of CLK_ADC.

In free running mode, the sampling rate R_S is calculated by

$R_S = f_{CLK_ADC} / (n_{SAMPLING} + n_{OFFCOMP} + n_{DATA})$

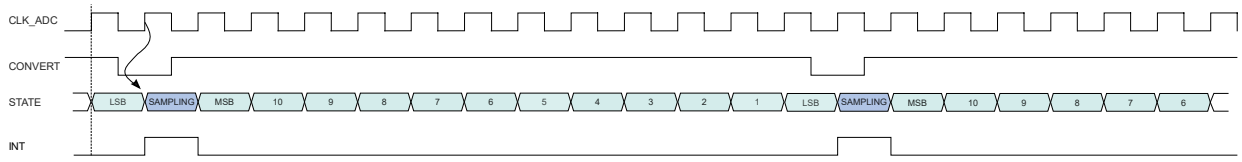
Here, $n_{SAMPLING}$ is the sampling duration in CLK_ADC cycles, $n_{OFFCOMP}$ is the offset compensation duration in clock cycles, and n_{DATA} is the bit resolution. f_{CLK_ADC} is the ADC clock frequency from the internal prescaler: $f_{CLK_ADC} = f_{GCLK_ADC} / 2^{(1 + CTRLB.PRESCALER)}$

Figure 41-5. ADC Timing for One Conversion with Offset Compensation, 12-bit



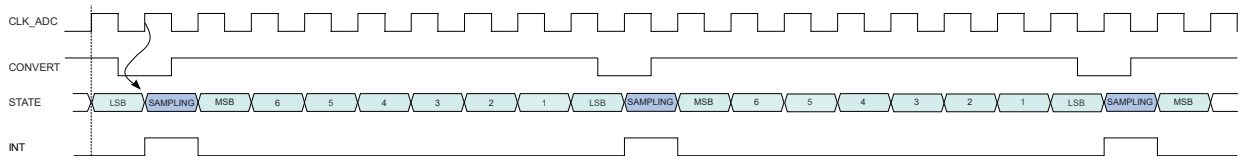
The impact of resolution on the sampling rate is seen in the next two figures, where free-running sampling in 12-bit and 8-bit resolution are compared.

Figure 41-6. ADC Timing for Free Running in 12-bit Resolution



Downloaded from Arrow.com.

Figure 41-7. ADC Timing for Free Running in 8-bit Resolution



The propagation delay of an ADC measurement is given by:

$$\text{PropagationDelay} = \frac{1 + \text{Resolution}}{f_{\text{ADC}}}$$

> **Example.** In order to obtain 1MSPS in 12-bit resolution with a sampling time length of four CLK_ADC cycles, f_{CLK_ADC} must be 1MSPS * (4 + 12) = 16MHz. As the minimal division factor of the prescaler is 2, GCLK_ADC must be 32MHz.

41.6.2.9 Accumulation

The result from multiple consecutive conversions can be accumulated. The number of samples to be accumulated is specified by the Sample Number field in the Average Control register (AVGCTRL.SAMPLENUM). When accumulating more than 16 samples, the result will be too large to match the 16-bit RESULT register size. To avoid overflow, the result is right shifted automatically to fit within the available register size. The number of automatic right shifts is specified in the table below.

Note: To perform the accumulation of two or more samples, the Conversion Result Resolution field in the Control C register (CTRLC.RESSEL) must be set.

Table 41-1. Accumulation

Number of Accumulated Samples	AVGCTRL. SAMPLENUM	Number of Automatic Right Shifts	Final Result Precision	Automatic Division Factor
1	0x0	0	12 bits	0
2	0x1	0	13 bits	0
4	0x2	0	14 bits	0
8	0x3	0	15 bits	0
16	0x4	0	16 bits	0
32	0x5	1	16 bits	2
64	0x6	2	16 bits	4
128	0x7	3	16 bits	8
256	0x8	4	16 bits	16
512	0x9	5	16 bits	32
1024	0xA	6	16 bits	64
Reserved	0xB –0xF		12 bits	0

Downloaded from Arrow.com.

41.6.2.10 Averaging

Averaging is a feature that increases the sample accuracy, at the cost of a reduced sampling rate. This feature is suitable when operating in noisy conditions.

Averaging is done by accumulating m samples, as described in 41.6.2.9 Accumulation, and dividing the result by m. The averaged result is available in the RESULT register. The number of samples to be accumulated is specified by writing to AVGCTRL.SAMPLENUM as shown in Table 41-2.

The division is obtained by a combination of the automatic right shift described above, and an additional right shift that must be specified by writing to the Adjusting Result/Division Coefficient field in AVGCTRL (AVGCTRL.ADJRES), as described in Table 41-2.

Note: To perform the averaging of two or more samples, the Conversion Result Resolution field in the Control C register (CTRLC.RESSEL) must be set.

Averaging AVGCTRL.SAMPLENUM samples will reduce the un-averaged sampling rate by a factor $\frac{1}{\text{AVGCTRL.SAMPLENUM}}$.

When the averaged result is available, the INTFLAG.RESRDY bit will be set.

Table 41-2. Averaging

Number of Accumulated Samples	AVGCTRL. SAMPLENUM	Intermediate Result Precision	Number of Automatic Right Shifts	Division Factor	AVGCTRL.ADJRES	Total Number of Right Shifts	Final Result Precision	Automatic Division Factor
1	0x0	12 bits	0	1	0x0		12 bits	0
2	0x1	13	0	2	0x1	1	12 bits	0
4	0x2	14	0	4	0x2	2	12 bits	0
8	0x3	15	0	8	0x3	3	12 bits	0
16	0x4	16	0	16	0x4	4	12 bits	0
32	0x5	17	1	16	0x4	5	12 bits	2
64	0x6	18	2	16	0x4	6	12 bits	4
128	0x7	19	3	16	0x4	7	12 bits	8
256	0x8	20	4	16	0x4	8	12 bits	16
512	0x9	21	5	16	0x4	9	12 bits	32
1024	0xA	22	6	16	0x4	10	12 bits	64
Reserved	0xB –0xF				0x0		12 bits	0

41.6.2.11 Oversampling and Decimation

By using oversampling and decimation, the ADC resolution can be increased from 12 bits up to 16 bits, for the cost of reduced effective sampling rate.

To increase the resolution by n bits, 4^n samples must be accumulated. The result must then be right-shifted by n bits. This right-shift is a combination of the automatic right-shift and the value written to AVGCTRL.ADJRES. To obtain the correct resolution, the ADJRES must be configured as described in the table below. This method will result in n bit extra LSB resolution.

Downloaded from Arrow.com.

Table 41-3. Configuration Required for Oversampling and Decimation

Result Resolution	Number of Samples to Average	AVGCTRL.SAMPLENUM[3:0]	Number of Automatic Right Shifts	AVGCTRL.ADJRES[2:0]
13 bits	$4^1 = 4$	0x2	0	0x1
14 bits	$4^2 = 16$	0x4	0	0x2
15 bits	$4^3 = 64$	0x6	2	0x1
16 bits	$4^4 = 256$	0x8	4	0x0

41.6.2.12 Automatic Sequences

The ADC has the ability to automatically sequence a series of conversions. This means that each time the ADC receives a start-of-conversion request, it can perform multiple conversions automatically. All of the 32 positive inputs can be included in a sequence by writing to corresponding bits in the Sequence Control register (SEQCTRL). The order of the conversion in a sequence is the lower positive MUX selection to upper positive MUX (AIN0, AIN1, AIN2 ...). In differential mode, the negative inputs selected by MUXNEG field, will be used for the entire sequence.

When a sequence starts, the Sequence Busy status bit in Sequence Status register (SEQSTATUS.SEQBUSY) will be set. When the sequence is complete, the Sequence Busy status bit will be cleared.

Each time a conversion is completed, the Sequence State bit in Sequence Status register (SEQSTATUS.SEQSTATE) will store the input number from which the conversion is done. The result will be stored in the RESULT register, and the Result Ready Interrupt Flag (INTFLAG.RESRDY) is set.

If additional inputs must be scanned, the ADC will automatically start a new conversion on the next input present in the sequence list.

Note that if SEQCTRL register has no bits set to '1', the conversion is done with the selected MUXPOS input.

41.6.2.13 Window Monitor

The window monitor feature allows the conversion result in the RESULT register to be compared to predefined threshold values. The window mode is selected by setting the Window Monitor Mode bits in the Control C register (CTRLC.WINMODE). Threshold values must be written in the Window Monitor Lower Threshold register (WINLT) and Window Monitor Upper Threshold register (WINUT).

If differential input is selected, the WINLT and WINUT are evaluated as signed values. Otherwise they are evaluated as unsigned values. The significant WINLT and WINUT bits are given by the precision selected in the Conversion Result Resolution bit group in the Control C register (CTRLC.RESSEL). This means that for example in 8-bit mode, only the eight lower bits will be considered. In addition, in differential mode, the eighth bit will be considered as the sign bit, even if the ninth bit is zero.

The INTFLAG.WINMON interrupt flag will be set if the conversion result matches the window monitor condition.

41.6.2.14 Offset and Gain Correction

Inherent gain and offset errors affect the absolute accuracy of the ADC.

The offset error is defined as the deviation of the actual ADC transfer function from an ideal straight line at zero input voltage. The offset error cancellation is handled by the Offset Correction register

Downloaded from Arrow.com.

(OFFSETCORR). The offset correction value is subtracted from the converted data before writing the Result register (RESULT).

The gain error is defined as the deviation of the last output step's midpoint from the ideal straight line, after compensating for offset error. The gain error cancellation is handled by the Gain Correction register (GAINCORR).

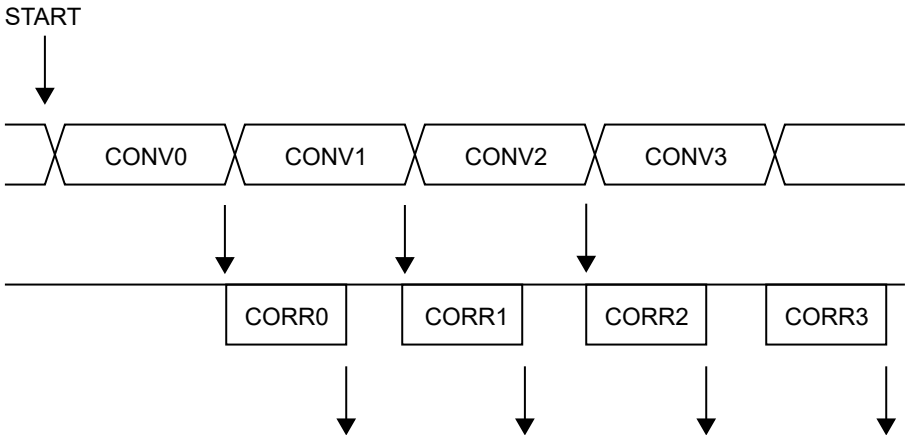
To correct these two errors, the Digital Correction Logic Enabled bit in the Control C register (CTRLC.CORREN) must be set.

Offset and gain error compensation results are both calculated according to:

$$\text{Result} = (\text{Conversion value} + - \text{OFFSETCORR}) \cdot \text{GAINCORR}$$

The correction will introduce a latency of 13 CLK_ADC clock cycles. In free running mode this latency is introduced on the first conversion only, since its duration is always less than the propagation delay. In single conversion mode this latency is introduced for each conversion.

Figure 41-8. ADC Timing Correction Enabled



41.6.2.15 Reference Buffer Compensation Offset

A hardware compensation using a reference buffer can be used.

When the REFCTRL.REFCOMP bit is set, the offset of the reference buffer is sensed during the ADC sampling phase. This offset will be then cancelled during the conversion phase. This feature allows to decrease the overall gain error of the ADC.

There is also a digital gain correction (refer to Offset and gain correction chapter) but contrary to that digital gain correction, the hardware compensation won't introduce any latency.

41.6.3 Additional Features

41.6.3.1 Double Buffering

The following registers are double buffered:

- Input Control (INPUTCTRL)
- Control C (CTRLC)
- Average Control (AVGCTRL)
- Sampling Time Control (SAMPCTRL)
- Window Monitor Lower Threshold (WINLT)
- Window Monitor Upper Threshold (WINUT)

Downloaded from Arrow.com.

- Gain Correction (GAINCORR)
- Offset Correction (OFFSETCORR)

When one of these registers is written, the data is stored in the corresponding buffer as long as the current conversion is not impacted, and the corresponding busy status will be set in the Synchronization Busy register (SYNCBUSY). When a new RESULT is available, data stored in the buffer registers will be transfered to the ADC and a new conversion can start.

41.6.3.2 Device Temperature Measurement

Principle

The device has an integrated temperature sensor which is part of the Supply Controller (SUPC). The analog signal of that sensor can be converted into a digital value by the ADC. The digital value can be converted into a temperature in °C by following the steps in this section.

Configuration and Conditions

In order to conduct temperature measurements, configure the device according to these steps.

1. Configure the clocks and device frequencies according to the Electrical Characteristics chapters.
2. Configure the Voltage References System of the Supply Controller (SUPC):
   - 2.1. Enable the temperature sensor by writing a '1' to the Temperature Sensor Enable bit in the VREF Control register (SUPC.VREF.TSEN).
   - 2.2. Select the required voltage for the internal voltage reference INTREF by writing to the Voltage Reference Selection bits (SUPC.VREF.SEL). The required value can be found in the Electrical Characteristics chapters.
   - 2.3. Enable routing INTREF to the ADC by writing a '1' to the Voltage Reference Output Enable bit (SUPC.VREF.VREFOE).
3. Configure the ADC:
   - 3.1. Select the internal voltage reference INTREF as ADC reference voltage by writing to the Reference Control register (ADC.REFCTRL.REFSEL).
   - 3.2. Select the temperature sensor vs. internal GND as input by writing TEMP and GND to the positive and negative MUX Input Selection bit fields (ADC.INPUTCTRL.MUXNEG and .MUXPOS, respectively).
   - 3.3. Configure the remaining ADC parameters according to the Electrical Characteristics chapters.
   - 3.4. Enable the ADC and acquire a value, ADC_m.

Calculation Parameter Values

The temperature sensor behavior is linear, but it is sensitive to several parameters such as the internal voltage reference - which itself depends on the temperature. To take this into account, each device contains a Temperature Log row with individual calibration data measured and written during the production tests. These calibration values are read by software to infer the most accurate temperature readings possible.

The Temperature Log Row basically contains the following parameter set for two different temperatures ("ROOM" and "HOT"):

- Calibration temperatures in °C. One at room temperature $temp_R$, one at a higher temperature $temp_H$:
  - ROOM_TEMP_VAL_INT and ROOM_TEMP_VAL_DEC contain the measured temperature at room insertion, $temp_R$, in °C, separated in integer and decimal value.

Downloaded from Arrow.com.

Example: For ROOM_TEMP_VAL_INT=0x19=25 and ROOM_TEMP_VAL_DEC=2, the measured temperature at room insertion is 25.2°C.

  - HOT_TEMP_VAL_INT and HOT_TEMP_VAL_DEC contain the measured temperature at hot insertion, $temp_H$, in °C. The integer and decimal value are also separated.
- For each temperature, the corresponding sensor value at the ADC in 12-bit, ADC_R and ADC_H:
  - ROOM_ADC_VAL contains the 12-bit ADC value, ADC_R, corresponding to $temp_R$. Its conversion to Volt is denoted V_{ADCR}.
  - HOT_ADC_VAL contains the 12-bit ADC value, ADC_H, corresponding to $temp_H$. Its conversion to Volt is denoted V_{ADCH}.
- Actual reference voltages at each calibration temperature in Volt, $INT1V_R$ and $INT1V_H$, respectively:
  - ROOM_INT1V_VAL is the 2's complement of the internal 1V reference value at $temp_R$: $INT1V_R$.
  - HOT_INT1V_VAL is the 2's complement of the internal 1V reference value at $temp_H$: $INT1V_H$.
  - Both ROOM_INT1V_VAL and HOT_INT1V_VAL values are centered around 1V with a 0.001V step. In other words, the range of values [0,127] corresponds to [1V, 0.873V] and the range of values [-1, -127] corresponds to [1.001V, 1.127V]. INT1V == 1 - (VAL/1000) is valid for both ranges.

Calculating the Temperature by Linear Interpolation

Using the data pairs ($temp_R$, V_{ADCR}) and ($temp_H$, V_{ADCH}) for a linear interpolation, we have the following equation:

$$\left(\frac{V_{ADC} - V_{ADCR}}{temp - temp_R}\right) = \left(\frac{V_{ADCH} - V_{ADCR}}{temp_H - temp_R}\right)$$

The voltages V_x are acquired as 12-bit ADC values ADC_x, with respect to an internal reference voltage $INT1V_x$:

[Equation 1]

$$V_{ADCx} = ADC_x \cdot \frac{\mathrm{INT1V}_x}{2^{12} - 1}$$

For the measured value of the temperature sensor, ADC_m, the reference voltage is assumed to be perfect, i.e., $INT1V_m$=$INT1V_c$=1V. These substitutions yield a coarse value of the measured temperature $temp_C$:

[Equation 2]

$$temp_C = temp_R + \left[\frac{\left\{\left(ADC_m \cdot \frac{\mathrm{INT1V}_c}{\left(2^{12}-1\right)}\right) - \left(ADC_R \cdot \frac{\mathrm{INT1V}_R}{\left(2^{12}-1\right)}\right)\right\} \cdot (temp_H - temp_R)}{\left(ADC_H \cdot \frac{\mathrm{INT1V}_H}{\left(2^{12}-1\right)}\right) - \left(ADC_R \cdot \frac{\mathrm{INT1V}_R}{\left(2^{12}-1\right)}\right)}\right]$$

Or, after eliminating the 12-bit scaling factor (2^{12}-1):

[Equation 3]

$$temp_C = temp_R + \left[\frac{\{ADC_m \cdot \mathrm{INT1V}_c - (ADC_R \cdot \mathrm{INT1V}_R)\} \cdot (temp_H - temp_R)}{\{(ADC_H \cdot \mathrm{INT1V}_H) - (ADC_R \cdot \mathrm{INT1V}_R)\}}\right]$$

Downloaded from Arrow.com.

Equations 3 is a coarse value, because we assumed that $INT1V_c$=1V. To achieve a more accurate result, we replace $INT1V_c$ with an interpolated value $INT1V_m$. We use the two data pairs ($temp_R$, $INT1V_R$) and ($temp_H$, $INT1V_H$) and yield:

$$\left(\frac{\mathrm{INT1V}_m - \mathrm{INT1V}_R}{temp_m - temp_R}\right) = \left(\frac{\mathrm{INT1V}H - \mathrm{INT1V}V_R}{temp_H - temp_R}\right)$$

Using the coarse temperature value $temp_c$, we can infer a more precise $INT1V_m$ value during the ADC conversion as:

[Equation 4]

$$\mathrm{INT1V}_m = \mathrm{INT1V}_R + \left(\frac{(\mathrm{INT1V}_H - \mathrm{INT1V}_R) \cdot (temp_C - temp_R)}{(temp_H - temp_R)}\right)$$

Back to Equation 3, we replace the simple $INT1V_c$=1V by the more precise $INT1V_m$ of Equation 4, and find a more accurate temperature value $temp_f$:

[Equation 5]

$$temp_f = temp_R + \left[\frac{\{ADC_m \cdot \mathrm{INT1V}_m - (ADC_R \cdot \mathrm{INT1V}_R)\} \cdot (temp_H - temp_R)}{\{(ADC_H \cdot \mathrm{INT1V}_H) - (ADC_R \cdot \mathrm{INT1V}_R)\}}\right]$$

41.6.4 DMA Operation

The ADC generates the following DMA request:

- Result Conversion Ready (RESRDY): the request is set when a conversion result is available and cleared when the RESULT register is read. When the averaging operation is enabled, the DMA request is set when the averaging is completed and result is available.

41.6.5 Interrupts

The ADC has the following interrupt sources:

- Result Conversion Ready: RESRDY
- Window Monitor: WINMON
- Overrun: OVERRUN

These interrupts, except the OVERRUN interrupt, are asynchronous wake-up sources. See *Sleep Mode Controller* for details.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a one to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a one to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register. An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the ADC is reset. See INTFLAG register for details on how to clear interrupt flags. All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. Refer to *Nested Vector Interrupt Controller* for details. The user must read the INTFLAG register to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

22.6.3.3 Sleep Mode Controller

Downloaded from Arrow.com.

41.8.5 INTENCLR
41.8.6 INTENSET
41.8.7 INTFLAG

41.6.6 Events

The ADC can generate the following output events:

- Result Ready (RESRDY): Generated when the conversion is complete and the result is available. Refer to 41.8.4 EVCTRL for details.
- Window Monitor (WINMON): Generated when the window monitor condition match. Refer to 41.8.10 CTRLC for details.

Setting an Event Output bit in the Event Control Register (EVCTRL.xxEO=1) enables the corresponding output event. Clearing this bit disables the corresponding output event. Refer to the Event System chapter for details on configuring the event system.

The ADC can take the following actions on an input event:

- Start conversion (START): Start a conversion. Refer to 41.8.17 SWTRIG for details.
- Conversion flush (FLUSH): Flush the conversion. Refer to 41.8.17 SWTRIG for details.

Setting an Event Input bit in the Event Control register (EVCTRL.xxEI=1) enables the corresponding action on input event. Clearing this bit disables the corresponding action on input event.

The ADC uses only asynchronous events, so the asynchronous Event System channel path must be configured. By default, the ADC will detect a rising edge on the incoming event. If the ADC action must be performed on the falling edge of the incoming event, the event line must be inverted first. This is done by setting the corresponding Event Invert Enable bit in Event Control register (EVCTRL.xINV=1).

Note: If several events are connected to the ADC, the enabled action will be taken on any of the incoming events. If FLUSH and START events are available at the same time, the FLUSH event has priority.

Related Links

33. EVSYS – Event System

41.6.7 Sleep Mode Operation

The ONDEMAND and RUNSTDBY bits in the Control A register (CTRLA) control the behavior of the ADC during standby sleep mode, in cases where the ADC is enabled (CTRLA.ENABLE = 1). For further details on available options, refer to Table 41-4.

Note: When CTRLA.ONDEMAND=1, the analog block is powered-off when the conversion is complete. When a start request is detected, the system returns from sleep and starts a new conversion after the start-up time delay.

Table 41-4. ADC Sleep Behavior

CTRLA.RUNSTDBY	CTRLA.ONDEMAND	CTRLA.ENABLE	Description
x	x	0	Disabled
0	0	1	Run in all sleep modes except STANDBY.
0	1	1	Run in all sleep modes on request, except STANDBY.

Downloaded from Arrow.com.

CTRLA.RUNSTDBY	CTRLA.ONDEMAND	CTRLA.ENABLE	Description
1	0	1	Run in all sleep modes.
1	1	1	Run in all sleep modes on request.

41.6.8 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in Control A register (CTRLA.SWRST)
- Enable bit in Control A register (CTRLA.ENABLE)

The following registers are synchronized when written:

- Input Control register (INPUTCTRL)
- Control C register (CTRLC)
- Average control register (AVGCTRL)
- Sampling time control register (SAMPCTRL)
- Window Monitor Lower Threshold register (WINLT)
- Window Monitor Upper Threshold register (WINUT)
- Gain correction register (GAINCORR)
- Offset Correction register (OFFSETCORR)
- Software Trigger register (SWTRIG)

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Downloaded from Arrow.com.

41.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	ONDEMAND	RUNSTDBY					ENABLE	SWRST
0x01	CTRLB	7:0						PRESCALER[2:0]		
0x02	REFCTRL	7:0	REFCOMP				REFSEL[3:0]			
0x03	EVCTRL	7:0			WINMONEO	RESRDYEO	STARTINV	FLUSHINV	STARTEI	FLUSHEI
0x04	INTENCLR	7:0						WINMON	OVERRUN	RESRDY
0x05	INTENSET	7:0						WINMON	OVERRUN	RESRDY
0x06	INTFLAG	7:0						WINMON	OVERRUN	RESRDY
0x07	SEQSTATUS	7:0	SEQBUSY			SEQSTATE[4:0]				
0x08	INPUTCTRL	7:0				MUXPOS[4:0]				
		15:8				MUXNEG[4:0]				
0x0A	CTRLC	7:0			RESSEL[1:0]		CORREN	FREERUN	LEFTADJ	DIFFMODE
		15:8						WINMODE[2:0]		
0x0C	AVGCTRL	7:0		ADJRES[2:0]			SAMPLENUM[3:0]			
0x0D	SAMPCTRL	7:0	OFFCOMP		SAMPLEN[5:0]					
0x0E	WINLT	7:0	WINLT[7:0]							
		15:8	WINLT[15:8]							
0x10	WINUT	7:0	WINUT[7:0]							
		15:8	WINUT[15:8]							
0x12	GAINCORR	7:0	GAINCORR[7:0]							
		15:8					GAINCORR[11:8]			
0x14	OFFSETCORR	7:0	OFFSETCORR[7:0]							
		15:8					OFFSETCORR[11:8]			
0x16 ... 0x17	Reserved									
0x18	SWTRIG	7:0							START	FLUSH
0x19 ... 0x1B	Reserved									
0x1C	DBGCTRL	7:0								DBGRUN
0x1D ... 0x1F	Reserved									
0x20	SYNCBUSY	7:0	WINUT	WINLT	SAMPCTRL	AVGCTRL	CTRLC	INPUTCTRL	ENABLE	SWRST
		15:8						SWTRIG	OFFSETCOR R	GAINCORR
0x22 ... 0x23	Reserved									
0x24	RESULT	7:0	RESULT[7:0]							
		15:8	RESULT[15:8]							
0x26 ... 0x27	Reserved									

Downloaded from Arrow.com.

Offset	Name	Bit Pos.								
0x28	SEQCTRL	7:0	SEQENn[7:0]							
		15:8	SEQENn[15:8]							
		23:16	SEQENn[23:16]							
		31:24	SEQENn[31:24]							
0x2C	CALIB	7:0						BIASCOMP[2:0]		
		15:8						BIASREFBUF[2:0]		

41.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to the section on Synchronization.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to Synchronization section.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

Downloaded from Arrow.com.

41.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY					ENABLE	SWRST
Access	R/W	R/W					R/W	R/W
Reset	0	0					0	0

Bit 7 – ONDEMAND On Demand Control
The On Demand operation mode allows the ADC to be enabled or disabled, depending on other peripheral requests.

In On Demand operation mode, i.e., if the ONDEMAND bit has been previously set, the ADC will only be running when requested by a peripheral. If there is no peripheral requesting the ADC will be in a disable state.

If On Demand is disabled the ADC will always be running when enabled.

In standby sleep mode, the On Demand operation is still active if the CTRLA.RUNSTDBY bit is '1'. If CTRLA.RUNSTDBY is '0', the ADC is disabled.

This bit is not synchronized.

Value	Description
0	The ADC is always on , if enabled.
1	The ADC is enabled, when a peripheral is requesting the ADC conversion. The ADC is disabled if no peripheral is requesting it.

Bit 6 – RUNSTDBY Run in Standby
This bit controls how the ADC behaves during standby sleep mode.

This bit is not synchronized.

Value	Description
0	The ADC is halted during standby sleep mode.
1	The ADC is not stopped in standby sleep mode. If CTRLA.ONDEMAND=1, the ADC will be running when a peripheral is requesting it. If CTRLA.ONDEMAND=0, the ADC will always be running in standby sleep mode.

Bit 1 – ENABLE Enable
Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRL.ENABLE will read back immediately and the ENABLE bit in the SYNCBUSY register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

Value	Description
0	The ADC is disabled.
1	The ADC is enabled.

Downloaded from Arrow.com.

Bit 0 – SWRST Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the ADC, except DBGCTRL, to their initial state, and the ADC will be disabled.

Writing a '1' to CTRL.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

Downloaded from Arrow.com.

41.8.2 Control B

Name: CTRLB
Offset: 0x01
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
						PRESCALER[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:0 – PRESCALER[2:0] Prescaler Configuration
This field defines the ADC clock relative to the peripheral clock.

Value	Name	Description
0x0	DIV2	Peripheral clock divided by 2
0x1	DIV4	Peripheral clock divided by 4
0x2	DIV8	Peripheral clock divided by 8
0x3	DIV16	Peripheral clock divided by 16
0x4	DIV32	Peripheral clock divided by 32
0x5	DIV64	Peripheral clock divided by 64
0x6	DIV128	Peripheral clock divided by 128
0x7	DIV256	Peripheral clock divided by 256

Downloaded from Arrow.com.

41.8.3 Reference Control

Name: REFCTRL
Offset: 0x02
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
	REFCOMP				REFSEL[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit 7 – REFCOMP Reference Buffer Offset Compensation Enable
The gain error can be reduced by enabling the reference buffer offset compensation. This will increase the start-up time of the reference.

Value	Description
0	Reference buffer offset compensation is disabled.
1	Reference buffer offset compensation is enabled.

Bits 3:0 – REFSEL[3:0] Reference Selection
These bits select the reference for the ADC.

Value	Name	Description
0x0	INTREF	Internal variable reference voltage, refer to the SUPC.VREF register for voltage reference value
x01	INTVCC0	1/1.6 VDDANA
0x2	INTVCC1	1/2 VDDANA (only for VDDANA > 2.0V)
0x3	VREFA	External reference
0x4	VREFB	External reference
0x5	INTVCC2	VDDANA
0x6 - 0xF		Reserved

Downloaded from Arrow.com.

41.8.4 Event Control

Name: EVCTRL
Offset: 0x03
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
			WINMONEO	RESRDYEO	STARTINV	FLUSHINV	STARTEI	FLUSHEI
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – WINMONEO Window Monitor Event Out
This bit indicates whether the Window Monitor event output is enabled or not and an output event will be generated when the window monitor detects something.

Value	Description
0	Window Monitor event output is disabled and an event will not be generated.
1	Window Monitor event output is enabled and an event will be generated.

Bit 4 – RESRDYEO Result Ready Event Out
This bit indicates whether the Result Ready event output is enabled or not and an output event will be generated when the conversion result is available.

Value	Description
0	Result Ready event output is disabled and an event will not be generated.
1	Result Ready event output is enabled and an event will be generated.

Bit 3 – STARTINV Start Conversion Event Invert Enable

Value	Description
0	Start event input source is not inverted.
1	Start event input source is inverted.

Bit 2 – FLUSHINV Flush Event Invert Enable

Value	Description
0	Flush event input source is not inverted.
1	Flush event input source is inverted.

Bit 1 – STARTEI Start Conversion Event Input Enable

Value	Description
0	A new conversion will not be triggered on any incoming event.
1	A new conversion will be triggered on any incoming event.

Bit 0 – FLUSHEI Flush Event Input Enable

Downloaded from Arrow.com.

Value	Description
0	A flush and new conversion will not be triggered on any incoming event.
1	A flush and new conversion will be triggered on any incoming event.

Downloaded from Arrow.com.

41.8.5 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Bit	7	6	5	4	3	2	1	0
						WINMON	OVERRUN	RESRDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – WINMON Window Monitor Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Window Monitor Interrupt Enable bit, which disables the corresponding interrupt request.

Value	Description
0	The window monitor interrupt is disabled.
1	The window monitor interrupt is enabled, and an interrupt request will be generated when the Window Monitor interrupt flag is set.

Bit 1 – OVERRUN Overrun Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Overrun Interrupt Enable bit, which disables the corresponding interrupt request.

Value	Description
0	The Overrun interrupt is disabled.
1	The Overrun interrupt is enabled, and an interrupt request will be generated when the Overrun interrupt flag is set.

Bit 0 – RESRDY Result Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Result Ready Interrupt Enable bit, which disables the corresponding interrupt request.

Value	Description
0	The Result Ready interrupt is disabled.
1	The Result Ready interrupt is enabled, and an interrupt request will be generated when the Result Ready interrupt flag is set.

Downloaded from Arrow.com.

41.8.6 Interrupt Enable Set

Name: INTENSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Bit	7	6	5	4	3	2	1	0
						WINMON	OVERRUN	RESRDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – WINMON Window Monitor Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Window Monitor Interrupt bit, which enables the Window Monitor interrupt.

Value	Description
0	The Window Monitor interrupt is disabled.
1	The Window Monitor interrupt is enabled.

Bit 1 – OVERRUN Overrun Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Overrun Interrupt bit, which enables the Overrun interrupt.

Value	Description
0	The Overrun interrupt is disabled.
1	The Overrun interrupt is enabled.

Bit 0 – RESRDY Result Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Result Ready Interrupt bit, which enables the Result Ready interrupt.

Value	Description
0	The Result Ready interrupt is disabled.
1	The Result Ready interrupt is enabled.

Downloaded from Arrow.com.

41.8.7 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x06
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
						WINMON	OVERRUN	RESRDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – WINMON Window Monitor
This flag is cleared by writing a '1' to the flag or by reading the RESULT register.

This flag is set on the next GCLK_ADC cycle after a match with the window monitor condition, and an interrupt request will be generated if INTENCLR/SET.WINMON is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Window Monitor interrupt flag.

Bit 1 – OVERRUN Overrun
This flag is cleared by writing a '1' to the flag.

This flag is set if RESULT is written before the previous value has been read by CPU, and an interrupt request will be generated if INTENCLR/SET.OVERRUN=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Overrun interrupt flag.

Bit 0 – RESRDY Result Ready
This flag is cleared by writing a '1' to the flag or by reading the RESULT register.

This flag is set when the conversion result is available, and an interrupt will be generated if INTENCLR/SET.RESRDY=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Result Ready interrupt flag.

Downloaded from Arrow.com.

41.8.8 Sequence Status

Name: SEQSTATUS
Offset: 0x07
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	SEQBUSY			SEQSTATE[4:0]				
Access	R			R	R	R	R	R
Reset	0			0	0	0	0	0

Bit 7 – SEQBUSY Sequence busy
This bit is set when the sequence start.

This bit is clear when the last conversion in a sequence is done.

Bits 4:0 – SEQSTATE[4:0] Sequence State
These bit fields are the pointer of sequence. This value identifies the last conversion done in the sequence.

Downloaded from Arrow.com.

41.8.9 Input Control

Name: INPUTCTRL
Offset: 0x08
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
				MUXNEG[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				MUXPOS[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 12:8 – MUXNEG[4:0] Negative MUX Input Selection
These bits define the MUX selection for the negative ADC input.

Value	Name	Description
0x00	AIN0	ADC AIN0 pin
0x01	AIN1	ADC AIN1 pin
0x02	AIN2	ADC AIN2 pin
0x03	AIN3	ADC AIN3 pin
0x04	AIN4	ADC AIN4 pin
0x05	AIN5	ADC AIN5 pin
0x06	AIN6	ADC AIN6 pin
0x07	AIN7	ADC AIN7 pin
0x08 – 0x17	-	Reserved
0x18	GND	Internal ground
0x19 – 0x1F	-	Reserved

Bits 4:0 – MUXPOS[4:0] Positive MUX Input Selection
These bits define the MUX selection for the positive ADC input. If the internal bandgap voltage or temperature sensor input channel is selected, then the Sampling Time Length bit group in the Sampling Control register must be written with a corresponding value.

Value	Name	Description
0x00	AIN0	ADC AIN0 pin
0x01	AIN1	ADC AIN1 pin
0x02	AIN2	ADC AIN2 pin
0x03	AIN3	ADC AIN3 pin
0x04	AIN4	ADC AIN4 pin
0x05	AIN5	ADC AIN5 pin
0x06	AIN6	ADC AIN6 pin

Downloaded from Arrow.com.

Value	Name	Description
0x07	AIN7	ADC AIN7 pin
0x08	AIN8	ADC AIN8 pin
0x09	AIN9	ADC AIN9 pin
0x0A – 0x17	-	Reserved
0x18	TEMP	Temperature Sensor
0x19	BANDGAP	INTREF Voltage Reference
0x1A	SCALEDVDDCORE	1/4 Scaled VDDCORE Supply
0x1B	SCALEDVDDANA	1/4 Scaled VDDANA Supply
0x1C	DAC	DAC Output
0x1D	SCALEDVDDIO	1/4 Scaled VDDIO Supply
0x1E	OPAMP01	OPAMP0 or OPAMP1 output
0x1F	OPAMP2	OPAMP2 output

Downloaded from Arrow.com.

41.8.10 Control C

Name: CTRLC
Offset: 0x0A
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
						WINMODE[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
			RESSEL[1:0]		CORREN	FREERUN	LEFTADJ	DIFFMODE
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 10:8 – WINMODE[2:0] Window Monitor Mode
These bits enable and define the window monitor mode.

Value	Name	Description
0x0	DISABLE	No window mode (default)
0x1	MODE1	RESULT > WINLT
0x2	MODE2	RESULT < WINUT
0x3	MODE3	WINLT < RESULT < WINUT
0x4	MODE4	WINUT < RESULT < WINLT
0x5 – 0x7		Reserved

Bits 5:4 – RESSEL[1:0] Conversion Result Resolution
These bits define whether the ADC completes the conversion 12-, 10- or 8-bit result resolution.

Value	Name	Description
0x0	12BIT	12-bit result
0x1	16BIT	For averaging mode output
0x2	10BIT	10-bit result
0x3	8BIT	8-bit result

Bit 3 – CORREN Digital Correction Logic Enabled

Value	Description
0	Disable the digital result correction.
1	Enable the digital result correction. The ADC conversion result in the RESULT register is then corrected for gain and offset based on the values in the GAINCORR and OFFSETCORR registers. Conversion time will be increased by 13 cycles according to the value in the Offset Correction Value bit group in the Offset Correction register.

Bit 2 – FREERUN Free Running Mode

Downloaded from Arrow.com.

Value	Description
0	The ADC run in single conversion mode.
1	The ADC is in free running mode and a new conversion will be initiated when a previous conversion completes.

Bit 1 – LEFTADJ Left-Adjusted Result

Value	Description
0	The ADC conversion result is right-adjusted in the RESULT register.
1	The ADC conversion result is left-adjusted in the RESULT register. The high byte of the 12-bit result will be present in the upper part of the result register. Writing this bit to zero (default) will right-adjust the value in the RESULT register.

Bit 0 – DIFFMODE Differential Mode

Value	Description
0	The ADC is running in singled-ended mode.
1	The ADC is running in differential mode. In this mode, the voltage difference between the MUXPOS and MUXNEG inputs will be converted by the ADC.

Downloaded from Arrow.com.

41.8.11 Average Control

Name: AVGCTRL
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
		ADJRES[2:0]			SAMPLENUM[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 6:4 – ADJRES[2:0] Adjusting Result / Division Coefficient
These bits define the division coefficient in 2n steps.

Bits 3:0 – SAMPLENUM[3:0] Number of Samples to be Collected
These bits define how many samples are added together. The result will be available in the Result register (RESULT). Note: if the result width increases, CTRLC.RESSEL must be changed.

Value	Description
0x0	1 sample
0x1	2 samples
0x2	4 samples
0x3	8 samples
0x4	16 samples
0x5	32 samples
0x6	64 samples
0x7	128 samples
0x8	256 samples
0x9	512 samples
0xA	1024 samples
0xB - 0xF	Reserved

Downloaded from Arrow.com.

41.8.12 Sampling Time Control

Name: SAMPCTRL
Offset: 0x0D
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	OFFCOMP		SAMPLEN[5:0]					
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bit 7 – OFFCOMP Comparator Offset Compensation Enable
Setting this bit enables the offset compensation for each sampling period to ensure low offset and immunity to temperature or voltage drift. This compensation increases the sampling time by three clock cycles.

This bit must be set to zero to validate the SAMPLEN value. It's not possible to use OFFCOMP=1 and SAMPLEN>0.

Bits 5:0 – SAMPLEN[5:0] Sampling Time Length
These bits control the ADC sampling time in number of CLK_ADC cycles, depending of the prescaler value, thus controlling the ADC input impedance. Sampling time is set according to the equation:

$$\text{Sampling time} = (\text{SAMPLEN}+1) \cdot (\text{CLK}_{\text{ADC}})$$

Downloaded from Arrow.com.

41.8.13 Window Monitor Lower Threshold

Name: WINLT
Offset: 0x0E
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	WINLT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	WINLT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – WINLT[15:0] Window Lower Threshold
If the window monitor is enabled, these bits define the lower threshold value.

Downloaded from Arrow.com.

41.8.14 Window Monitor Upper Threshold

Name: WINUT
Offset: 0x10
Reset: 0x0000
Property: PAV Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	WINUT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	WINUT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – WINUT[15:0] Window Upper Threshold
If the window monitor is enabled, these bits define the upper threshold value.

Downloaded from Arrow.com.

41.8.15 Gain Correction

Name: GAINCORR
Offset: 0x12
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
					GAINCORR[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GAINCORR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 11:0 – GAINCORR[11:0] Gain Correction Value
If CTRLC.CORREN=1, these bits define how the ADC conversion result is compensated for gain error before being written to the result register. The gain correction is a fractional value, a 1-bit integer plus an 11-bit fraction, and therefore ½ <= GAINCORR < 2. GAINCORR values range from 0.10000000000 to 1.11111111111.

Downloaded from Arrow.com.

41.8.16 Offset Correction

Name: OFFSETCORR
Offset: 0x14
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
					OFFSETCORR[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OFFSETCORR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 11:0 – OFFSETCORR[11:0] Offset Correction Value
If CTRLC.CORREN=1, these bits define how the ADC conversion result is compensated for offset error before being written to the Result register. This OFFSETCORR value is in two's complement format.

Downloaded from Arrow.com.

41.8.17 Software Trigger

Name: SWTRIG
Offset: 0x18
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
							START	FLUSH
Access							W	W
Reset							0	0

Bit 1 – START ADC Start Conversion
Writing a '1' to this bit will start a conversion or sequence. The bit is cleared by hardware when the conversion has started. Writing a '1' to this bit when it is already set has no effect.

Writing a '0' to this bit will have no effect.

Bit 0 – FLUSH ADC Conversion Flush
Writing a '1' to this bit will flush the ADC pipeline. A flush will restart the ADC clock on the next peripheral clock edge, and all conversions in progress will be aborted and lost. This bit is cleared until the ADC has been flushed.

After the flush, the ADC will resume where it left off; i.e., if a conversion was pending, the ADC will start a new conversion.

Writing this bit to '0' will have no effect.

Downloaded from Arrow.com.

41.8.18 Debug Control

Name: DBGCTRL
Offset: 0x1C
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN Debug Run
This bit is not reset by a software reset.

This bit controls the functionality when the CPU is halted by an external debugger.

This bit should be written only while a conversion is not ongoing.

Value	Description
0	The ADC is halted when the CPU is halted by an external debugger.
1	The ADC continues normal operation when the CPU is halted by an external debugger.

Downloaded from Arrow.com.

41.8.19 Synchronization Busy

Name: SYNCBUSY
Offset: 0x20
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
						SWTRIG	OFFSETCORR	GAINCORR
Access						R	R	R
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	WINUT	WINLT	SAMPCTRL	AVGCTRL	CTRLC	INPUTCTRL	ENABLE	SWRST
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 10 – SWTRIG Software Trigger Synchronization Busy
This bit is cleared when the synchronization of SWTRIG register between the clock domains is complete.

This bit is set when the synchronization of SWTRIG register between clock domains is started.

Bit 9 – OFFSETCORR Offset Correction Synchronization Busy
This bit is cleared when the synchronization of OFFSETCORR register between the clock domains is complete.

This bit is set when the synchronization of OFFSETCORR register between clock domains is started.

Bit 8 – GAINCORR Gain Correction Synchronization Busy
This bit is cleared when the synchronization of GAINCORR register between the clock domains is complete.

This bit is set when the synchronization of GAINCORR register between clock domains is started.

Bit 7 – WINUT Window Monitor Lower Threshold Synchronization Busy
This bit is cleared when the synchronization of WINUT register between the clock domains is complete.

This bit is set when the synchronization of WINUT register between clock domains is started.

Bit 6 – WINLT Window Monitor Upper Threshold Synchronization Busy
This bit is cleared when the synchronization of WINLT register between the clock domains is complete.

This bit is set when the synchronization of WINLT register between clock domains is started.

Bit 5 – SAMPCTRL Sampling Time Control Synchronization Busy
This bit is cleared when the synchronization of SAMPCTRL register between the clock domains is complete.

This bit is set when the synchronization of SAMPCTRL register between clock domains is started.

Downloaded from Arrow.com.

Bit 4 – AVGCTRL Average Control Synchronization Busy
This bit is cleared when the synchronization of AVGCTRL register between the clock domains is complete.

This bit is set when the synchronization of AVGCTRL register between clock domains is started.

Bit 3 – CTRLC Control C Synchronization Busy
This bit is cleared when the synchronization of CTRLC register between the clock domains is complete.

This bit is set when the synchronization of CTRLC register between clock domains is started.

Bit 2 – INPUTCTRL Input Control Synchronization Busy
This bit is cleared when the synchronization of INPUTCTRL register between the clock domains is complete.

This bit is set when the synchronization of INPUTCTRL register between clock domains is started.

Bit 1 – ENABLE ENABLE Synchronization Busy
This bit is cleared when the synchronization of ENABLE register between the clock domains is complete.

This bit is set when the synchronization of ENABLE register between clock domains is started.

Bit 0 – SWRST SWRST Synchronization Busy
This bit is cleared when the synchronization of SWRST register between the clock domains is complete.

This bit is set when the synchronization of SWRST register between clock domains is started

Downloaded from Arrow.com.

41.8.20 Result

Name: RESULT
Offset: 0x24
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
	RESULT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RESULT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RESULT[15:0] Result Conversion Value
These bits will hold up to a 16-bit ADC conversion result, depending on the configuration.

In single conversion mode without averaging, the ADC conversion will produce a 12-bit result, which can be left- or right-shifted, depending on the setting of CTRLC.LEFTADJ.

If the result is left-adjusted (CTRLC.LEFTADJ), the high byte of the result will be in bit position [15:8], while the remaining 4 bits of the result will be placed in bit locations [7:4]. This can be used only if an 8-bit result is needed; i.e., one can read only the high byte of the entire 16-bit register.

If the result is not left-adjusted (CTRLC.LEFTADJ) and no oversampling is used, the result will be available in bit locations [11:0], and the result is then 12 bits long. If oversampling is used, the result will be located in bit locations [15:0], depending on the settings of the Average Control register.

Downloaded from Arrow.com.

41.8.21 Sequence Control

Name: SEQCTRL
Offset: 0x28
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	SEQENn[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SEQENn[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SEQENn[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SEQENn[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – SEQENn[31:0] Enable Positive Input in the Sequence
For details on available positive mux selection, refer to INPUTCTRL.MUXENG.

The sequence start from the lowest input, and go to the next enabled input automatically when the conversion is done. If no bits are set the sequence is disabled.

Value	Description
0	Disable the positive input mux n selection from the sequence.
1	Enable the positive input mux n selection to the sequence.

Downloaded from Arrow.com.

41.8.22 Calibration

Name: CALIB
Offset: 0x2C
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
						BIASREFBUF[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
						BIASCOMP[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 10:8 – BIASREFBUF[2:0] Bias Reference Buffer Scaling
This value from production test must be loaded from the NVM software calibration row into the CALIB register by software to achieve the specified accuracy.

The value must be copied only, and must not be changed.

Bits 2:0 – BIASCOMP[2:0] Bias Comparator Scaling
This value from production test must be loaded from the NVM software calibration row into the CALIB register by software to achieve the specified accuracy.

The value must be copied only, and must not be changed

Downloaded from Arrow.com.

42. AC – Analog Comparators

42.1 Overview

The Analog Comparator (AC) supports two individual comparators. Each comparator (COMP) compares the voltage levels on two inputs, and provides a digital output based on this comparison. Each comparator may be configured to generate interrupt requests and/or peripheral events upon several different combinations of input change.

Hysteresis and propagation delay can be adjusted to achieve the optimal operation for each application.

The input selection includes four shared analog port pins and several internal signals. Each comparator output state can also be output on a pin for use by external devices.

The comparators are grouped in pairs on each port. The AC peripheral implements one pair of comparators . These are called Comparator 0 (COMP0) and Comparator 1 (COMP1) They have identical behaviors, but separate control registers. The pair can be set in window mode to compare a signal to a voltage range instead of a single voltage level.

42.2 Features

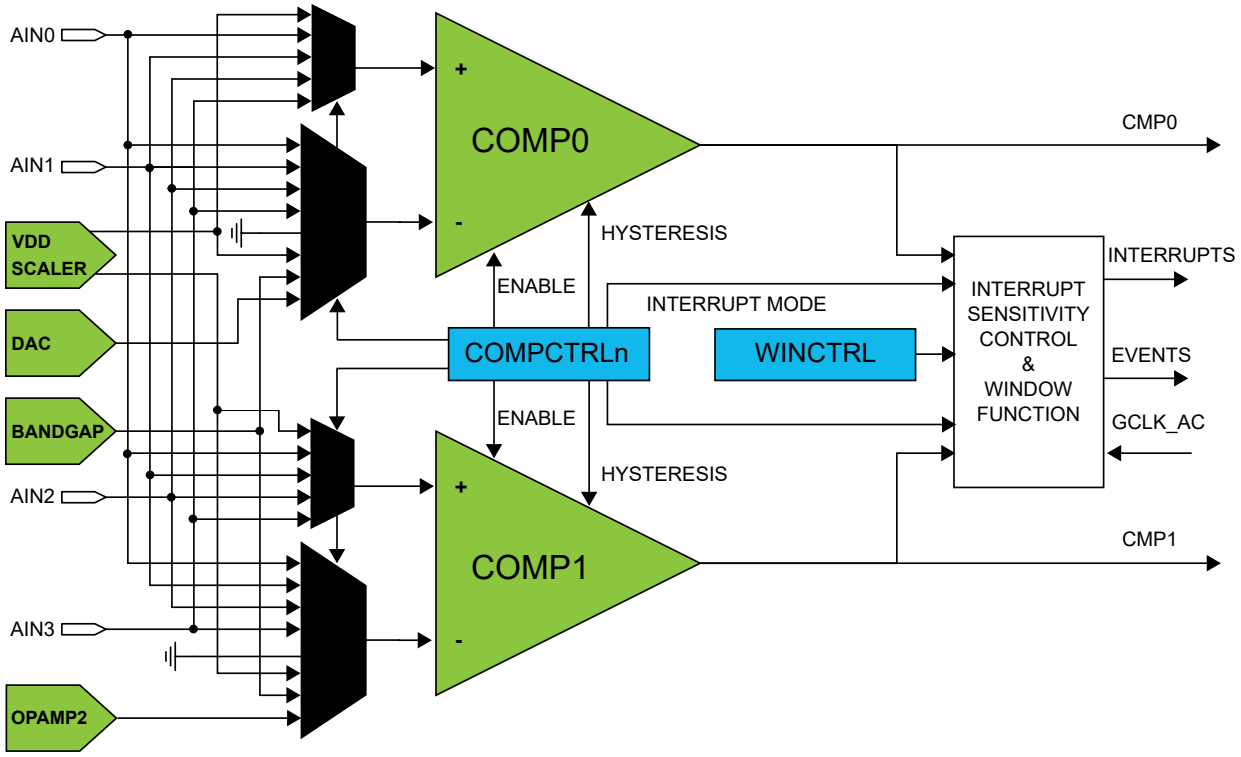
- Two individual comparators
- Selectable propagation delay versus current consumption
- Selectable hysteresis: 4-level On, or Off
- Analog comparator outputs available on pins
  - Asynchronous or synchronous
- Flexible input selection:
  - Four pins selectable for positive or negative inputs
  - Ground (for zero crossing)
  - Bandgap reference voltage
  - 64-level programmable VDD scaler per comparator
  - DAC
  - OPAMP2
- Interrupt generation on:
  - Rising or falling edge
  - Toggle
  - End of comparison
- Window function interrupt generation on:
  - Signal above window
  - Signal inside window
  - Signal below window
  - Signal outside window
- Event generation on:
  - Comparator output
  - Window function inside/outside window

Downloaded from Arrow.com.

- Optional digital filter on comparator output

42.3 Block Diagram

Figure 42-1. Analog Comparator Block Diagram



42.4 Signal Description

Signal	Description	Type
AIN[3..0]	Analog input	Comparator inputs
CMP[1..0]	Digital output	Comparator outputs

Refer to *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

42.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

42.5.1 I/O Lines

Using the AC's I/O lines requires the I/O pins to be configured. Refer to *PORT - I/O Pin Controller* for details.

Related Links

32. PORT - I/O Pin Controller

Downloaded from Arrow.com.

42.5.2 Power Management

The AC will continue to operate in any sleep mode where the selected source clock is running. The AC's interrupts can be used to wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes.

Related Links

22. PM – Power Manager

42.5.3 Clocks

The AC bus clock (CLK_AC_APB) can be enabled and disabled in the Main Clock module, MCLK (see *MCLK - Main Clock*, and the default state of CLK_AC_APB can be found in *Peripheral Clock Masking*.

A generic clock (GCLK_AC) is required to clock the AC. This clock must be configured and enabled in the generic clock controller before using the AC. Refer to the Generic Clock Controller chapter for details.

This generic clock is asynchronous to the bus clock (CLK_AC_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

Related Links

22. PM – Power Manager

42.5.4 DMA

Not applicable.

42.5.5 Interrupts

The interrupt request lines are connected to the interrupt controller. Using the AC interrupts requires the interrupt controller to be configured first. Refer to *Nested Vector Interrupt Controller* for details.

42.5.6 Events

The events are connected to the Event System. Refer to *EVSYS – Event System* for details on how to configure the Event System.

Related Links

33. EVSYS – Event System

42.5.7 Debug Operation

When the CPU is halted in debug mode, the AC will halt normal operation after any on-going comparison is completed. The AC can be forced to continue normal operation during debugging. Refer to DBGCTRL for details. If the AC is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

42.5.8 Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Control B register (CTRLB)
- Interrupt Flag register (INTFLAG)

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

Downloaded from Arrow.com.

Related Links

15. PAC - Peripheral Access Controller

42.5.9 SAM L11 TrustZone Specific Register Access Protection

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

42.5.10 Analog Connections

Each comparator has up to four I/O pins that can be used as analog inputs. Each pair of comparators shares the same four pins. These pins must be configured for analog operation before using them as comparator inputs.

Any internal reference source, such as a bandgap voltage reference, OPAMP2,or DAC must be configured and enabled prior to its use as a comparator input.

The analog signals of AC, ADC, DAC and OPAMP can be interconnected. The AC and ADC peripheral can request the OPAMP using an analog ONDEMAND functionality.

See *Analog Connections of Peripherals* for details.

42.6 Functional Description

42.6.1 Principle of Operation

Each comparator has one positive input and one negative input. Each positive input may be chosen from a selection of analog input pins. Each negative input may be chosen from a selection of both analog input pins and internal inputs, such as a bandgap voltage reference.

The digital output from the comparator is '1' when the difference between the positive and the negative input voltage is positive, and '0' otherwise.

The individual comparators can be used independently (normal mode) or paired to form a window comparison (window mode).

42.6.2 Basic Operation

42.6.2.1 Initialization

Some registers are enable-protected, meaning they can only be written when the module is disabled.

The following register is enable-protected:

- Event Control register (EVCTRL)

Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

42.6.2.2 Enabling, Disabling and Resetting

The AC is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The AC is disabled writing a '0' to CTRLA.ENABLE.

Downloaded from Arrow.com.

The AC is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the AC will be reset to their initial state, and the AC will be disabled. Refer to *CTRLA* for details.

42.6.2.3 Comparator Configuration

Each individual comparator must be configured by its respective Comparator Control register (COMPCTRLx) before that comparator is enabled. These settings cannot be changed while the comparator is enabled.

- Select the desired measurement mode with COMPCTRLx.SINGLE. See Starting a Comparison for more details.
- Select the desired hysteresis with COMPCTRLx.HYSTEN and COMPCTRLx.HYST. See Input Hysteresis for more details.
- Select the comparator speed versus power with COMPCTRLx.SPEED. See Propagation Delay vs. Power Consumption for more details.
- Select the interrupt source with COMPCTRLx.INTSEL.
- Select the positive and negative input sources with the COMPCTRLx.MUXPOS and COMPCTRLx.MUXNEG bits. See Selecting Comparator Inputs for more details.
- Select the filtering option with COMPCTRLx.FLEN.
- Select standby operation with Run in Standby bit (COMPCTRLx.RUNSTDBY).

The individual comparators are enabled by writing a '1' to the Enable bit in the Comparator x Control registers (COMPCTRLx.ENABLE). The individual comparators are disabled by writing a '0' to COMPCTRLx.ENABLE. Writing a '0' to CTRLA.ENABLE will also disable all the comparators, but will not clear their COMPCTRLx.ENABLE bits.

42.6.2.4 Starting a Comparison

Each comparator channel can be in one of two different measurement modes, determined by the Single bit in the Comparator x Control register (COMPCTRLx.SINGLE):

- Continuous measurement
- Single-shot

After being enabled, a start-up delay is required before the result of the comparison is ready. This start-up time is measured automatically to account for environmental changes, such as temperature or voltage supply level, and is specified in the *Electrical Characteristics* chapters. During the start-up time, the COMP output is not available.

The comparator can be configured to generate interrupts when the output toggles, when the output changes from '0' to '1' (rising edge), when the output changes from '1' to '0' (falling edge) or at the end of the comparison. An end-of-comparison interrupt can be used with the Single-Shot mode to chain further events in the system, regardless of the state of the comparator outputs. The Interrupt mode is set by the Interrupt Selection bit group in the Comparator Control register (COMPCTRLx.INTSEL). Events are generated using the comparator output state, regardless of whether the interrupt is enabled or not.

42.6.2.4.1 Continuous Measurement

Continuous measurement is selected by writing COMPCTRLx.SINGLE to zero. In continuous mode, the comparator is continuously enabled and performing comparisons. This ensures that the result of the latest comparison is always available in the Current State bit in the Status A register (STATUSA.STATEx).

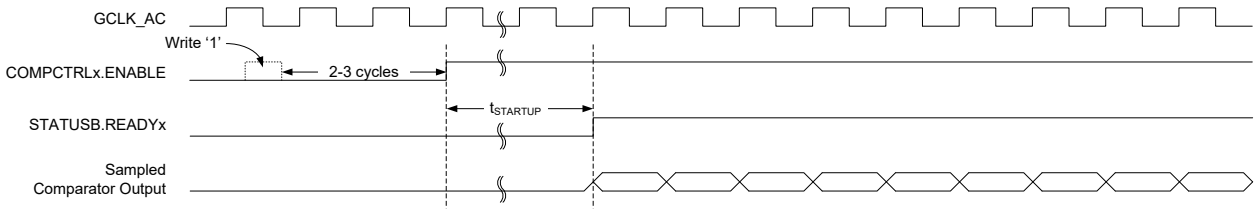
After the start-up time has passed, a comparison is done and STATUSA is updated. The Comparator x Ready bit in the Status B register (STATUSB.READYx) is set, and the appropriate peripheral events and

Downloaded from Arrow.com.

interrupts are also generated. New comparisons are performed continuously until the COMPCTRLx.ENABLE bit is written to zero. The start-up time applies only to the first comparison.

In continuous operation, edge detection of the comparator output for interrupts is done by comparing the current and previous sample. The sampling rate is the GCLK_AC frequency. An example of continuous measurement is shown in the Figure 42-2.

Figure 42-2. Continuous Measurement Example



For low-power operation, comparisons can be performed during sleep modes without a clock. The comparator is enabled continuously, and changes of the comparator state are detected asynchronously. When a toggle occurs, the Power Manager will start GCLK_AC to register the appropriate peripheral events and interrupts. The GCLK_AC clock is then disabled again automatically, unless configured to wake up the system from sleep.

42.6.2.4.2 Single-Shot

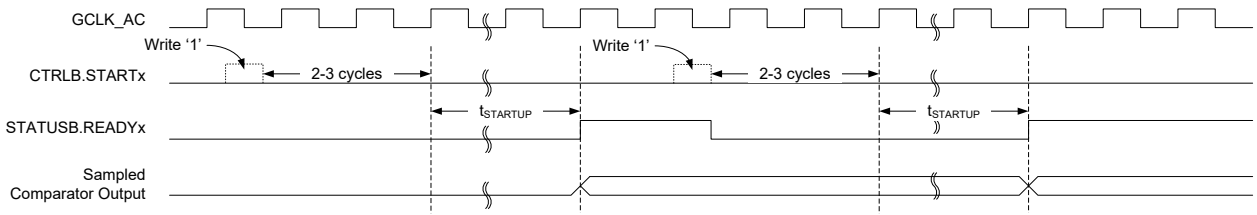
Single-shot operation is selected by writing COMPCTRLx.SINGLE to '1'. During single-shot operation, the comparator is normally idle. The user starts a single comparison by writing '1' to the respective Start Comparison bit in the write-only Control B register (CTRLB.STARTx). The comparator is enabled, and after the start-up time has passed, a single comparison is done and STATUSA is updated. Appropriate peripheral events and interrupts are also generated. No new comparisons will be performed.

Writing '1' to CTRLB.STARTx also clears the Comparator x Ready bit in the Status B register (STATUSB.READYx). STATUSB.READYx is set automatically by hardware when the single comparison has completed.

A single-shot measurement can also be triggered by the Event System. Setting the Comparator x Event Input bit in the Event Control Register (EVCTRL.COMPEIx) enables triggering on incoming peripheral events. Each comparator can be triggered independently by separate events. Event-triggered operation is similar to user-triggered operation; the difference is that a peripheral event from another hardware module causes the hardware to automatically start the comparison and clear STATUSB.READYx.

To detect an edge of the comparator output in single-shot operation for the purpose of interrupts, the result of the current measurement is compared with the result of the previous measurement (one sampling period earlier). An example of single-shot operation is shown in Figure 42-3.

Figure 42-3. Single-Shot Example



For low-power operation, event-triggered measurements can be performed during sleep modes. When the event occurs, the Power Manager will start GCLK_AC. The comparator is enabled, and after the startup time has passed, a comparison is done and appropriate peripheral events and interrupts are also generated. The comparator and GCLK_AC are then disabled again automatically, unless configured to wake up the system from sleep.

Downloaded from Arrow.com.

42.6.3 Selecting Comparator Inputs

Each comparator has one positive and one negative input. The positive input is one of the external input pins (AINx). The negative input can be fed either from an external input pin (AINx) or from one of the several internal reference voltage sources common to all comparators. The user selects the input source as follows:

- The positive input is selected by the Positive Input MUX Select bit group in the Comparator Control register (COMPCTRLx.MUXPOS)
- The negative input is selected by the Negative Input MUX Select bit group in the Comparator Control register (COMPCTRLx.MUXNEG)

In the case of using an external I/O pin, the selected pin must be configured for analog use in the PORT Controller by disabling the digital input and output. The switching of the analog input multiplexers is controlled to minimize crosstalk between the channels. The input selection must be changed only while the individual comparator is disabled.

Note: For internal use of the comparison results by the CCL, this bit must be 0x1 or 0x2.

42.6.4 Window Operation

Each comparator pair can be configured to work together in window mode. In this mode, a voltage range is defined, and the comparators give information about whether an input signal is within this range or not. Window mode is enabled by the Window Enable x bit in the Window Control register (WINCTRL.WENx). Both comparators in a pair must have the same measurement mode setting in their respective Comparator Control Registers (COMPCTRLx.SINGLE).

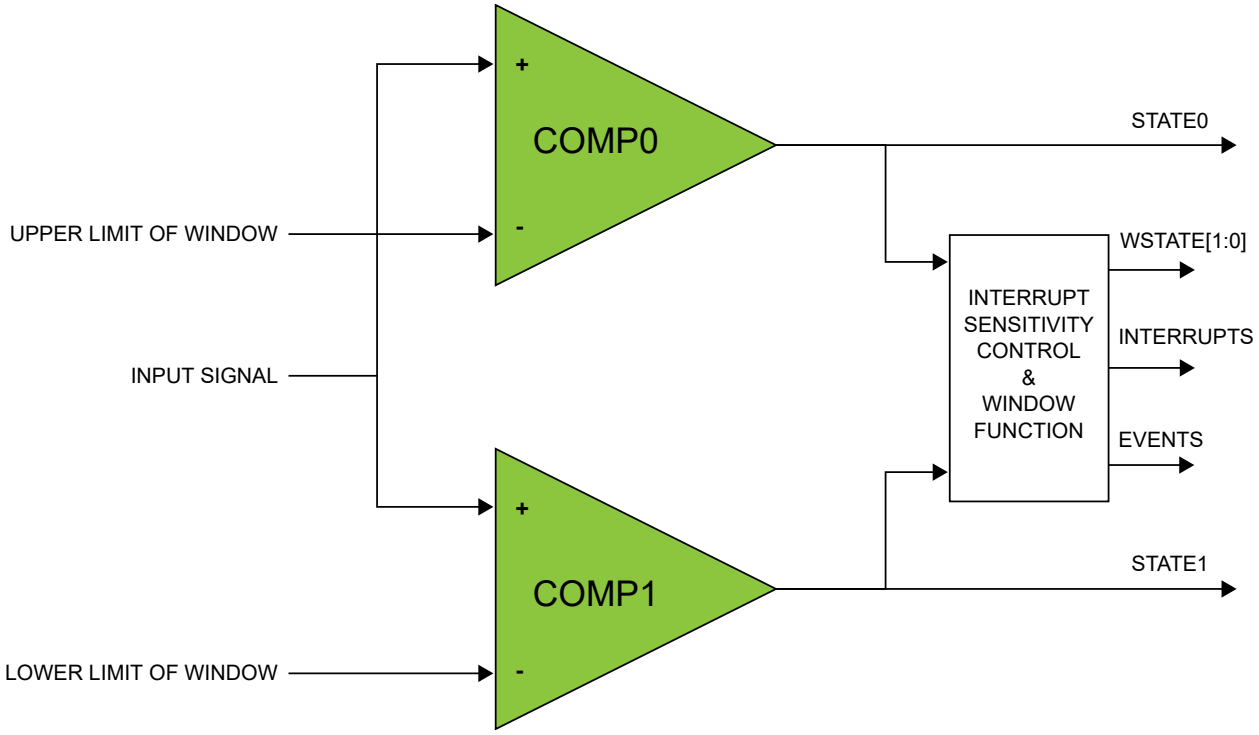
To physically configure the pair of comparators for window mode, the same I/O pin must be chosen as positive input for each comparator, providing a shared input signal. The negative inputs define the range for the window. In Figure 42-4, COMP0 defines the upper limit and COMP1 defines the lower limit of the window, as shown but the window will also work in the opposite configuration with COMP0 lower and COMP1 higher. The current state of the window function is available in the Window x State bit group of the Status register (STATUS.WSTATEx).

Window mode can be configured to generate interrupts when the input voltage changes to below the window, when the input voltage changes to above the window, when the input voltage changes into the window or when the input voltage changes outside the window. The interrupt selections are set by the Window Interrupt Selection bit field in the Window Control register (WINCTRL.WINTSEL). Events are generated using the inside/outside state of the window, regardless of whether the interrupt is enabled or not. Note that the individual comparator outputs, interrupts and events continue to function normally during window mode.

When the comparators are configured for window mode and single-shot mode, measurements are performed simultaneously on both comparators. Writing '1' to either Start Comparison bit in the Control B register (CTRLB.STARTx) will start a measurement. Likewise either peripheral event can start a measurement.

Downloaded from Arrow.com.

Figure 42-4. Comparators in Window Mode

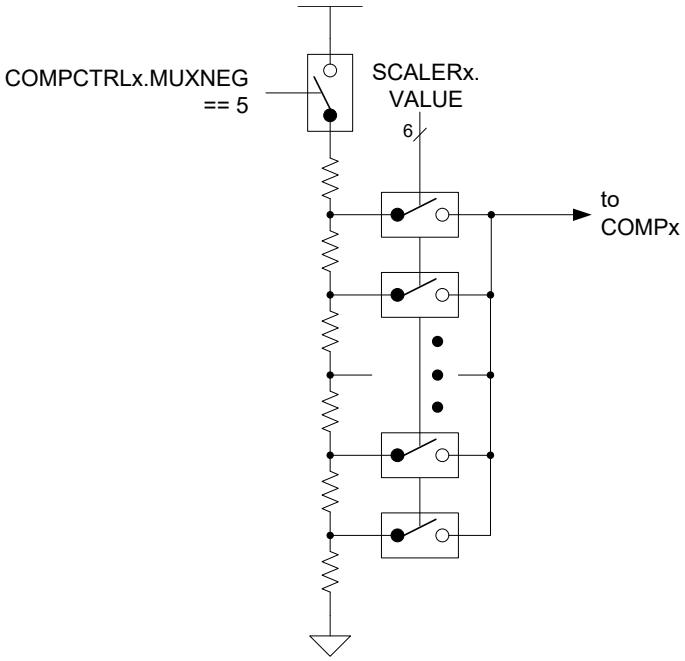


42.6.5 VDD Scaler

The VDD scaler generates a reference voltage that is a fraction of the device's supply voltage, with 64 levels. One independent voltage channel is dedicated for each comparator. The scaler of a comparator is enabled when the Negative Input Mux bit field in the respective Comparator Control register (COMPCTRLx.MUXNEG) is set to 0x5 and the comparator is enabled. The voltage of each channel is selected by the Value bit field in the Scaler x registers (SCALERx.VALUE).

Downloaded from Arrow.com.

Figure 42-5. VDD Scaler



42.6.6 Input Hysteresis

Application software can selectively enable/disable hysteresis for the comparison. Applying hysteresis will help prevent constant toggling of the output, which can be caused by noise when the input signals are close to each other.

Hysteresis is enabled for each comparator individually by the Hysteresis Enable bit in the Comparator x Control register (COMPCTRLx.HYSTEN). Furthermore, when enabled, the level of hysteresis is programmable through the Hysteresis Level bits also in the Comparator x Control register (COMPCTRLx.HYST). Hysteresis is available only in continuous mode (COMPCTRLx.SINGLE=0).

42.6.7 Propagation Delay vs. Power Consumption

It is possible to trade off comparison speed for power efficiency to get the shortest possible propagation delay or the lowest power consumption. The speed setting is configured for each comparator individually by the Speed bit group in the Comparator x Control register (COMPCTRLx.SPEED). The Speed bits select the amount of bias current provided to the comparator, and as such will also affect the start-up time.

42.6.8 Filtering

The output of the comparators can be filtered digitally to reduce noise. The filtering is determined by the Filter Length bits in the Comparator Control x register (COMPCTRLx.FLEN), and is independent for each comparator. Filtering is selectable from none, 3-bit majority (N=3) or 5-bit majority (N=5) functions. Any change in the comparator output is considered valid only if N/2+1 out of the last N samples agree. The filter sampling rate is the GCLK_AC frequency.

Note that filtering creates an additional delay of N-1 sampling cycles from when a comparison is started until the comparator output is validated. For continuous mode, the first valid output will occur when the required number of filter samples is taken. Subsequent outputs will be generated every cycle based on the current sample plus the previous N-1 samples, as shown in Figure 42-6. For single-shot mode, the comparison completes after the Nth filter sample, as shown in Figure 42-7.

Downloaded from Arrow.com.

Figure 42-6. Continuous Mode Filtering

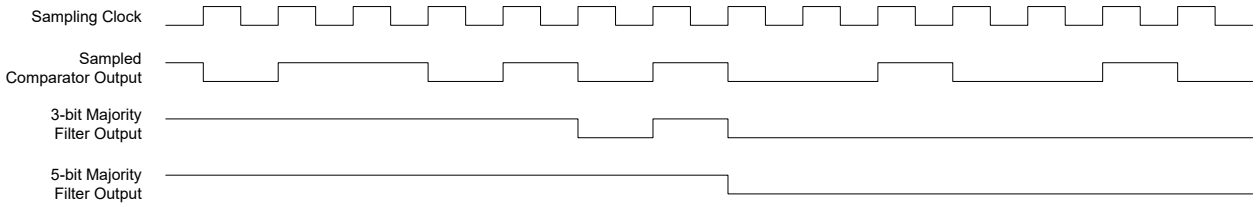
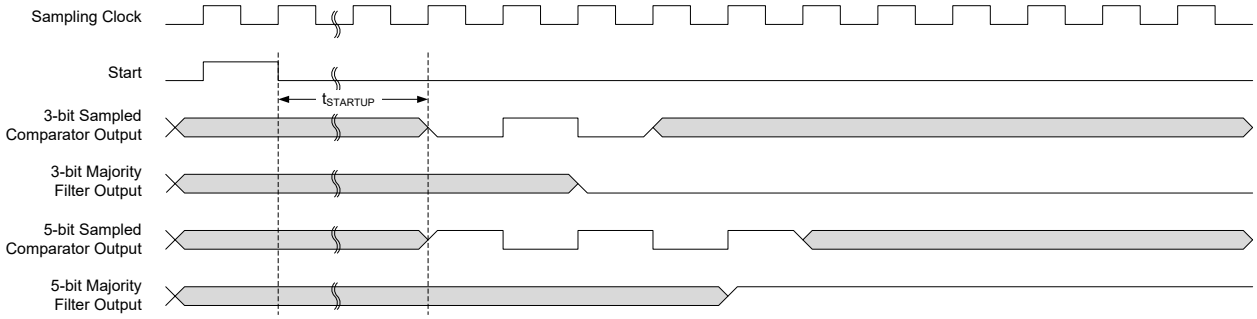


Figure 42-7. Single-Shot Filtering



During sleep modes, filtering is supported only for single-shot measurements. Filtering must be disabled if continuous measurements will be done during sleep modes, or the resulting interrupt/event may be generated incorrectly.

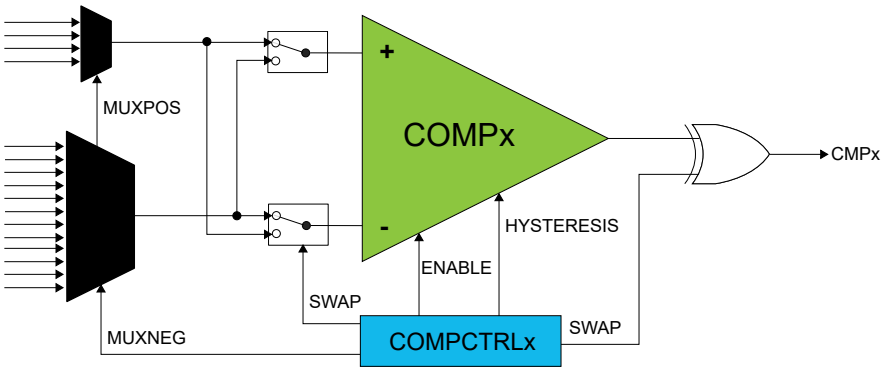
42.6.9 Comparator Output

The output of each comparator can be routed to an I/O pin by setting the Output bit group in the Comparator Control x register (COMPCTRLx.OUT). This allows the comparator to be used by external circuitry. Either the raw, non-synchronized output of the comparator or the CLK_AC-synchronized version, including filtering, can be used as the I/O signal source. The output appears on the corresponding CMP[x] pin.

42.6.10 Offset Compensation

The Swap bit in the Comparator Control registers (COMPCTRLx.SWAP) controls switching of the input signals to a comparator's positive and negative terminals. When the comparator terminals are swapped, the output signal from the comparator is also inverted, as shown in Figure 42-8. This allows the user to measure or compensate for the comparator input offset voltage. As part of the input selection, COMPCTRLx.SWAP can be changed only while the comparator is disabled.

Figure 42-8. Input Swapping for Offset Compensation



42.6.11 DMA Operation

Not applicable.

Downloaded from Arrow.com.

42.6.12 Interrupts

The AC has the following interrupt sources:

- Comparator (COMP0, COMP1): Indicates a change in comparator status.
- Window (WIN0): Indicates a change in the window status.

Comparator interrupts are generated based on the conditions selected by the Interrupt Selection bit group in the Comparator Control registers (COMPCTRLx.INTSEL). Window interrupts are generated based on the conditions selected by the Window Interrupt Selection bit group in the Window Control register (WINCTRL.WINTSEL[1:0]).

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a one to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a one to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register. An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the AC is reset. See INFLAG register for details on how to clear interrupt flags. All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. The user must read the INTFLAG register to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated.

42.6.13 Events

The AC can generate the following output events:

- Comparator (COMP0, COMP1): Generated as a copy of the comparator status
- Window (WIN0): Generated as a copy of the window inside/outside status

Writing a one to an Event Output bit in the Event Control Register (EVCTRL.xxEO) enables the corresponding output event. Writing a zero to this bit disables the corresponding output event. Refer to the Event System chapter for details on configuring the event system.

The AC can take the following action on an input event:

- Start comparison (START0, START1): Start a comparison.

Writing a one to an Event Input bit into the Event Control register (EVCTRL.COMPEIx) enables the corresponding action on input event. Writing a zero to this bit disables the corresponding action on input event. Note that if several events are connected to the AC, the enabled action will be taken on any of the incoming events. Refer to the Event System chapter for details on configuring the event system.

When EVCTRL.COMPEIx is one, the event will start a comparison on COMPx after the start-up time delay. In normal mode, each comparator responds to its corresponding input event independently. For a pair of comparators in window mode, either comparator event will trigger a comparison on both comparators simultaneously.

42.6.14 Sleep Mode Operation

The Run in Standby bits in the Comparator x Control registers (COMPCTRLx.RUNSTDBY) control the behavior of the AC during standby sleep mode. Each RUNSTDBY bit controls one comparator. When the bit is zero, the comparator is disabled during sleep, but maintains its current configuration. When the bit is one, the comparator continues to operate during sleep. Note that when RUNSTDBY is zero, the analog blocks are powered off for the lowest power consumption. This necessitates a start-up time delay when the system returns from sleep.

Downloaded from Arrow.com.

For Window Mode operation, both comparators in a pair must have the same RUNSTDBY configuration.

When RUNSTDBY is one, any enabled AC interrupt source can wake up the CPU. The AC can also be used during sleep modes where the clock used by the AC is disabled, provided that the AC is still powered (not in shutdown). In this case, the behavior is slightly different and depends on the measurement mode, as listed in Table 42-1.

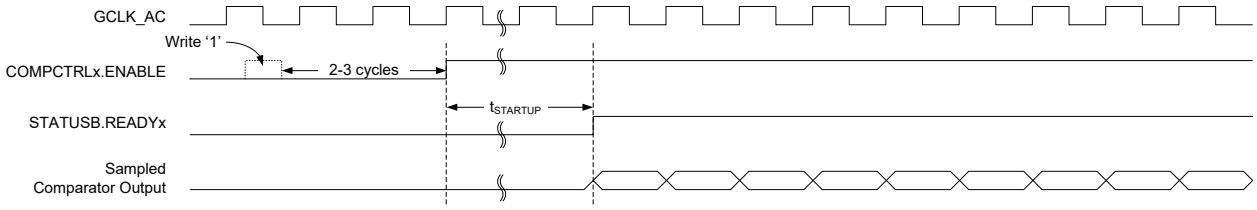
Table 42-1. Sleep Mode Operation

COMPCTRLx.MODE	RUNSTDBY=0	RUNSTDBY=1
0 (Continuous)	COMPx disabled	GCLK_AC stopped, COMPx enabled
1 (Single-shot)	COMPx disabled	GCLK_AC stopped, COMPx enabled only when triggered by an input event

42.6.14.1 Continuous Measurement during Sleep

When a comparator is enabled in continuous measurement mode and GCLK_AC is disabled during sleep, the comparator will remain continuously enabled and will function asynchronously. The current state of the comparator is asynchronously monitored for changes. If an edge matching the interrupt condition is found, GCLK_AC is started to register the interrupt condition and generate events. If the interrupt is enabled in the Interrupt Enable registers (INTENCLR/SET), the AC can wake up the device; otherwise GCLK_AC is disabled until the next edge detection. Filtering is not possible with this configuration.

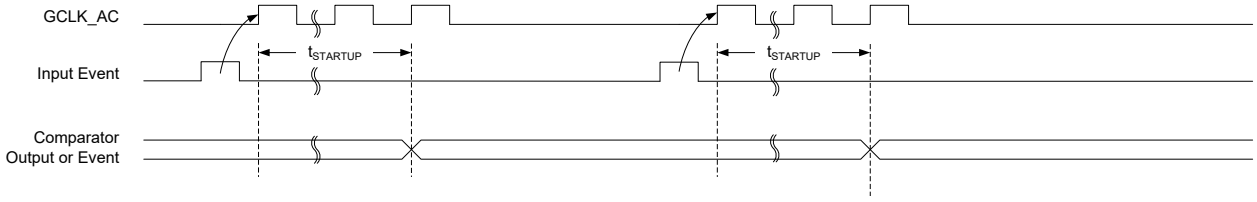
Figure 42-9. Continuous Mode SleepWalking



42.6.14.2 Single-Shot Measurement during Sleep

For low-power operation, event-triggered measurements can be performed during sleep modes. When the event occurs, the Power Manager will start GCLK_AC. The comparator is enabled, and after the start-up time has passed, a comparison is done, with filtering if desired, and the appropriate peripheral events and interrupts are also generated, as shown in Figure 42-10. The comparator and GCLK_AC are then disabled again automatically, unless configured to wake the system from sleep. Filtering is allowed with this configuration.

Figure 42-10. Single-Shot SleepWalking



42.6.15 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

Downloaded from Arrow.com.

- Software Reset bit in control register (CTRLA.SWRST)
- Enable bit in control register (CTRLA.ENABLE)
- Enable bit in Comparator Control register (COMPCTRLn.ENABLE)

The following registers are synchronized when written:

- Window Control register (WINCTRL)

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Downloaded from Arrow.com.

42.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0							ENABLE	SWRST
0x01	CTRLB	7:0							STARTx	STARTx
0x02	EVCTRL	7:0				WINEO0			COMPEOx	COMPEOx
		15:8			INVEIx	INVEIx			COMPEIx	COMPEIx
0x04	INTENCLR	7:0				WIN0			COMPx	COMPx
0x05	INTENSET	7:0				WIN0			COMPx	COMPx
0x06	INTFLAG	7:0				WIN0			COMPx	COMPx
0x07	STATUSA	7:0			WSTATE0[1:0]				STATEx	STATEx
0x08	STATUSB	7:0							READYx	READYx
0x09	DBGCTRL	7:0								DBGRUN
0x0A	WINCTRL	7:0						WINTSEL0[1:0]		WEN0
0x0B	Reserved									
0x0C	SCALER0	7:0			VALUE[5:0]					
0x0D	SCALER1	7:0			VALUE[5:0]					
0x0E ... 0x0F	Reserved									
0x10	COMPCTRL0	7:0		RUNSTDBY		INTSEL[1:0]		SINGLE	ENABLE	
		15:8	SWAP	MUXPOS[2:0]				MUXNEG[2:0]		
		23:16			HYST[1:0]		HYSTEN		SPEED[1:0]	
		31:24			OUT[1:0]			FLEN[2:0]		
0x14	COMPCTRL1	7:0		RUNSTDBY		INTSEL[1:0]		SINGLE	ENABLE	
		15:8	SWAP	MUXPOS[2:0]				MUXNEG[2:0]		
		23:16			HYST[1:0]		HYSTEN		SPEED[1:0]	
		31:24			OUT[1:0]			FLEN[2:0]		
0x18 ... 0x1F	Reserved									
0x20	SYNCBUSY	7:0				COMPCTRLx	COMPCTRLx	WINCTRL	ENABLE	SWRST
		15:8								
		23:16								
		31:24								

42.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

Downloaded from Arrow.com.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to Synchronization.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

Downloaded from Arrow.com.

42.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
							ENABLE	SWRST
Access							R/W	W
Reset							0	0

Bit 1 – ENABLE Enable
Due to synchronization, there is delay from updating the register until the peripheral is enabled/disabled. The value written to CTRL.ENABLE will read back immediately and the corresponding bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE is cleared when the peripheral is enabled/disabled.

Value	Description
0	The AC is disabled.
1	The AC is enabled. Each comparator must also be enabled individually by the Enable bit in the Comparator Control register (COMPCTRLn.ENABLE).

Bit 0 – SWRST Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the AC to their initial state, and the AC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Due to synchronization, there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

Downloaded from Arrow.com.

42.8.2 Control B

Name: CTRLB
Offset: 0x01
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
							STARTx	STARTx
Access							R/W	R/W
Reset							0	0

Bits 1,0 – STARTx Comparator x Start Comparison
Writing a '0' to this field has no effect.

Writing a '1' to STARTx starts a single-shot comparison on COMPx if both the Single-Shot and Enable bits in the Comparator x Control Register are '1' (COMPCTRLx.SINGLE and COMPCTRLx.ENABLE). If comparator x is not implemented, or if it is not enabled in single-shot mode, Writing a '1' has no effect.

This bit always reads as zero.

Downloaded from Arrow.com.

42.8.3 Event Control

Name: EVCTRL
Offset: 0x02
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
			INVEIx	INVEIx			COMPEIx	COMPEIx
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	7	6	5	4	3	2	1	0
				WINEO0			COMPEOx	COMPEOx
Access				R/W			R/W	R/W
Reset				0			0	0

Bits 13,12 – INVEIx Inverted Event Input Enable x

Value	Description
0	Incoming event is not inverted for comparator x.
1	Incoming event is inverted for comparator x.

Bits 9,8 – COMPEIx Comparator x Event Input
Note that several actions can be enabled for incoming events. If several events are connected to the peripheral, the enabled action will be taken for any of the incoming events. There is no way to tell which of the incoming events caused the action.

These bits indicate whether a comparison will start or not on any incoming event.

Value	Description
0	Comparison will not start on any incoming event.
1	Comparison will start on any incoming event.

Bit 4 – WINEO0 Window 0 Event Output Enable
These bits indicate whether the window 0 function can generate a peripheral event or not.

Value	Description
0	Window 0 Event is disabled.
1	Window 0 Event is enabled.

Bits 1,0 – COMPEOx Comparator x Event Output Enable
These bits indicate whether the comparator x output can generate a peripheral event or not.

Value	Description
0	COMPx event generation is disabled.
1	COMPx event generation is enabled.

Downloaded from Arrow.com.

42.8.4 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Bit	7	6	5	4	3	2	1	0
				WIN0			COMPx	COMPx
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – WIN0 Window 0 Interrupt Enable
Reading this bit returns the state of the Window 0 interrupt enable.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit disables the Window 0 interrupt.

Value	Description
0	The Window 0 interrupt is disabled.
1	The Window 0 interrupt is enabled.

Bits 1,0 – COMPx Comparator x Interrupt Enable
Reading this bit returns the state of the Comparator x interrupt enable.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit disables the Comparator x interrupt.

Value	Description
0	The Comparator x interrupt is disabled.
1	The Comparator x interrupt is enabled.

Downloaded from Arrow.com.

42.8.5 Interrupt Enable Set

Name: INTENSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Bit	7	6	5	4	3	2	1	0
				WIN0			COMPx	COMPx
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – WIN0 Window 0 Interrupt Enable
Reading this bit returns the state of the Window 0 interrupt enable.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit enables the Window 0 interrupt.

Value	Description
0	The Window 0 interrupt is disabled.
1	The Window 0 interrupt is enabled.

Bits 1,0 – COMPx Comparator x Interrupt Enable
Reading this bit returns the state of the Comparator x interrupt enable.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Ready interrupt bit and enable the Ready interrupt.

Value	Description
0	The Comparator x interrupt is disabled.
1	The Comparator x interrupt is enabled.

Downloaded from Arrow.com.

42.8.6 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x06
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
				WIN0			COMPx	COMPx
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – WIN0 Window 0
This flag is set according to the Window 0 Interrupt Selection bit group in the WINCTRL register (WINCTRL.WINTSELx) and will generate an interrupt if INTENCLR/SET.WINx is also one.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Window 0 interrupt flag.

Bits 1,0 – COMPx Comparator x
Reading this bit returns the status of the Comparator x interrupt flag. If comparator x is not implemented, COMPx always reads as zero.

This flag is set according to the Interrupt Selection bit group in the Comparator x Control register (COMPCTRLx.INTSEL) and will generate an interrupt if INTENCLR/SET.COMPx is also one.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Comparator x interrupt flag.

Downloaded from Arrow.com.

42.8.7 Status A

Name: STATUSA
Offset: 0x07
Reset: 0x00
Property: Read-Only

Bit	7	6	5	4	3	2	1	0
			WSTATE0[1:0]				STATEx	STATEx
Access			R	R			R	R
Reset			0	0			0	0

Bits 5:4 – WSTATE0[1:0] Window 0 Current State
These bits show the current state of the signal if the window 0 mode is enabled.

Value	Name	Description
0x0	ABOVE	Signal is above window
0x1	INSIDE	Signal is inside window
0x2	BELOW	Signal is below window
0x3		Reserved

Bits 1,0 – STATEx Comparator x Current State
This bit shows the current state of the output signal from COMPx. STATEx is valid only when STATUSB.READYx is one.

Downloaded from Arrow.com.

42.8.8 Status B

Name: STATUSB
Offset: 0x08
Reset: 0x00
Property: Read-Only

Bit	7	6	5	4	3	2	1	0
							READYx	READYx
Access							R	R
Reset							0	0

Bits 1,0 – READYx Comparator x Ready
This bit is cleared when the comparator x output is not ready.
This bit is set when the comparator x output is ready.

Downloaded from Arrow.com.

42.8.9 Debug Control

Name: DBGCTRL
Offset: 0x09
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN Debug Run
This bit is not reset by a software reset.

This bits controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The AC is halted when the CPU is halted by an external debugger. Any on-going comparison will complete.
1	The AC continues normal operation when the CPU is halted by an external debugger.

Downloaded from Arrow.com.

42.8.10 Window Control

Name: WINCTRL
Offset: 0x0A
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
						WINTSEL0[1:0]		WEN0
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:1 – WINTSEL0[1:0] Window 0 Interrupt Selection
These bits configure the interrupt mode for the comparator window 0 mode.

Value	Name	Description
0x0	ABOVE	Interrupt on signal above window
0x1	INSIDE	Interrupt on signal inside window
0x2	BELOW	Interrupt on signal below window
0x3	OUTSIDE	Interrupt on signal outside window

Bit 0 – WEN0 Window 0 Mode Enable

Value	Description
0	Window mode is disabled for comparators 0 and 1.
1	Window mode is enabled for comparators 0 and 1.

Downloaded from Arrow.com.

42.8.11 Scaler n

Name: SCALER
Offset: 0x0C + n*0x01 [n=0..1]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
			VALUE[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 5:0 – VALUE[5:0] Scaler Value
These bits define the scaling factor for channel n of the V_{DD} voltage scaler. The output voltage, V_{SCALE}, is:

$$V_{SCALE} = \frac{V_{DD} \cdot (VALUE+1)}{64}$$

Downloaded from Arrow.com.

42.8.12 Comparator Control n

Name: COMPCTRL
Offset: 0x10 + n*0x04 [n=0..1]
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
			OUT[1:0]			FLEN[2:0]		
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bit	23	22	21	20	19	18	17	16
			HYST[1:0]		HYSTEN		SPEED[1:0]	
Access			R/W	R/W	R/W		R/W	R/W
Reset			0	0	0		0	0

Bit	15	14	13	12	11	10	9	8
	SWAP	MUXPOS[2:0]				MUXNEG[2:0]		
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY		INTSEL[1:0]		SINGLE	ENABLE	
Access		R/W		R/W	R/W	R/W	R/W	
Reset		0		0	0	0	0	

Bits 29:28 – OUT[1:0] Output
These bits configure the output selection for comparator n. COMPCTRLn.OUT can be written only while COMPCTRLn.ENABLE is zero.

Note: For internal use of the comparison results by the CCL, this bit must be 0x1 or 0x2.

These bits are not synchronized.

Value	Name	Description
0x0	OFF	The output of COMPn is not routed to the COMPn I/O port
0x1	ASYNC	The asynchronous output of COMPn is routed to the COMPn I/O port
0x2	SYNC	The synchronous output (including filtering) of COMPn is routed to the COMPn I/O port
0x3	N/A	Reserved

Bits 26:24 – FLEN[2:0] Filter Length
These bits configure the filtering for comparator n. COMPCTRLn.FLEN can only be written while COMPCTRLn.ENABLE is zero.

These bits are not synchronized.

Downloaded from Arrow.com.

Value	Name	Description
0x0	OFF	No filtering
0x1	MAJ3	3-bit majority function (2 of 3)
0x2	MAJ5	5-bit majority function (3 of 5)
0x3-0x7	N/A	Reserved

Bits 21:20 – HYST[1:0] Hysteresis Level
These bits indicate the hysteresis level of comparator n when hysteresis is enabled (COMPCTRLn.HYSTEN=1). Hysteresis is available only for continuous mode (COMPCTRLn.SINGLE=0). COMPCTRLn.HYST can be written only while COMPCTRLn.ENABLE is zero.

These bits are not synchronized.

Value	Name	Description
0x0	HYST50	50mV
0x1	HYST70	70mV
0x2	HYST90	90mV
0x3	HYST110	110mV

Bit 19 – HYSTEN Hysteresis Enable
This bit indicates the hysteresis mode of comparator n. Hysteresis is available only for continuous mode (COMPCTRLn.SINGLE=0). COMPCTRLn.HYST can be written only while COMPCTRLn.ENABLE is zero.

This bit is not synchronized.

Value	Description
0	Hysteresis is disabled.
1	Hysteresis is enabled.

Bits 17:16 – SPEED[1:0] Speed Selection
This bit indicates the speed/propagation delay mode of comparator n. COMPCTRLn.SPEED can be written only while COMPCTRLn.ENABLE is zero.

These bits are not synchronized.

Value	Name	Description
0x0	LOW	Low speed
0x1	MEDLOW	Medium low speed
0x2	MEDHIGH	Medium high speed
0x3	HIGH	High speed

Bit 15 – SWAP Swap Inputs and Invert
This bit swaps the positive and negative inputs to COMPn and inverts the output. This function can be used for offset cancellation. COMPCTRLn.SWAP can be written only while COMPCTRLn.ENABLE is zero.

These bits are not synchronized.

Downloaded from Arrow.com.

Value	Description
0	The output of MUXPOS connects to the positive input, and the output of MUXNEG connects to the negative input.
1	The output of MUXNEG connects to the positive input, and the output of MUXPOS connects to the negative input.

Bits 14:12 – MUXPOS[2:0] Positive Input Mux Selection
These bits select which input will be connected to the positive input of comparator n. COMPCTRLn.MUXPOS can be written only while COMPCTRLn.ENABLE is zero.

These bits are not synchronized.

Value	Name	Description
0x0	PIN0	I/O pin 0
0x1	PIN1	I/O pin 1
0x2	PIN2	I/O pin 2
0x3	PIN3	I/O pin 3
0x4	VSCALE	VDD scaler
0x5-0x7	-	Reserved

Bits 10:8 – MUXNEG[2:0] Negative Input Mux Selection
These bits select which input will be connected to the negative input of comparator n. COMPCTRLn.MUXNEG can only be written while COMPCTRLn.ENABLE is zero.

These bits are not synchronized.

Value	Name	Description
0x0	PIN0	I/O pin 0
0x1	PIN1	I/O pin 1
0x2	PIN2	I/O pin 2
0x3	PIN3	I/O pin 3
0x4	GND	Ground
0x5	VSCALE	VDD scaler
0x6	BANDGAP	Internal bandgap voltage
0x7	DAC / OPAMP	DAC0 output (COMP0) OPAMP2 output (COMP1)

Bit 6 – RUNSTDBY Run in Standby
This bit controls the behavior of the comparator during standby sleep mode.

This bit is not synchronized

Value	Description
0	The comparator is disabled during sleep.
1	The comparator continues to operate during sleep.

Bits 4:3 – INTSEL[1:0] Interrupt Selection
These bits select the condition for comparator n to generate an interrupt or event. COMPCTRLn.INTSEL can be written only while COMPCTRLn.ENABLE is zero.

These bits are not synchronized.

Downloaded from Arrow.com.

Value	Name	Description
0x0	TOGGLE	Interrupt on comparator output toggle
0x1	RISING	Interrupt on comparator output rising
0x2	FALLING	Interrupt on comparator output falling
0x3	EOC	Interrupt on end of comparison (single-shot mode only)

Bit 2 – SINGLE Single-Shot Mode
This bit determines the operation of comparator n. COMPCTRLn.SINGLE can be written only while COMPCTRLn.ENABLE is zero.

These bits are not synchronized.

Value	Description
0	Comparator n operates in continuous measurement mode.
1	Comparator n operates in single-shot mode.

Bit 1 – ENABLE Enable
Writing a zero to this bit disables comparator n.
Writing a one to this bit enables comparator n.

Due to synchronization, there is delay from updating the register until the comparator is enabled/disabled. The value written to COMPCTRLn.ENABLE will read back immediately after being written. SYNCBUSY.COMPCTRLn is set. SYNCBUSY.COMPCTRLn is cleared when the peripheral is enabled/disabled.

Writing a one to COMPCTRLn.ENABLE will prevent further changes to the other bits in COMPCTRLn. These bits remain protected until COMPCTRLn.ENABLE is written to zero and the write is synchronized.

Downloaded from Arrow.com.

42.8.13 Synchronization Busy

Name: SYNCBUSY
Offset: 0x20
Reset: 0x00000000
Property: Read-Only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				COMPCTRLx	COMPCTRLx	WINCTRL	ENABLE	SWRST
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bits 4,3 – COMPCTRLx COMPCTRLx Synchronization Busy
This bit is cleared when the synchronization of the COMPCTRLx register between the clock domains is complete.

This bit is set when the synchronization of the COMPCTRLx register between clock domains is started.

Bit 2 – WINCTRL WINCTRL Synchronization Busy
This bit is cleared when the synchronization of the WINCTRL register between the clock domains is complete.

This bit is set when the synchronization of the WINCTRL register between clock domains is started.

Bit 1 – ENABLE Enable Synchronization Busy
This bit is cleared when the synchronization of the CTRLA.ENABLE bit between the clock domains is complete.

This bit is set when the synchronization of the CTRLA.ENABLE bit between clock domains is started.

Bit 0 – SWRST Software Reset Synchronization Busy
This bit is cleared when the synchronization of the CTRLA.SWRST bit between the clock domains is complete.

This bit is set when the synchronization of the CTRLA.SWRST bit between clock domains is started.

Downloaded from Arrow.com.

43. DAC – Digital-to-Analog Converter

43.1 Overview

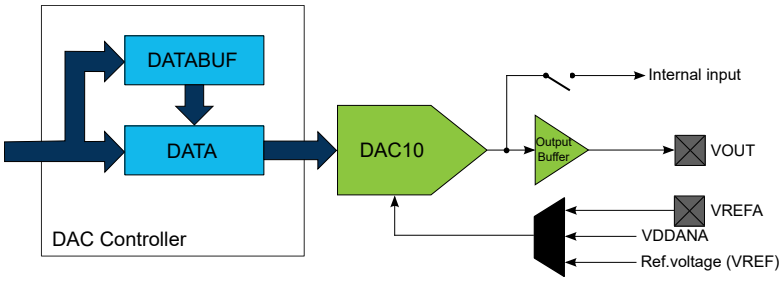
The Digital-to-Analog Converter (DAC) converts a digital value to a voltage. The DAC has one channel with 10-bit resolution, and it is capable of converting up to 350,000 samples per second (350ksps).

43.2 Features

- DAC with 10-bit resolution
- Up to 350ksps conversion rate
- Hardware support for 14-bit using dithering
- Multiple trigger sources
- High-drive capabilities
- Output can be used as input to the Analog Comparator (AC), ADC
- DMA support

43.3 Block Diagram

Figure 43-1. DAC Block Diagram



43.4 Signal Description

Signal Name	Type	Description
VOUT	Analog output	DAC output
VREFA	Analog input	External reference

43.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

43.5.1 I/O Lines

Using the DAC Controller's I/O lines requires the I/O pins to be configured using the port configuration (PORT).

Related Links

Downloaded from Arrow.com.

32. PORT - I/O Pin Controller

43.5.2 Power Management

The DAC will continue to operate in any Sleep mode where the selected source clock is running.

The DAC interrupts can be used to wake up the device from sleep modes.

Events connected to the event system can trigger other operations in the system without exiting sleep modes.

Related Links

22. PM – Power Manager

43.5.3 Clocks

The DAC bus clock (CLK_DAC_APB) can be enabled and disabled by the Main Clock module, and the default state of CLK_DAC_APB can be found in the *Peripheral Clock Masking* section.

A generic clock (GCLK_DAC) is required to clock the DAC Controller. This clock must be configured and enabled in the Generic Clock Controller before using the DAC Controller. Refer to *GCLK – Generic Clock Controller* for details.

This generic clock is asynchronous to the bus clock (CLK_DAC_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to 43.6.7 Synchronization for further details.

Related Links

18. GCLK - Generic Clock Controller

43.5.4 DMA

The DMA request line is connected to the DMA Controller (DMAC). Using the DAC Controller DMA requests requires to configure the DMAC first.

Related Links

28. DMAC – Direct Memory Access Controller

43.5.5 Interrupts

The interrupt request line is connected to the interrupt controller. Using the DAC Controller interrupt(s) requires the interrupt controller to be configured first.

43.5.6 Events

The events are connected to the Event System.

Related Links

33. EVSYS – Event System

43.5.7 Debug Operation

When the CPU is halted in debug mode the DAC will halt normal operation. Any on-going conversions will be completed. The DAC can be forced to continue normal operation during debugging. If the DAC is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

43.5.8 Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except the following registers:

Downloaded from Arrow.com.

- Interrupt Flag Status and Clear (INTFLAG) register
- Data Buffer (DATABUF) register

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger

Related Links

15. PAC - Peripheral Access Controller

43.5.9 SAM L11 TrustZone Specific Register Access Protection

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

43.5.10 Analog Connections

The DAC has one output pin (VOUT) and one analog input pin (VREFA) that must be configured first.

When internal input is used, it must be enabled before DAC Controller is enabled.

43.6 Functional Description

43.6.1 Principle of Operation

The DAC converts the digital value located in the Data register (DATA) into an analog voltage on the DAC output (VOUT).

A conversion is started when new data is written to the Data register. The resulting voltage is available on the DAC output after the conversion time. A conversion can also be started by input events from the Event System.

43.6.2 Basic Operation

43.6.2.1 Initialization

The following registers are enable-protected, meaning they can only be written when the DAC is disabled (CTRLA.ENABLE is zero):

- Control B register (CTRLB)
- Event Control register (EVCTRL)

Enable-protection is denoted by the Enable-Protected property in the register description.

Before enabling the DAC, it must be configured by selecting the voltage reference using the Reference Selection bits in the Control B register (CTRLB.REFSEL).

43.6.2.2 Enabling, Disabling and Resetting

The DAC Controller is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The DAC Controller is disabled by writing a '0' to CTRLA.ENABLE.

Downloaded from Arrow.com.

The DAC Controller is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the DAC will be reset to their initial state, and the DAC Controller will be disabled. Refer to the CTRLA register for details.

43.6.2.3 Enabling the Output Buffer

To enable the DAC output on the V_{OUT} pin, the output driver must be enabled by writing a one to the External Output Enable bit in the Control B register (CTRLB.EOEN).

The DAC output buffer provides a high-drive-strength output, and is capable of driving both resistive and capacitive loads. To minimize power consumption, the output buffer should be enabled only when external output is needed.

43.6.2.4 Digital to Analog Conversion

The DAC converts a digital value (stored in the DATA register) into an analog voltage. The conversion range is between GND and the selected DAC voltage reference. The default voltage reference is the internal reference voltage. Other voltage reference options are the analog supply voltage (VDDANA) and the external voltage reference (VREFA). The voltage reference is selected by writing to the Reference Selection bits in the Control B register (CTRLB.REFSEL).

The output voltage from the DAC can be calculated using the following formula:

$$V_{\mathrm{OUT}} = \frac{\mathrm{DATA}}{0x3\mathrm{FF}} \cdot \mathrm{VREF}$$

A new conversion starts as soon as a new value is loaded into DATA. DATA can either be loaded via the APB bus during a CPU write operation, using DMA, or from the DATABUF register when a START event occurs. Refer to 43.6.5 Events for details. As there is no automatic indication that a conversion is done, the sampling period must be greater than or equal to the specified conversion time.

43.6.3 DMA Operation

The DAC generates the following DMA request:

- Data Buffer Empty (EMPTY): The request is set when data is transferred from DATABUF to the internal data buffer of DAC. The request is cleared when DATABUF register is written, or by writing a one to the EMPTY bit in the Interrupt Flag register (INTFLAG.EMPTY).

For each Start Conversion event, DATABUF is transferred into DATA and the conversion starts. When DATABUF is empty, the DAC generates the DMA request for new data. As DATABUF is initially empty, a DMA request is generated whenever the DAC is enabled.

If the CPU accesses the registers that are the source of a DMA request set/clear condition, the DMA request can be lost or the DMA transfer can be corrupted, if enabled.

43.6.4 Interrupts

The DAC Controller has the following interrupt sources:

- Data Buffer Empty (EMPTY): Indicates that the internal data buffer of the DAC is empty.
- Underrun (UNDERRUN): Indicates that the internal data buffer of the DAC is empty and a DAC start of conversion event occurred. Refer to 43.6.5 Events for details.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a one to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing a one to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

Downloaded from Arrow.com.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled or the DAC is reset. See INTFLAG register for details on how to clear interrupt flags.

All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. The user must read the INTFLAG register to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated..

43.6.5 Events

The DAC Controller can generate the following output events:

- Data Buffer Empty (EMPTY): Generated when the internal data buffer of the DAC is empty. Refer to DMA Operation for details.

Writing a '1' to an Event Output bit in the Event Control register (EVCTRL.EMPTYEO) enables the corresponding output event. Writing a '0' to this bit disables the corresponding output event.

The DAC can take the following action on an input event:

- Start Conversion (START): DATABUF value is transferred into DATA as soon as the DAC is ready for the next conversion, and then conversion is started. START is considered as asynchronous to GCLK_DAC thus it is resynchronized in DAC Controller. Refer to 43.6.2.4 Digital to Analog Conversion for details.

Writing a '1' to an Event Input bit in the Event Control register (EVCTRL.STARTEI) enables the corresponding action on an input event. Writing a '0' to this bit disables the corresponding action on input event.

Note: When several events are connected to the DAC Controller, the enabled action will be taken on any of the incoming events.

By default, DAC Controller detects rising edge events. Falling edge detection can be enabled by writing a '1' to EVCTRL.INVEIx.

Related Links

33. EVSYS – Event System

43.6.6 Sleep Mode Operation

The generic clock for the DAC is running in idle sleep mode. If the Run In Standby bit in the Control A register (CTRLA.RUNSTDBY) is one, the DAC output buffer will keep its value in standby sleep mode. If CTRLA.RUNSTDBY is zero, the DAC output buffer will be disabled in standby sleep mode.

43.6.7 Synchronization

Due to the asynchronicity between main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read. A register can require:

- Synchronization when written
- Synchronization when read
- Synchronization when written and read
- No synchronization

When executing an operation that requires synchronization, the corresponding status bit in the Synchronization Busy register (SYNCBUSY) will be set immediately, and cleared when synchronization is complete.

Downloaded from Arrow.com.

If an operation that requires synchronization is executed while its busy bit is one, the operation is discarded and an error is generated.

The following bits need synchronization when written:

- Software Reset bit in the Control A register (CTRLA.SWRST)
- Enable bit in the Control A register (CTRLA.ENABLE)
- All bits in the Data register (DATA)
- All bits in the Data Buffer register (DATABUF)

Write-synchronization is denoted by the Write-Synchronized property in the register description.

No bits need synchronization when read.

43.6.8 Additional Features

43.6.8.1 DAC as an Internal Reference

The DAC output can be internally enabled as input to the analog comparator. This is enabled by writing a one to the Internal Output Enable bit in the Control B register (CTRLB.IOEN). It is possible to have the internal and external output enabled simultaneously.

The DAC output can also be enabled as input to the Analog-to-Digital Converter. In this case, the output buffer must be enabled.

43.6.8.2 Data Buffer

The Data Buffer register (DATABUF) and the Data register (DATA) are linked together to form a two-stage FIFO. The DAC uses the Start Conversion event to load data from DATABUF into DATA and start a new conversion. The Start Conversion event is enabled by writing a one to the Start Event Input bit in the Event Control register (EVCTRL.STARTEI). If a Start Conversion event occurs when DATABUF is empty, an Underrun interrupt request is generated if the Underrun interrupt is enabled.

The DAC can generate a Data Buffer Empty event when DATABUF becomes empty and new data can be loaded to the buffer. The Data Buffer Empty event is enabled by writing a one to the Empty Event Output bit in the Event Control register (EVCTRL.EMPTYEO). A Data Buffer Empty interrupt request is generated if the Data Buffer Empty interrupt is enabled.

43.6.8.3 Voltage Pump

When the DAC is used at operating voltages lower than 2.5V, the voltage pump must be enabled. This enabling is done automatically, depending on operating voltage.

The voltage pump can be disabled by writing a one to the Voltage Pump Disable bit in the Control B register (CTRLB.VPD). This can be used to reduce power consumption when the operating voltage is above 2.5V.

The voltage pump uses the asynchronous GCLK_DAC clock, and requires that the clock frequency be at least four times higher than the sampling period.

43.6.8.4 Dithering mode

In dithering mode, DATA is a 14-bit signed value where DATA[13:4] is the 10-bit data converted by DAC and DATA[3:0] the dither bits, used to minimize the quantization error.

The principle is to make 16 sub-conversions of DATA[13:4] value or (DATA[13:4] + 1) value so that by averaging those 2 values, the 14-bit value (DATA[13:0]) conversion is accurate.

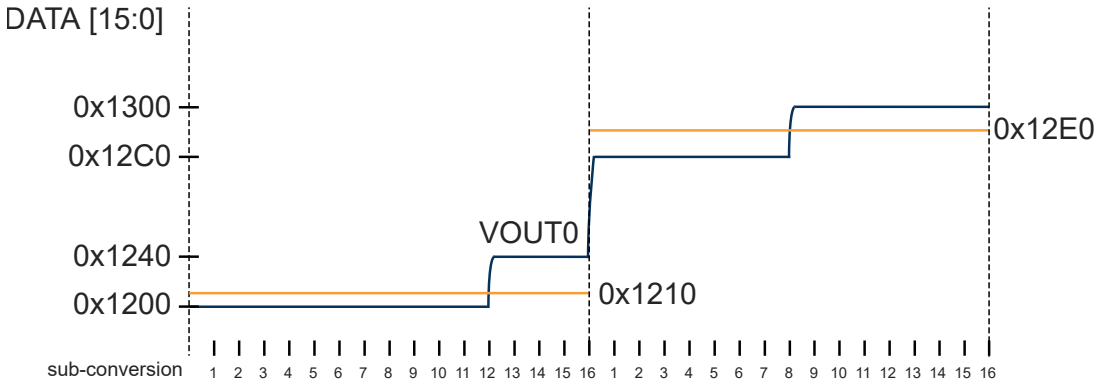
To operate, START event must be configured to generate 16 events for each DATA[15:0] conversion and DATABUF must be loaded every 16 DAC conversions. EMPTY event and DMA request are therefore generated every 16 DATABUF to DATA transfer.

Downloaded from Arrow.com.

Writing a one to the Left Adjust bit in Control B register (CTRLB.LEFTADJ) change the data to DATA[15:6] and the dithering bits to DATA[5:2]. Refer to 43.8.8 DATA description for further details.

Following timing diagram shows examples with DATA[15:0] = 0x1210 then DATA[15:0] = 0x12E0 and CTRLB.LEFTADJ=1.

Figure 43-2. DAC Conversions in Dithering Mode (CTRLB.LEFTADJ=1)



Downloaded from Arrow.com.

43.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0		RUNSTDBY					ENABLE	SWRST
0x01	CTRLB	7:0	REFSEL[1:0]		DITHER		VPD	LEFTADJ	IOEN	EOEN
0x02	EVCTRL	7:0						INVEI	EMPTYEO	STARTEI
0x03	Reserved									
0x04	INTENCLR	7:0							EMPTY	UNDERRUN
0x05	INTENSET	7:0							EMPTY	UNDERRUN
0x06	INTFLAG	7:0							EMPTY	UNDERRUN
0x07	STATUS	7:0								READY
0x08	DATA	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
0x0A ... 0x0B	Reserved									
0x0C	DATABUF	7:0	DATABUF[7:0]							
		15:8	DATABUF[15:8]							
0x0E ... 0x0F	Reserved									
0x10	SYNCBUSY	7:0					DATABUF	DATA	ENABLE	SWRST
		15:8								
		23:16								
		31:24								
0x14 ... 0x17	Reserved									
0x18	DBGCTRL	7:0								DBGRUN

43.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to 43.5.8 Register Access Protection.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to 43.6.7 Synchronization.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

Downloaded from Arrow.com.

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

Downloaded from Arrow.com.

43.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY					ENABLE	SWRST
Access		R/W					R/W	R/W
Reset		0					0	0

Bit 6 – RUNSTDBY Run in Standby
This bit is not synchronized

Value	Description
0	The DAC output buffer is disabled in standby sleep mode.
1	The DAC output buffer can be enabled in standby sleep mode.

Bit 1 – ENABLE Enable DAC Controller
Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the corresponding bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled or being enabled.

Bit 0 – SWRST Software Reset
Writing '0' to this bit has no effect.

Writing '1' to this bit resets all registers in the DAC to their initial state, and the DAC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

Downloaded from Arrow.com.

43.8.2 Control B

Name: CTRLB
Offset: 0x01
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
	REFSEL[1:0]		DITHER		VPD	LEFTADJ	IOEN	EOEN
Access	R/W	R/W	R/W		R/W	R/W	R/W	R/W
Reset	0	0	0		0	0	0	0

Bits 7:6 – REFSEL[1:0] Reference Selection
This bit field selects the Reference Voltage for the DAC.

Important: For the SAM L10/L11 devices, the 1.1V voltage reference typical value must be selected in case of an internal voltage reference (INTREF). Refer to the *Supply Controller VREF* register.

Value	Name	Description
0x0	INTREF	Internal voltage reference
0x1	VDDANA	Analog voltage supply
0x2	VREFA	External reference
0x3		Reserved

Bit 5 – DITHER Dithering Mode
This bit controls dithering operation according to 43.6.8.4 Dithering mode.

Value	Description
0	Dithering mode is disabled.
1	Dithering mode is enabled.

Bit 3 – VPD Voltage Pump Disabled
This bit controls the behavior of the voltage pump.

Value	Description
0	Voltage pump is turned on/off automatically
1	Voltage pump is disabled.

Bit 2 – LEFTADJ Left-Adjusted Data
This bit controls how the 10-bit conversion data is adjusted in the Data and Data Buffer registers.

Value	Description
0	DATA and DATABUF registers are right-adjusted.
1	DATA and DATABUF registers are left-adjusted.

Bit 1 – IOEN Internal Output Enable

Downloaded from Arrow.com.

Value	Description
0	Internal DAC output not enabled.
1	Internal DAC output enabled to be used by the AC or ADC.

Bit 0 – EOEN External Output Enable

Value	Description
0	The DAC output is turned off.
1	The high-drive output buffer drives the DAC output to the V_{OUT} pin.

Downloaded from Arrow.com.

43.8.3 Event Control

Name: EVCTRL
Offset: 0x02
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
						INVEI	EMPTYEO	STARTEI
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – INVEI Enable Inversion Data Buffer Empty Event Output
This bit defines the edge detection of the input event for STARTEI.

Value	Description
0	Rising edge.
1	Falling edge.

Bit 1 – EMPTYEO Data Buffer Empty Event Output
This bit indicates whether or not the Data Buffer Empty event is enabled and will be generated when the Data Buffer register is empty.

Value	Description
0	Data Buffer Empty event is disabled and will not be generated.
1	Data Buffer Empty event is enabled and will be generated.

Bit 0 – STARTEI Start Conversion Event Input
This bit indicates whether or not the Start Conversion event is enabled and data are loaded from the Data Buffer register to the Data register upon event reception.

Value	Description
0	A new conversion will not be triggered on any incoming event.
1	A new conversion will be triggered on any incoming event.

Downloaded from Arrow.com.

43.8.4 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Bit	7	6	5	4	3	2	1	0
							EMPTY	UNDERRUN
Access							R/W	R/W
Reset							0	0

Bit 1 – EMPTY Data Buffer Empty Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Data Buffer Empty Interrupt Enable bit, which disables the Data Buffer Empty interrupt.

Value	Description
0	The Data Buffer Empty interrupt is disabled.
1	The Data Buffer Empty interrupt is enabled.

Bit 0 – UNDERRUN Underrun Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Data Buffer Underrun Interrupt Enable bit, which disables the Data Buffer Underrun interrupt.

Value	Description
0	The Data Buffer Underrun interrupt is disabled.
1	The Data Buffer Underrun interrupt is enabled.

Downloaded from Arrow.com.

43.8.5 Interrupt Enable Set

Name: INTENSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Bit	7	6	5	4	3	2	1	0
							EMPTY	UNDERRUN
Access							R/W	R/W
Reset							0	0

Bit 1 – EMPTY Data Buffer Empty Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Data Buffer Empty Interrupt Enable bit, which enables the Data Buffer Empty interrupt.

Value	Description
0	The Data Buffer Empty interrupt is disabled.
1	The Data Buffer Empty interrupt is enabled.

Bit 0 – UNDERRUN Underrun Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Data Buffer Underrun Interrupt Enable bit, which enables the Data Buffer Underrun interrupt.

Value	Description
0	The Data Buffer Underrun interrupt is disabled.
1	The Data Buffer Underrun interrupt is enabled.

Downloaded from Arrow.com.

43.8.6 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x06
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
							EMPTY	UNDERRUN
Access							R/W	R/W
Reset							0	0

Bit 1 – EMPTY Data Buffer Empty
This flag is cleared by writing a '1' to it or by writing new data to DATABUF.

This flag is set when data is transferred from DATABUF to DATA, and the DAC is ready to receive new data in DATABUF, and will generate an interrupt request if INTENCLR/SET.EMPTY is one.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Data Buffer Empty interrupt flag.

Bit 0 – UNDERRUN Underrun
This flag is cleared by writing a '1' to it.

This flag is set when a start conversion event occurs when DATABUF is empty, and will generate an interrupt request if INTENCLR/SET.UNDERRUN is one.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Underrun interrupt flag.

Downloaded from Arrow.com.

43.8.7 Status

Name: STATUS
Offset: 0x07
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
								READY
Access								R
Reset								0

Bit 0 – READY DAC Ready

Value	Description
0	DAC is not ready for conversion.
1	Startup time has elapsed, DAC is ready for conversion.

Downloaded from Arrow.com.

43.8.8 Data DAC

Name: DATA
Offset: 0x08
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – DATA[15:0] Data value to be converted
DATA register contains the 10-bit value that is converted to a voltage by the DAC. The adjustment of these 10 bits within the 16-bit register is controlled by CTRLB.LEFTADJ.

Four additional bits are also used for the dithering feature according to 43.6.8.4 Dithering mode.

Table 43-1. Valid Data Bits

CTRLB.DITHER	CTRLB.LEFTADJ	DATA	Description
0	0	DATA[9:0]	Right adjusted, 10-bits
0	1	DATA[15:6]	Left adjusted, 10-bits
1	0	DATA[13:4], DATA[3:0]	Right adjusted, 14-bits
1	1	DATA[15:6], DATA[5:2]	Left adjusted, 14-bits

Downloaded from Arrow.com.

43.8.9 Data Buffer

Name: DATABUF
Offset: 0x0C
Reset: 0x0000
Property: Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	DATABUF[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATABUF[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – DATABUF[15:0] Data Buffer
DATABUF contains the value to be transferred into DATA register.

Downloaded from Arrow.com.

43.8.10 Synchronization Busy

Name: SYNCBUSY
Offset: 0x10
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					DATABUF	DATA	ENABLE	SWRST
Access					R	R	R	R
Reset					0	0	0	0

Bit 3 – DATABUF Data Buffer DAC0
This bit is set when DATABUF register is written.

This bit is cleared when DATABUF synchronization is completed.

Value	Description
0	No ongoing synchronized access.
1	Synchronized access is ongoing.

Bit 2 – DATA Data
This bit is set when DATA register is written.

This bit is cleared when DATA synchronization is completed.

Value	Description
0	No ongoing synchronized access.
1	Synchronized access is ongoing.

Bit 1 – ENABLE DAC Enable Status
This bit is set when CTRLA.ENABLE bit is written.

This bit is cleared when CTRLA.ENABLE synchronization is completed.

Downloaded from Arrow.com.

Value	Description
0	No ongoing synchronization.
1	Synchronization is ongoing.

Bit 0 – SWRST Software Reset
This bit is set when CTRLA.SWRST bit is written.

This bit is cleared when CTRLA.SWRST synchronization is completed.

Value	Description
0	No ongoing synchronization.
1	Synchronization is ongoing.

Downloaded from Arrow.com.

43.8.11 Debug Control

Name: DBGCTRL
Offset: 0x18
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								
Reset								0

Bit 0 – DBGRUN Debug Run
This bit is not reset by a software reset.

This bits controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The DAC is halted when the CPU is halted by an external debugger. Any ongoing conversion will complete.
1	The DAC continues normal operation when the CPU is halted by an external debugger.

Downloaded from Arrow.com.

44. OPAMP – Operational Amplifier Controller

44.1 Overview

The Operational Amplifier (OPAMP) Controller configures and controls three low power, general purpose operational amplifiers offering a high degree of flexibility and rail-to-rail inputs.

Most common inverting or non-inverting programmable gain and hysteresis configurations can be selected by software - no external components are required for these configurations.

The OPAMPs can be cascaded for both standalone mode and built-in configurations.

Each OPAMP can be used as a standalone amplifier. External pins are available for filter configurations or other applications. A reference can be generated from the DAC to be used as selectable reference for inverting PGA (programmable gain amplifier) or instrumentation amplifier. Each OPAMP can be used as buffer or PGA for the ADC or an AC. The OPAMP offset voltage can be compensated when it is used in combination with the ADC.

Four modes are available to select the trade-off between speed and power consumption to best fit the application requirements and optimize the power consumption.

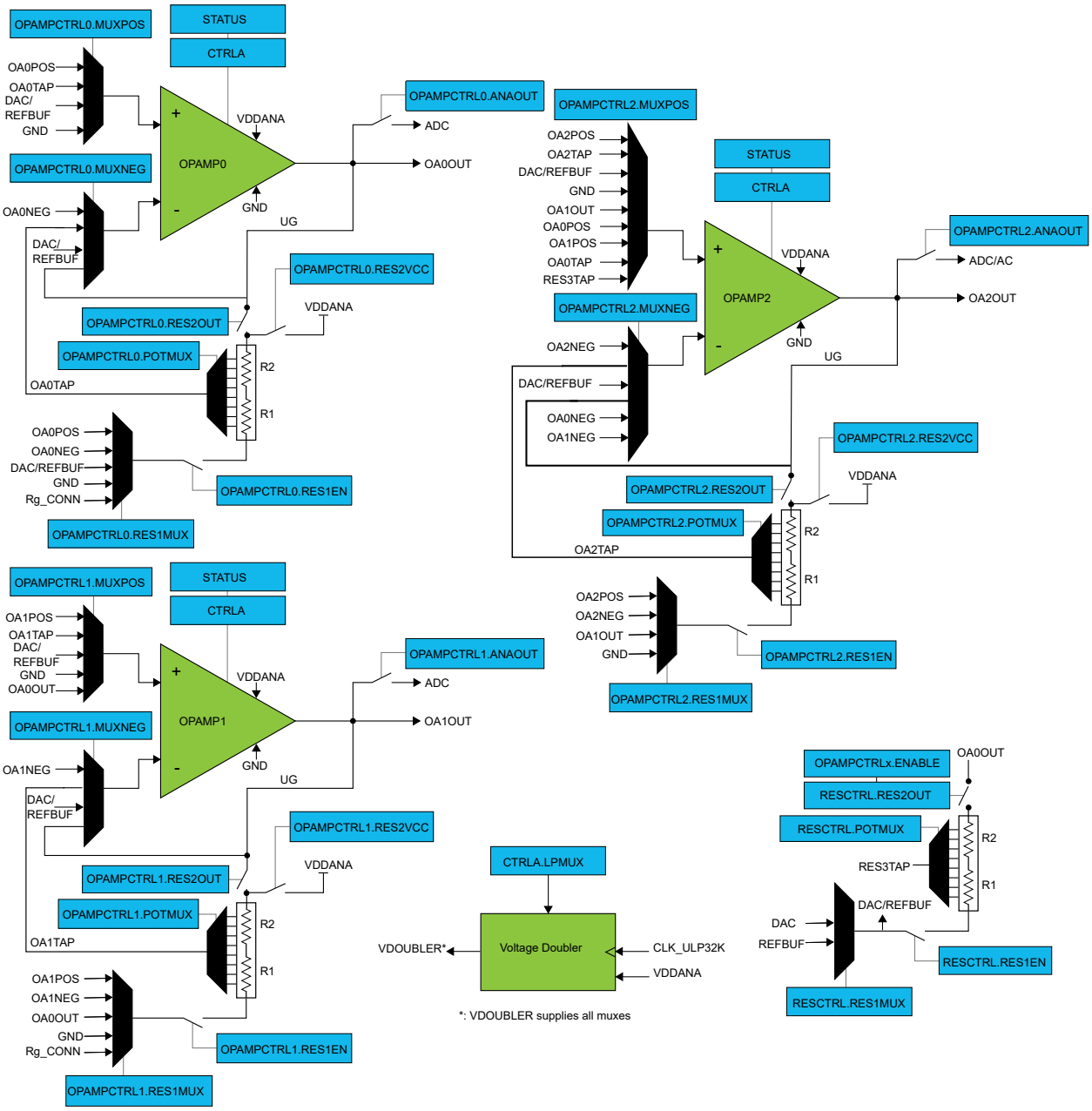
44.2 Features

- Three individually configurable low power OPAMPs
- Rail-to-rail inputs
- Configurable resistor ladders for internal feedback
- Selectable configurations
  - Standalone OPAMP with flexible inputs
  - Unity gain amplifier
  - Non-inverting / inverting Programmable Gain Amplifier (PGA)
  - Cascaded PGAs
  - Instrumentation amplifier
  - Comparator with programmable hysteresis
- OPAMP output:
  - On I/O pins
  - As input for AC or ADC
- Flexible input selection:
  - I/O pins
  - DAC
  - Ground
- Low power options:
  - Selectable voltage doubler and propagation delay versus current consumption
  - On demand start-up for ADC and AC operations
- Offset/Gain measurement for calibration when used with the ADC

Downloaded from Arrow.com.

44.3 Block Diagram

Figure 44-1. OPAMP Block Diagram



44.4 Signal Description

Signal	Description	Type
OA0POS	OPAMP0 positive input	Analog input
OA0NEG	OPAMP0 negative input	Analog input
OA1POS	OPAMP1 positive input	Analog input

Downloaded from Arrow.com.

Signal	Description	Type
OA1NEG	OPAMP1 negative input	Analog input
OA2POS	OPAMP2 positive input	Analog input
OA2NEG	OPAMP2 negative input	Analog input
OA0OUT	OPAMP0 output	Analog output
OA1OUT	OPAMP1 output	Analog output
OA2OUT	OPAMP2 output	Analog output

One signal can be mapped on several pins.

Important:
When an analog peripheral is enabled, the analog output of the peripheral will interfere with the alternative functions of the output pads. This is also true even when the peripheral is used for internal purposes.

Analog inputs do not interfere with alternative pad functions.

44.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

44.5.1 I/O Lines

Using the OPAMP I/O lines requires the I/O pins to be configured. Refer to the *PORT - I/O Pin Controller* chapter for details.

44.5.2 Power Management

The OPAMP can operate in idle and standby sleep mode, according to the settings of the Run in Standby and On Demand bits in the OPAMP Control x registers (OPAMPCTRLx.RUNSTDBY and OPAMPCTRLx.ONDEMAND), as well as the Enable bit in the Control A register (CTRLA.ENABLE). Refer to *PM – Power Manager* for details on the different sleep modes.

Related Links

22. PM – Power Manager

44.5.3 Clocks

The OPAMP bus clock (CLK_OPAMP_APB) can be enabled and disabled in the Power Manager, and the default state of CLK_OPAMP_APB can be found in the *Peripheral Clock Masking*.

A clock (CLK_ULP32K) is required by the voltage doubler for low voltage operation (VCC < 2.5V). The CLK_ULP32K is a 32KHz clock which is provided by the OSCULP32K oscillator in the OSC32KCTRL module.

Related Links

19.6.2.6 Peripheral Clock Masking

44.5.4 DMA

Not applicable.

Downloaded from Arrow.com.

44.5.5 Interrupts

Not applicable.

44.5.6 Events

Not applicable.

44.5.7 Debug Operation

When the CPU is halted in debug mode the OPAMP continues normal operation. If the OPAMP is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

44.5.8 Register Access Protection

All registers with write-access can be write-protected optionally by the peripheral access controller (PAC). Refer to *PAC - Peripheral Access Controller* for details.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

Related Links

15. PAC - Peripheral Access Controller

44.5.9 SAM L11 TrustZone Specific Register Access Protection

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

44.5.10 Analog Connections

Each OPAMP has two I/O pins that can be used as analog inputs. These pins must be configured for analog operation before using them as OPAMP inputs.

If the DAC is to be used as OPAMP input, the DAC must be configured and enabled first.

Each OPAMP has one I/O pin that can be used as analog output. This pin must be configured for analog operation before using it as OPAMP output.

The analog signals of AC, ADC, DAC and OPAMP can be interconnected. The AC and ADC peripheral can request the OPAMP using an analog ONDEMAND functionality.

See *Analog Connections of Peripherals* for details.

44.5.11 Other dependencies

Not applicable.

Downloaded from Arrow.com.

44.6 Functional Description

44.6.1 Principle of Operation

Each OPAMP has one positive and one negative input. Each input may be chosen from either a selection of analog input pins, or internal inputs such as the DAC, the resistor ladder, and the ground and output of another OPAMP.

Each OPAMP can be configured with built-in feedback to support various functions with programmable or unity gain.

I/O pins are externally accessible so that the operational amplifier can be configured with external feedback.

All OPAMPs can be cascaded to support circuits such as differential amplifiers.

44.6.2 Basic Operation

Each operational amplifier can be configured in different modes, selected by the OPAMP Control x register (OPAMPCTRLx):

- Standalone operational amplifier
- Operational amplifier with built-in feedback

After being enabled, a start-up delay is added before the output of the operational amplifier is available. This start-up time is measured internally to account for environmental changes such as temperature or voltage supply level.

When the OPAMP is ready, the respective Ready x bit in the Status register is set (STATUS.READYx=1).

If the supply voltage is below 2.5V, the start-up time is also dependent on the voltage doubler. If the supply voltage is always above 2.5V, the voltage doubler can be disabled by setting the Low-Power Mux bit in the Control A Register (CTRLA.LPMUX).

44.6.2.1 Initialization

The OPAMP must be configured with the desired properties and inputs before it is enabled.

The asynchronous clocks CLK_ULP32K must be configured in the OSC32KCTRL module before enabling individual OPAMPs. See *OSC32KCTRL – 32KHz Oscillators Controller* for further details.

Related Links

24. OSC32KCTRL – 32KHz Oscillators Controller

44.6.2.2 Enabling, Disabling, and Resetting

The OPAMP is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The OPAMP is disabled by writing a '0' to CTRLA.ENABLE.

Each OPAMP sub-module is enabled by writing a '1' to the Enable bit in the OPAMP Control x register (OPAMPCTRLx.ENABLE). Each OPAMP sub-module is disabled by writing a '0' to OPAMPCTRLx.ENABLE.

The OPAMP module is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the OPAMP will be reset to their initial state, and the OPAMP will be disabled. Refer to 44.8.1 CTRLA for details.

44.6.3 DMA Operation

Not applicable.

Downloaded from Arrow.com.

44.6.4 Interrupts

Not applicable.

44.6.5 Events

Not applicable.

44.6.6 Sleep Mode Operation

The OPAMPs can also be used during sleep modes. The 32KHz clock source used by the voltage doubler must remain active. See Voltage Doubler for more details.

Each OPAMP x can be configured to behave differently in different sleep modes. The behavior is determined by the individual Run in Standby and On Demand bits in the OPAMP Control x registers (OPAMPCTRLx.RUNSTDBY, and OPAMPCTRLx.ONDEMAND), as well as the common Enable bit in the Control A register (CTRLA.ENABLE).

Table 44-1. Individual OPAMP Sleep Mode Operation

OPAMPCTRLx.RUNSTDBY	OPAMPCTRLx.ONDEMAND	CTRLA.ENABLE	Sleep Behavior
-	-	0	Disabled
0	0	1	Always run in all sleep modes except STANDBY sleep mode
0	1	1	Only run in all sleep modes except STANDBY sleep mode if requested by a peripheral.
1	0	1	Always run in all sleep mode
1	1	1	Only run in all sleep modes if requested by a peripheral.

Note:
When OPAMPCTRLx.ONDEMAND=1, the analog block is powered off for the lowest power consumption if it is not requested.

When requested, a start-up time delay is necessary when the system returns from sleep. The start-up time is depending on the Bias Selection bits in the OPAMP Control x register (OPAMPCTRLx.BIAS) and the corresponding speed/current consumption requirements.

44.6.7 Synchronization

Not applicable.

44.6.8 Configuring the Operational Amplifiers

Each individual operational amplifier is configured by its respective Operational Amplifier Control x register (OPAMPCTRLx). These settings must be configured before the amplifier is started.

- Select the positive input in OPAMPCTRLx.MUXPOS
- Select the negative input in OPAMPCTRLx.MUXNEG
- Select RES1EN if resistor ladder is used
- Select the input for the resistor ladder in OPAMPCTRLx.RES1MUX
- Select the potentiometer selection of the resistor ladder in OPAMPCTRLx.POTMUX
- Select the VCC input for the resistor ladder in OPAMPCTRLx.RES2VCC
- Connect the operational amplifier output to the resistor ladder using OPAMPCTRLx.RES2OUT
- Select the trade-off between speed and energy consumption in OPAMPCTRLx.BIAS

Downloaded from Arrow.com.

- Select RES3TAP as positive input for the OPAMP2 and connect the resistor ladder to OA0OUT by setting the RESCTRL.RES2OUT bit only if OPAMPs are configured as a High Gain Instrumentation Amplifier

44.6.9 Standalone Mode

Each operational amplifier can be used as standalone amplifier. In this mode, positive input, negative input and the output are routed from/to external I/Os, requiring external feedback. OPAMPs can also be cascaded to support multiple OPAMP configurations. Refer to Operational Amplifier Control x register (OPAMPCTRLx) for further details on how to configure OPAMP I/Os.

44.6.10 Built-in Modes

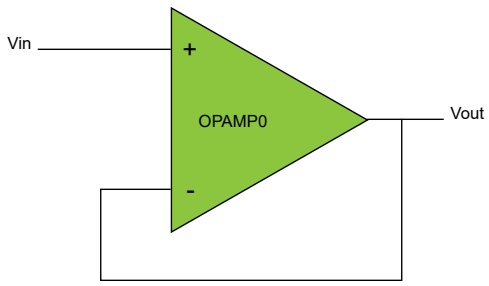
44.6.10.1 Voltage Follower

In this mode the unity gain path is selected for the negative input. The OPAMPCTRLx register can be configured as follows:

Table 44-2. Configuration - Three Independent Unitary Gain Followers

	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	0000	011	011	000	0	0	0	0
OPAMP1	0000	011	011	000	0	0	0	0
OPAMP2	0000	011	011	000	0	0	0	0

Figure 44-2. Voltage follower



44.6.10.2 Inverting PGA

For inverting programmable gain amplifier operation, the OPAMPCTRLx registers can be configured as follows:

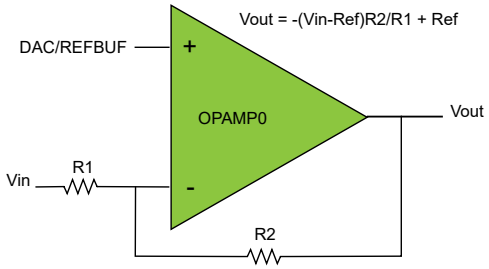
Table 44-3. Configuration - Three Independent Inverting PGAs

	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	0010	001	001	100	0	1	1	0
OPAMP1	0010	001	001	100	0	1	1	0
OPAMP2	0010	001	001	100	0	1	1	0

Inverting PGA (Example: Vout=-3.Vin, R1=4R, R2=12R)

Downloaded from Arrow.com.

Figure 44-3. Inverting PGA



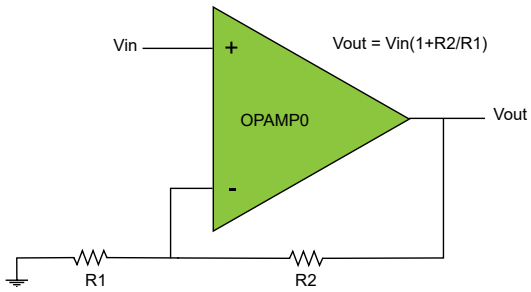
44.6.10.3 Non-Inverting PGA

For non-inverting programmable gain amplifier operation, the OPAMPCTRLx registers can be configured as follows:

Table 44-4. Configuration - Three Independent Non-Inverting PGAs (Example: Vout=4.Vin, R1=4R, R2=12R)

	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	000	001	11	100	0	1	1	0
OPAMP1	000	001	11	100	0	1	1	0
OPAMP2	000	001	11	100	0	1	1	0

Figure 44-4. Non-Inverting PGA



44.6.10.4 Cascaded Inverting PGA

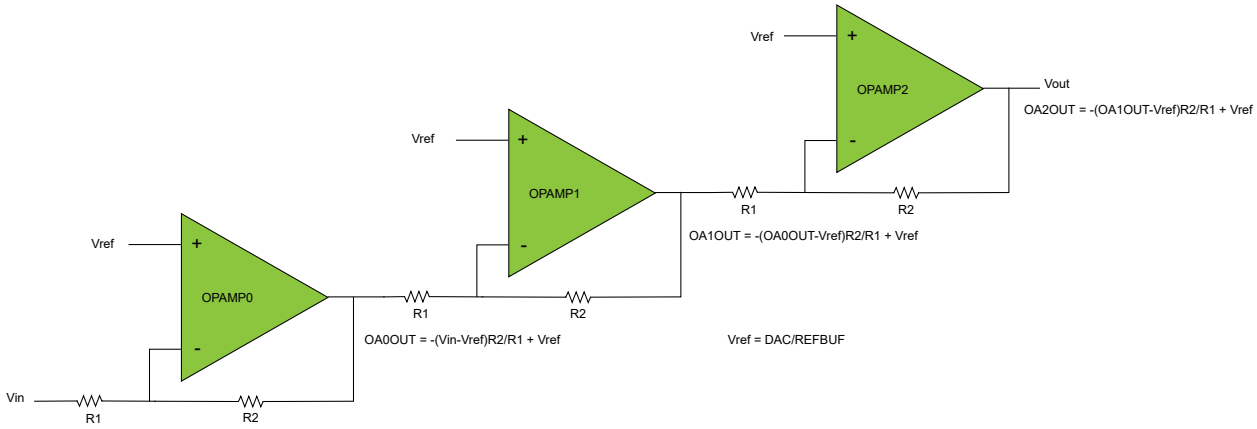
The OPAMPs can be configured as three cascaded, inverting PGAs using these settings in OPAMPCTRLx:

Table 44-5. Cascade of three inverting PGAs (Example: R1=4R, R2=12R)

	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	0010	001	001	100	0	1	1	0
OPAMP1	0010	001	010	100	0	1	1	0
OPAMP2	0010	001	010	100	0	1	1	0

Downloaded from Arrow.com.

Figure 44-5. Cascaded Inverting PGA



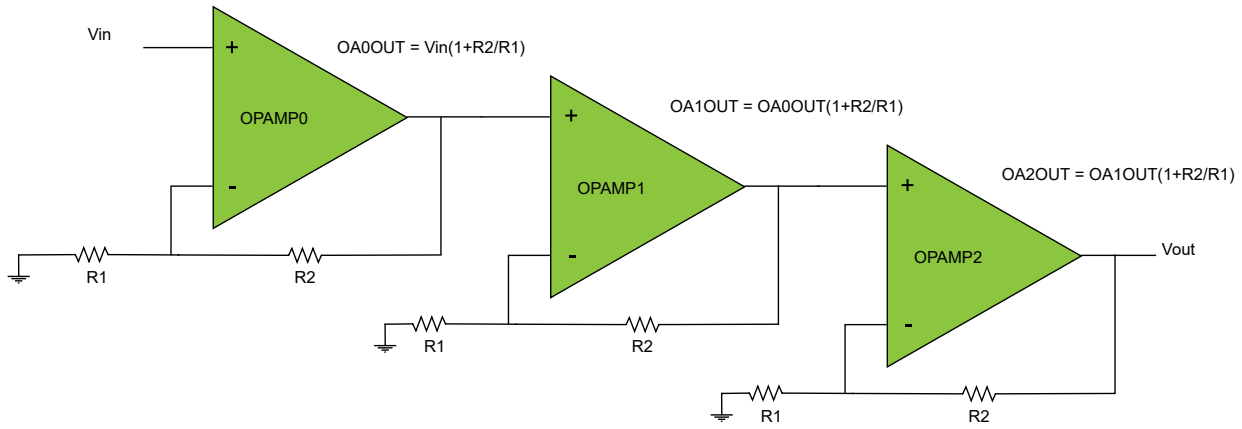
44.6.10.5 Cascaded Non-Inverting PGA

The OPAMPs can be configured as three cascaded, non-inverting PGAs using these settings in OPAMPCTRLx:

Table 44-6. Cascaded Non-Inverting PGA (Exemple: R1=4R, R2=12R)

	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	0000	001	011	100	0	1	1	0
OPAMP1	0100	001	011	100	0	1	1	0
OPAMP2	0100	001	011	100	0	1	1	0

Figure 44-6. Cascaded Non-Inverting PGA



44.6.10.6 Two OPAMPs Differential Amplifier

In this mode, OPAMP0 can be coupled with OPAMP1 or OPAMP1 with OPAMP2 in order to amplify a differential signal.

To configure OPAMP0 and OPAMP1 as differential amplifier, the OPAMPCTRLx register can be configured as follows:

Downloaded from Arrow.com.

Table 44-7. OPAMP0 OPAMP1 Differential Amplifier (Example: R1=4R, R2=12R)

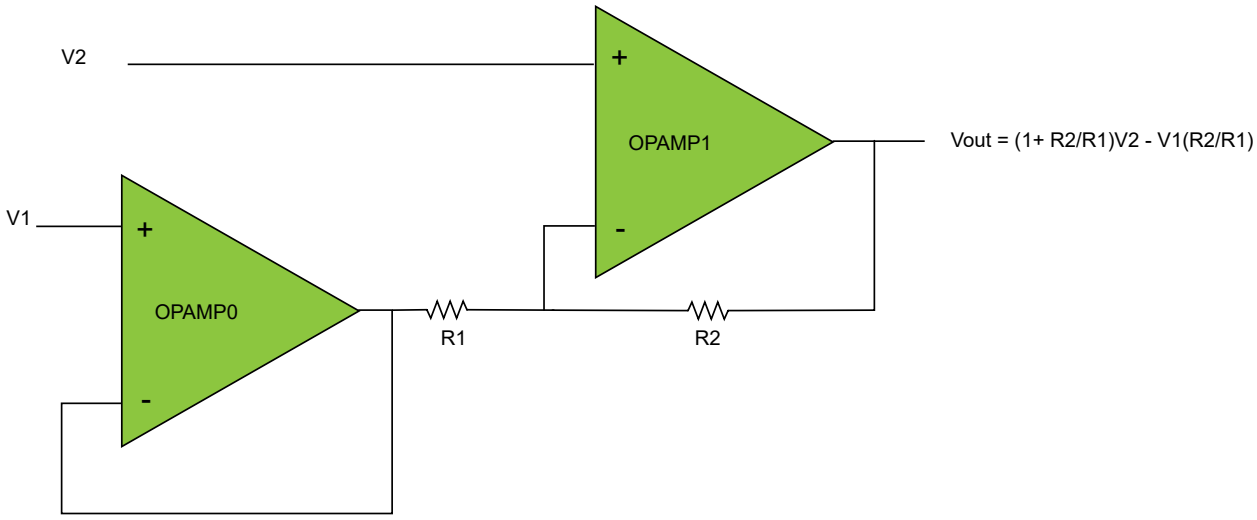
	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	0000	011	011	000	0	0	0	0
OPAMP1	0000	001	010	100	0	1	1	0
OPAMP2	0000	000	000	000	0	0	0	0

To configure OPAMP1 and OPAMP2 as differential amplifier, the OPAMPCTRLx register can be configured as follows:

Table 44-8. OPAMP1 OPAMP2 Differential Amplifier (Example: R1=4R, R2=12R)

	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	0000	000	000	000	0	0	0	0
OPAMP1	0000	011	011	000	0	0	0	0
OPAMP2	0000	001	010	100	0	1	1	0

Figure 44-7. OPAMP0 OPAMP1 Differential Amplifier



44.6.10.7 Instrumentation Amplifier

In this mode, OPAMP0 and OPAMP1 are configured as voltage followers. The OPAMPCTRLx register can be configured as follows:

Table 44-9. Instrumentation Amplifier Configuration

	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	0000	011	010	010	0	1	1	0
OPAMP1	0000	011	011	000	0	0	0	0
OPAMP2	0111	001	010	010	0	1	1	0

The resistor ladders associated with OPAMP0 and OPAMP2 must be configured as follows in order to select the appropriate gain:

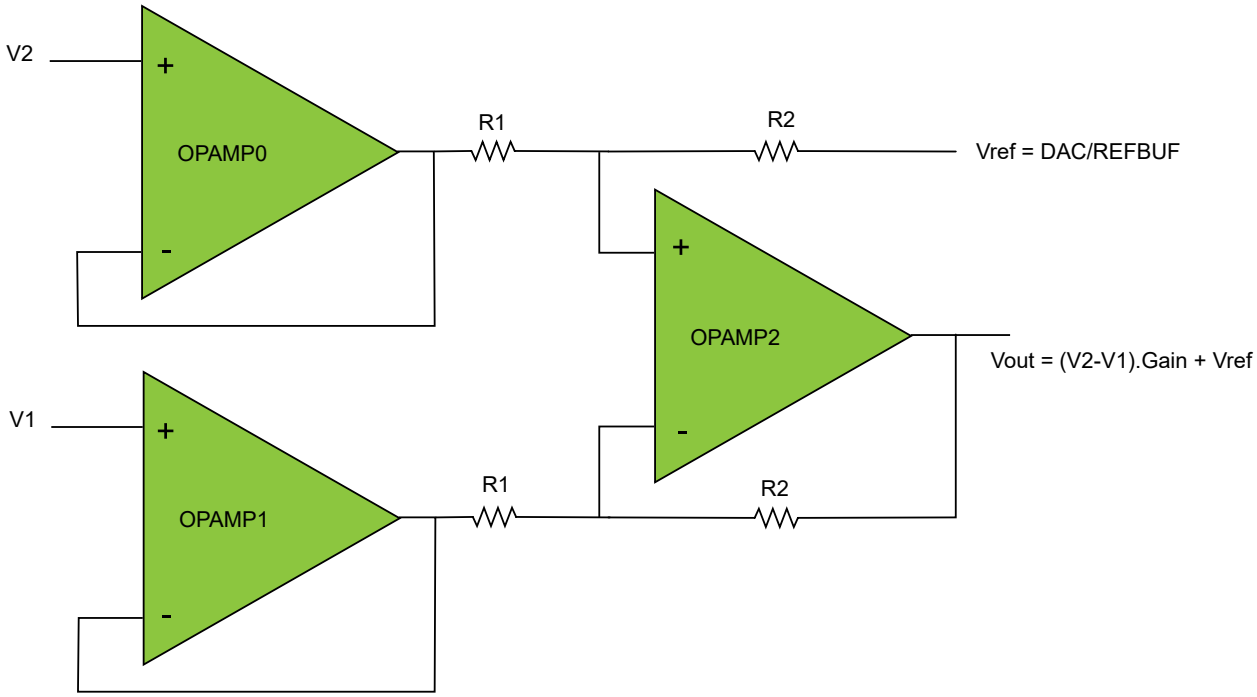
Downloaded from Arrow.com.

Table 44-10. Instrumentation Amplifier Gain Selection

OPAMPCTRL0.POTMUX	OPAMPCTRL2.POTMUX	GAIN
0x7	Reserved	Reserved
0x6	0x0	1/7
0x5	Reserved	Reserved
0x4	0x1	1/3
0x3	Reserved	Reserved
0x2	0x2	1
0x1	0x4	3
0x0	0x6	7

Note: Either the DAC or GND must be the reference, selected by the OPAMPCTRL0.RES1MUX bits. Refer to the OPAMP Control 'x' register (44.8.3 OPAMPCTRL) for details.

Figure 44-8. Instrumentation amplifier



44.6.10.8 High Gain Instrumentation Amplifier

In this mode, OPAMP0 and OPAMP1 are configured as non inverting amplifiers [Stage 1]. OPAMP2 is configured as difference amplifier[Stage 2]. Total gain of the instrumentation amplifier is product of gains of stage 1 and stage 2. The OPAMPCTRLx and RESCTRL registers can be configured as follows:

Downloaded from Arrow.com.

Table 44-11. Instrumentation Amplifier Configuration

	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	0000	001	100	111	0	1	1	0
OPAMP1	0000	001	100	111	0	1	1	0
OPAMP2	1000	001	010	111	0	1	1	0

Table 44-12. Resister Control Configuration

RES1MUX	POTMUX	RES2OUT	RES1EN
0 (DAC)	111	1	1
1 (REFBUF)	111	1	1

The resistor ladders associated with OPAMP0, OPAMP1, OPAMP2 and RESCTRL must be configured as follows in order to select the appropriate gain:

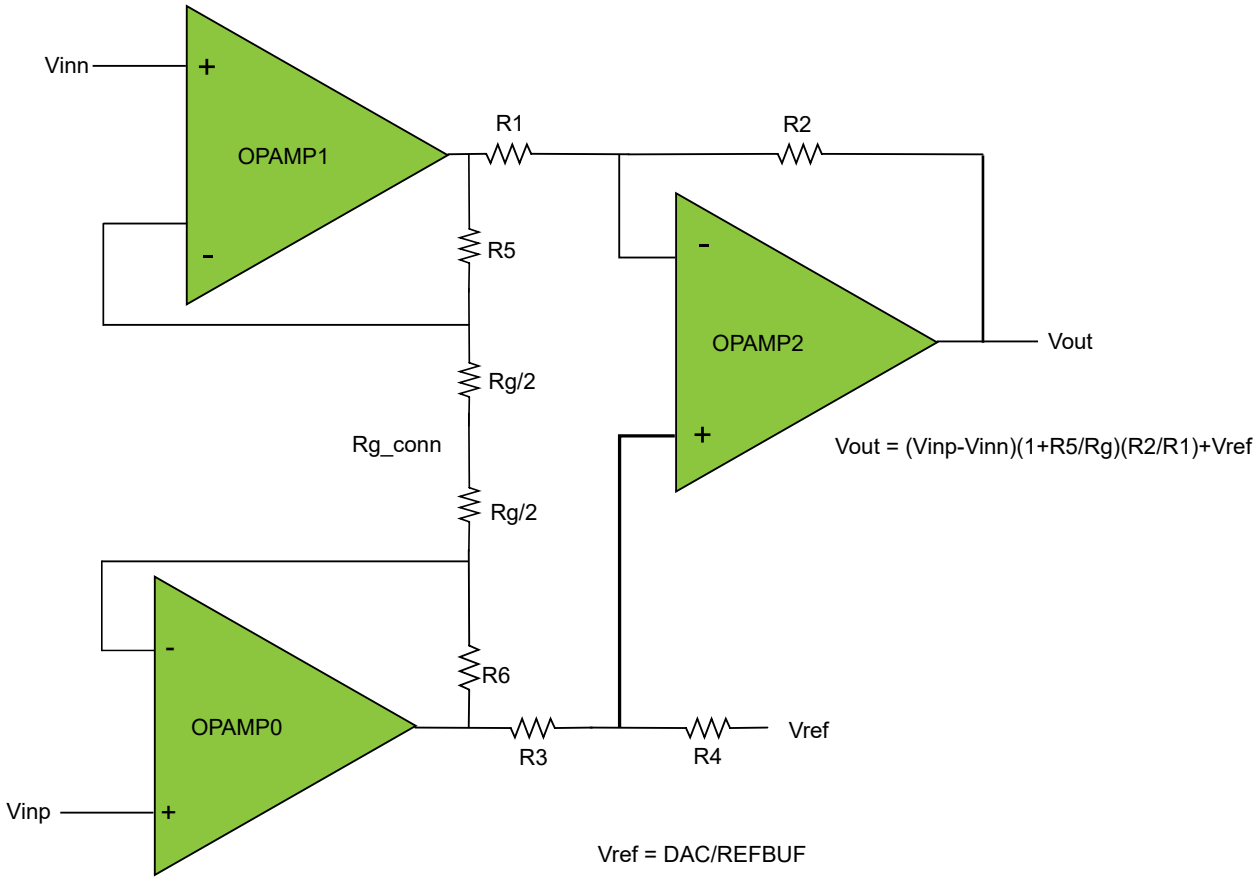
Table 44-13. Instrumentation Amplifier Gain Selection

OPAMPCTRL0.POTMUX	OPAMPCTRL1.POTMUX	STAGE 1 GAIN	OPAMPCTRL2.POTMUX	RESCTRL1.POTMUX	STAGE 2 GAIN
0x7	0x7	16	0x7	0x7	15
0x6	0x6	8	0x6	0x6	7
0x5	0x5	5 + 1/3	0x5	0x5	4 + 1/3
0x4	0x4	4	0x4	0x4	3
0x3	0x3	2 + 2/3	0x3	0x3	1 + 2/3
0x2	0x2	2	0x2	0x2	1
0x1	0x1	1 + 1/3	0x1	0x1	1/3
0x0	0x0	1 + 1/7	0x0	0x0	1/7

The RES3TAP potentiometer can be selected as a positive input for the OPAMP2 and the resistor ladder can be connected to OA0OUT by setting the RESCTRL.RES2OUT bit in order to configure OPAMPs as a High Gain Instrumentation Amplifier.

Downloaded from Arrow.com.

Figure 44-9. High Gain Instrumentation amplifier



44.6.10.9 Transimpedance amplifier

Each OPAMP can be configured as a transimpedance amplifier (current to voltage converter). In this mode the positive input is connected to ground. The negative input is connected to the output through the resistor ladder. The OPAMPCTRLx register can be configured as follows:

Table 44-14. Transimpedance Amplifier

	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	0010	000	001	000	0	1	1	0
OPAMP1	0010	000	001	000	0	1	1	0
OPAMP2	0010	000	001	000	0	1	1	0

Downloaded from Arrow.com.

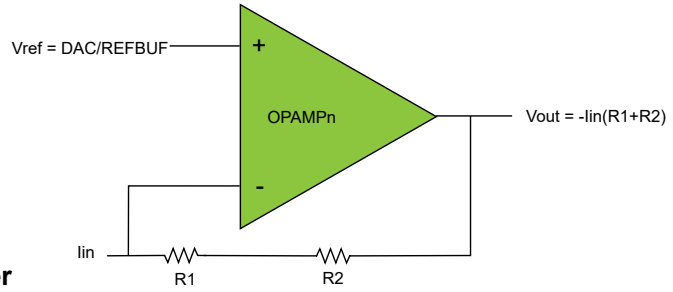


Figure 44-10. Transimpedance Amplifier

44.6.10.10 Programmable Hysteresis

OPAMP can be configured as an inverting or non-inverting comparator with programmable hysteresis. Applying hysteresis will prevent constant toggling of the output, caused by noise when the input signals are close to each other.

In both inverting and non-inverting comparator configurations the positive input is connected to the resistor ladder. When OPAMP is configured as an inverting comparator with programmable hysteresis, the input voltage must be applied to the negative input and RES1MUX must be connected to the ground. When an OPAMP is configured as a non-inverting comparator with programmable hysteresis, the input voltage must be applied to RES1MUX and the negative input must be connected to the ground.

To configure an OPAMP as an inverting comparator with programmable hysteresis, the OPAMPCTRLx register can be configured as follows:

Table 44-15. Configuration of Input Multiplexes for OPAMP0 (Example: Vth = 3/4*Vcc, Ref = GND)

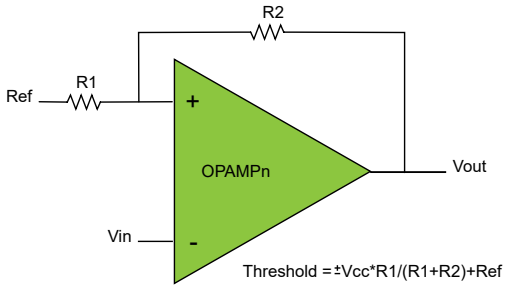
	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	0001	000	010	001	0	1	1	0

Table 44-16. POTMUX [2:0]: Potentiometer Selection

Value	R1	R2	Threshold = Vcc * R1 / (R1 + R2)
0x0	14R	2R	7/8 * Vcc
0x1	12R	4R	3/4 * Vcc
0x2	8R	8R	1/2 * Vcc
0x3	6R	10R	3/8 * Vcc
0x4	4R	12R	1/4 * Vcc
0x5	3R	13R	3/16 * Vcc
0x6	2R	14R	1/8 * Vcc
0x7	R	15R	1/16 * Vcc

Downloaded from Arrow.com.

Figure 44-11. Inverting comparator with programmable hysteresis



To configure an OPAMP as a non-inverting comparator with programmable hysteresis, the OPAMPCTRLx register can be configured as follows:

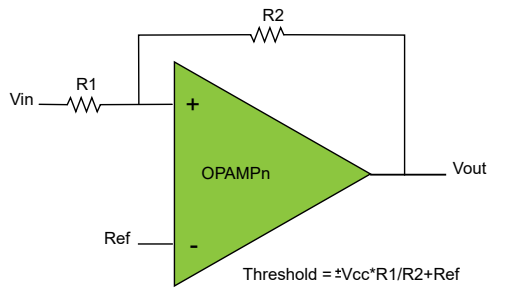
Table 44-17. Configuration of Input Multiplexes for OPAMP0 and OPAMP1 (Example: Vth = 1/3*Vcc, Ref = Gnd)

	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	0001	010	000	100	0	1	1	0
OPAMP1	0001	010	000	100	0	1	1	0
OPAMP2	0001	010	000	100	0	1	1	0

Table 44-18. POTMUX [2:0]: Potentiometer Selection

Value	R1	R2	Threshold = Vcc * R1 / R2
0x0	14R	2R	Vcc * 7 (unused)
0x1	12R	4R	Vcc * 3 (unused)
0x2	8R	8R	Vcc (unused)
0x3	6R	10R	0.6* Vcc
0x4	4R	12R	1/3 * Vcc
0x5	3R	13R	3/13 *Vcc
0x6	2R	14R	1/7 * Vcc
0x7	R	15R	1/15 * Vcc

Figure 44-12. Non-Inverting comparator with programmable hysteresis



Downloaded from Arrow.com.

44.6.11 ADC Driver

44.6.11.1 Buffer/PGA for ADC

Each OPAMP can be configured as a buffer or a PGA for the other modules (such as ADC or AC). OPAMPs can also be cascaded to increase the programmable gain.

The output to the OPAMP must be enabled by writing a '1' to the Analog Output bit in the Operational Amplifier x Control register (OPAMPCTRLx.ANAOUT). The ADC input mux must be configured to select OPAMP as input. Refer to *ADC – Analog-to-Digital Converter* for details on configuring the ADC.

Related Links

41. ADC – Analog-to-Digital Converter

44.6.11.2 Offset and Gain Compensation

When the OPAMP is used in combination with the ADC, the OPAMP offset and gain errors can be compensated. To calculate offset and gain error compensation values

1. Configure OPAMP as Voltage Follower
2. Route the OPAMP output to the ADC:
   - Write a '1' to the Analog Output bit in the Operational Amplifier x Control register (OPAMPCTRLx.ANAOUT)
   - Select the OPAMP as input for the ADC, see *ADC – Analog-to-Digital Converter*.
3. Measure and set the Offset Correction value for the ADC OFFSETCORR register as in 44.6.11.3 Offset Compensation.
4. Measure and set the Gain Correction value for the ADC GAINCORR register as in 44.6.11.4 Gain Compensation.

The offset error compensation must be determined before gain error compensation.

The relation for offset and gain error compensation is shown in this equation:

Result = (converted value + OFFSETCORR)*GAINCORR

Related Links

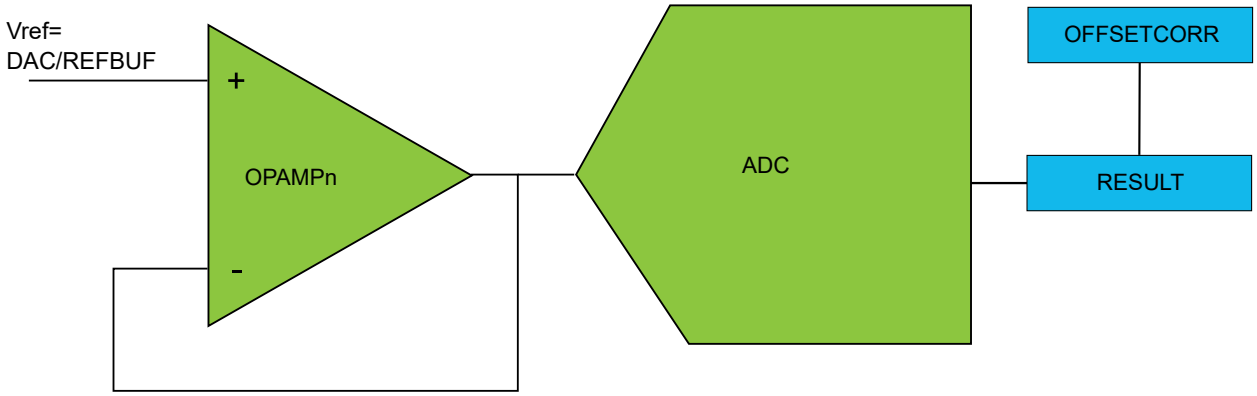
41. ADC – Analog-to-Digital Converter

44.6.11.3 Offset Compensation

To determine the offset compensation value, the positive input must be tied to ground. The result of the ADC conversion gives directly the offset compensation value that must be written in the ADC OFFSETCORR register.

Downloaded from Arrow.com.

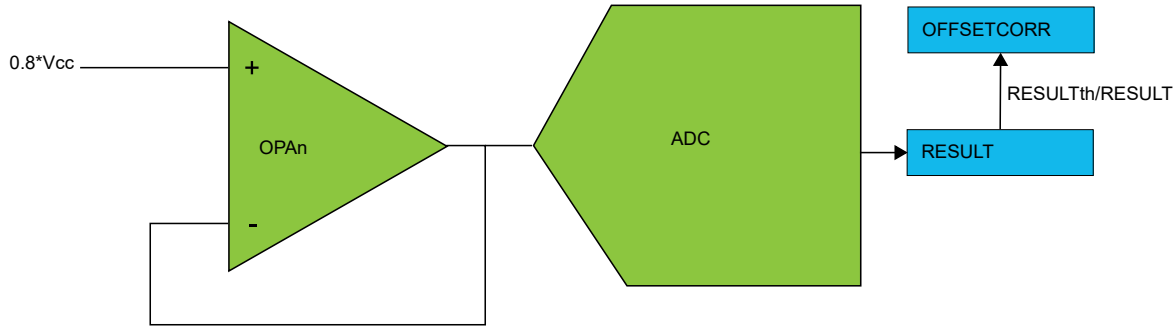
Figure 44-13. Offset Compensation



44.6.11.4 Gain Compensation

To perform gain compensation positive input must be close to VDD, but less (e.g. 0.8*Vref for instance) to avoid ADC saturation. The value for gain error compensation is obtained by dividing the theoretical ADC conversion result by the result from measurement. The obtained value for gain error compensation must be written in the ADC GAINCORR register.

Figure 44-14. Gain Compensation



44.6.12 AC Driver

One or several OPAMPs can be configured as input for the AC. The AC input mux must be appropriately configured to select OPAMP as input.

Related Links

42. AC – Analog Comparators

44.6.13 Input Connection to DAC

The DAC can be used as a reference. This is configured by the corresponding OPAMPCTRLx.MUXPOS and OPAMPCTRLx.RES1MUX bits.

44.6.14 Voltage Doubler

The OPAMP peripheral contains a voltage doubler for the analog multiplexer switches to ensure proper operation for a supply voltage below 2.5V. Aside from the multiplexers, no other supply voltages are affected by the voltage doubler.

The voltage doubler is normally switched on/off automatically, based on the supply level. If the supply voltage is guaranteed to be above 2.5V, the voltage doubler can be completely disabled by writing the Low-Power Mux bit in the Control Register (CTRLA.LPMUX).

Downloaded from Arrow.com.

When enabling OPAMPs, additional start-up time is required for the voltage doubler to settle. Disabling the voltage doubler saves power and reduces the startup time.

44.6.15 Performance vs. Power Consumption

It is possible to tradeoff speed versus power efficiency to get the shortest possible propagation delay or the lowest power consumption.

The speed setting is configured for each amplifier individually by the Bias Control field in the Operational Amplifier x Control register (OPAMPCTRLx.BIAS). The BIAS bits select the amount of bias current provided to the operational amplifiers. This will also affect the start-up time.

Downloaded from Arrow.com.

44.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	LPMUX						ENABLE	SWRST
0x01	Reserved									
0x02	STATUS	7:0						READYx	READYx	READYx
0x03	Reserved									
0x04	OPAMPCTRL0	7:0	ONDEMAND	RUNSTDBY	RES2VCC	BIAS[1:0]		ANAOUT	ENABLE	
		15:8	POTMUX[2:0]			RES1MUX[2:0]			RES1EN	RES2OUT
		23:16	MUXNEG[3:0]				MUXPOS[3:0]			
		31:24								
0x08	OPAMPCTRL1	7:0	ONDEMAND	RUNSTDBY	RES2VCC	BIAS[1:0]		ANAOUT	ENABLE	
		15:8	POTMUX[2:0]			RES1MUX[2:0]			RES1EN	RES2OUT
		23:16	MUXNEG[3:0]				MUXPOS[3:0]			
		31:24								
0x0C	OPAMPCTRL2	7:0	ONDEMAND	RUNSTDBY	RES2VCC	BIAS[1:0]		ANAOUT	ENABLE	
		15:8	POTMUX[2:0]			RES1MUX[2:0]			RES1EN	RES2OUT
		23:16	MUXNEG[3:0]				MUXPOS[3:0]			
		31:24								
0x10	RESCTRL	7:0	REFBUFLEVEL[1:0]			POTMUX[2:0]		RES1MUX	RES1EN	RES2OUT

44.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

Downloaded from Arrow.com.

44.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	LPMUX						ENABLE	SWRST
Access	R/W						R/W	R/W
Reset	0						0	0

Bit 7 – LPMUX Low-Power Mux

Value	Description
0	The analog input muxes have low resistance, but consume more power at lower voltages (e.g., are driven by the voltage doubler).
1	The analog input muxes have high resistance, but consume less power at lower voltages (e.g., the voltage doubler is disabled).

Bit 1 – ENABLE Enable

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled. Each OPAMP must also be enabled individually by the Enable bit in the corresponding OPAMP Control register (OPAMPCTRLx.ENABLE).

Bit 0 – SWRST Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the MODULE to their initial state, and the OPAMP will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

Downloaded from Arrow.com.

44.8.2 Status

Name: STATUS
Offset: 0x02
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
						READYx	READYx	READYx
Access						R	R	R
Reset						0	0	0

Bits 2,1,0 – READYx OPAMP x Ready

This bit is set when the OPAMPx output is ready.

This bit is cleared when the output of OPAMPx is not ready.

Downloaded from Arrow.com.

44.8.3 OPAMP Control x

Name: OPAMPCTRL
Offset: 0x04 + n*0x04 [n=0..2]
Reset: 0x00000080
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	MUXNEG[3:0]				MUXPOS[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	POTMUX[2:0]			RES1MUX[2:0]			RES1EN	RES2OUT
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY	RES2VCC	BIAS[1:0]		ANAOUT	ENABLE	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	

Bits 23:20 – MUXNEG[3:0] Negative Input Mux Selection
Selection on negative input for operational amplifier x.

Value	OPAMPx	Name	Description
0x0	x=0,1,2	OAxNEG	OPAMPx Negative Input
0x1	x=0,1,2	OAxTAP	OPAMPx Resistor Ladder Taps
0x2	x=0,1,2	REFERENCE	REFERENCE[DAC/REFBUF]
0x3	x=0,1,2	OAxOUT	
0x4	x=0,1	Reserved	
	x=2	OA0NEG	OPAMP0 Negative Input
0x5	x=0,1	Reserved	
	x=2	OA1NEG	OPAMP1 Negative Input

Bits 19:16 – MUXPOS[3:0] Positive Input Mux Selection
Selection on positive input for operational amplifier x.

Downloaded from Arrow.com.

Value	OPAMPx	Name	Description
0x0	x=0,1,2	OAxPOS	OPAMPx Positive Input
0x1	x=0,1,2	OAxTAP	OPAMPx Resistor Ladder Taps
0x2	x=0,1,2	REFERENCE	REFERENCE[DAC/REFBUF]
0x3	x=0,1,2	GND	Ground
0x4	x=0	Reserved	
	x=1	OA0OUT	OPAMP0 output
	x=2	OA1OUT	OPAMP1 output
0x5	x=0,1	Reserved	
	x=2	OA0POS	OPAMP0 Positive Input
0x6	x=0,1	Reserved	
	x=2	OA1POS	OPAMP1 Positive Input
0x7	x=0,1	Reserved	
	x=2	OA0TAP	OPAMP0 Resistor Ladder Taps
0x8	x=0,1	Reserved	
	x=2	RES3TAP	

Bits 15:13 – POTMUX[2:0] Potentiometer selection
Resistor selection bits control a numeric potentiometer with eight fixed values.

Value	R1	R2
0x0	14R	2R
0x1	12R	4R
0x2	8R	8R
0x3	6R	10R
0x4	4R	12R
0x5	3R	13R
0x6	2R	14R
0x7	R	15R

Bits 12:10 – RES1MUX[2:0] Resistor 1 Mux
These bits select the connection of R1 resistor of the potentiometer.

Value	OPAMPx	Name	Description
0x0	x=0,1,2	OAxPOS	OPAMPx Positive Input
0x1	x=0,1,2	OAxNEG	OPAMPx Negative Input

Downloaded from Arrow.com.

Value	OPAMPx	Name	Description
0x2	x=0	REFERENCE	REFERENCE[DAC/REFBUF]
	x=1	OA0OUT	OPAMP0 output
	x=2	OA1OUT	OPAMP1 output
0x3	x=0,1,2	GND	
0x4	x=0,1	RG_CONN	
	x=2	Reserved	

Bit 9 – RES1EN Resistor 1 Enable

Value	Description
0	R1 disconnected from RES1MUX.
1	R1 connected to RES1MUX.

Bit 8 – RES2OUT Resistor ladder To Output

Value	Description
0	Swith open.
1	Switch closed.

Bit 7 – ONDEMAND On Demand Control
The On Demand operation mode allows the OPAMPx to be enabled or disabled, depending on other peripheral requests.

Value	Description
0	The OPAMPx is always on, if enabled.
1	The OPAMPx is enabled when a peripheral is requesting the OPAMPx to be used as an input. The OPAMPx is disabled if no peripheral is requesting it as an input.

Bit 6 – RUNSTDBY Run in Standby
This bit controls how the OPAMPx behaves during standby sleep mode:

Value	Description
0	The OPAMPx is disabled in standby sleep mode.
1	The OPAMPx is not stopped in standby sleep mode. If OPAMPCTRLx.ONDEMAND=1, the OPAMPx will be running when a peripheral is requesting it as an input. If OPAMPCTRLx.ONDEMAND=0, OPAMPx will always be running in standby sleep mode.

Bit 5 – RES2VCC Resistor ladder To VCC

Value	Description
0	Swith open.
1	Switch closed.

Bits 4:3 – BIAS[1:0] Bias Selection
These bits are used to select the bias mode.

Downloaded from Arrow.com.

Value	Name	Description
0x0	Mode 0	Minimum current consumption, but the slowest mode
0x1	Mode 1	Low current consumption, slow speed
0x2	Mode 2	High current consumption, fast speed
0x3	Mode 3	Maximum current consumption but the fastest mode

Bit 2 – ANAOUT Analog Output
This bit controls a switch connected to the OPAMP output.

Value	Description
0	Switch open. No ADC or AC connection.
1	Switch closed. OPAMP output is connected to the ADC or AC input.

Bit 1 – ENABLE Operational Amplifier Enable

Value	Description
0	The OPAMPx is disabled
1	The OPAMPx is enabled

Downloaded from Arrow.com.

44.8.4 Resistor Control

Name: RESCTRL
Offset: 0x10
Reset: 0x0
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	REFBUFLEVEL[1:0]		POTMUX[2:0]			RES1MUX	RES1EN	RES2OUT
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:6 – REFBUFLEVEL[1:0] Reference output voltage level select
Reference output voltage level select.

Value	Reference Level
0x0	1.1v
0x1	1.25v
0x2	1.6v
0x3	Reserved

Bits 5:3 – POTMUX[2:0] Potentiometer selection
Resistor selection bits control a numeric potentiometer with eight fixed values.

Value	R1	R2
0x0	14R	2R
0x1	12R	4R
0x2	8R	8R
0x3	6R	10R
0x4	4R	12R
0x5	3R	13R
0x6	2R	14R
0x7	R	15R

Bit 2 – RES1MUX Resistor 1 Mux
These bits select the connection of R1 resistor of the potentiometer.

Value	Name	Description
0x0	DAC	DACOUT
0x1	REFBUF	REFBUF

Downloaded from Arrow.com.

Bit 1 – RES1EN Resistor 1 Enable
RES1EN = 1 is required in order to provide DAC/REFBUF to POSMUX, NEGMUX and RES1MUX.

Value	Description
0	R1 disconnected from RES1MUX.
1	R1 connected to RES1MUX.

Bit 0 – RES2OUT Resistor ladder To Output
RES2OUT switch can only be closed if all OPAMPs are enabled.

Value	Description
0	Swith open.
1	Switch closed.

Downloaded from Arrow.com.

45. PTC - Peripheral Touch Controller

45.1 Overview

The Peripheral Touch Controller (PTC) acquires signals in order to detect touch on capacitive sensors. The external capacitive touch sensor is typically formed on a PCB, and the sensor electrodes are connected to the analog front end of the PTC through the I/O pins in the device. The PTC supports both self- and mutual-capacitance sensors.

In mutual-capacitance mode, sensing is done using capacitive touch matrices in various X-Y configurations, including indium tin oxide (ITO) sensor grids. The PTC requires one pin per X-line and one pin per Y-line.

In self-capacitance mode, the PTC requires only one pin (Y-line) for each touch sensor.

The number of available pins and the assignment of X- and Y-lines is depending on both package type and device configuration. Refer to the Configuration Summary and I/O Multiplexing table for details.

Related Links

1. Configuration Summary

45.2 Features

- Low-power, high-sensitivity, environmentally robust capacitive touch buttons, sliders, and wheels
- Supports wake-up on touch from standby Sleep mode
- Supports mutual capacitance and self-capacitance sensing
  - Mix-and-match mutual-and self-capacitance sensors
- One pin per electrode – no external components
- Load compensating charge sensing
  - Parasitic capacitance compensation and adjustable gain for superior sensitivity
- Zero drift over the temperature and V_{DD} range
  - Auto calibration and recalibration of sensors
- Single-shot and free-running charge measurement
- Hardware noise filtering and noise signal desynchronization for high conducted immunity
- Polarity control, allowing Parallel Acquisition (through the QTouch Library) individually controls the polarity of each line
- Driven Shield Plus for better noise immunity and moisture tolerance
  - Any PTC X/Y line can be used for the driven shield
  - All enabled sensors will be driven at the same potential as the sensor scanned
- Selectable channel change delay allows choosing the settling time on a new channel, as required
- Acquisition-start triggered by command or through auto-triggering feature
- Low CPU utilization through interrupt on acquisition-complete

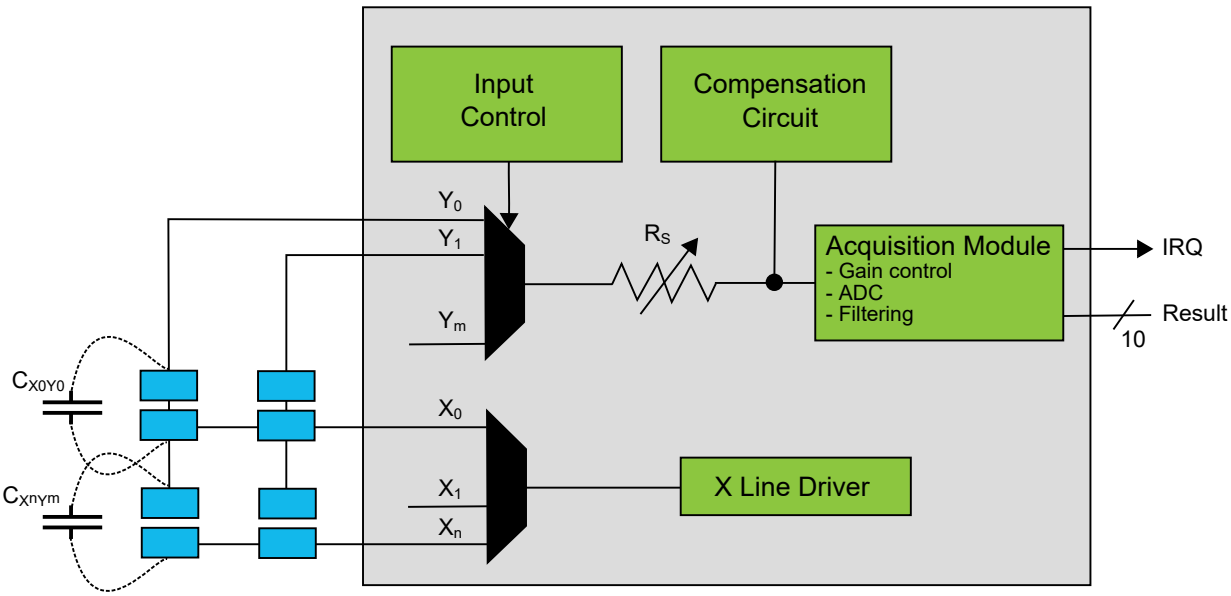
Related Links

1. Configuration Summary

Downloaded from Arrow.com.

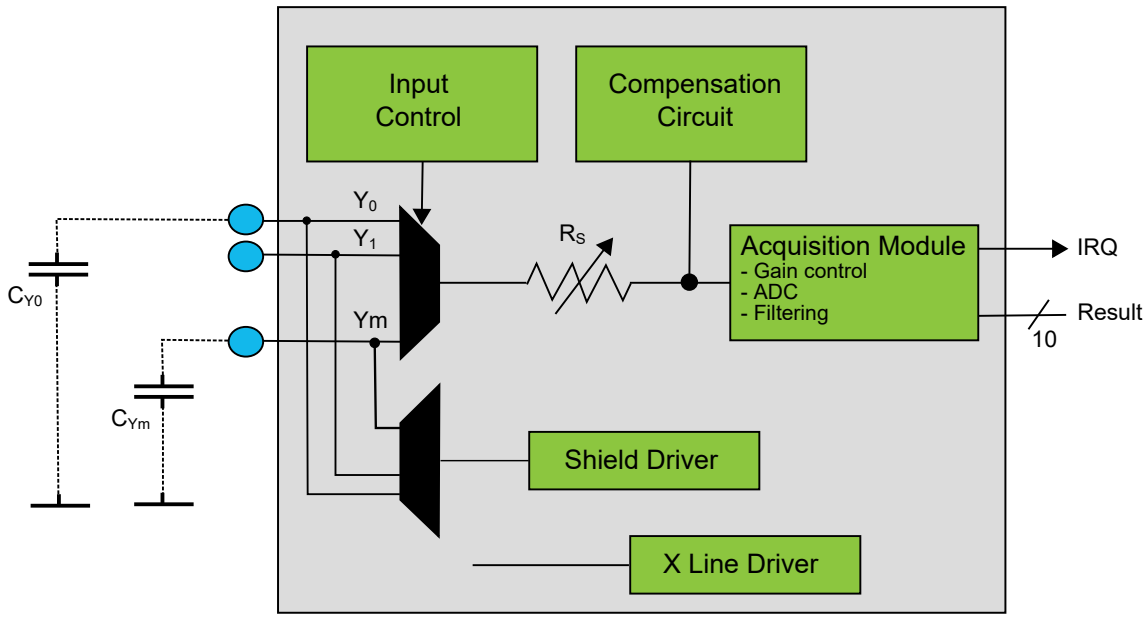
45.3 Block Diagram

Figure 45-1. PTC Block Diagram Mutual Capacitance



Note: For SAM L10/L11 the R_S = 100 KΩ.

Figure 45-2. PTC Block Diagram Self Capacitance



Note: For SAM L10/L11 the RS = 100 KΩ.

Downloaded from Arrow.com.

45.4 Signal Description

Table 45-1. Signal Description for PTC

Name	Type	Description
Y[m:0]	Analog	Y-line (Input/Output)
X[n:0]	Digital	X-line (Output)

Note: The number of X- and Y-lines are device dependent. Refer to *Configuration Summary* for details.

Refer to *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

Related Links

1. Configuration Summary

45.5 System Dependencies

In order to use this peripheral, configure the other components of the system as described in the following sections.

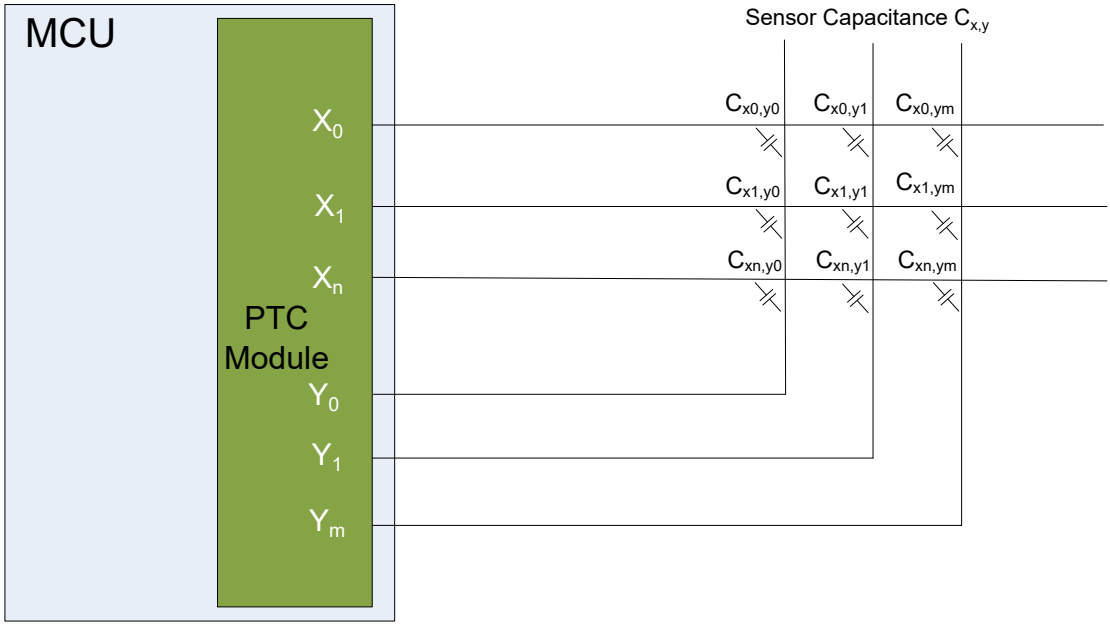
45.5.1 I/O Lines

The I/O lines used for analog X-lines and Y-lines must be connected to external capacitive touch sensor electrodes. External components are not required for normal operation. However, to improve the EMC performance, a series resistor of 1kΩ or more can be used on X-lines and Y-lines.

45.5.1.1 Mutual-Capacitance Sensor Arrangement

A mutual-capacitance sensor is formed between two I/O lines - an X electrode for transmitting and Y electrode for sensing. The mutual capacitance between the X and Y electrode is measured by the Peripheral Touch Controller.

Figure 45-3. Mutual Capacitance Sensor Arrangement

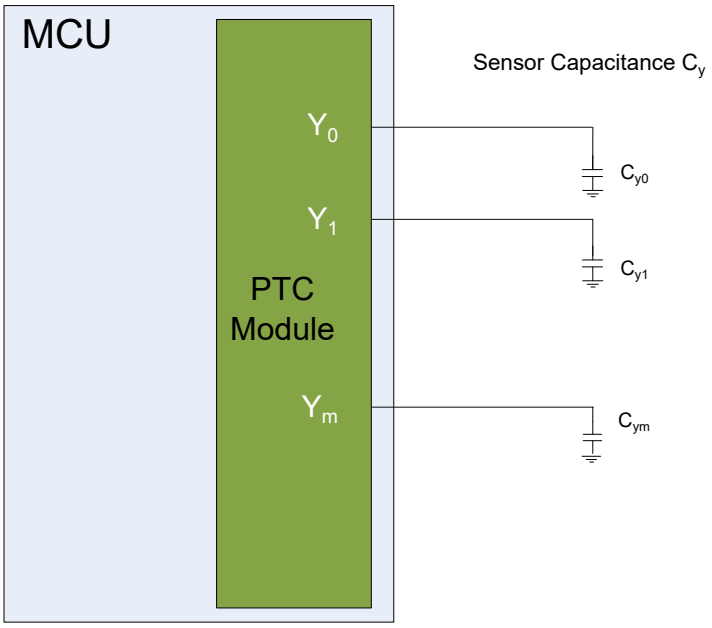


Downloaded from Arrow.com.

45.5.1.2 Self-Capacitance Sensor Arrangement

A self-capacitance sensor is connected to a single pin on the Peripheral Touch Controller through the Y electrode for sensing the signal. The sense electrode capacitance is measured by the Peripheral Touch Controller.

Figure 45-4. Self-capacitance Sensor Arrangement



For more information about designing the touch sensor, refer to Buttons, Sliders and Wheels Touch Sensor Design Guide.

45.5.2 Clocks

The PTC is clocked by the GCLK_PTC clock. The PTC operates from an asynchronous clock source and the operation is independent of the main system clock and its derivative clocks, such as the peripheral bus clock (CLK_APB). A number of clock sources can be selected as the source for the asynchronous GCLK_PTC. The clock source is selected by configuring the Generic Clock Selection ID in the Generic Clock Control register. For more information about selecting the clock sources, refer to *GCLK - Generic Clock Controller*.

The selected clock must be enabled in the Power Manager, before it can be used by the PTC. By default these clocks are disabled. The frequency range of GCLK_PTC is 400kHz to 4MHz.

Related Links

18. GCLK - Generic Clock Controller
22. PM – Power Manager

45.5.3 SAM L11 TrustZone Specific Register Access Protection

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Downloaded from Arrow.com.

Refer to *Peripherals Security Attribution* for more information.

45.6 Functional Description

In order to access the PTC, the user must use the Atmel Start QTouch Configurator to configure and link the QTouch Library firmware with the application software. QTouch Library can be used to implement buttons, sliders, wheels in a variety of combinations on a single interface.

For more information about QTouch Library, refer to the QTouch Library Peripheral Touch Controller User Guide.

Downloaded from Arrow.com.

46. Electrical Characteristics

This section provides an overview of the SAM L10 and SAM L11 electrical characteristics.

Specifications for Extended Temperature Devices (-40ºC to +125ºC) that are different from the specifications in this section are provided in 47. 125°C Electrical Characteristics.

46.1 Disclaimer

All typical values are measured at T = 25°C unless otherwise specified. All minimum and maximum values are valid across operating temperature and voltage unless otherwise specified.

46.2 Thermal Considerations

46.2.1 Thermal Resistance Data

The following table summarizes the thermal resistance data depending on the package.

Table 46-1. Thermal Resistance Data

Package Type	θ_{JA}	θ_{JC}
32-pin TQFP	65.1°C/W	16.3°C/W
32-pin QFN	40.4°C/W	15.8°C/W
24-pin QFN	59.1°C/W	21.1°C/W
24-pin SSOP	76.3°C/W	16.0°C/W
32-pin WLCSP	76.89°C/W	12.15°C/W

46.2.2 Junction Temperature

The average chip-junction temperature, T_J, in °C can be obtained from the following:

1. $T_J = T_A + (P_D \times \theta_{JA})$
2. $T_J = T_A + (P_D \times (\theta_{HEATSINK} + \theta_{JC}))$

where:

- θ_{JA} = Package thermal resistance, Junction-to-ambient (°C/W), see Thermal Resistance Data
- θ_{JC} = Package thermal resistance, Junction-to-case thermal resistance (°C/W), see Thermal Resistance Data
- $\theta_{HEATSINK}$ = Thermal resistance (°C/W) specification of the external cooling device
- P_D = Device power consumption (W)
- T_A = Ambient temperature (°C)

From the first equation, the user can derive the estimated lifetime of the chip and decide if a cooling device is necessary or not. If a cooling device is to be fitted on the chip, the second equation should be used to compute the resulting average chip-junction temperature T_J in °C.

Downloaded from Arrow.com.

46.3 Absolute Maximum Ratings

Stresses beyond those listed in the following table may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Table 46-2. Absolute Maximum Ratings

Symbol	Description	Min.	Max.	Units
V_{DD}	Power supply voltage	0	3.8	V
I_{VDD}	Current into a V_{DD} pin	-	46(1)	mA
I_{GND}	Current out of a GND pin	-	65(1)	mA
V_{PIN}	Pin voltage with respect to GND and V_{DD}	GND-0.6V	VDD+0.6V	V
$T_{storage}$	Storage temperature	-60	150	°C

Note:
1. Maximum source current is 46 mA and maximum sink current is 65 mA per cluster. A cluster is a group of GPIOs.

Also note that each VDD\GND pair is connected to two clusters, hence current consumption through the pair will be a sum of the clusters' source/sink currents.

⚠ CAUTION This device is sensitive to electrostatic discharges (ESD). Improper handling may lead to permanent performance degradation or malfunctioning.
Handle the device following best practice ESD protection rules: Be aware that the human body can accumulate charges large enough to impair functionality or destroy the device.

46.4 General Operating Ratings

The device must operate within the ratings listed in the following table for all other electrical characteristics and typical characteristics of the device to be valid.

Table 46-3. General Operating Conditions

Symbol	Description	Min.	Typ.	Max.	Units
V_{DDIO}	IO Supply Voltage	1.62	3.3	3.63	V
V_{DDANA}	Analog supply voltage	1.62	3.3	3.63	V
T_A	Temperature range	-40	25	85	°C
T_J	Junction temperature	-	-	105	°C

Downloaded from Arrow.com.

46.5 Supply Characteristics

Table 46-4. Supply Characteristics

Symbol	Voltage		
	Min.	Max.	Units
V_{DDIO}	1.62	3.63	V
V_{DDANA}	1.62	3.63	V

Table 46-5. Supply Slew Rates(1)

Symbol	Fall Rate	Rise Rate	Units
	Max.	Max.	
V_{DDIO}	0.05	0.1	V/µs
V_{DDANA}	0.05	0.1	V/µs

Note: 1. These values are based on simulation. They are not covered by production test limits or characterization.

46.6 Maximum Clock Frequencies

Table 46-6. Maximum GCLK Generator Output Frequencies(1)

Symbol	Description	Conditions	Fmax		Units
			PL0	PL2	
$F_{gclkgen}[0:2]$	GCLK Generator output Frequency	-	24	96	MHz
$F_{gclkgen}[3:4]$		-	12	48	MHz

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.

Table 46-7. Maximum Peripheral Clock Frequencies(1)

Symbol	Description	Conditions	Fmax.		Units
			PL0	PL2	
f_{CPU}	CPU clock frequency	-	8	32	MHz
f_{AHB}	AHB clock frequency	-	8	32	MHz
f_{APBA}	APBA clock frequency	-	8	32	MHz
f_{APBB}	APBB clock frequency	-			
f_{APBC}	APBC clock frequency	-			
$f_{GCLK_DFLLULP}$	DFLLULP Reference clock frequency	-	1	4	MHz

Downloaded from Arrow.com.

Symbol	Description	Conditions	Fmax.		Units
			PL0	PL2	
f_{GCLK_DPLL}	FDPLL96M Reference clock frequency	-	2	2	MHz
$f_{GCLK_DPLL_32K}$	FDPLL96M 32K clock frequency	-	32.768	32.768	kHz
f_{GCLK_EIC}	EIC input clock frequency	-	12	48	MHz
$f_{GCLK_EVSYS_CHANNEL_0}$	EVSYS channel 0 input clock frequency	-	12	48	MHz
$f_{GCLK_EVSYS_CHANNEL_1}$	EVSYS channel 1 input clock frequency	-			
$f_{GCLK_EVSYS_CHANNEL_2}$	EVSYS channel 2 input clock frequency	-			
$f_{GCLK_EVSYS_CHANNEL_3}$	EVSYS channel 3 input clock frequency	-			
$f_{GCLK_SERCOM0_SLOW}$	Common SERCOM0 slow input clock frequency	-	1	5	MHz
$f_{GCLK_SERCOM1_SLOW}$	Common SERCOM1 slow input clock frequency	-			
$f_{GCLK_SERCOM2_SLOW}$	Common SERCOM2 slow input clock frequency	-			
$f_{GCLK_SERCOM0_CORE}$	SERCOM0 input clock frequency	-	12	48	MHz
$f_{GCLK_SERCOM1_CORE}$	SERCOM1 input clock frequency	-			
$f_{GCLK_SERCOM2_CORE}$	SERCOM2 input clock frequency	-			
f_{GCLK_TC0}	TC0 input clock frequency	-	12	48	MHz
f_{GCLK_TC1}	TC1 input clock frequency	-			
f_{GCLK_TC2}	TC2 input clock frequency	-			
f_{GCLK_ADC}	ADC input clock frequency	-	12	48	MHz
f_{GCLK_AC}	AC digital input clock frequency	-			
f_{GCLK_DAC}	DAC input clock frequency	-			
$f_{GCLK_FREQM_MSR}$	FREQM measured clock frequency	-			
$f_{GCLK_FREQM_REF}$	FREQM reference clock frequency	-			
f_{GCLK_PTC}	PTC input clock frequency	-			
f_{GCLK_CCL}	CCL input clock frequency	-			
f_{GCLKin}	External GCLK input clock frequency	-			

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.

Downloaded from Arrow.com.

46.7 Power Consumption

The values in this section are measured values of power consumption under the following conditions, except where noted:

- **Operating Conditions**
  - V_{DDIO} = 3.3V or 1.8V
  - CPU is running on Flash with required Wait states, as recommended in the NVM Characteristics section
  - Low-power cache is enabled
  - BOD33 is disabled
  - I/Os are configured with digital input trigger disabled (default Reset configuration)
- **Oscillators**
  - XOSC (crystal oscillator) stopped
  - XOSC32K (32.768 kHz crystal oscillator) running with external 32.768 kHz crystal
  - When in Active mode with Performance Level 2 (PL2), DPLL is running at 32 MHz and using XOSC32K as reference
  - When in Active mode on DFLLULP, the DFLLULP is configured in Closed Loop mode using XOSC32K as reference clock and MCLK.CTRLA.CKSEL = 1

Table 46-8. Active Current Consumption

Mode	Conditions	Regulator	PL	CPU Clock	Vcc	Ta	Typ.	Max	Units
ACTIVE	COREMARK/ FIBONACCI	LDO	PL0	DFLLULP at 8MHz	1.8V	Max at 85°C Typ at 25°	64.1	82	µA/Mhz
					3.3V		64.4	84	
				OSC 8MHz	1.8V		66.6	81	
					3.3V		70.3	83	
				OSC 4MHz	1.8V		74.1	102	
					3.3V		77.8	106	
			PL2	FDPLL96M at 32MHz	1.8V		82.0	89	
					3.3V		82.5	89	
				DFLLULP at 32MHz	1.8V		75.8	99	
					3.3V		75.8	96	
		BUCK	PL0	DFLLULP at 8MHz	1.8V		40.0	53	
					3.3V		25.3	34	
				OSC 8MHz	1.8V		43.8	53	
					3.3V		32.1	39	
				OSC 4MHz	1.8V		50.3	68	
					3.3V		38.9	52	
			PL2	FDPLL96M at 32MHz	1.8V		59.9	66	

Downloaded from Arrow.com.

Mode	Conditions	Regulator	PL	CPU Clock	Vcc	Ta	Typ.	Max	Units
					3.3V		35.3	39	
				DFLLULP at 32MHz	1.8V		55.3	68	
					3.3V		32.6	41	
	WHILE1	LDO	PL0	DFLLULP at 8MHz	1.8V		44.3	61	
					3.3V		44.4	62	
				OSC 8MHz	1.8V		47.6	60	
					3.3V		50.1	63	
				OSC 4MHz	1.8V		54.6	83	
					3.3V		57.7	86	
			PL2	FDPLL96M at 32MHz	1.8V		56.9	61	
					3.3V		57.2	62	
				DFLLULP at 32MHz	1.8V		50.8	66	
					3.3V		51.0	64	
		BUCK	PL0	DFLLULP at 8MHz	1.8V		28.1	40	
					3.3V		18.5	27	
				OSC 8MHz	1.8V		32.2	41	
					3.3V		25.3	32	
				OSC 4MHz	1.8V		38.4	57	
					3.3V		31.9	45	
			PL2	FDPLL96M at 32MHz	1.8V		41.5	46	
					3.3V		24.6	28	
				DFLLULP at 32MHz	1.8V		37.1	47	
					3.3V		22.0	28	
IDLE		LDO	PL0	DFLLULP at 8MHz	1.8V		16.0	32	
					3.3V		16.2	33	
				OSC 8MHz	1.8V		19.8	33	
					3.3V		22.0	36	
				OSC 4MHz	1.8V		26.2	55	
					3.3V		29.2	59	
			PL2	FDPLL96M at 32MHz	1.8V		20.3	25	
					3.3V		20.4	26	

Downloaded from Arrow.com.

Mode	Conditions	Regulator	PL	CPU Clock	Vcc	Ta	Typ.	Max	Units
				DFLLULP at 32MHz	1.8V		14.3	19	
					3.3V		14.4	19	
		BUCK	PL0	DFLLULP at 8MHz	1.8V		11.1	21	
					3.3V		8.3	16	
				OSC 8MHz	1.8V		15.5	24	
					3.3V		15.2	21	
				OSC 4MHz	1.8V		21.3	39	
					3.3V		21.6	35	
			PL2	FDPLL96M at 32MHz	1.8V		14.9	19	
					3.3V		9.1	12	
				DFLLULP at 32MHz	1.8V		10.6	14	
					3.3V		6.7	9	

Table 46-9. Standby and Off Mode Current Consumption

Mode	Conditions	Regulator Mode	Vcc	Ta	Typ.	Max.	Units
STANDBY	All 16 kB RAM retained, PDSW domain in active state	LPVREG with LPEFF Disable	1.8V	25°C	1.3	3.5	µA
				85°C	18.4	66.0	
		LPVREG with LPEFF Enable	3.3V	25°C	1.1	3.0	
				85°C	14.2	41.8	
		BUCK in standby with MAINVREG in PL0 mode (VREG.RUNSTDBY=1 and VREG.STDBYPL0=1)	1.8V	25°C	1.2	2.9	
				85°C	14.6	42.9	
			3.3V	25°C	1.1	2.2	
				85°C	9.6	28.6	
	All 16 kB RAM retained, PDSW domain in retention	LPVREG with LPEFF Disable	1.8V	25°C	0.6	1.1	
				85°C	5.1	14.9	
		LPVREG with LPEFF Enable	3.3V	25°C	0.5	1.0	
				85°C	4.3	12.1	
		BUCK in standby with MAINVREG in PL0 mode (VREG.RUNSTDBY=1 and VREG.STDBYPL0=1)	1.8V	25°C	0.8	1.1	
				85°C	4.3	11.9	
			3.3V	25°C	0.8	1.5	
				85°C	3.4	8.5	

Downloaded from Arrow.com.

Mode	Conditions	Regulator Mode	Vcc	Ta	Typ.	Max.	Units
	12 kB RAM retained,PDSW domain in retention	LPVREG with LPEFF Disable	1.8V	25°C	0.6	1.1	
				85°C	4.7	13.6	
		LPVREG with LPEFF Enable	3.3V	25°C	0.5	1.0	
				85°C	4.0	11.1	
		BUCK in standby with MAINVREG in PL0 mode (VREG.RUNSTDBY=1 and VREG.STDBYPL0=1)	1.8V	25°C	0.7	1.1	
				85°C	4.1	11.0	
			3.3V	25°C	0.8	1.5	
				85°C	3.2	8.0	
	8kB RAM retained,PDSW domain in retention	LPVREG with LPEFF Disable	1.8V	25°C	0.5	1.0	
				85°C	4.4	12.6	
		LPVREG with LPEFF Enable	3.3V	25°C	0.5	0.9	
				85°C	3.8	10.3	
		BUCK in standby with MAINVREG in PL0 mode (VREG.RUNSTDBY=1 and VREG.STDBYPL0=1)	1.8V	25°C	0.7	1.0	
				85°C	3.8	10.1	
			3.3V	25°C	0.7	1.4	
				85°C	3.0	7.9	
	4kB RAM retained,PDSW domain in retention	LPVREG with LPEFF Disable	1.8V	25°C	0.5	0.9	
				85°C	4.0	11.2	
		LPVREG with LPEFF Enable	3.3V	25°C	0.5	0.9	
				85°C	3.5	9.3	
		BUCK in standby with MAINVREG in PL0 mode (VREG.RUNSTDBY=1 and VREG.STDBYPL0=1)	1.8V	25°C	0.7	1.0	
				85°C	3.5	9.1	
			3.3V	25°C	0.8	1.5	
				85°C	2.9	6.8	
	4kB RAM retained,PDSW domain in retention and RTC running on XOSC32K	LPVREG with LPEFF Disable	1.8V	25°C	0.9	1.3	
				85°C	4.5	11.7	
		LPVREG with LPEFF Enable	3.3V	25°C	0.8	1.2	
				85°C	4.0	9.8	
		BUCK in standby with MAINVREG in PL0 mode (VREG.RUNSTDBY=1 and VREG.STDBYPL0=1)	1.8V	25°C	1.0	1.3	
				85°C	4.0	9.6	
			3.3V	25°C	1.1	1.7	
				85°C	3.3	7.3	

Downloaded from Arrow.com.

Mode	Conditions	Regulator Mode	Vcc	Ta	Typ.	Max.	Units
OFF			1.8V	25°C	34.6	54.4	nA
				85°C	595.7	1197.3	
			3.3V	25°C	61.2	89.1	
				85°C	796.1	1622.8	

46.8 Wake-Up Time

Conditions:

- VDDIO/VDDANA = 3.3V
- LDO Regulation mode
- CPU clock = OSC16M @ 4 MHz
- One Wait-state
- Cache enabled
- Flash Fast Wake-up enabled (NVMCTRL.CTRLB.FWUP = 1)
- Flash in WAKEUPINSTANT mode (NVMCTRL.CTRLB.SLEEPPRM = 1)

Measurement Method:

For Idle and Standby, the CPU sets an I/O by writing PORT->IOBUS without jumping in an interrupt handler (Cortex M23 register PRIMASK = 1). The wake-up time is measured between the edge of the wake-up input signal and the edge of the GPIO pin.

For Off mode, the exit of the mode is done through the reset pin, the time is measured between the falling edge of the RESETN signal (with the minimum reset pulse length), and the set of the I/O which is done by the first executed instructions after Reset.

Table 46-10. Wake-Up Timing (1)

Sleep Mode	Condition		Typ	Unit
Idle	PL2 or PL0		1.5	µs
Standby	PL0	PDSW domain in retention	5.3	
		PDSW domain in active	2.6	
	PL2 Voltage scaling at default values: SUPC >VREG.VSVSTEP=0 SUPC > VREG.VSPER=0	PDSW domain in retention	76	
		PDSW domain in active	75	
	PL2 Voltage scaling at fastest setting: SUPC > VREG.VSVSTEP=15 SUPC > VREG.VSPER=0	PDSW domain in retention	16	
		PDSW domain in active	15	
OFF	L10 with BOOTOPT=0		3.2	ms

Downloaded from Arrow.com.

Sleep Mode	Condition	Typ	Unit
	L11 with BOOTOPT=0	4.1	
	L10 or L11 with BOOTOPT=1, BS = 0x40	210	
	L10 or L11 with BOOTOPT=1, BS = 0x80	410	
	L10 or L11 with BOOTOPT=2, BS = 0x40	210	
	L10 or L11 with BOOTOPT=2, BS = 0x80	410	

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.

46.9 I/O Pin Characteristics

There are two different pin types with three different speeds: Normal and High Sink(2).

The Drive Strength bit is located in the Pin Configuration register of the PORT (PORT.PINCFG.DRVSTR).

Table 46-11. I/O Pins Common Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
V_{IL}	Input low-level voltage	V_{DD}=1.62V-2.7V	-	-	0.25*V_{DD}	V
		V_{DD}=2.7V-3.63V	-	-	0.3*V_{DD}	
V_{IH}	Input high-level voltage	V_{DD}=1.62V-2.7V	0.7*V_{DD}	-	-	
		V_{DD}=2.7V-3.63V	0.55*V_{DD}	-	-	
V_{OL}	Output low-level voltage	V_{DD}>1.62V, I_{OL} max	-	0.1*V_{DD}	0.2*V_{DD}	
V_{OH}	Output high-level voltage	V_{DD}>1.62V, I_{OH} max	0.75*V_{DD}	0.85*V_{DD}	-	
R_{PULL}	Pull-up - Pull-down resistance		20	40	63	kΩ
I_{LEAK}	Input leakage current	Pull-up resistors disabled	-1	±0.015	1	µA

Table 46-12. I/O Pins Maximum Output Current

Symbol	Parameter	Conditions	Normal Pins	High Sink Pins(2)	Normal Pins	High Sink Pins(2)	Units
			DRVSTR=0		DRVSTR=1		
I_{OL}	Maximum Output low-level current	V_{DD}=1.62V-3V	1	2	2	4	mA
		V_{DD}=3V-3.63V	2.5	6	6	12	
I_{OH}	Maximum Output high-level current	V_{DD}=1.62V-3V	0.7	1.5	1.5	3	
		V_{DD}=3V-3.63V	2	5	5	10	

Downloaded from Arrow.com.

Table 46-13. I/O Pins Dynamic Characteristics(1)

Symbol	Parameter	Conditions	Normal Pins	High Sink Pins(2)	Normal Pins	High Sink Pins(2)	Units
			DRVSTR=0		DRVSTR=1		
t_{RISE}	Maximum Rise time	V_{DD}=3.3V, load=20 pF, 10%/90%	13	6	6	4.5	ns
t_{FALL}	Maximum Fall time	V_{DD}=3.3V, load=20 pF, 10%/90%	12	7	7	4.5	

The pins with I^2C alternative mode available are compliant(2) with I^2C norms. All I^2C pins support Standard mode (Sm), Fast mode (Fm), Fast plus mode (Fm+), and High speed mode (Hs, up to 3.4 MHz). The available I^2C pins are listed in the I/O Multiplexing section.

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. The following pins are High Sink pins and have different properties than normal pins: PA16, PA17, PA22,PA23, and PA31.

46.10 Injection Current

Stresses beyond those listed in the table below may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Table 46-14. Injection Current(1)

Symbol	Description	min	max	Unit
I_{inj1} (2)	I/O pin injection current	-1	+1	mA
I_{inj2} (3)	I/O pin injection current	-15	+15	mA
$I_{injtotal}$	Sum of I/O pins injection current	-45	+45	mA

Note:

1. Injecting current may have an effect on the accuracy of Analog blocks
2. Conditions for V_{pin}: V_{pin} < GND-0.6V or 3.6V<V_{pin}≤4.2V.

   Conditions for V_{DD}: 3V<V_{DD}≤3.6V.

   If V_{pin} is lower than GND-0.6V, a current limiting resistor is required. The negative DC injection current limiting resistor *R* is calculated as $R = |(GND-0.6V - V_{pin})/I_{inj1}|$.

   If V_{pin} is greater than V_{DD}+0.6V, a current limiting resistor is required. The positive DC injection current limiting resistor *R* is calculated as $R = (V_{pin}-(V_{DD}+0.6))/I_{inj1}$.
3. Conditions for V_{pin}: V_{pin} < GND-0.6V or V_{pin}≤3.6V.

   Conditions for V_{DD}: V_{DD}≤3V.

Downloaded from Arrow.com.

If V_{pin} is lower than GND-0.6V, a current limiting resistor is required. The negative DC injection current limiting resistor *R* is calculated as $R = |(GND-0.6V - V_{pin})/I_{inj2}|$.

If V_{pin} is greater than V_{DD}+0.6V, a current limiting resistor is required. The positive DC injection current limiting resistor *R* is calculated as $R = (V_{pin}-(V_{DD}+0.6))/I_{inj2}$.

46.11 Analog Characteristics

46.11.1 Voltage Regulator Characteristics

46.11.1.1 Buck Converter

Table 46-15. Buck Converter Electrical Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
P_{EFF}	Power Efficiency(2)	I_{OUT} = 5mA	-	86	-	%
		I_{OUT} = 50mA	-	85	-	%
VREGSCAL	Voltage scaling(1)	min step size for PLx to PLy transition	-	5	-	mV
		Voltage Scaling Period	-	1	-	µs

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. These values are based on characterization.

Table 46-16. External Components Requirements in Switching Mode(1)

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
C_{IN}	Input regulator capacitor	Tantalum or Electrolitic Dielectric	-	10	-	µF
		Ceramic dielectric	-	100	-	nF
C_{OUT}	Output regulator capacitor	Tantalum or Electrolitic Dielectric	-	1	-	µF
		Ceramic dielectric	-	100	-	nF
L_{EXT}	External inductance	Murata LQH3NPN100MJ0	-	10	-	µH
$R_{SERIE_L_{EXT}}$	Serial resistance of Lext	-	-	-	0.7	Ω
$R_{SERIE_C_{OUT}}$	Serial resistance of C_{OUT} (including $C_{OUT\ ESR}$)	-	-	-	0.5	Ω
SFR_L_{EXT}	Self-resonant frequency of L_{EXT}	-	30	-	-	MHz
$I_{SAT_L_{EXT}}$	Saturation current	-	275	-	-	mA

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.

Downloaded from Arrow.com.

46.11.1.2 LDO Regulator

Table 46-17. LDO Regulator Electrical Characteristics (1)

Symbol	Parameter	Conditions	Typ.	Units
VREGSCAL	Voltage scaling	min step size for PLx to Ply transition	5	mV
		Voltage Scaling Period	1	µs

Note: 1. These values are based on simulation. They are not covered by production test limits or characterization.

Table 46-18. External Components Requirements in Linear Mode

Symbol	Parameter	Conditions	Typ.	Units
C_{IN}	Input regulator capacitor	Tantalum or electrolitic dielectric	10	µF
		Ceramic dielectric X7R	100	nF
C_{OUT}	Output regulator capacitor	Tantalum or electrolitic dielectric	1	µF
		Ceramic dielectric X7R	100	nF

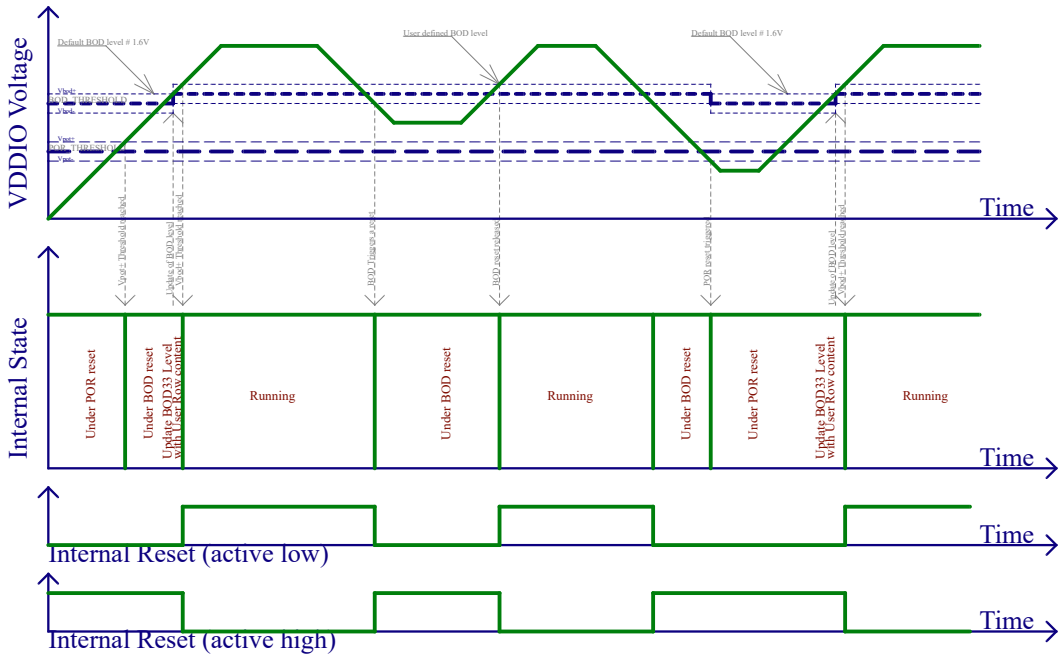
46.11.2 Power-On Reset (POR) Characteristics

Table 46-19. POR33 Characteristics

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
V_{POT+}	Voltage threshold Level on V_{DDIO} rising	-	1.52	1.57	1.63	V
V_{POT-}	Voltage threshold Level on V_{DDIO} falling		0.6	1.04	1.39	V

Downloaded from Arrow.com.

Figure 46-1. BOD Reset Behavior at Startup and Default Levels



46.11.3 Brown-Out Detectors (BOD) Characteristics

Table 46-20. BOD33 Characteristics (BOD33.VREFSEL = 0)

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
V_{BOD+}(2)	BOD33 high threshold level	BOD33.LEVEL = 6	1.66	1.68	1.70	V
		BOD33.LEVEL = 7	1.70	1.72	1.74	
		BOD33.LEVEL = 39	2.79	2.84	2.89	
		BOD33.LEVEL = 48	3.12	3.18	3.20	
V_{BOD-} / V_{BOD}(2)	BOD33 low threshold level	BOD33.LEVEL = 6	1.61	1.64	1.65	V
		BOD33.LEVEL = 7	1.65	1.67	1.68	
		BOD33.LEVEL = 39	2.74	2.78	2.80	
		BOD33.LEVEL = 48	3.04	3.09	3.11	
-	Step size	-	-	34	-	mV
V_{HYS}	Hysteresis (V_{BOD+} - V_{BOD-})	BOD33.LEVEL = 0x0 to 0x3F	40	-	180	mV
T_{START}(1)	Startup time	time from enable to RDY	-	3.2	-	µs

Downloaded from Arrow.com.

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. With BOD33.VREFSEL = 0 and no hysteresis configured, BOD levels can be given as: VBOD+ = VBOD- = 1.43 + BOD33.LEVEL * Step_size

Table 46-21. BOD33 Characteristics (BOD33.VREFSEL = 1)

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
V_{BOD+}(2)	BOD33 high threshold level	BOD33.LEVEL = 17	1.62	1.70	1.79	V
		BOD33.LEVEL = 18	1.65	1.73	1.81	
		BOD33.LEVEL = 59	2.86	2.96	3.08	
		BOD33.LEVEL = 63	2.97	3.08	3.20	
V_{BOD-} / V_{BOD}(2)	BOD33 low threshold level	BOD33.LEVEL = 17	1.59	1.65	1.72	V
		BOD33.LEVEL = 18	1.62	1.68	1.75	
		BOD33.LEVEL = 59	2.73	2.83	2.94	
		BOD33.LEVEL = 63	2.83	2.94	3.06	
-	Step size	-	-	28	-	mV
V_{HYS}	Hysteresis (V_{BOD+} - V_{BOD-})	BOD33.LEVEL = 0x0 to 0x3F	30	-	150	mV
T_{START}(1)	Startup time	time from enable to RDY	-	3.2	-	µs

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. With BOD33.VREFSEL = 1 and no hysteresis configured, BOD levels can be given as: Vbod+ = Vbod- = 1.17 + BOD33.LEVEL * Step_size

Table 46-22. BOD33 Power Consumption

Symbol	Parameters	Conditions	Ta	Min.	Typ.	Max	Units
IDD	IDLE, mode CONT	VCC = 1.8V	Max. at 85°C Typ. at 25°C	-	17.4	20.9	µA
		VCC = 3.3V		-	28.5	34.7	
	IDLE, mode SAMPL	VCC = 1.8V		-	0.02	0.09	
		VCC = 3.3V		-	0.04	0.10	

Downloaded from Arrow.com.

Symbol	Parameters	Conditions	Ta	Min.	Typ.	Max	Units
	STDBY, mode SAMPL	VCC = 1.8V		-	0.11	0.12	
		VCC = 3.3V		-	0.23	0.28	

46.11.4 Analog-to-Digital Converter (ADC) Characteristics

Table 46-23. Operating Conditions

Symbol	Parameters	Conditions	Min	Typ	Max	Unit
Res	Resolution		-	-	12	bits
Rs	Sampling rate		10	-	1000	kSPS
	Differential mode Number of ADC clock cycles SAMPCTRL.OFFCOMP=1	resolution 12 bit (RESEL=0)	-	16	-	cycles
		resolution 10 bit (RESEL=2)		14		
		resolution 8 bit (RESEL=3)		12		
	Differential mode Number of ADC clock cycles SAMPCTRL.OFFCOMP=0 SAMPLEN corresponds to the decimal value of SAMPLEN[5:0] register	resolution 12 bit (RESEL=0)	-	SAMPLEN +13	-	cycles
		resolution 10 bit (RESEL=2)		SAMPLEN +11		
		resolution 8 bit (RESEL=3)		SAMPLEN +9		
	Single-ended mode Number of ADC clock cycles SAMPCTRL.OFFCOMP=1	resolution 12 bit (RESEL=0)	-	16	-	cycles
		resolution 10 bit (RESEL=2)		15		
		resolution 8 bit (RESEL=3)		13		
	Single-ended mode Number of ADC clock cycles SAMPCTRL.OFFCOMP=0 SAMPLEN corresponds to the decimal value of SAMPLEN[5:0] register	resolution 12 bit (RESEL=0)	-	SAMPLEN +13	-	cycles
		resolution 10 bit (RESEL=2)		SAMPLEN +12		
		resolution 8 bit (RESEL=3)		SAMPLEN +10		
fadc	ADC Clock frequency		160	-	16000	kHz
Ts	Sampling time	SAMPCTRL.OFFCOMP=1	250	-	25000	ns
		SAMPCTRL.OFFCOMP=0	76	-	7692	
	Sampling time with DAC as input (MUXPOS = 0x1C)	SAMPCTRL.OFFCOMP=1	3000	-	25000	
		SAMPCTRL.OFFCOMP=0	3000	-	7692	
-	Conversion range	Diff mode	-Vref	-	Vref	V
-	Conversion range	Single-ended mode	0	-	Vref	
Vref	Reference input		1		VDDANA-0.6	V

Downloaded from Arrow.com.

Symbol	Parameters	Conditions	Min	Typ	Max	Unit
Vin	Input channel range	-	0	-	VDDANA	V
Vcmin	Input common mode voltage	For Vref > 1.0V	0.7	-	Vref-0.7	V
		For Vref=1.0V	0.3	-	Vref-0.3	V
CSAMPLE(1)	Input sampling capacitance		-	2.8	3.2	pF
RSAMPLE(1)	Input sampling on-resistance		-	-	1715	Ω
Rref(1)	Reference input source resistance	REFCOMP = 1	-	-	5	kΩ

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.

Table 46-24. Differential Mode (1)

Symbol	Parameters	Conditions		Measurements			Unit
				Min	Typ	Max	
ENOB	Effective Number of bits	Fadc = 1Msps	Vref=2.0V Vddana=3.0V	9.1	10.2	10.8	bits
			Vref=1.0V Vddana=1.6V to 3.6V	9.0	10.1	10.6	
			Vref=Vddana=1.6V to 3.6V	8.9	9.9	11.0	
			Bandgap Reference, Vddana=1.6V to 3.6V	9.0	9.8	10.6	
TUE	Total Unadjusted Error	without offset and gain compensation	Vref=Vddana=1.6V to 3.6V	-	7	32	LSB
INL	Integral Non Linearity	without offset and gain compensation	Vref=Vddana=1.6V to 3.6V	-	+/-1.9	+/-4	
DNL	Differential Non Linearity	without offset and gain compensation	Vref=Vddana=1.6V to 3.6V	-	+0.94/-1	+1.85/-1	
Gain	Gain Error	without gain compensation	Vref=1V Vddana=1.6V to 3.6V	-	+/-0.38	+/-1.9	%
			Vref=3V Vddana=1.6V to 3.6V	-	+/-0.14	+/-0.9	
			Bandgap Reference	-	+/-0.64	+/-5.4	
			Vref=Vddana=1.6V to 3.6V	-	+/-0.15	+/-0.9	
Offset	Offset Error	without offset compensation	Vref=1V Vddana=1.6V to 3.6V	-	+/-0.13	+/-15.8	mV

Downloaded from Arrow.com.

Symbol	Parameters	Conditions		Measurements			Unit
				Min	Typ	Max	
			Vref=3V Vddana=1.6V to 3.6V	-	+/-1.82	+/-14.9	
			Bandgap Reference	-	+/-2.07	+/-15.8	
			Vref=Vddana=1.6V to 3.6V	-	+/-1.82	+/-15.3	
SFDR	Spurious Free Dynamic Range	Fs = 1MHz / Fin = 13 kHz / Full range Input signal	Vref=2.0V Vddana=3.0V	58.1	70.5	77.5	dB
SINAD	Signal to Noise and Distortion ratio			56.7	63.4	66.5	
SNR	Signal to Noise ratio			56.5	64.4	67.1	
THD	Total Harmonic Distortion			-74.7	-68.7	-57.7	
-	Noise RMS	External Reference voltage		-	0.42	-	mV

Note:

1. These are given without any ADC oversampling and decimation features enabled.

Table 46-25. Single-Ended Mode (1)

Symbol	Parameters	Conditions		Measurements			Unit
				Min	Typ	Max	
ENOB	Effective Number of bits	Fadc = 1Msps	Vref=2.0V Vddana=3.0V	8.0	9.3	9.7	bits
			Vref=1.0V Vddana=1.6V to 3.6V	7.9	8.2	9.4	
			Vref=Vddana=1.6V to 3.6V	8.6	9.2	9.9	
			Bandgap Reference, Vddana=1.6V to 3.6V	7.8	8.4	8.9	
TUE	Total Unadjusted Error	without offset and gain compensation	Vref=2.0V Vddana=3.0V	-	12	63	LSB
INL	Integral Non Linearity	without offset and gain	Vref=2.0V Vddana=3.0V	-	+/-3.4	+/-8.9	

Downloaded from Arrow.com.

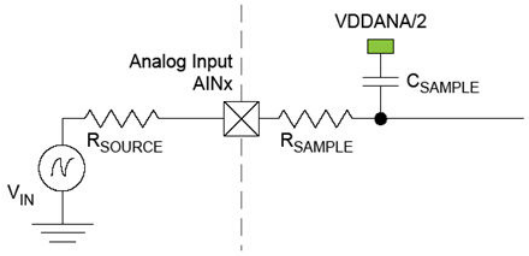
Symbol	Parameters	Conditions		Measurements			Unit
				Min	Typ	Max	
		compensation					
DNL	Differential Non Linearity	without offset and gain compensation	Vref=2.0V Vddana=3.0V	-	+0.9/-1	+1.8/-1	
Gain	Gain Error	without gain compensation	Vref=1V Vddana=1.6V to 3.6V	-	+/-0.3	+/-5.1	%
			Vref=3V Vddana=1.6V to 3.6V	-	+/-0.3	+/-5.1	
			Bandgap Reference	-	+/-0.4	+/-5.1	
			Vref=Vddana=1.6V to 3.6V	-	+/-0.2	+/-0.8	
Offset	Offset Error	without offset compensation	Vref=1V Vddana=1.6V to 3.6V	-	+/-2.6	+/-45	mV
			Vref=3V Vddana=1.6V to 3.6V	-	+/-2.6	+/-45	
			Bandgap Reference	-	+/-1.3	+/-34	
			Vref=Vddana=1.6V to 3.6V	-	+/-1.8	+/-37	
SFDR	Spurious Free Dynamic Range	Fs = 1MHz / Fin = 13 kHz / Full range Input signal	Vref=2.0V Vddana=3.0V	56.1	63.8	72.6	dB
SINAD	Signal to Noise and Distortion ratio			50.0	57.7	60.1	
SNR	Signal to Noise ratio			51.9	58.3	59.8	
THD	Total Harmonic Distortion			-72.5	-62.4	-52.3	
	Noise RMS	External Reference voltage		-	0.80	-	mV

Note:

1. These are given without any ADC oversampling and decimation features enabled.

Downloaded from Arrow.com.

Figure 46-2. ADC Analog Input AINx



The minimum sampling time $t_{samplehold}$ for a given R_{source} can be found using this formula:

$$t_{samplehold} \geq \left(R_{sample} + R_{source}\right) \times C_{sample} \times \left(n + 2\right) \times \ln\left(2\right)$$

For 12-bit accuracy:

$$t_{samplehold} \geq \left(R_{sample} + R_{source}\right) \times C_{sample} \times 9.7$$

where $t_{samplehold} \geq \frac{1}{2 \times f_{ADC}}$.

46.11.5 Digital-to-Analog Converter (DAC) Characteristics

Table 46-26. Operating Conditions(1)

Symbol	Parameters	Conditions	Min	Typ	Max	Unit
AVREF	External reference voltage		1	-	VDDANA-0.6	V
	Internal reference voltage 1		-	1	-	V
	Internal reference voltage 2		-	VDDANA	-	V
	Linear output voltage range		0.05	-	VDDANA-0.05	V
	Minimum resistive load		5	-	-	kOhm
	Maximum capacitance load		-	-	100	pF
IDD	DC supply current(2)	Voltage pump disabled	-	175	247	µA

Note:

1. The values in this table are based on specifications otherwise noted.
2. These values are based on characterization. These values are not covered in test limits in production.

Table 46-27. Clock and Timing

Parameter	Conditions		Min.	Typ.	Max.	Units
Conversion rate	Cload=100pF Rload > 5kOhm	Normal mode			350	ksps
		For DDATA=+/-1			1000	
Startup time	VDDANA > 2.6V	VDDANA > 2.6V	-	-	2.85	µs
	VDDANA < 2.6V	VDDANA < 2.6V	-	-	10	µs

Note: These values are based on simulation. These values are not covered by test limits in production or characterization.

Downloaded from Arrow.com.

Table 46-28. Accuracy Characteristics

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Units
RES	Input resolution			-	-	10	Bits
INL	Integral non-linearity	VREF= Ext 1.0V	VDD = 1.62V	±0.2	±0.5	±1.4	LSB
			VDD = 3.63V	±0.2	±0.4	± 1.2	
		VREF = VDDANA	VDD = 1.62V	±0.2	±0.6	± 2.1	
			VDD = 3.63V	± 0.2	±0.5	± 1.9	
		VREF= INT1V	VDD = 1.62V	± 0.4	±0.7	± 3.5	
			VDD = 3.63V	± 0.4	±0.8	± 6	
DNL	Differential non-linearity	VREF= Ext 1.0V	VDD = 1.62V	± 0.1	±0.3	± 1.5	LSB
			VDD = 3.63V	± 0.1	±0.3	± 1.2	
		VREF= VDDANA	VDD = 1.62V	± 0.1	±0.2	± 1.7	
			VDD = 3.63V	± 0.1	±0.2	± 1.5	
		VREF= INT1V	VDD = 1.62V	± 0.3	±0.6	± 3	
			VDD = 3.63V	± 0.3	±0.8	± 7	
Gain	Gain error	VREF= Ext 1.0V		-	±4	± 16	mV
		VREF= VDDANA		-	±12	± 60	mV
		VREF= INT1V		-	±1	± 22	mV
Offset	Offset error	VREF= Ext 1.0V		-	±1	± 13	mV
		VREF= VDDANA		-	±2.5	± 21	mV
		VREF= INT1V		-	±1.5	± 20	mV

46.11.6 Analog Comparator (AC) Characteristics

Table 46-29. Electrical and Timing

Symbol	Parameters	Conditions	Min.	Typ	Max.	Unit
PNIVR	Positive and Negative input range voltage		0	-	V_{DDANA}	V
ICMR	Input common mode range		0	-	V_{DDANA}-0.1	V
Off	Offset	COMPCTRLn.SPEED=0x0	-70	-4.5/+1.5	70	mV
		COMPCTRLn.SPEED=0x1	-55	-4.5/+1.5	55	
		COMPCTRLn.SPEED=0x2	-48	-4.5/+1.5	48	
		COMPCTRLn.SPEED=0x3	-42	-4.5/+1.5	42	
V_{Hys}	Hysteresis	COMPCTRLn.HYST=0x0	10	45	74	mV
		COMPCTRLn.HYST=0x1	22	70	106	

Downloaded from Arrow.com.

Symbol	Parameters	Conditions	Min.	Typ	Max.	Unit
		COMPCTRLn.HYST=0x2	37	90	129	
		COMPCTRLn.HYST=0x3	49	105	150	
T_{pd}	Propagation Delay $V_{cm}=V_{DDANA}/2$, V_{in}= +-100 mV overdrive from V_{cm} (see note 2)	COMPCTRLn.SPEED=0x0	-	4	12.3	µs
		COMPCTRLn.SPEED=0x1	-	0.97	2.6	
		COMPCTRLn.SPEED=0x2	-	0.56	1.4	
		COMPCTRLn.SPEED=0x3	-	0.33	0.77	
T_{start}	Start-up time	COMPCTRLn.SPEED=0x0	-	17	71	µs
		COMPCTRLn.SPEED=0x1	-	0.85	4.5(1)	
		COMPCTRLn.SPEED=0x2	-	0.55	3.2(1)	
		COMPCTRLn.SPEED=0x3	-	0.45	2.7(1)	
V_{scale}	INL		-	0.4	-	LSB
	DNL		-	0.1	-	
	Offset Error		-	0.1	-	
	Gain Error		-	1.3	-	

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. Vcm: Common Mode Voltage.

Table 46-30. Power Consumption

Symbol	Parameters	Conditions	Ta	Min.	Typ	Max.	Unit
I_{DDANA}	Current consumption voltage scaler disabled.	COMPCTRLn.SPEED=0x0, V_{DDANA}=3.3V	Max.85°C Typ.25°C	-	51	118	nA
		COMPCTRLn.SPEED=0x1, V_{DDANA}=3.3V		-	233	481	
		COMPCTRLn.SPEED=0x2, V_{DDANA}=3.3V		-	456	885	
		COMPCTRLn.SPEED=0x3, V_{DDANA}=3.3V		-	879	1604	
	Current consumption voltage scaler only	V_{DDANA}=3.3V		-	13	18	µA

Downloaded from Arrow.com.

46.11.7 DETREF Characteristics

Table 46-31. Reference Voltage Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
ADC/DAC Ref	ADC/DAC internal reference	nom. 1.0V, V_{CC}=3.0V, T= 25°C	0.976	1.0	1.022	V
		nom. 1.1V, V_{CC}=3.0V, T= 25°C	1.077	1.1	1.127	
		nom. 1.2V, V_{CC}=3.0V, T= 25°C	1.174	1.2	1.234	
		nom. 1.25V, V_{CC}=3.0V, T= 25°C	1.221	1.25	1.287	
		nom. 2.0V, V_{CC}=3.0V, T= 25°C	1.945	2.0	2.030	
		nom. 2.2V, V_{CC}=3.0V, T= 25°C	2.143	2.2	2.242	
		nom. 2.4V, V_{CC}=3.0V, T= 25°C	2.335	2.4	2.457	
		nom. 2.5V, V_{CC}=3.0V, T= 25°C	2.428	2.5	2.563	
	Ref Temperature coefficient	drift over [-40, +25]°C	-	-0.01/+0.015	-	%/°C
		drift over [+25, +85]°C	-	-0.01/+0.005	-	
	Ref Supply coefficient	drift over [1.6, 3.63]V	-	+/-0.35	-	%/V
AC Ref	AC Ref Accuracy	V_{CC}=3.0V, T=25°C	1.086	1.1	1.128	V
	Ref Temperature coefficient	drift over [-40, +25]°C	-	+/-0.01	-	%/°C
		drift over [+25, +85]°C	-	-0.005/+0.001	-	%/°C
	Ref Supply coefficient	drift over [1.6, 3.63]V	-	-0.35/+0.35	-	%/V

46.11.8 OPAMP Characteristics

Table 46-32. Operating Conditions

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
V_{CC}	Power Supply	All power modes	1.6	3	3.63	V
V_{in}	Input voltage range		0	-	Vcc	V
V_{out}	Output voltage range		0.15	-	Vcc-0.15	V
C_{load}	Maximum capacitance load		-	-	50	pF
R_{load} (1)	Minimum resistive load	Output Range[0.15V;Vcc-0.15V]	3.5	-	-	kΩ
		Output Range[0.3V;Vcc-0.3V]	0.5	-	-	
I_{load}(1)	DC output current load	Output Range[0.15V;Vcc-0.15V]	-	-	1	mA
		Output Range[0.3V;Vcc-0.3V]	-	-	6.9	

Note: 1. These values are based on simulation. They are not covered by production test limits or characterization.

Downloaded from Arrow.com.

Table 46-33. Power Consumption(1)

Symbol	Parameters	Conditions	Ta	Min.	Typ.	Max.	Unit
IDD	DC supply current (Voltage Doubler OFF)	Mode 3,VCC =3.3V	Max 85°C Typ 25°C	-	235	400	µA
		Mode 2,VCC =3.3V		-	94	166	
		Mode 1,VCC =3.3V		-	26	47	
		Mode 0 ,VCC =3.3V		-	7	13	
	Voltage Doubler consumption	VCC =3.3V		-	0.70	1.4	

Note: 1. These values are based on simulation. They are not covered by production test limits or characterization.

Table 46-34. Static Characteristics in 1X Gain(1)

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
G0	Open loop gain	Mode 3	-	114.5	-	dB
		Mode 2	-	117.6	-	
		Mode 1	-	116.8	-	
		Mode 0	-	108.5	-	
GBW	Gain Bandwidth	Mode 3	-	7.1	-	MHz
		Mode 2	-	2.8	-	
		Mode 1	-	0.85	-	
		Mode 0	-	0.2	-	
Φ_m	Phase margin	Mode 3	-	71.5	-	deg
		Mode 2	-	64	-	
		Mode 1	-	56	-	
		Mode 0	-	52	-	
T_{r1}	Response Time at 240µV (X1 gain)	Mode 3	-	1.3	-	µs
		Mode 2	-	3.3	-	
		Mode 1	-	13	-	
		Mode 0	-	52	-	
ΔT_{r1}	Response Time Variation for 10mV	Mode 3	-	100	-	ns
T_{start}	Start-up time (Enable to Ready), (Voltage Doubler OFF)	Mode 3	-	2.7	-	µs
		Mode 2	-	6.35	-	
		Mode 1	-	21.5	-	
		Mode 0	-	88.5	-	
Oe	Input Offset Voltage		-	-	+-3.5	mV
SR	Slew rate	Mode 3	-	- 2.8/2.6	-	V/µs
		Mode 2	-	-1.2/1.1	-	

Downloaded from Arrow.com.

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
		Mode 1	-	-0.3/0.3	-	
		Mode 0	-	-0.09/0.07	-	
CMRR	1X gain	Mode 3	-	83	-	dB
		Mode 2	-	84	-	
		Mode 1	-	84	-	
		Mode 0	-	83	-	
PSRR	1X gain	Mode 3	-	76	-	dB
		Mode 2	-	76	-	
		Mode 1	-	76	-	
		Mode 0	-	75	-	
-	Integrated Noise, BW=[0.1Hz-10kHz], x1 gain - V_{OUT}=1V	Mode 3	-	7.9	-	μV_{RMS}
		Mode 2	-	8.3	-	
		Mode 1	-	9.9	-	
		Mode 0	-	12.7	-	
-	Integrated Noise, BW=[0.1Hz-1MHz], x1 gain - V_{OUT}=1V	Mode 3	-	18.2	-	μV_{RMS}
		Mode 2	-	22.8	-	
		Mode 1	-	36.7	-	
		Mode 0	-	44.4	-	

Note: 1.These values are based on simulation. They are not covered by production test limits or characterization.

Table 46-35. PGA Electrical Characteristics(1)

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
-	Gain accuracy	16X Gain	-	-	+/-2.4	%
		4X Gain	-	-	+/-1.1	
		1X Gain	-	-	+/-2.6	
THD	Total Harmonic Distortion @ 10kHz - mode 3	16X Gain	-	-77	-	dB
		4X Gain	-	-72.8	-	
		1X Gain	-	-82.6	-	
-	Integrated Noise, BW=[0.1Hz-10 kHz], 16X gain - V_{OUT}=1V	Mode 3	-	147	-	µVrms
		Mode 2	-	147	-	
		Mode 1	-	162	-	
		Mode 0	-	191	-	
-	Integrated Noise, BW=[0.1Hz-1MHz], 16X gain - V_{OUT}=1V	Mode 3	-	262	-	µVrms
		Mode 2	-	247	-	
		Mode 1	-	235	-	
		Mode 0	-	235	-	

Downloaded from Arrow.com.

Note: 1.These values are based on simulation. They are not covered by production test limits or characterization.

46.11.9 Peripheral Touch Controller (PTC) Characteristics

Table 46-36. Sensor Load Capacitance

Symbol	Mode	PTC channel	Min PCB External (1)	Max Sensor Load (2)	Units
Cload	Self-capacitance	Y0	5	54	pF
		Y1	-		
		Y2	-	50	
		Y3	5	54	
		Y4	5		
		Y5	-		
		Y6	5		
		Y7	5		
		Y8	5		
		Y9	5		
		Y10	-	45	
		Y11	5	54	
		Y12	5		
		Y13	-		
		Y14	-		
		Y15	5	50	
		Y16	5	54	
		Y17	5		
		Y18	5		
		Y19	5		
	Mutual-capacitance	All	-	31	

Note:

1. Minimum external capacitance must be added on PCB design per PTC channel. Ensure that the PCB and sensor design add enough parasitic capacitance on each PTC channel, otherwise, an external capacitor must be added.
2. Capacitance load, the PTC circuitry, can compensate for each channel.

Downloaded from Arrow.com.

Table 46-37. Analog Gain Settings (1)

Symbol	Setting	Average
Gain	GAIN_1	1.0
	GAIN_2	2.0
	GAIN_4	3.9
	GAIN_8	8.1

Note:

1. Analog Gain is a parameter of the QTouch Library. Refer to the QTouch Library Peripheral Touch Controller User Guide.

Power Consumption

The values in the Power Consumption table below are measured values of power consumption under the following conditions:

Operating Conditions

- VDD = 3.3V

Clocks

- OSC16M divided to 4MHz used as main clock source
- CPU is running on flash with 0 wait states, at 4MHz
- PTC running at 4MHz
- Voltage Regulator mode: LPEFF enabled

PTC Configuration

- Mutual-capacitance mode
- One touch channel

System Configuration

- Standby sleep mode enabled
- RTC running on OSCULP32K: used to define the PTC scan rate, through the event system
- Drift Calibration disabled: no interrupts, PTC scans are performed in standby mode
- Drift Calibration enabled: RTC interrupts (wakeup) the CPU to perform PTC scans. PTC drift calibration is performed every 1.5 sec.

Table 46-38. Power Consumption (1)

Symbol	Parameters	Drift Calibration	PTC scan rate (msec)	Oversamples	Ta	Typ.	Max.	Units
IDD	Current Consumption	Disabled	10	4	Max. 85°C Typ. 25°C	6.2	49.2	µA
				16		12.7	58.1	
			50	4		2.3	43.7	
				16		3.7	45.5	

Downloaded from Arrow.com.

Symbol	Parameters	Drift Calibration	PTC scan rate (msec)	Oversamples	Ta	Typ.	Max.	Units
			100	4		1.7	43.2	
				16		2.4	43.9	
			200	4		1.4	42.8	
				16		1.8	43.2	
		Enabled	10	4		8.3	51.7	
				16		14.2	60.5	
			50	4		3.0	44.8	
				16		4.8	47.0	
			100	4		2.3	44.5	
				16		2.8	45.4	
			200	4		1.9	43.9	
				16		2.4	44.2	

Note:

1. These are based on characterization.

46.12 NVM Characteristics

Table 46-39. NVM Max Speed Characteristics (1)

	Conditions	CPU Fmax (MHz)		
		0WS	1WS	2WS
PL0 (-40/85°C) (-40/125°C)	V_{DDIO}>1.62 V	6	8	8
	V_{DDIO}>2.7 V	7.5	8	8
PL2 (-40/85°C) (-40/125°C)	V_{DDIO}>1.62 V	14	28	32
	V_{DDIO}>2.7 V	14	32	32

Table 46-40. NVM Timing Characteristics (1)

Symbol	Timings	Max	Units
t_{FPP}	Page Write	2.5	ms
t_{FRE}	Row erase	6	

Note:

Downloaded from Arrow.com.

1. For this Flash technology, a maximum number of 8 consecutive writes is allowed per row. Once this number is reached, a row erase is mandatory.

Table 46-41. Flash Erase and Programming Current

Symbol	Parameter	Typ.	Units
IDD_{NVM}	Maximum current (peak) during whole programming or erase operation	10	mA

Table 46-42. NVM Reliability Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Units
Ret_{NVM25k}	Retention after up to 25k	Average ambient 55°C	10	50	Years
$Ret_{NVM2.5k}$	Retention after up to 2.5k	Average ambient 55°C	20	100	Years
Ret_{NVM100}	Retention after up to 100	Average ambient 55°C	25	>100	Years
Cyc_{NVM}	Cycling Endurance(1)	-40°C < Ta < 85°C -40°C < Ta < 125°C	25K	100K	Cycles
	Cycling Endurance using Tamper Erase(1)		50	100	Cycles

Note: 1. An endurance cycle is a write and an erase operation.

46.13 Oscillators Characteristics

46.13.1 Crystal Oscillator (XOSC) Characteristics

Digital Clock Characteristics

The following table describes the characteristics for the oscillator when a digital clock is applied on XIN.

Table 46-43. Digital Clock Characteristics

Symbol	Parameter	Min.	Typ.	Max.	Units
F_{XIN}	XIN clock frequency	-	-	24	MHz
DC_{XIN}(1)	XIN clock duty cycle	40	50	60	%

Note:

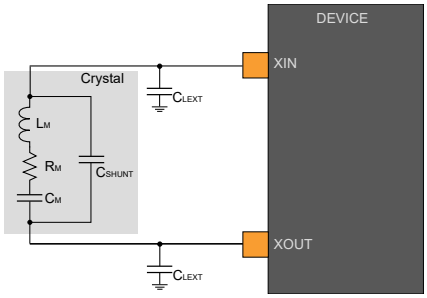
1. These values are based on simulation. They are not covered by production test limits or characterization.

Crystal Oscillator Characteristics

The following Table describes the characteristics for the oscillator when a crystal is connected between XIN and XOUT.

Downloaded from Arrow.com.

Figure 46-3. Oscillator Connection



The user must choose a crystal oscillator where the crystal load capacitance C_L is within the range given in the Table. The exact value of C_L can be found in the crystal data sheet. The capacitance of the external capacitors (C_{LEXT}) can then be computed as follows:

$C_{LEXT} = 2(C_L - C_{PARA} - C_{PCB} - C_{SHUNT})$

Where:

- C_{PARA} is the internal load capacitor parasitic between XIN and XOUT ($C_{PARA} = (C_{XIN} * C_{XOUT})/(C_{XIN} + C_{XOUT})$)
- C_{PCB} is the capacitance of the PCB
- C_{SHUNT} is the shunt capacity of the crystal.

Table 46-44. Multi Crystal Oscillator Electrical Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max	Units
Fout	Crystal oscillator frequency		0.4	-	32	MHz
ESR(2)	Crystal Equivalent Series Resistance - SF = 3	F = 0,4MHz - CL=100 pF XOSC,GAIN=0	-	-	5.6K	Ω
		F = 2MHz - CL=20 pF XOSC,GAIN=0, Cshunt=3.3pF	-	-	330	
		F = 4MHz - CL=20 pF XOSC,GAIN=1, Cshunt=2.5pF	-	-	240	
		F = 8MHz - CL=20 pF XOSC,GAIN=2, Cshunt=5.5pF	-	-	105	
		F = 16MHz - CL=20 pF XOSC,GAIN=3, Cshunt=4pF	-	-	60	
		F = 32MHz - CL=20 pF XOSC,GAIN=4, Cshunt=3.9pF	-	-	55	
Cxin(2)	Parasitic load capacitor		-	6.7	-	pF
Cxout(2)			-	4.2	-	pF
Tstart(2)	Startup time	F = 2MHz - CL=20 pF XOSC,GAIN=0, Cshunt=3.3pF	-	15.6K	81.6K	Cycles
		F = 4MHz - CL=20 pF XOSC,GAIN=1, Cshunt=2.5pF	-	6.3K	25.2K	

Downloaded from Arrow.com.

Symbol	Parameter	Conditions	Min.	Typ.	Max	Units
		F = 8MHz - CL=20 pF XOSC,GAIN=2, Cshunt=5.5pF	-	6.2K	27.2K	
		F = 16MHz - CL=20 pF XOSC,GAIN=3, Cshunt=4pF	-	7.7K	27.3K	
		F = 32MHz - CL=20 pF XOSC,GAIN=4, Cshunt=3.9pF	-	6.0K	21K	
CL(1)	Crystal load capacitance		10	-	20	pF
Pon(1)	Drive Level	AMPGC=ON	-	-	100	uW

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. These are based on characterization.

Table 46-45. Power Consumption

Symbol	Parameter	Conditions		Ta	Min.	Typ.	Max.	Units
I_{DD}	Current consumption	F=2MHz - CL=20pF XOSC,GAIN=0, VCC=3.3V	AMPGC=OFF	Max 85°C Typ 25°C	-	66	85	µA
			AMPGC=ON		-	62	99	
		F=4MHz - CL=20pF XOSC,GAIN=1, VCC=3.3V	AMPGC=OFF		-	107	140	
			AMPGC=ON		-	70	101	
		F=8MHz - CL=20pF XOSC,GAIN=2, VCC=3.3V	AMPGC=OFF		-	200	261	
			AMPGC=ON		-	118	153	
		F=16MHz - CL=20pF XOSC,GAIN=3, VCC=3.3V	AMPGC=OFF		-	436	581	
			AMPGC=ON		-	247	329	
		F=32MHz - CL=20pF XOSC,GAIN=4, VCC=3.3V	AMPGC=OFF		-	1303	1902	
			AMPGC=ON		-	627	940	

46.13.2 External 32KHz Crystal Oscillator (XOSC32K) Characteristics

Digital Clock Characteristics

The following table describes the characteristics for the oscillator when a digital clock is applied on XIN32 pin.

Downloaded from Arrow.com.

Table 46-46. Digital Clock Characteristics(1)

Symbol	Parameter	Min.	Typ.	Max.	Units
$f_{CPXIN32}$	XIN32 clock frequency		32.768	1000	kHz
DC_{XIN}	XIN32 clock duty cycle	40	50	60	%

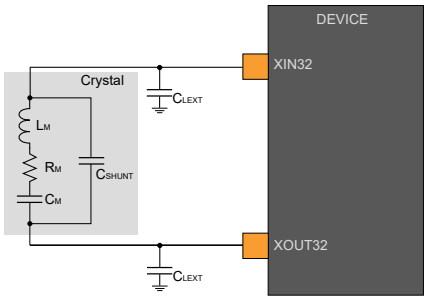
Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.

Crystal Oscillator Characteristics

The following section describes the characteristics for the oscillator when a crystal is connected between XIN32 and XOUT32 pins.

Figure 46-4. Oscillator Crystal Connection



The user must choose a crystal oscillator where the crystal load capacitance C_L is within the range given in the table. The exact value of C_L can be found in the crystal data sheet. The capacitance of the external capacitors (C_{LEXT}) can then be computed as follows:

$C_{LEXT} = 2(C_L - C_{PARA} - C_{PCB} - C_{SHUNT})$

Where:

- C_{PARA} is the internal load capacitor parasitic between XIN and XOUT ($C_{PARA} = (C_{XIN32K} C_{XOUT32K})/(C_{XIN32K} + C_{XOUT32K})$)
- C_{PCB} is the capacitance of the PCB
- C_{SHUNT} is the shunt capacity of the crystal.

Table 46-47. 32 KHz Crystal Oscillator Electrical Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
F_{OUT}	Crystal oscillator frequency	-	-	32.768	-	kHz
$CL^{(1)}$	Crystal load capacitance	-	7	-	9	pF
$C_{SHUNT}^{(1)}$	Crystal shunt capacitance	-	0.6	-	2	pF
$C_M^{(1)}$	Motional capacitance	-	0.6	-	3	fF
$ESR^{(2)}$	Crystal Equivalent Series Resistance - SF=3	f=32.768kHz, C_L= 9pF	-	-	70	kΩ
$C_{XIN32k}^{(2)}$	Parasitic load capacitor	-	-	3.2	-	pF
$C_{XOUT32k}^{(2)}$		-	-	3.4	-	

Downloaded from Arrow.com.

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
$t_{STARTUP}$(2)	Startup time	f=32.768kHz, C_L= 9pF	-	10	43	KCycles
P_{on}(1)	Drive Level	-	-	-	0.1	µW

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. These are based on characterization.

Table 46-48. Power Consumption

Symbol	Parameter	Conditions	Ta	Min.	Typ.	Max.	Units
I_{DD}	Current consumption	VCC=3.3V	Max 85°C Typ 25°C	-	309	606	nA

46.13.3 Ultra Low-Power Internal 32 kHz RC Oscillator (OSCULP32K) Characteristics

Table 46-49. Ultra Low-Power Internal 32 kHz RC Oscillator Electrical Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
F_{OUT}	Output frequency	at 25°C, at V_{DDIO}=3.3V	30.84	32.768	34.51	kHz
		at 25°C, over [1.62, 3.63]V	30.84	32.768	34.74	kHz
		over[-40,+85]°C, over [1.62, 3.63]V	25.17	32.768	39.10	kHz
Duty	Duty Cycle		-	50	-	%

46.13.4 16 MHz RC Oscillator (OSC16M) Characteristics

Table 46-50. Multi-RC Oscillator Electrical Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
F_{OUT}	Output frequency	V_{DD}=3.3V, T=25°C Calibrated against a 4/8/12/16 MHz reference	3.96	4.00	4.04	MHz
			7.92	8.00	8.08	
			11.88	12.00	12.12	
			15.84	16.00	16.16	
TempDrift	Freq vs. temperature drift	VDD=3.3V over temperature [-40°C-85°C], versus calibration reference at 25°C	-5	-	5	%
SupplyDrift	Freq vs. supply drift	Temperature =25°C over voltage [1.62V-3.63V], versus calibration reference at 3.3V	-1.5	-	1.5	
T_{WUP}(2)	Wake up time - 1st clock edge after enable	F_{OUT} = 4MHz	-	0.13	0.28	µs
		F_{OUT} = 8MHz	-	0.13	0.28	

Downloaded from Arrow.com.

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
		F_{OUT} = 12MHz	-	0.13	0.28	
		F_{OUT} = 16MHz	-	0.13	0.27	
$T_{STARTUP}$(2)	Startup time	F_{OUT} = 4MHz	-	1.16	2.96	µs
		F_{OUT} = 8MHz	-	1.29	2.74	
		F_{OUT} = 12MHz	-	1.34	2.95	
		F_{OUT} = 16MHz	-	1.39	3.11	
Duty(1)	Duty Cycle	-	45	50	55	%

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. These are based on characterization.

Table 46-51. Power Consumption

Symbol	Parameter	Conditions	Ta	Min.	Typ.	Max.	Units
I_{DD}	Current consumption	F_{out}=4MHz, V_{CC}=3.3V	Max.85°C Typ.25°C	-	73	139	µA
		F_{out}=8MHz, V_{CC}=3.3V		-	106	169	
		F_{out}=12MHz, V_{CC}=3.3V		-	135	195	
		F_{out}=16MHz, V_{CC}=3.3V		-	166	225	

46.13.5 Digital Frequency Locked Loop (DFLLULP) Characteristics

Table 46-52. Digital Frequency Locked Loop Characteristics(2)

Symbol	Parameter		Min	Typ	Max	Unit
FIN	Input Clock Frequency		32	-	33	kHz
FOUT	Output Clock Frequency	PL2	-	32	-	MHz
		PL0	-	8	-	
Jp	Period jitter	PL0, Fin= 32 kHz 50 ppm, Fout = 8MHz	-4	-	4	%
		PL2, Fin= 32 kHz 50 ppm, Fout = 32 MHz	-4.3	-	4.3	
tLOCK	Lock Time	After startup, time to get lock signal Fin = 32768 Hz, Fout = 8MHz, PL0 Binary Search mode enabled	-	362	-	µs
		After startup, time to get lock signal Fin = 32768 Hz, Fout = 32 MHz, PL2	-	362	-	µs

Downloaded from Arrow.com.

Symbol	Parameter		Min	Typ	Max	Unit
		Binary Search mode enabled				
Duty	Duty cycle(1)		40	50	60	%

Note:

1. These values are based on simulation. These values are not covered by test or characterization.
2. These characteristics are only applicable in LDO regulator mode.

Table 46-53. Power Consumption(1)

Symbol	Parameters	Conditions	Ta	Min	Typ.	Max	Units
IDD	Current Consumption	Fout=8MHz (PL0) - VCC=3.3V	Max 85°C Typ 25°C	-	33	93	µA
		Fout=32MHz (PL2) - VCC=3.3V		-	144	223	

Note: 1. These characteristics are only applicable in LDO regulator mode.

46.13.6 Digital Phase Lock Loop (DPLL) Characteristics

Table 46-54. Fractional Digital Phase Lock Loop Characteristics(2)

Symbol	Parameter		Min.	Typ.	Max.	Unit
FIN	Input Clock Frequency		32	-	2000	kHz
FOUT	Output Clock Frequency	PL2	32	-	96	MHz
		PL0	32	-	48	MHz
Jp	Period jitter	PL0, Fin = 32 kHz, Fout = 32 MHz	-	3	6	%
		PL2, Fin = 32 kHz, Fout = 32 MHz		2	5	
		PL0, Fin = 32 kHz, Fout = 48 MHz	-	3	4	
		PL2, Fin = 32 kHz, Fout = 48 MHz		2	6	
		PL2, Fin = 32 kHz, Fout = 96 MHz	-	3	4	
		PL0, Fin = 32 kHz, Fout = 32 MHz	-	3	5	
		PL2, Fin = 32 kHz, Fout = 32 MHz		3	6	
		PL0, Fin = 2 MHz, Fout = 48 MHz	-	5	7	
		PL2, Fin = 2 MHz, Fout = 48 MHz		3	6	
		PL2, Fin = 2 MHz, Fout = 96 MHz	-	4	10	
tLOCK	Lock Time	After startup, time to get lock signal Fin = 32 kHz, Fout = 96 MHz	-	1.1	1.5	ms
		After startup, time to get lock signal Fin = 2 MHz, Fout = 96 MHz	-	24	35	µs
Duty	Duty cycle(1)		40	50	60	%

Downloaded from Arrow.com.

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. These characteristics are applicable only in LDO regulator mode and with a XOSC or XOSC32K reference.

Table 46-55. Power Consumption(1)

Symbol	Parameter	Conditions	TA	Min.	Typ.	Max.	Units
I_{DD}	Current Consumption	Fout = 48 MHz (PL0) - VDD=3.3V	Max. 85°C Typ. 25°C	-	339	432	µA
		Fout = 96 MHz (PL2) - VDD=3.3V		-	678	777	

Note: 1. These characteristics are only applicable in LDO regulator mode.

46.14 Timing Characteristics

46.14.1 External Reset Pin

Table 46-56. External Reset Characteristics(1)

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
t_{EXT}	Minimum reset pulse width		1	-	-	µs

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.

46.14.2 SERCOM in SPI Mode in PL0

Table 46-57. SPI Timing Characteristics and Requirements (1)

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Units
tSCK	SCK period when tSOV=0 on the slave side	Master	Reception	2*(tMIS +tSLAVE_OUT) (3)	-	-	ns
		Master	Transmission	2*(tMOV +tSLAVE_IN) (4)	-	-	
tSCKW	SCK high/low width	Master		-	0,5*tSCK	-	
tSCKR	SCK rise time(2)	Master		-	0,25*tSCK	-	
tSCKF	SCK fall time(2)	Master		-	0,25*tSCK	-	
tMIS	MISO setup to SCK	Master, VDD>2,70V		90	-	-	
		Master, VDD>1,62V		98.1	-	-	
tMIH	MISO hold after SCK	Master, VDD>2,70V		0	-	-	

Downloaded from Arrow.com.

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Units
		Master, VDD>1,62V		0	-	-	
tMOV	MOSI output valid after SCK	Master, VDD>2,70V		-	-	34.5	ns
		Master, VDD>1,62V		-	-	38.6	
tMOH	MOSI hold after SCK	Master, VDD>2,70V		9.7	-	-	
		Master, VDD>1,62V		9.7	-	-	
tSSCK	Slave SCK Period when tMIS=0 on the master side	Slave	Reception	2*(tSIS +tMASTER_OUT) (5)	-	-	
		Slave	Transmission	2*(tSOV +tMASTER_IN) (6)	-	-	
tSSCKW	SCK high/low width	Slave		-	0,5*tSCK	-	
tSSCKR	SCK rise time(2)	Slave		-	0,25*tSCK	-	
tSSCKF	SCK fall time(2)	Slave		-	0,25*tSCK	-	
tSIS	MOSI setup to SCK	Slave, VDD>2,70V		25.6	-	-	ns
		Slave, VDD>1,62V		26.2	-	-	
tSIH	MOSI hold after SCK	Slave, VDD>2,70V		13.2	-	-	
		Slave, VDD>1,62V		13.9	-	-	
tSSS	SS setup to SCK	Slave	PRELOADEN=1	tSOSS+tEXT_MIS +2*tAPBC (8) (9)	-	-	
			PRELOADEN=0	tSOSS+tEXT_MIS (8)	-	-	
tSSH	SS hold after SCK	Slave		0.5*tSSCK	-	-	
tSOV	MISO output valid after SCK	Slave, VDD>2,70V		-	-	69	
		Slave, VDD>1,62V		-	-	78.4	
tSOH	MISO hold after SCK	Slave, VDD>2,70V		20.2	-	-	
		Slave, VDD>1,62V		20.2	-	-	
tSOSS	MISO setup after SS low	Slave, VDD>2,70V		-	-	1* tSCK	
		Slave, VDD>1,62V		-	-	1* tSCK	

Downloaded from Arrow.com.

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Units
tSOSH	MISO hold after SS high	Slave, VDD>2,70V		15	-	-	
		Slave, VDD>1,62V		15	-	-	

Note:

1. These values are based on simulation. These values are not covered by test limits in production.
2. See I/O Pin Characteristics.
3. Where tSLAVE_OUT is the slave external device output response time, generally tEXT_SOV +tLINE_DELAY. (7)
4. Where tSLAVE_IN is the slave external device input constraint, generally tEXT_SIS+tLINE_DELAY. (7)
5. Where tMASTER_OUT is the master external device output response time, generally tEXT_MOV +tLINE_DELAY. (7)
6. Where tMASTER_IN is the master external device input constraint, generally tEXT_MIS +tLINE_DELAY. (7)
7. tLINE_DELAY is the transmission line time delay.
8. tEXT_MIS is the input constraint for the master external device.
9. tAPBC is the APB period for SERCOM.

Figure 46-5. SPI Timing Requirements in Master Mode

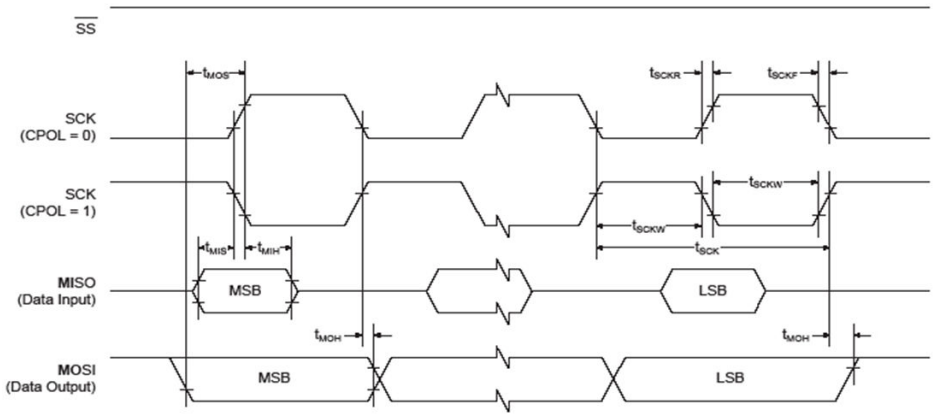
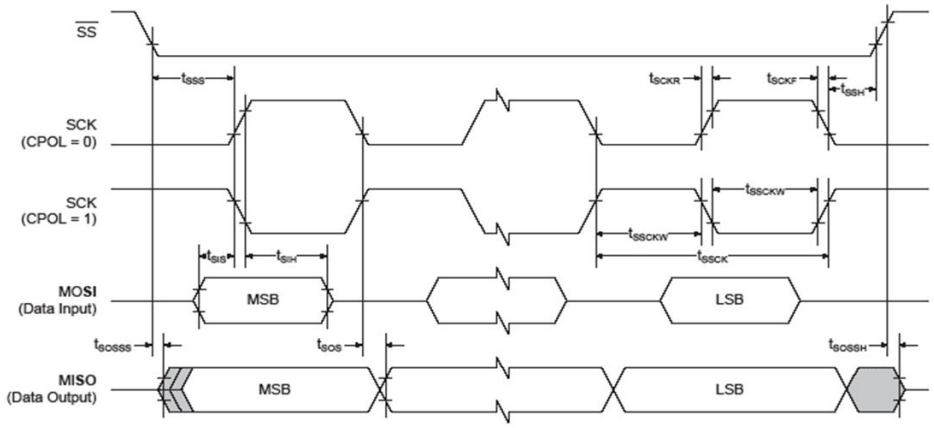


Figure 46-6. SPI Timing Requirements in Slave Mode



Maximum SPI Frequency

Downloaded from Arrow.com.

- Master Mode
  f_{SCKmax} = 1/2*(t_{MIS} + t_{valid}), where t_{valid} is the slave time response to output data after detecting an SCK edge. For a non-volatile memory with t_{valid} = 12 ns Max, $f_{SPCKMax}$ = 3.7 MHz @ VDDIO > 2.7V
- Slave Mode
  f_{SCKmax} = 1/2*(t_{SOV} + t_{su}), where t_{su} is the setup time from the master before sampling data. With a perfect master (t_{su}=0), $f_{SPCKMax}$ = 6 MHz @ VDDIO > 2.7V

46.14.3 SERCOM in SPI Mode in PL2

Table 46-58. SPI Timing Characteristics and Requirements (1)

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Units
tSCK	SCK period when tSOV=0 on the slave side	Master	Reception	2*(tMIS +tSLAVE_OUT) (3)	-	-	ns
		Master	Transmission	2*(tMOV +tSLAVE_IN) (4)	-	-	
tSCKW	SCK high/low width	Master		-	0,5*tSCK	-	
tSCKR	SCK rise time(2)	Master		-	0,25*tSCK	-	
tSCKF	SCK fall time(2)	Master		-	0,25*tSCK	-	
tMIS	MISO setup to SCK	Master, VDD>2,70V		42.5	-	-	
		Master, VDD>1,62V		52.5	-	-	
tMIH	MISO hold after SCK	Master, VDD>2,70V		0	-	-	
		Master, VDD>1,62V		0	-	-	
tMOV	MOSI output valid after SCK	Master, VDD>2,70V		-	-	17.1	
		Master, VDD>1,62V		-	-	21.2	
tMOH	MOSI hold after SCK	Master, VDD>2,70V		6.3	-	-	
		Master, VDD>1,62V		6.3	-	-	
tSSCK	Slave SCK Period when tMIS =0 on the master side	Slave	Reception	2*(tSIS +tMASTER_OUT) (5)	-	-	
		Slave	Transmission	2*(tSOV +tMASTER_IN) (6)	-	-	
tSSCKW	SCK high/low width	Slave		-	0,5*tSCK	-	ns
tSSCKR	SCK rise time(2)	Slave		-	0,25*tSCK	-	
tSSCKF	SCK fall time(2)	Slave		-	0,25*tSCK	-	

Downloaded from Arrow.com.

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Units
tSIS	MOSI setup to SCK	Slave, VDD>2,70V		10.3	-	-	
		Slave, VDD>1,62V		11.1	-	-	
tSIH	MOSI hold after SCK	Slave, VDD>2,70V		6.1	-	-	
		Slave, VDD>1,62V		6.9	-	-	
tSSS	SS setup to SCK	Slave	PRELOADEN=1	tSOSS+tEXT_MIS +2*tAPBC (8) (9)	-	-	
			PRELOADEN=0	tSOSS+tEXT_MIS (8)	-	-	ns
tSSH	SS hold after SCK	Slave		0.5*tSSCK	-	-	
tSOV	MISO output valid after SCK	Slave, VDD>2,70V		-	-	35	
		Slave, VDD>1,62V		-	-	44.8	
tSOH	MISO hold after SCK	Slave, VDD>2,70V		13.4	-	-	
		Slave, VDD>1,62V		13.4	-	-	
tSOSS	MISO setup after SS low	Slave, VDD>2,70V		-	-	1* tSCK	
		Slave, VDD>1,62V		-	-	1* tSCK	
tSOSH	MISO hold after SS high	Slave, VDD>2,70V		8.6	-	-	
		Slave, VDD>1,62V		8.6	-	-	

Note:

1. These values are based on simulation. These values are not covered by test limits in production.
2. See I/O Pin Characteristics.
3. Where tSLAVE_OUT is the slave external device output response time, generally tEXT_SOV +tLINE_DELAY. (7)
4. Where tSLAVE_IN is the slave external device input constraint, generally tEXT_SIS+tLINE_DELAY. (7)
5. Where tMASTER_OUT is the master external device output response time, generally tEXT_MOV +tLINE_DELAY. (7)
6. Where tMASTER_IN is the master external device input constraint, generally tEXT_MIS +tLINE_DELAY. (7)
7. tLINE_DELAY is the transmission line time delay.
8. tEXT_MIS is the input constraint for the master external device.
9. tAPBC is the APB period for SERCOM.

Downloaded from Arrow.com.

Figure 46-7. SPI Timing Requirements in Master Mode

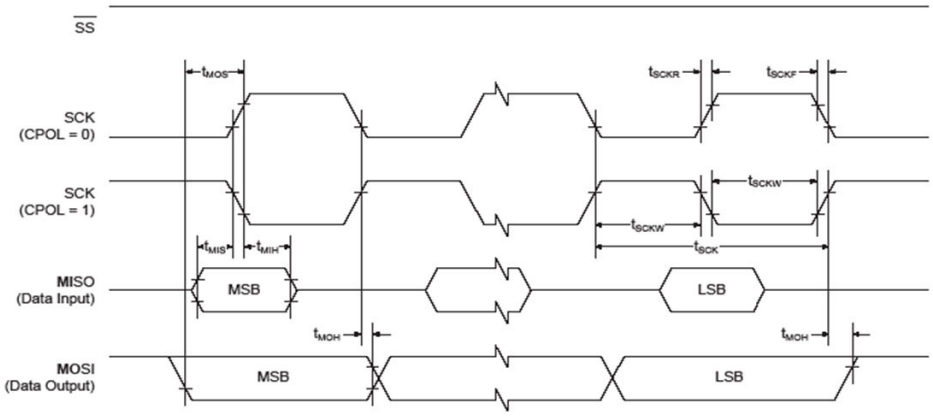
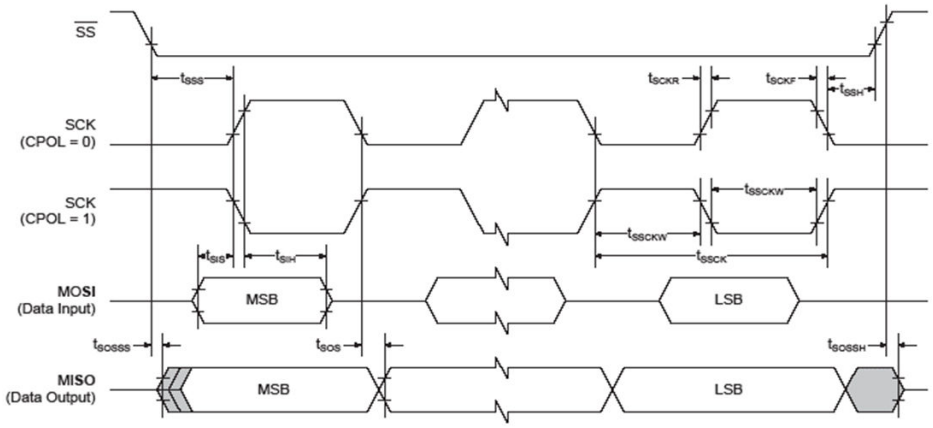


Figure 46-8. SPI Timing Requirements in Slave Mode



Maximum SPI Frequency

- Master Mode
  f_{SCKmax} = 1/2*(t_{MIS} + t_{valid}), where t_{valid} is the slave time response to output data after detecting an SCK edge. For a non-volatile memory with t_{valid} = 12ns Max, $f_{SPCKMax}$ = 9.8 MHz @ VDDIO > 2.7V
- Slave Mode
  f_{SCKmax} = 1/2*(t_{SOV} + t_{su}), where t_{su} is the setup time from the master before sampling data. With a perfect master (t_{su}=0), $f_{SPCKMax}$ = 16.3MHz @ VDDIO > 2.7V

Downloaded from Arrow.com.

47. 125°C Electrical Characteristics

This section provides an overview of the SAM L10 and SAM L11 electrical characteristics, which are specific for devices running up to 125ºC.

47.1 Disclaimer

All typical values are measured at T = 25°C unless otherwise specified. All minimum and maximum values are valid across operating temperature and voltage unless otherwise specified.

47.2 General Operating Ratings

The device must operate within the ratings listed in the following table for all other electrical characteristics and typical characteristics of the device to be valid.

Table 47-1. General Operating Conditions

Symbol	Description	Min.	Typ.	Max.	Units
V_{DDIO}	IO Supply Voltage	1.62	3.3	3.63	V
V_{DDANA}	Analog supply voltage	1.62	3.3	3.63	V
T_A	Temperature range	-40	25	125	°C
T_J	Junction temperature	-	-	145	°C

47.3 Power Consumption

The values in this section are measured values of power consumption under the following conditions, except where noted:

- **Operating Conditions**
  - V_{DDIO} = 3.3V or 1.8V
  - CPU is running on Flash with required Wait states, as recommended in the NVM Characteristics section.
  - Low power cache is enabled
  - BOD33 is disabled
  - I/Os are configured with digital input trigger disabled (default Reset configuration)
- **Oscillators**
  - XOSC (crystal oscillator) stopped
  - XOSC32K (32.768 kHz crystal oscillator) running with external 32.768 kHz crystal
  - When in active PL2 mode on FDPLL96M at 32 MHZ, DPLL is using XOSC32K as reference clock and running at 32 MHz
  - When in Active mode on DFLLULP, the DFLLULP is configured in Closed Loop mode using XOSC32K as reference clock and MCLK.CTRLA.CKSEL = 1

Downloaded from Arrow.com.

Table 47-2. Active Current Consumption

Mode	Conditions	Regulator	PL	CPU Clock	Vcc	Ta	Typ.	Max.	Units
ACTIVE	COREMARK / FIBONACCI	LDO	PL0	DFLLUP at 8 MHz	1.8V	Max at 125°C Typ at 25°C	64.1	129	uA/MHz
					3.3V		64.4	131	
				OSC 8 MHz	1.8V		66.6	130	
					3.3V		70.3	132	
				OSC 4 MHz	1.8V		74.1	203	
					3.3V		77.8	206	
			PL2	FDPLL96 at 32 MHz	1.8V		82.0	98	
					3.3V		82.5	99	
				DFLLULP at 32 MHz	1.8V		75.8	109	
					3.3V		75.8	107	
		BUCK	PL0	DFLLUP at 8 MHz	1.8V		40.0	84	
					3.3V		25.3	54	
				OSC 8 MHz	1.8V		43.8	84	
					3.3V		32.1	58	
				OSC 4 MHz	1.8V		50.3	131	
					3.3V		38.9	92	
			PL2	FDPLL96 at 32 MHz	1.8V		59.9	70	
					3.3V		35.3	43	
				DFLLULP at 32 MHz	1.8V		55.3	78	
					3.3V		32.6	46	
	WHILE1	LDO	PL0	DFLLUP at 8 MHz	1.8V		44.3	110	
					3.3V		44.4	111	
				OSC 8 MHz	1.8V		47.6	111	
					3.3V		50.1	113	
				OSC 4 MHz	1.8V		54.6	184	
					3.3V		57.7	187	
			PL2	FDPLL96 at 32 MHz	1.8V		56.9	79	
					3.3V		57.2	80	
				DFLLULP at 32 MHz	1.8V		50.8	72	
					3.3V		51.0	72	

Downloaded from Arrow.com.

Mode	Conditions	Regulator	PL	CPU Clock	Vcc	Ta	Typ.	Max.	Units
		BUCK	PL0	DFLLUP at 8 MHz	1.8V		28.1	72	
					3.3V		18.5	47	
				OSC 8 MHz	1.8V		32.2	73	
					3.3V		25.3	51	
				OSC 4 MHz	1.8V		38.4	121	
					3.3V		31.9	86	
			PL2	FDPLL96 at 32 MHz	1.8V		41.5	55	
					3.3V		24.6	34	
				DFLLULP at 32 MHz	1.8V		37.1	53	
					3.3V		22.0	32	
IDLE		LDO	PL0	DFLLUP at 8 MHz	1.8V		16.0	81	
					3.3V		16.2	82	
				OSC 8 MHz	1.8V		19.8	82	
					3.3V		22.0	85	
				OSC 4 MHz	1.8V		26.2	152	
					3.3V		29.2	157	
			PL2	FDPLL96 at 32 MHz	1.8V		20.3	54	
					3.3V		20.4	54	
				DFLLULP at 32 MHz	1.8V		14.3	32	
					3.3V		14.4	33	
		BUCK	PL0	DFLLUP at 8 MHz	1.8V		11.1	52	
					3.3V		8.3	35	
				OSC 8 MHz	1.8V		15.5	55	
					3.3V		15.2	40	
				OSC 4 MHz	1.8V		21.3	100	
					3.3V		21.6	73	
			PL2	FDPLL96 at 32 MHz	1.8V		14.9	30	
					3.3V		9.1	19	
				DFLLULP at 32 MHz	1.8V		10.6	24	
					3.3V		6.7	15	

Downloaded from Arrow.com.

Table 47-3. Standby and Off Mode Current Consumption

Mode	Conditions	Regulator Mode	Vcc	Ta	Typ.	Max.	Units
STANDBY	All 16kB RAM retained, PDSW domain in active state	LPEFF Disable	1,8V	25°C	1.3	3.5	µA
				125°C	121.7	304.8	
		LPEFF Enable	3,3V	25°C	1.1	3.0	
				125°C	74.5	282.6	
		BUCK in standby PL0 (VREG.RUNSTDBY=1 and VREG.STDBYPL0=1)	1,8V	25°C	1.2	2.9	
				125°C	78.0	188.7	
			3,3V	25°C	1.1	2.2	
				125°C	50.9	122.9	µA
	All 16kB RAM retained, PDSW domain in retention	LPEFF Disable	1,8V	25°C	0.6	1.1	
				125°C	27.1	81.0	
		LPEFF Enable	3,3V	25°C	0.5	1.0	
				125°C	23.1	52.8	
		BUCK in standby PL0 (VREG.RUNSTDBY=1 and VREG.STDBYPL0=1)	1,8V	25°C	0.8	1.1	
				125°C	23.0	53.7	
			3,3V	25°C	0.8	1.5	
				125°C	17.3	37.6	
	12 kB RAM retained,PDSW domain in retention	LPEFF Disable	1,8V	25°C	0.6	1.1	
				125°C	25.5	73.7	
		LPEFF Enable	3,3V	25°C	0.5	1.0	
				125°C	21.6	48.8	
		Buck in standby PL0 (VREG.RUNSTDBY=1 and VREG.STDBYPL0=1)	1,8V	25°C	0.7	1.1	
				125°C	21.5	50.5	
			3,3V	25°C	0.8	1.5	
				125°C	16.4	35.4	
	8kB RAM retained,PDSW domain in retention	LPEFF Disable	1,8V	25°C	0.5	1.0	µA
				125°C	23.8	67.1	
		LPEFF Enable	3,3V	25°C	0.5	0.9	
				125°C	20.2	45.4	
		BUCK in standby PL0 (VREG.RUNSTDBY=1 and VREG.STDBYPL0=1)	1,8V	25°C	0.7	1.0	
				125°C	19.9	46.5	
			3,3V	25°C	0.7	1.4	

Downloaded from Arrow.com.

Mode	Conditions	Regulator Mode	Vcc	Ta	Typ.	Max.	Units
				125°C	15.5	33.2	
	4kB RAM retained,PDSW domain in retention	LPEFF Disable	1,8V	25°C	0.5	0.9	
				125°C	22.0	58.9	
		LPEFF Enable	3,3V	25°C	0.5	0.9	
				125°C	18.7	41.5	
		BUCK in standby PL0 (VREG.RUNSTDBY=1 and VREG.STDBYPL0=1)	1,8V	25°C	0.7	1.0	
				125°C	18.4	42.7	µA
			3,3V	25°C	0.8	1.5	
				125°C	14.6	31.0	
	4kB RAM retained,PDSW domain in retention and RTC running on XOSC32k	LPEFF Disable	1,8V	25°C	0.9	1.3	
				125°C	22.6	59.8	
		LPEFF Enable	3,3V	25°C	0.8	1.2	
				125°C	19.3	42.1	
		BUCK in standby PL0 (VREG.RUNSTDBY=1 and VREG.STDBYPL0=1)	1,8V	25°C	1.0	1.3	
				125°C	19.0	43.3	
			3,3V	25°C	1.1	1.7	
				125°C	15.2	31.6	
OFF			1,8V	25°C	34.6	54.4	nA
				125°C	4385.0	8291.5	
			3,3V	25°C	61.2	89.1	
				125°C	5489.5	10564.7	

47.4 Analog Characteristics

47.4.1 Brown-Out Detectors (BOD) Characteristics

Table 47-4. BOD33 Characteristics with BOD33.VREFSEL = 0

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
VBOD+ (2)	BOD33 high threshold level	BOD33.LEVEL = 6	1.66	1.68	1.70	V
		BOD33.LEVEL = 7	1.70	1.72	1.74	
		BOD33.LEVEL = 39	2.79	2.84	2.89	

Downloaded from Arrow.com.

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
		BOD33.LEVEL = 48	3.11	3.18	3.20	
VBOD- / VBOD(2)	BOD33 low threshold level	BOD33.LEVEL = 6	1.61	1.64	1.65	
		BOD33.LEVEL = 7	1.65	1.67	1.68	
		BOD33.LEVEL = 39	2.74	2.78	2.80	
		BOD33.LEVEL = 48	3.04	3.09	3.11	
-	Step size			34		mV
VHys	Hysteresis (VBOD+ - VBOD-)	BOD33.LEVEL = 0x0 to 0x3F	30	-	180	mV
Tstart	Startup time(1)	time from enable to RDY	-	3.2	-	us

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. With BOD33.VREF_SEL = 0 and no hysteresis configured, BOD levels can be given as: VBOD+ = VBOD- = 1.43 + BOD Setting * Step_size

Table 47-5. BOD33 Characteristics with BOD33.VREFSEL = 1

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
VBOD+ (2)	-	BOD33.LEVEL = 17	1.62	1.70	1.79	V
		BOD33.LEVEL = 18	1.65	1.73	1.81	
		BOD33.LEVEL = 59	2.86	2.96	3.09	
		BOD33.LEVEL = 63	2.97	3.08	3.20	
VBOD- / VBOD(2)	-	BOD33.LEVEL = 17	1.59	1.65	1.72	V
		BOD33.LEVEL = 18	1.62	1.68	1.75	
		BOD33.LEVEL = 59	2.73	2.83	2.94	

Downloaded from Arrow.com.

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
		BOD33.LEVEL = 63	2.83	2.94	3.06	
-	Step size			28		mV
VHys	Hysteresis (VBOD + - VBOD-) BOD33.LEVEL = 0x0 to 0x3F		30	-	150	mV
Tstart	Startup time(1)	Time from enable to RDY	-	3.2	-	us

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. With BOD33.VREF_SEL = 0 and no hysteresis configured, BOD levels can be given as: Vbod+ = Vbod- = 1.17 + Bod setting * Step_size

Table 47-6. Power Consumption

Symbol	Parameters	Conditions	Ta	Min.	Typ.	Max.	Units
IDD	IDLE, Mode CONT	VCC = 1.8V	Max 125°C Typ 25°C	-	17.4	21.8	µA
		VCC = 3.3V		-	28.5	37.5	
	IDLE, Mode SAMPL	VCC = 1.8V		-	0.02	0.17	
		VCC = 3.3V		-	0.04	0.13	
	STANDBY, Mode SAMPL	VCC = 1.8V		-	0.11	0.17	
		VCC = 3.3V		-	0.23	0.29	

47.4.2 Analog to Digital (ADC) Characteristics

Table 47-7. Operating Conditions

Symbol	Parameters	Conditions	Min	Typ	Max	Unit
Res	Resolution		-	-	12	bits
Rs	Sampling rate		10	-	1000	kSPS
	Differential mode Number of ADC clock cycles SAMPCTRL.OFFCOMP=1	resolution 12 bit (RESEL=0)	-	16	-	cycles
		resolution 10 bit (RESEL=2)		14		
		resolution 8 bit (RESEL=3)		12		
	Differential mode	resolution 12 bit (RESEL=0)	-	SAMPLEN +13	-	cycles

Downloaded from Arrow.com.

Symbol	Parameters	Conditions	Min	Typ	Max	Unit
	Number of ADC clock cycles SAMPCTRL.OFFCOMP=0 SAMPLEN corresponds to the decimal value of SAMPLEN[5:0] register	resolution 10 bit (RESEL=2)		SAMPLEN +11		
		resolution 8 bit (RESEL=3)		SAMPLEN +9		
	Single-ended mode Number of ADC clock cycles SAMPCTRL.OFFCOMP=1	resolution 12 bit (RESEL=0)	-	16	-	cycles
		resolution 10 bit (RESEL=2)		15		
		resolution 8 bit (RESEL=3)		13		
	Single-ended mode Number of ADC clock cycles SAMPCTRL.OFFCOMP=0 SAMPLEN corresponds to the decimal value of SAMPLEN[5:0] register	resolution 12 bit (RESEL=0)	-	SAMPLEN +13	-	cycles
		resolution 10 bit (RESEL=2)		SAMPLEN +12		
		resolution 8 bit (RESEL=3)		SAMPLEN +10		
fadc	ADC Clock frequency		160	-	16000	kHz
Ts	Sampling time	SAMPCTRL.OFFCOMP=1	250	-	25000	ns
		SAMPCTRL.OFFCOMP=0	76	-	7692	
	Sampling time with DAC as input (MUXPOS = 0x1C)	SAMPCTRL.OFFCOMP=1	3000	-	25000	
		SAMPCTRL.OFFCOMP=0	3000	-	7692	
	Conversion range	Diff mode	-VREF	-	+VREF	V
	Conversion range	Single-ended mode	0	-	VREF	
Vref	Reference input		1	-	VDDANA-0.6	V
Vin	Input channel range	-	0	-	VDDANA	V
Vcmin	Input common mode voltage	For Vref > 1.0V	0.7	-	VREF-0.7	V
		For Vref=1.0V	0.3	-	VREF-0.3	V
CSAMPLE (1)	Input sampling capacitance		-	2.8	3.2	pF
RSAMPLE (1)	Input sampling on-resistance		-	-	1715	Ω
Rref (1)	Reference input source resistance	REFCOMP = 1	-	-	5	kΩ

Downloaded from Arrow.com.

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.

Table 47-8. Differential Mode (1)

Symbol	Parameters	Conditions		Measurements			Unit
				Min	Typ	Max	
ENOB	Effective Number of bits	Fadc = 1Msps	Vref=2.0V Vddana=3.0V	9.1	10.2	10.8	bits
			Vref=1.0V Vddana=1.6V to 3.6V	9.0	10.1	10.6	
			Vref=Vddana=1.6V to 3.6V	8.9	9.9	11.0	
			Bandgap Reference, Vddana=1.6V to 3.6V	9.0	9.8	10.6	
TUE	Total Unadjusted Error	without offset and gain compensation	Vref=Vddana=1.6V to 3.6V	-	7	32	LSB
INL	Integral Non Linearity	without offset and gain compensation	Vref=Vddana=1.6V to 3.6V	-	+/-1.9	+/-4.8	
DNL	Differential Non Linearity	without offset and gain compensation	Vref=Vddana=1.6V to 3.6V	-	+0.94/-1	+1.85/-1	
Gain	Gain Error	without gain compensation	Vref=1V Vddana=1.6V to 3.6V	-	+/-0.38	+/-1.9	%
			Vref=3V Vddana=1.6V to 3.6V	-	+/-0.14	+/-0.9	
			Bandgap Reference	-	+/-0.64	+/-5.4	

Downloaded from Arrow.com.

Symbol	Parameters	Conditions		Measurements			Unit
				Min	Typ	Max	
			Vref=Vddana=1.6V to 3.6V	-	+/-0.15	+/-0.9	
Offset	Offset Error	without offset compensation	Vref=1V Vddana=1.6V to 3.6V	-	+/-0.13	+/-15.8	mV
			Vref=3V Vddana=1.6V to 3.6V	-	+/-1.82	+/-14.9	
			Bandgap Reference	-	+/-2.07	+/-15.8	
			Vref=Vddana=1.6V to 3.6V	-	+/-1.82	+/-15.3	
SFDR	Spurious Free Dynamic Range	Fs = 1MHz / Fin = 13 kHz / Full range Input signal	Vref=2.0V Vddana=3.0V	58.1	70.5	77.5	dB
SINAD	Signal to Noise and Distortion ratio			56.7	63.4	66.5	
SNR	Signal to Noise ratio			56.5	64.4	67.1	
THD	Total Harmonic Distortion			-74.7	-68.7	-57.7	
	Noise RMS	External Reference voltage	External Reference voltage	-	0.42	-	mV

Note:

1. These are given without any ADC oversampling and decimation features enabled.

Downloaded from Arrow.com.

Table 47-9. Single Ended Mode (1)

Symbol	Parameters	Conditions		Measurements			Unit
				Min	Typ	Max	
ENOB	Effective Number of bits	Fadc = 1Msps	Vref=2.0V Vddana=3.0V	8.0	9.3	9.7	bits
			Vref=1.0V Vddana=1.6V to 3.6V	7.9	8.2	9.4	
			Vref=Vddana=1.6V to 3.6V	8.6	9.2	9.9	
			Bandgap Reference, Vddana=1.6V to 3.6V	7.8	8.4	8.9	
TUE	Total Unadjusted Error	without offset and gain compensation	Vref=2.0V Vddana=3.0V	-	12	66	LSB
INL	Integral Non Linearity	without offset and gain compensation	Vref=2.0V Vddana=3.0V	-	+/-3.4	+/-9.1	
DNL	Differential Non Linearity	without offset and gain compensation	Vref=2.0V Vddana=3.0V	-	+0.9/-1	+1.8/-1	
Gain	Gain Error	without gain compensation	Vref=1V Vddana=1.6V to 3.6V	-	+/-0.3	+/-5.1	%
			Vref=3V Vddana=1.6V to 3.6V	-	+/-0.3	+/-5.1	
			Bandgap Reference	-	+/-0.4	+/-5.1	
			Vref=Vddana=1.6V to 3.6V	-	+/-0.2	+/-0.8	

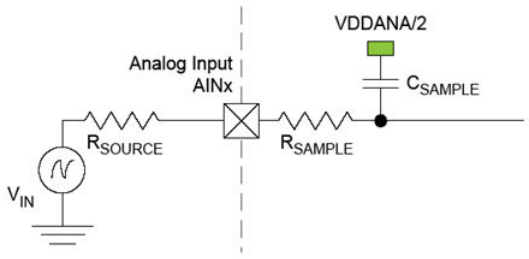
Downloaded from Arrow.com.

Symbol	Parameters	Conditions		Measurements			Unit
				Min	Typ	Max	
Offset	Offset Error	without offset compensation	Vref=1V Vddana=1.6V to 3.6V	-	+/-2.6	+/-48	mV
			Vref=3V Vddana=1.6V to 3.6V	-	+/-2.6	+/-48	
			Bandgap Reference	-	+/-1.3	+/-35	
			Vref=Vddana=1.6V to 3.6V	-	+/-1.8	+/-38	
SFDR	Spurious Free Dynamic Range	Fs = 1MHz / Fin = 13 kHz / Full range Input signal	Vref=2.0V Vddana=3.0V	56.1	63.8	72.6	dB
SINAD	Signal to Noise and Distortion ratio			50.0	57.7	60.1	
SNR	Signal to Noise ratio			51.9	58.3	59.8	
THD	Total Harmonic Distortion			-72.5	-62.4	-52.3	
	Noise RMS	External Reference voltage	External Reference voltage	-	0.80	-	mV

Note:

1. These are given without any ADC oversampling and decimation features enabled.

Figure 47-1. ADC Analog Input AINx



The minimum sampling time $t_{samplehold}$ for a given R_{source} can be found using this formula:

$$t_{samplehold} \geq \left(R_{sample} + R_{source}\right) \times \left(C_{sample}\right) \times \left(n + 2\right) \times \ln\left(2\right)$$

Downloaded from Arrow.com.

For 12-bit accuracy:

$t_{\text{samplehold}} \geq \left(R_{\text{sample}} + R_{\text{source}}\right) \times \left(C_{\text{sample}}\right) \times 9.7$

where $t_{\text{samplehold}} = \frac{1}{2 \times f_{\text{ADC}}}$.

47.4.3 Digital-to-Analog Converter (DAC) Characteristics

Table 47-10. Operating Conditions (1)

Symbol	Parameters	Conditions	Min	Typ	Max	Unit
AVREF	External reference voltage		1	-	VDDANA-0.6	V
	Internal reference voltage 1		-	1	-	V
	Internal reference voltage 2		-	VDDANA	-	V
	Linear output voltage range		0.05	-	VDDANA-0.05	V
	Minimum resistive load		5	-	-	kOhm
	Maximum capacitance load		-	-	100	pF
IDD	DC supply current(2)	Voltage pump disabled	-	175	270	µA

Note:

1. The values in this table are based on specifications otherwise noted.
2. These values are based on characterization. These values are not covered in test limits in production.

Table 47-11. Clock and Timing (1)

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Units
	Conversion rate	Cload=100pF Rload > 5 kOhm	Normal mode	-	-	350	ksps
			For DDATA=+/-1	-	-	1000	
	Startup time	VDDANA > 2.6V	VDDANA > 2.6V	-	-	2.85	µs
		VDDANA < 2.6V	VDDANA < 2.6V	-	-	10	µs

Note:

1. These values are based on simulation. These values are not covered by test limits in production or characterization.

Table 47-12. Accuracy Characteristics (1)

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Units
RES	Input resolution			-	-	10	Bits
INL	Integral non-linearity	VREF= Ext 1.0V	VDD = 1.62V	+/-0,2	+/-0,5	+/-1.4	LSB
			VDD = 3.63V	+/-0,2	+/-0,4	+/-1,2	
		VREF = VDDANA	VDD = 1.62V	+/-0,2	+/-0,6	+/-2.1	
			VDD = 3.63V	+/-0,2	+/-0,5	+/-1,9	

Downloaded from Arrow.com.

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Units
		VREF= INT1V	VDD = 1.62V	+/-0,4	+/-0,7	+/-4.2	
			VDD = 3.63V	+/-0,4	+/-0,8	+/-6	
DNL	Differential non-linearity	VREF= Ext 1.0V	VDD = 1.62V	+/-0,1	+/-0,3	+/-2	LSB
			VDD = 3.63V	+/-0,1	+/-0,3	+/-1.5	
		VREF= VDDANA	VDD = 1.62V	+/-0,1	+-0,2	+/-3.0	
			VDD = 3.63V	+/-0,1	+/-0,2	+/-1.6	
		VREF= INT1V	VDD = 1.62V	+/-0,3	+/-0,6	+/-4.3	
			VDD = 3.63V	+/-0,3	+/-0,8	+/-7	
	Gain error	VREF= Ext 1.0V		-	+/-4	+/-16	mV
		VREF= VDDANA		-	+/-12	+/-60	mV
		VREF= INT1V		-	+/-1	+/-23	mV
	Offset error	VREF= Ext 1.0V		-	+/-1	+/-13	mV
		VREF= VDDANA		-	+/-2.5	+/-32	mV
		VREF= INT1V		-	+/-1.5	+/-30	mV

Note:

1. All values measured using a conversion rate of 350ksps.

47.4.4 Analog Comparator Characteristics

Table 47-13. Electrical and Timing (1)

Symbol	Parameters	Conditions	Min.	Typ	Max.	Unit
PNIVR	Positive and Negative input range voltage		0	-	V_{DDANA}	V
ICMR	Input common mode range		0	-	V_{DDANA}-0.1	V
Off	Offset	COMPCTRLn.SPEED=0x0	-70	-4.5/+1.5	70	mV
		COMPCTRLn.SPEED=0x1	-55	-4.5/+1.5	55	
		COMPCTRLn.SPEED=0x2	-48	-4.5/+1.5	48	
		COMPCTRLn.SPEED=0x3	-42	-4.5/+1.5	42	
V_{Hys}	Hysteresis	COMPCTRLn.HYST=0x0	10	45	79	mV
		COMPCTRLn.HYST=0x1	22	70	115	
		COMPCTRLn.HYST=0x2	37	90	138	
		COMPCTRLn.HYST=0x3	49	105	159	

Downloaded from Arrow.com.

Symbol	Parameters	Conditions	Min.	Typ	Max.	Unit
T_{pd}	Propagation Delay $V_{cm}=V_{ddana}/2$, V_{in}= +-100mV overdrive from V_{CM}	COMPCTRLn.SPEED=0x0	-	4	12.3	µs
		COMPCTRLn.SPEED=0x1	-	0.97	2.6	
		COMPCTRLn.SPEED=0x2	-	0.56	1.4	
		COMPCTRLn.SPEED=0x3	-	0.33	0.77	
T_{start}	Start-up time	COMPCTRLn.SPEED=0x0	-	17	71	µs
		COMPCTRLn.SPEED=0x1	-	0.85	4.5(1)	
		COMPCTRLn.SPEED=0x2	-	0.55	3.2(1)	
		COMPCTRLn.SPEED=0x3	-	0.45	2.7(1)	
V_{scale}	INL		-	0.4	-	LSB
	DNL		-	0.1	-	
	Offset Error		-	0.1	-	
	Gain Error		-	1.3	-	

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.

Table 47-14. Power Consumption

Symbol	Parameters	Conditions	Ta	Min.	Typ	Max.	Unit
I_{DDANA}	Current consumption $V_{CM}=V_{DDANA}/2$, +/-100mV overdrive from V_{CM}, Voltage scaler disabled	COMPCTRLn.SPEED=0x0, V_{DDANA}=3.3V	Max.125°C Typ.25°C	-	51	232	nA
		COMPCTRLn.SPEED=0x1, V_{DDANA}=3.3V		-	233	604	
		COMPCTRLn.SPEED=0x2, V_{DDANA}=3.3V		-	456	1009	
		COMPCTRLn.SPEED=0x3, V_{DDANA}=3.3V		-	879	1756	
	Current consumption Voltage Scaler only	V_{DDANA}=3.3V		-	13	19	µA

47.4.5 DETREF Characteristics

Table 47-15. Reference Voltage Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
ADC/DAC Ref	ADC/DAC internal reference	nom. 1.0V, V_{CC}=3.0V, T= 25°C	0.976	1.0	1.022	V
		nom. 1.1V, V_{CC}=3.0V, T= 25°C	1.077	1.1	1.127	
		nom. 1.2V, V_{CC}=3.0V, T= 25°C	1.174	1.2	1.234	

Downloaded from Arrow.com.

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
		nom. 1.25V, V_{CC}=3.0V, T= 25°C	1.221	1.25	1.287	
		nom. 2.0V, V_{CC}=3.0V, T= 25°C	1.945	2.0	2.030	
		nom. 2.2V, V_{CC}=3.0V, T= 25°C	2.143	2.2	2.242	
		nom. 2.4V, V_{CC}=3.0V, T= 25°C	2.335	2.4	2.457	
		nom. 2.5V, V_{CC}=3.0V, T= 25°C	2.428	2.5	2.563	
	Ref Temperature coefficient	drift over [-40, +25]°C	-	-0.01/+0.015	-	%/°C
		drift over [+25, +125]°C	-	-0.006/+0.003	-	
	Ref Supply coefficient	drift over [1.6, 3.63]V	-	+/-0.35	-	%/V
AC Ref	AC Ref Accuracy	V_{CC}=3.0V, T=25°C	1.086	1.1	1.128	V
	Ref Temperature coefficient	drift over [-40, +25]°C	-	+/-0.01	-	%/°C
		drift over [+25, +125]°C	-	-0.005/+0.001	-	%/°C
	Ref Supply coefficient	drift over [1.6, 3.63]V	-	-0.35/+0.35	-	%/V

47.4.6 OPAMP Characteristics

Table 47-16. Power Consumption(1)

Symbol	Parameters	Conditions	Ta	Min.	Typ.	Max.	Unit
IDD	DC supply current (Voltage Doubler OFF)	Mode 3,VCC =3.3V	Max 125°C Typ 25°C	-	235	415	µA
		Mode 2,VCC =3.3V		-	94	173	
		Mode 1,VCC =3.3V		-	26	50	
		Mode 0 ,VCC =3.3V		-	7	14	
	Voltage Doubler consumption	VCC =3.3V		-	0.70	1.5	

Note: 1. These values are based on simulation. They are not covered by production test limits or characterization.

47.4.7 Peripheral Touch Controller (PTC) Characteristics

Power Consumption

The values in the Power Consumption table below are measured values of power consumption under the following conditions:

Operating Conditions

- VDD = 3.3V

Clocks

- OSC16M divided to 4MHz used as main clock source

Downloaded from Arrow.com.

- CPU is running on Flash with 0 wait states, at 4MHz
- PTC running at 4MHz
- Voltage Regulator mode: LPEFF enabled

PTC Configuration

- Mutual-Capacitance mode
- One touch channel

System Configuration

- Standby Sleep mode enabled
- RTC running on OSCULP32K: used to define the PTC scan rate, through the event system
- Drift Calibration disabled: no interrupts, PTC scans are performed in Standby mode
- Drift Calibration enabled: RTC interrupts (wake-up) the CPU to perform PTC scans. PTC drift calibration is performed every 1.5 sec.

Table 47-17. Power Consumption (1)

Symbol	Parameters	Drift Calibration	PTC scan rate (msec)	Oversamples	Ta	Typ.	Max.	Units
IDD	Current Consumption	Disabled	10	4	Max 125°C Typ 25°C	6.2	292.0	µA
				16		12.7	300.5	
			50	4		2.3	286.1	
				16		3.7	290.3	
			100	4		1.7	286.1	
				16		2.4	286.2	
			200	4		1.4	285.5	
				16		1.8	286.2	
		Enabled	10	4		8.3	293.9	
				16		14.2	304.9	
			50	4		3.0	289.2	
				16		4.8	290.5	
			100	4		2.3	289.2	
				16		2.8	289.5	
			200	4		1.9	287.9	
				16		2.4	289.0	

Note:

1. These are based on characterization.

Downloaded from Arrow.com.

47.5 Oscillators Characteristics

47.5.1 Crystal Oscillator (XOSC) Characteristics

Table 47-18. Power Consumption

Symbol	Parameter	Conditions		Ta	Min.	Typ.	Max.	Units
I_{DD}	Current consumption	F=2MHz - CL=20pF XOSC,GAIN=0, VCC=3.3V	AMPGC=OFF	Max 125°C Typ 25°C	-	66	106	µA
			AMPGC=ON		-	62	107	
		F=4MHz - CL=20pF XOSC,GAIN=1, VCC=3.3V	AMPGC=OFF		-	107	164	
			AMPGC=ON		-	70	132	
		F=8MHz - CL=20pF XOSC,GAIN=2, VCC=3.3V	AMPGC=OFF		-	200	307	
			AMPGC=ON		-	118	180	
		F=16MHz - CL=20pF XOSC,GAIN=3, VCC=3.3V	AMPGC=OFF		-	436	630	
			AMPGC=ON		-	247	382	
		F=32MHz - CL=20pF XOSC,GAIN=4, VCC=3.3V	AMPGC=OFF		-	1303	2251	
			AMPGC=ON		-	627	1116	

47.5.2 External 32KHz Crystal Oscillator (XOSC32K) Characteristics

Table 47-19. Power Consumption

Symbol	Parameter	Conditions	Ta	Min.	Typ.	Max.	Units
I_{DD}	Current consumption	VCC=3.3V	Max 125°C Typ 25°C	-	309	767	nA

47.5.3 Ultra Low-Power Internal 32 kHz RC Oscillator (OSCULP32K) Characteristics

Table 47-20. Ultra Low-Power Internal 32 kHz RC Oscillator Electrical Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
F_{OUT}	Output frequency	at 25°C, at V_{DDIO}=3.3V	30.84	32.768	34.51	kHz
		at 25°C, over [1.62, 3.63]V	30.84	32.768	34.74	kHz
		over[-40,+125]°C, over [1.62, 3.63]V	25.17	32.768	41.76	kHz
Duty	Duty Cycle		-	50	-	%

47.5.4 16 MHz RC Oscillator (OSC16M) Characteristics

Table 47-21. Multi-RC Oscillator Electrical Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
F_{OUT}	Output frequency	V_{DD}=3.3V, T=25°C	3.96	4.00	4.04	MHz

Downloaded from Arrow.com.

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
		Calibrated against a 4/8/12/16 MHz reference	7.92	8.00	8.08	
			11.88	12.00	12.12	
			15.84	16.00	16.16	
TempDrift	Freq vs. temperature drift	VDD=3.3V over temperature [-40°C-125°C], versus calibration reference at 25°C	-5		5	%
SupplyDrift	Freq vs. supply drift	Temperature =25°C over voltage [1.62V-3.63V], versus calibration reference at 3.3V	-1.5		1.5	
T_{WUP}(2)	Wake up time - 1st clock edge after enable	F_{OUT} = 4MHz	-	0.13	0.32	µs
		F_{OUT} = 8MHz	-	0.13	0.31	
		F_{OUT} = 12MHz	-	0.13	0.31	
		F_{OUT} = 16MHz	-	0.13	0.31	
$T_{STARTUP}$(2)	Startup time	F_{OUT} = 4MHz	-	1.16	2.96	µs
		F_{OUT} = 8MHz	-	1.29	2.74	
		F_{OUT} = 12MHz	-	1.34	2.95	
		F_{OUT} = 16MHz	-	1.39	3.11	
Duty(1)	Duty Cycle	-	45	50	55	%

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. These values are based on characterization. These values are not covered in test limits in production.

Table 47-22. Power Consumption

Symbol	Parameter	Conditions	Ta	Min.	Typ.	Max.	Units
I_{DD}	Current consumption	F_{out}=4MHz, V_{CC}=3.3V	Max.125°C Typ.25°C	-	73	370	µA
		F_{out}=8MHz, V_{CC}=3.3V		-	106	400	
		F_{out}=12MHz, V_{CC}=3.3V		-	135	425	
		F_{out}=16MHz, V_{CC}=3.3V		-	166	455	

Downloaded from Arrow.com.

47.5.5 Digital Frequency Locked Loop (DFLLULP) Characteristics

Table 47-23. Power Consumption(1)

Symbol	Parameters	Conditions	Ta	Min.	Typ.	Max	Units
IDD	Current Consumption	Fout = 8 MHz (PL0) - VCC = 3.3V	Max 125°C Typ 25°C	-	33	284	µA
		Fout = 32 MHz (PL2) - VCC=3.3V		-	144	458	

Note: These characteristics are only applicable in LDO Regulator mode

47.5.6 Digital Phase Lock Loop (DPLL) Characteristics

Table 47-24. Fractional Digital Phase Lock Loop(2)

Symbol	Parameter		Min.	Typ.	Max.	Unit
FIN	Input Clock Frequency		32	-	2000	kHz
FOUT	Output Clock Frequency	PL2	32	-	96	MHz
		PL0	32	-	48	MHz
Jp	Period jitter	PL0, Fin = 32 kHz, Fout = 32 MHz	-	3	6	%
		PL2, = 32 kHz, Fout = 32 MHz	-	2	6	
		PL0, Fin = 32 kHz, Fout = 48 MHz	-	3	4	
		PL2, Fin = 32 kHz, Fout = 48 MHz	-	2	6	
		PL2, Fin = 32 kHz, Fout = 96 MHz	-	3	4	
		PL0, Fin = 32 kHz, Fout = 32 MHz	-	3	5	
		PL2, Fin = 32 kHz, Fout = 32 MHz	-	3	6	
		PL0, Fin = 2 MHz, Fout = 48 MHz	-	5	7	
		PL2, Fin = 2 MHz, Fout = 48 MHz	-	3	6	
		PL2, Fin= 2 MHz, Fout= 96 MHz	-	4	10	
tLOCK	Lock Time	After startup, time to get lock signal Fin = 32 kHz, Fout = 96 MHz	-	1.1	1.5	ms
		After startup, time to get lock signal Fin = 2 MHz, Fout = 96 MHz	-	24	35	µs
Duty	Duty cycle (1)		40	50	60	%

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. These characteristics are applicable only in LDO Regulator mode and with a XOSC or XOSC32K reference.

Downloaded from Arrow.com.

Table 47-25. Power Consumption(1)

Symbol	Parameter	Conditions	TA	Min.	Typ.	Max.	Units
I_{DD}	Current Consumption	Fout = 48 MHz (PL0) - VDD = 3.3V	Max. 125°C Typ. 25°C	-	339	618	µA
		Fout = 96 MHz (PL2) - VDD = 3.3V		-	678	1005	

Note: These characteristics are only applicable in LDO regulator mode.

47.6 Timing Characteristics

47.6.1 SERCOM in SPI Mode in PL0

Table 47-26. SPI Timing Characteristics and Requirements (1)

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Units
tSCK	SCK period when tSOV=0 on the slave side	Master	Reception	2*(tMIS +tSLAVE_OUT) (3)	-	-	ns
		Master	Transmission	2*(tMOV +tSLAVE_IN) (4)	-	-	
tSCKW	SCK high/low width	Master		-	0,5*tSCK	-	
tSCKR	SCK rise time(2)	Master		-	0,25*tSCK	-	
tSCKF	SCK fall time(2)	Master		-	0,25*tSCK	-	
tMIS	MISO setup to SCK	Master, VDD>2,70V		86	-	-	
		Master, VDD>1,62V		95	-	-	
tMIH	MISO hold after SCK	Master, VDD>2,70V		0	-	-	
		Master, VDD>1,62V		0	-	-	
tMOV	MOSI output valid after SCK	Master, VDD>2,70V		-	-	33.3	ns
		Master, VDD>1,62V		-	-	49.6	
tMOH	MOSI hold after SCK	Master, VDD>2,70V		9.7	-	-	
tMOH	MOSI hold after SCK	Master, VDD>1,62V		9.7	-	-	
tSSCK	Slave SCK Period when tMIS=0 on the master side	Slave	Reception	2*(tSIS +tMASTER_OUT) (5)	-	-	
		Slave	Transmission	2*(tSOV +tMASTER_IN) (6)	-	-	
tSSCKW	SCK high/low width	Slave		-	0,5*tSCK	-	

Downloaded from Arrow.com.

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Units
tSSCKR	SCK rise time(2)	Slave		-	0,25*tSCK	-	
tSSCKF	SCK fall time(2)	Slave		-	0,25*tSCK	-	
tSIS	MOSI setup to SCK	Slave, VDD>2,70V		24.2	-	-	
		Slave, VDD>1,62V		24.9	-	-	
tSIH	MOSI hold after SCK	Slave, VDD>2,70V		12.9	-	-	
		Slave, VDD>1,62V		13.5	-	-	
tSSS	SS setup to SCK	Slave	PRELOADEN=1	tSOSS+tEXT_MIS +2*tAPBC (8) (9)	-	-	ns
			PRELOADEN=0	tSOSS+tEXT_MIS (8)	-	-	
tSSH	SS hold after SCK	Slave		0.5*tSSCK	-	-	
tSOV	MISO output valid after SCK	Slave, VDD>2,70V		-	-	66.9	
		Slave, VDD>1,62V		-	-	76.6	
tSOH	MISO hold after SCK	Slave, VDD>2,70V		22.7	-	-	
		Slave, VDD>1,62V		20.3	-	-	
tSOSS	MISO setup after SS low	Slave, VDD>2,70V		-	-	1* tSCK	
		Slave, VDD>1,62V		-	-	1* tSCK	
tSOSH	MISO hold after SS high	Slave, VDD>2,70V		15	-	-	
		Slave, VDD>1,62V		15	-	-	

Note:

1. These values are based on simulation. These values are not covered by test limits in production.
2. See I/O Pin Characteristics.
3. Where tSLAVE_OUT is the slave external device output response time, generally tEXT_SOV +tLINE_DELAY (See Note 7).
4. Where tSLAVE_IN is the slave external device input constraint, generally tEXT_SIS+tLINE_DELAY (See Note 7).
5. Where tMASTER_OUT is the master external device output response time, generally tEXT_MOV +tLINE_DELAY (See Note 7).
6. Where tMASTER_IN is the master external device input constraint, generally tEXT_MIS +tLINE_DELAY (See Note 7).
7. tLINE_DELAY is the transmission line time delay.
8. tEXT_MIS is the input constraint for the master external device.
9. tAPBC is the APB period for SERCOM.

Downloaded from Arrow.com.

Figure 47-2. SPI Timing Requirements in Master Mode

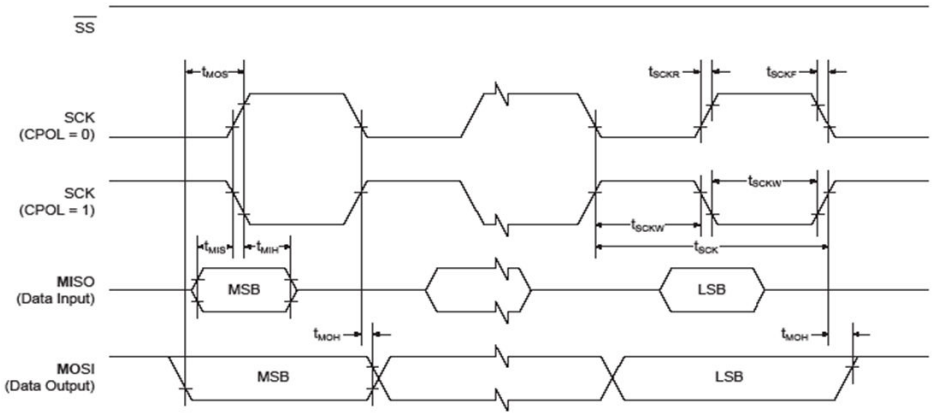
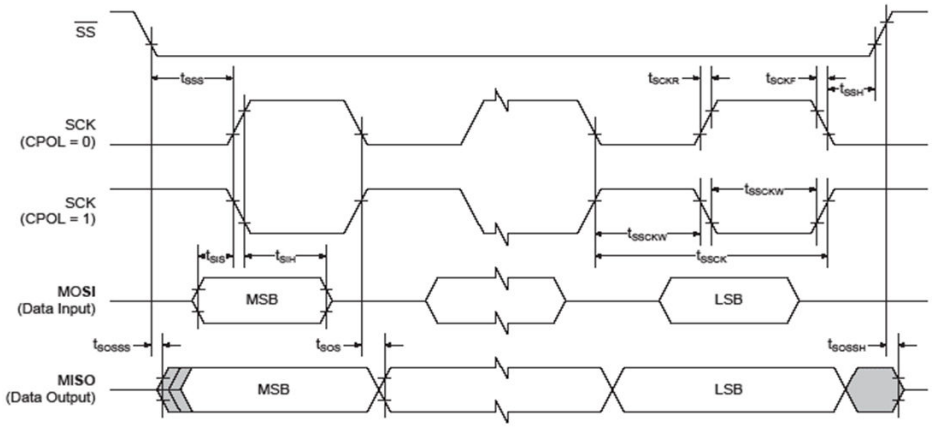


Figure 47-3. SPI Timing Requirements in Slave Mode



Maximum SPI Frequency

- Master mode:
  f_{SCKmax} = 1/2*(t_{MIS} + t_{valid}), where t_{valid} is the slave time response to output data after detecting an SCK edge. For a non-volatile memory with t_{valid} = 12 ns Max, $f_{SPCKMax}$ = 3.7 MHz @ VDDIO > 2.7V
- Slave mode:
  f_{SCKmax} = 1/2*(t_{SOV} + t_{su}), where t_{su} is the setup time from the master before sampling data. With a perfect master (t_{su}=0), $f_{SPCKMax}$ = 6 MHz @ VDDIO > 2.7V

47.6.2 SERCOM in SPI Mode in PL2

Table 47-27. SPI Timing Characteristics and Requirements (1)

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Units
tSCK	SCK period when tSOV=0 on the slave side	Master	Reception	2*(tMIS +tSLAVE_OUT) (3)	-	-	ns
		Master	Transmission	2*(tMOV +tSLAVE_IN) (4)	-	-	
tSCKW	SCK high/low width	Master		-	0,5*tSCK	-	
tSCKR	SCK rise time(2)	Master		-	0,25*tSCK	-	

Downloaded from Arrow.com.

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Units
tSCKF	SCK fall time(2)	Master		-	0,25*tSCK	-	
tMIS	MISO setup to SCK	Master, VDD>2,70V		43.8	-	-	
		Master, VDD>1,62V		54.1	-	-	
tMIH	MISO hold after SCK	Master, VDD>2,70V		0	-	-	ns
		Master, VDD>1,62V		0	-	-	
tMOV	MOSI output valid after SCK	Master, VDD>2,70V		-	-	17.5	
		Master, VDD>1,62V		-	-	21.2	
tMOH	MOSI hold after SCK	Master, VDD>2,70V		6.32	-	-	
		Master, VDD>1,62V		6.32	-	-	
tSSCK	Slave SCK Period when tMIS=0 on the master side	Slave	Reception	2*(tSIS +tMASTER_OUT) (5)	-	-	
		Slave	Transmission	2*(tSOV +tMASTER_IN) (6)	-	-	
tSSCKW	SCK high/low width	Slave		-	0,5*tSCK	-	
tSSCKR	SCK rise time(2)	Slave		-	0,25*tSCK	-	
tSSCKF	SCK fall time(2)	Slave		-	0,25*tSCK	-	
tSIS	MOSI setup to SCK	Slave, VDD>2,70V		10.7	-	-	ns
		Slave, VDD>1,62V		11.4	-	-	
tSIH	MOSI hold after SCK	Slave, VDD>2,70V		6.4	-	-	
		Slave, VDD>1,62V		7.1	-	-	
tSSS	SS setup to SCK	Slave	PRELOADEN=1	tSOSS+tEXT_MIS +2*tAPBC (8) (9)	-	-	
			PRELOADEN=0	tSOSS+tEXT_MIS (8)	-	-	
tSSH	SS hold after SCK	Slave		0.5*tSSCK	-	-	
tSOV	MISO output valid after SCK	Slave, VDD>2,70V		-	-	36.1	
		Slave, VDD>1,62V		-	-	46.4	
tSOH	MISO hold after SCK	Slave, VDD>2,70V		13.4	-	-	

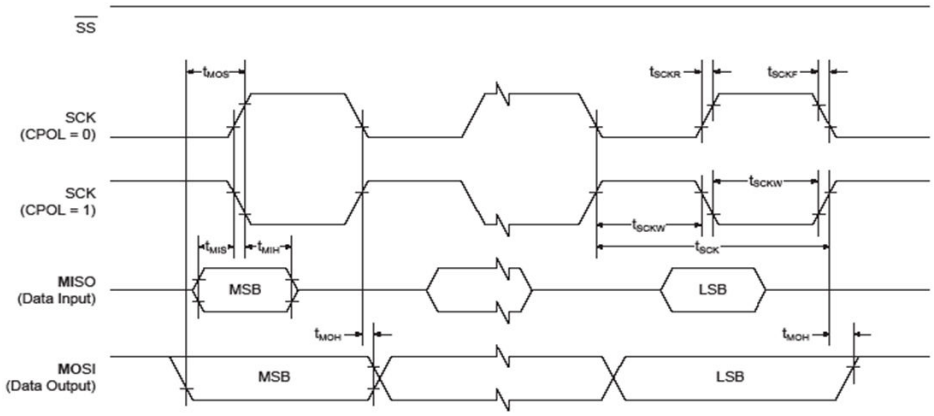
Downloaded from Arrow.com.

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Units
		Slave, VDD>1,62V		13.4	-	-	
tSOSS	MISO setup after SS low	Slave, VDD>2,70V			-	1* tSCK	
		Slave, VDD>1,62V			-	1* tSCK	
tSOSH	MISO hold after SS high	Slave, VDD>2,70V		8.7	-	-	
		Slave, VDD>1,62V		8.7	-	-	

Note:

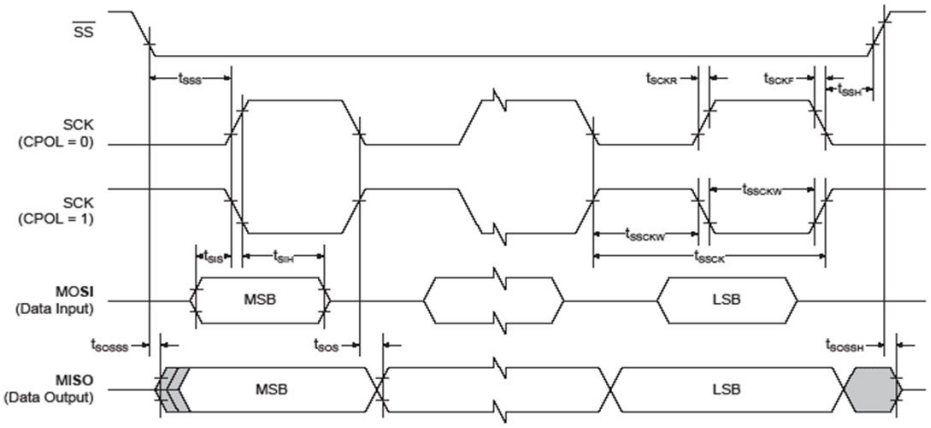
1. These values are based on simulation. These values are not covered by test limits in production.
2. See I/O Pin Characteristics.
3. Where tSLAVE_OUT is the slave external device output response time, generally tEXT_SOV +tLINE_DELAY (See Note 7).
4. Where tSLAVE_IN is the slave external device input constraint, generally tEXT_SIS+tLINE_DELAY (See Note 7).
5. Where tMASTER_OUT is the master external device output response time, generally tEXT_MOV +tLINE_DELAY (See Note 7).
6. Where tMASTER_IN is the master external device input constraint, generally tEXT_MIS +tLINE_DELAY (See Note 7).
7. tLINE_DELAY is the transmission line time delay.
8. tEXT_MIS is the input constraint for the master external device.
9. tAPBC is the APB period for SERCOM.

Figure 47-4. SPI Timing Requirements in Master Mode



Downloaded from Arrow.com.

Figure 47-5. SPI Timing Requirements in Slave Mode



Maximum SPI Frequency

- Master mode:
  $f_{SCKmax} = 1/2*(t_{MIS} + t_{valid})$, where t_{valid} is the slave time response to output data after detecting an SCK edge. For a non-volatile memory with t_{valid} = 12 ns Max, $f_{SPCKMax}$ = 9.8 MHz @ VDDIO > 2.7V
- Slave mode:
  $f_{SCKmax} = 1/2*(t_{SOV} + t_{su})$, where t_{su} is the setup time from the master before sampling data. With a perfect master (t_{su}=0), $f_{SPCKMax}$ = 16.3 MHz @ VDDIO > 2.7V

Downloaded from Arrow.com.

48. AC and DC Characteristics Graphs

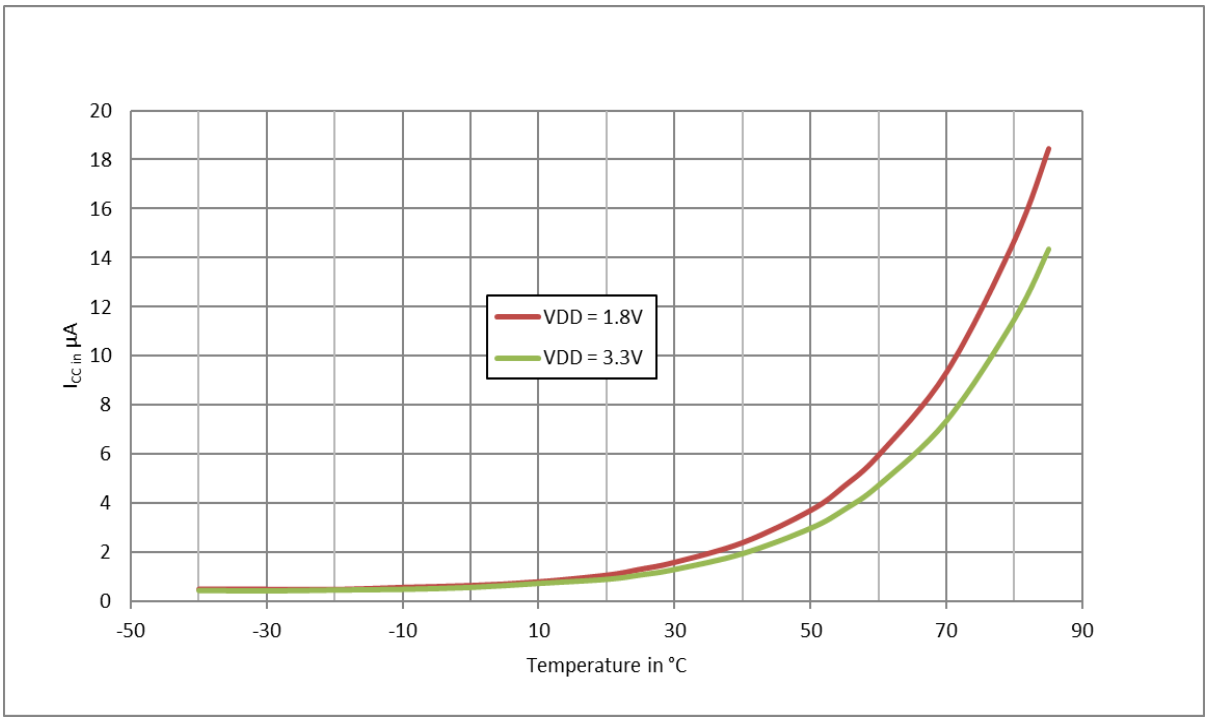
48.1 Typical Power Consumption over Temperature in Sleep Modes - 85°C

Power Consumption in Standby Sleep Mode with PDSW in Active state

Operating conditions:

- VDDIO = 3.3V or 1.8V
- No RTC running
- BOD33 is disabled
- LPVREG with LPEFF Enable
- All 16 kB SRAM retained
- PDSW Domain in Active state

Figure 48-1. Power Consumption over Temperature in Standby Sleep Mode with PDSW in Active state



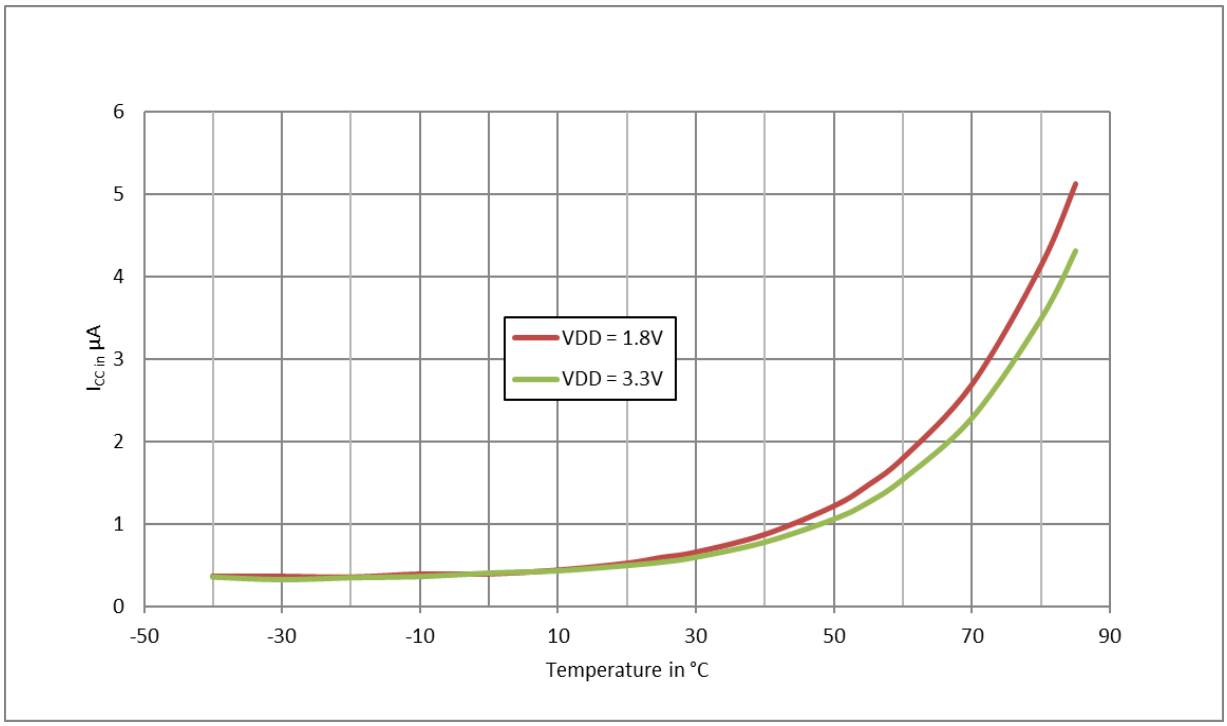
Power Consumption in Standby Sleep Mode with PDSW in Retention state

Operating conditions:

- VDDIO = 3.3V or 1.8V
- No RTC running
- BOD33 is disabled
- LPVREG with LPEFF Enable
- All 16 kB SRAM retained
- PDSW Domain in Retention state

Downloaded from Arrow.com.

Figure 48-2. Power Consumption over Temperature in Standby Sleep Mode with PDSW in Retention state

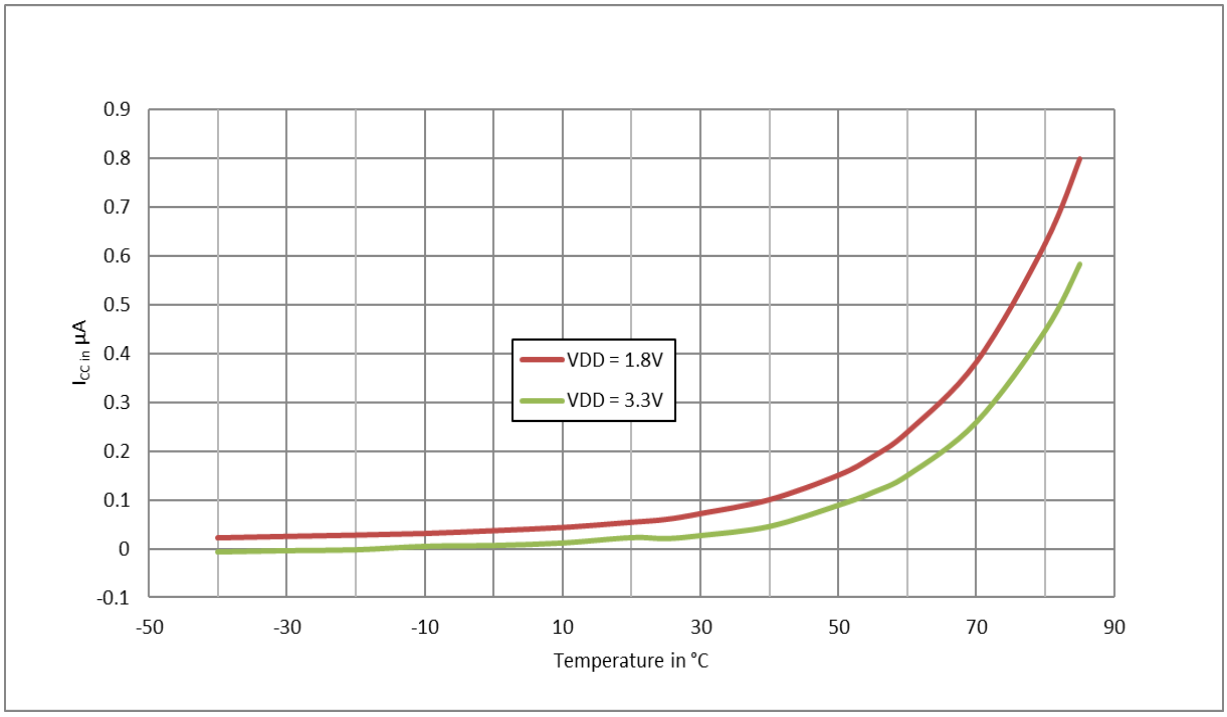


Power Consumption in Off Sleep Mode

Operating conditions:

- VDDIO = 3.3V or 1.8V

Figure 48-3. Power Consumption over Temperature in Off Sleep Mode



Downloaded from Arrow.com.

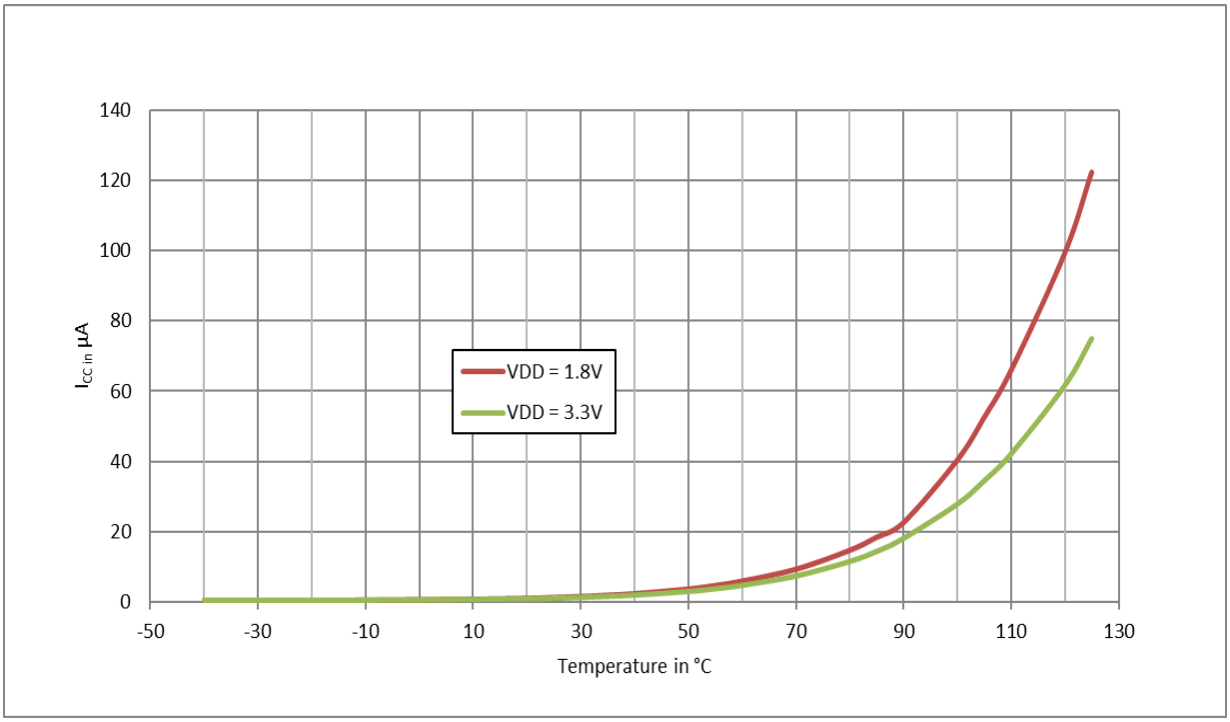
48.2 Typical Power Consumption over Temperature in Sleep Modes - 125°C

Power Consumption in Standby Sleep mode with PDSW in Active state

Operating conditions:

- VDDIO = 3.3V or 1.8V
- No RTC running
- BOD33 is disabled
- LPVREG with LPEFF Enable
- All 16 kB SRAM retained
- PDSW Domain in Active state

Figure 48-4. Power Consumption over Temperature in Standby Sleep Mode with PDSW in Active state



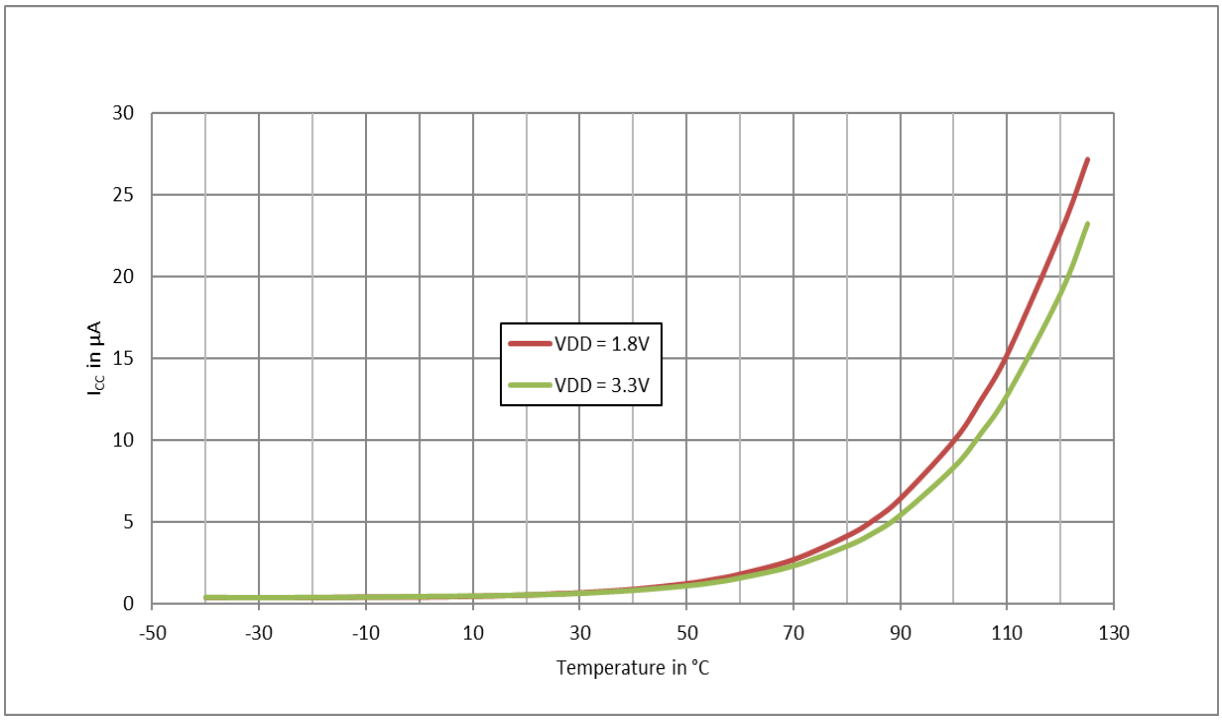
Power Consumption in Standby Sleep Mode with PDSW in Retention state

Operating conditions:

- VDDIO = 3.3V or 1.8V
- No RTC running
- BOD33 is disabled
- LPVREG with LPEFF Enable
- All 16 kB SRAM retained
- PDSW Domain in Retention state

Downloaded from Arrow.com.

Figure 48-5. Power Consumption over Temperature in Standby Sleep Mode with PDSW in Retention state

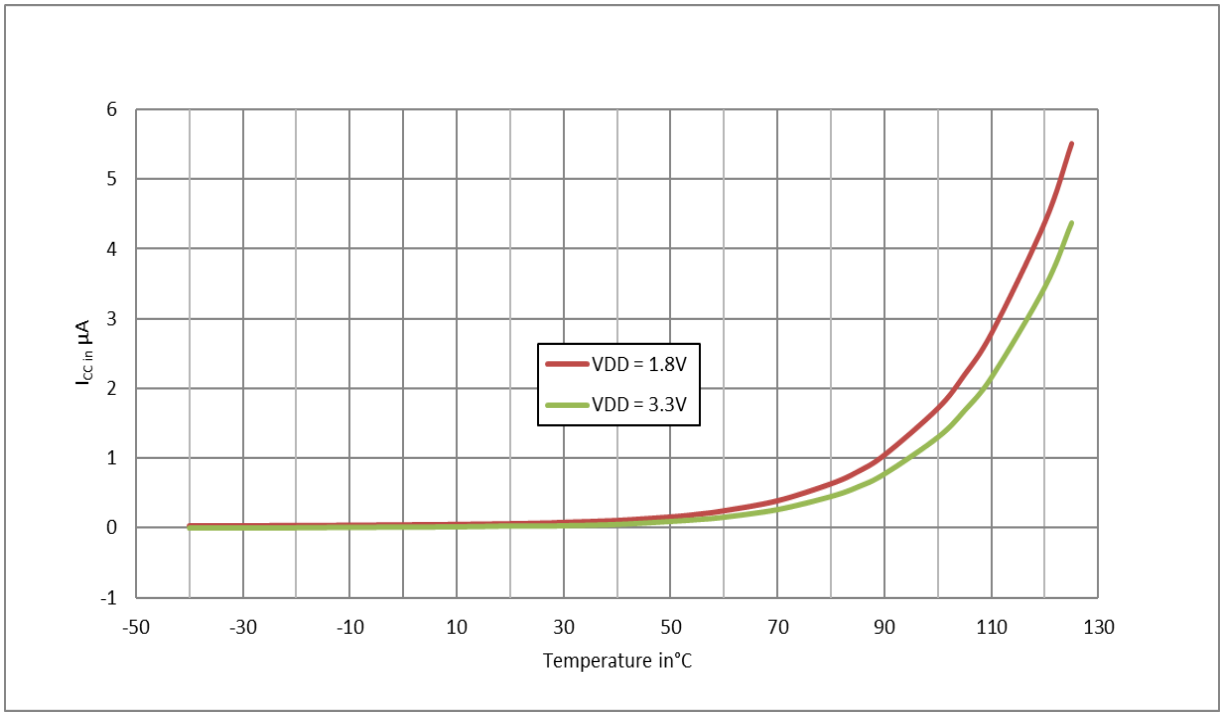


Power Consumption in Off Sleep Mode

Operating conditions:

- VDDIO = 3.3V or 1.8V

Figure 48-6. Power Consumption over Temperature in Off Sleep Mode



Downloaded from Arrow.com.

49. Packaging Information

49.1 Package Marking Information

All devices are marked with the Atmel logo, a shortened ordering code and additional marking (the two last lines)

YYWW R ARM

XXXXXX CC

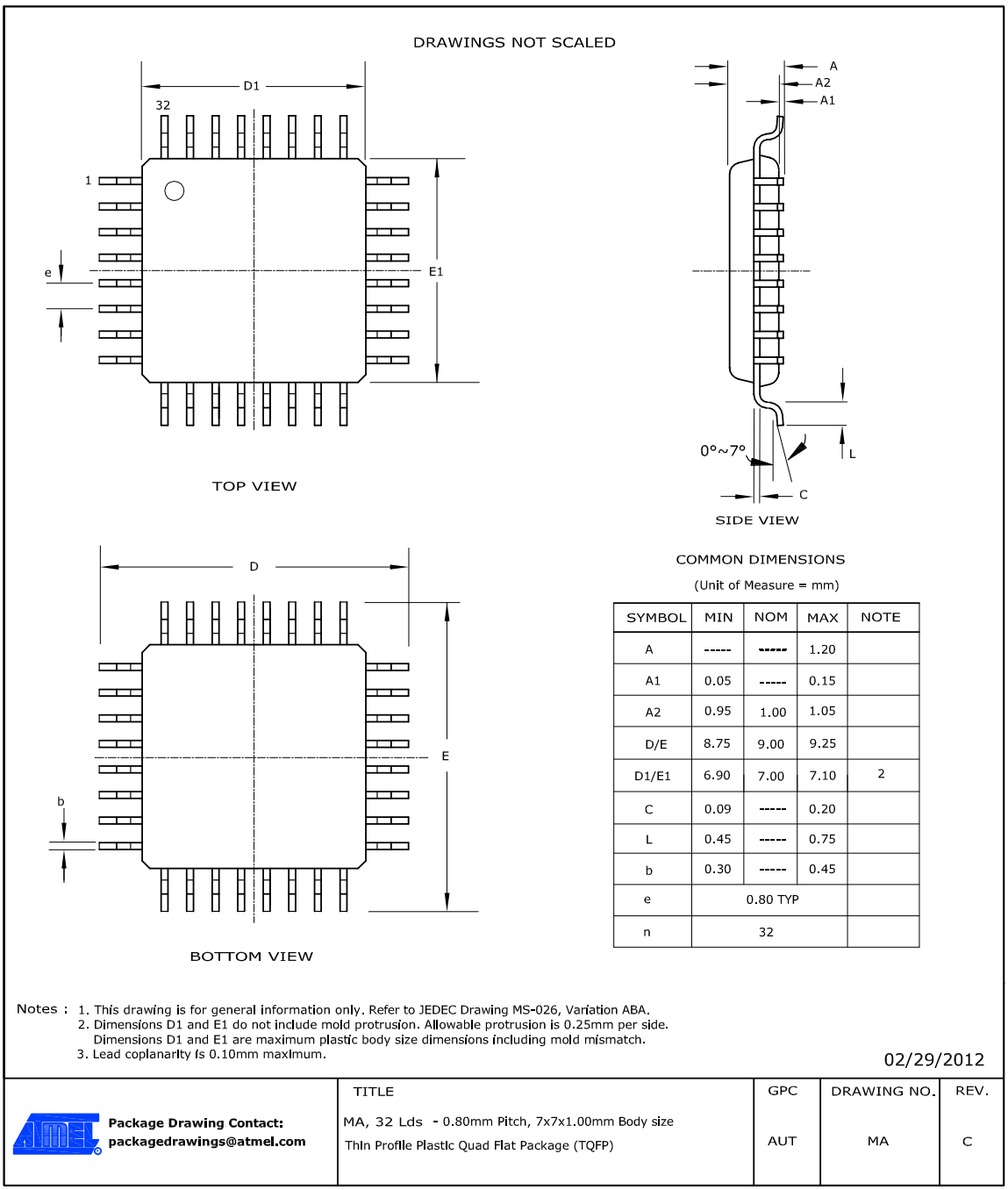
Where:

- "Y" or "YY": Manufacturing Year (last OR two last digit(s))
- "WW": Manufacturing Week
- "R": Revision
- "XXXXXX": Lot number
- "CC": Internal Code

Downloaded from Arrow.com.

49.2 Package Drawings

49.2.1 32-pin TQFP



COMMON DIMENSIONS

(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	-----	-----	1.20	
A1	0.05	-----	0.15	
A2	0.95	1.00	1.05	
D/E	8.75	9.00	9.25	
D1/E1	6.90	7.00	7.10	2
C	0.09	-----	0.20	
L	0.45	-----	0.75	
b	0.30	-----	0.45	
e	0.80 TYP			
n	32			

Notes : 1. This drawing is for general information only. Refer to JEDEC Drawing MS-026, Variation ABA.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.10mm maximum.

02/29/2012

Package Drawing Contact: packagedrawings@atmel.com	TITLE	GPC	DRAWING NO.	REV.
	MA, 32 Lds - 0.80mm Pitch, 7x7x1.00mm Body size Thin Profile Plastic Quad Flat Package (TQFP)	AUT	MA	C

Table 49-1. Device and Package Maximum Weight

100	mg

Table 49-2. Package Characteristics

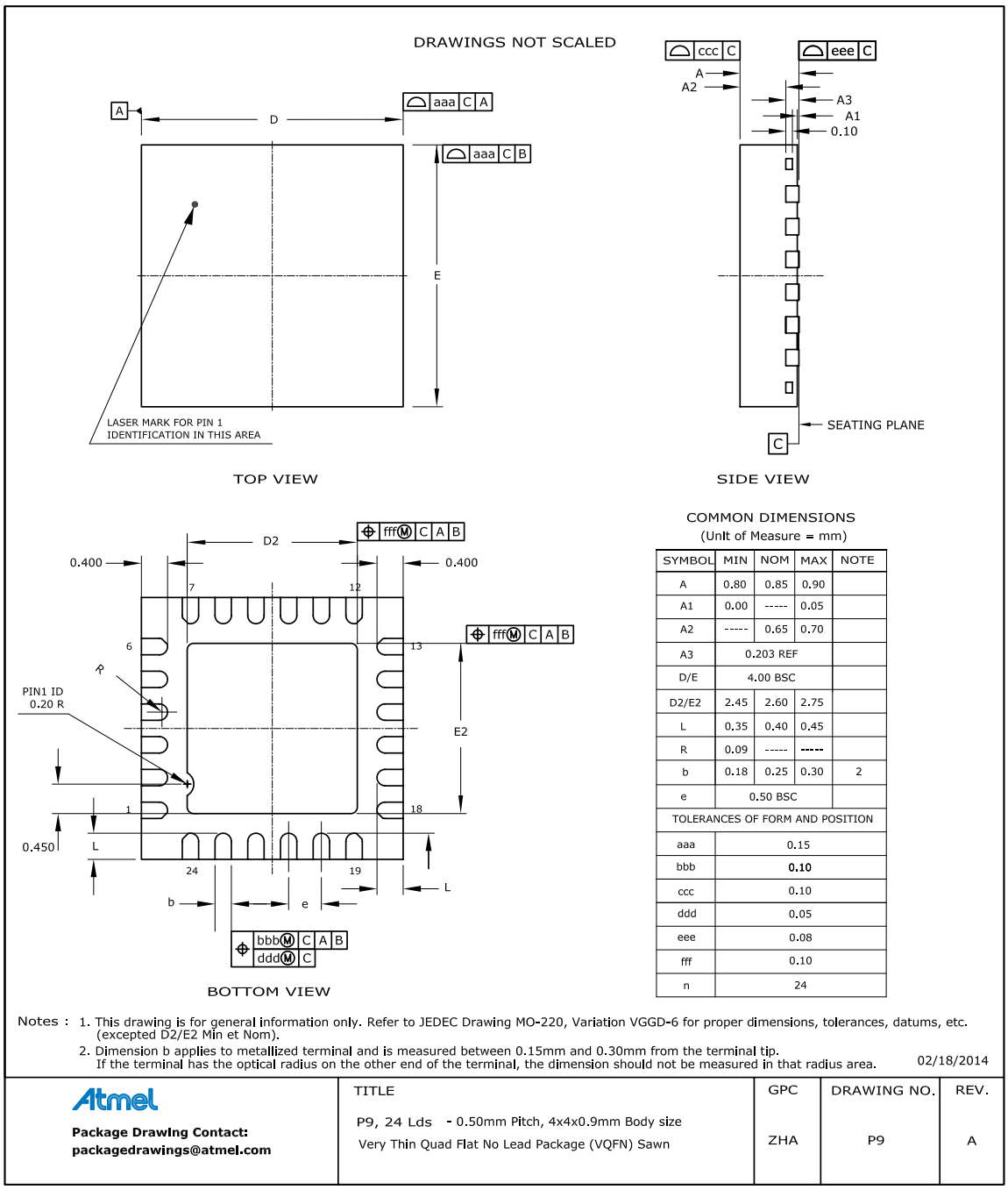
Moisture Sensitivity Level	MSL3

Downloaded from Arrow.com.

Table 49-3. Package Reference

JEDEC Drawing Reference	MS-026
JESD97 Classification	E3

49.2.2 24-pin VQFN



COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	0.80	0.85	0.90	
A1	0.00	-----	0.05	
A2	-----	0.65	0.70	
A3		0.203 REF		
D/E		4.00 BSC		
D2/E2	2.45	2.60	2.75	
L	0.35	0.40	0.45	
R	0.09	-----	-----	
b	0.18	0.25	0.30	2
e		0.50 BSC		
TOLERANCES OF FORM AND POSITION				
aaa		0.15		
bbb		0.10		
ccc		0.10		
ddd		0.05		
eee		0.08		
fff		0.10		
n		24		

Notes : 1. This drawing is for general information only. Refer to JEDEC Drawing MO-220, Variation VGGD-6 for proper dimensions, tolerances, datums, etc. (excepted D2/E2 Min et Nom).
2. Dimension b applies to metallized terminal and is measured between 0.15mm and 0.30mm from the terminal tip. If the terminal has the optical radius on the other end of the terminal, the dimension should not be measured in that radius area.

02/18/2014

	TITLE	GPC	DRAWING NO.	REV.
Atmel **Package Drawing Contact:** **packagedrawings@atmel.com**	P9, 24 Lds - 0.50mm Pitch, 4x4x0.9mm Body size Very Thin Quad Flat No Lead Package (VQFN) Sawn	ZHA	P9	A

Table 49-4. Device and Package Maximum Weight

36	mg

Downloaded from Arrow.com.

Table 49-5. Package Characteristics

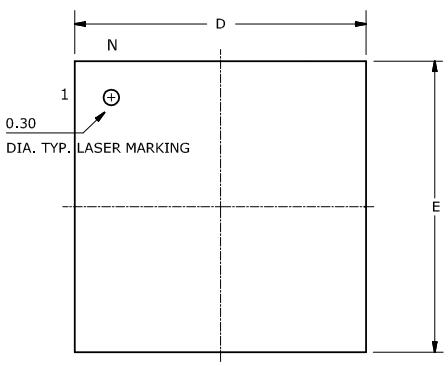
Moisture Sensitivity Level	MSL1

Table 49-6. Package Reference

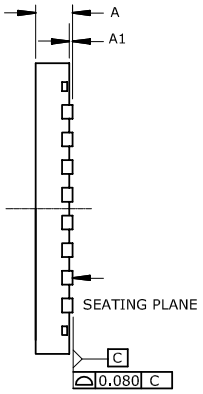
JEDEC Drawing Reference	MO-220
JESD97 Classification	E3

49.2.3 32-pin VQFN

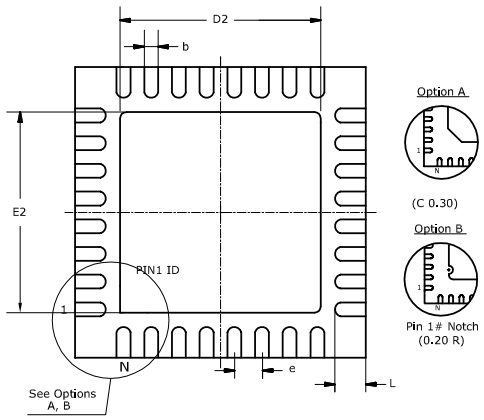
DRAWINGS NOT SCALED



TOP VIEW



SIDE VIEW



BOTTOM VIEW

COMMON DIMENSIONS

(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	0.80	-----	1.00	
A1	0.00	-----	0.05	
D/E		5.00 BSC		
D2/E2	3.50	3.60	3.70	
L	0.30	0.40	0.50	
b	0.18	0.25	0.30	2
e		0.50 BSC		
n		32		

Notes : 1. This drawing is for general information only. Refer to JEDEC Drawing MO-220, Variation VHHD-5, for proper dimensions, tolerances, datums, etc.
2. Dimension b applies to metallized terminal and is measured between 0.15mm and 0.30mm from the terminal tip.
If the terminal has the optical radius on the other end of the terminal, the dimension should not be measured in that radius area.

10/05/2015

Atmel	TITLE	GPC	DRAWING NO.	REV.
Package Drawing Contact: **packagedrawings@atmel.com**	PG, 32 Leads , 0.50mm pitch, 5x5x1.0mm Body Size Very Thin Quad Flat No Lead Package (VQFN) Sawn	ZKV	PG	C

Downloaded from Arrow.com.

Table 49-7. Device and Package Maximum Weight

90	mg

Table 49-8. Package Characteristics

Moisture Sensitivity Level	MSL3

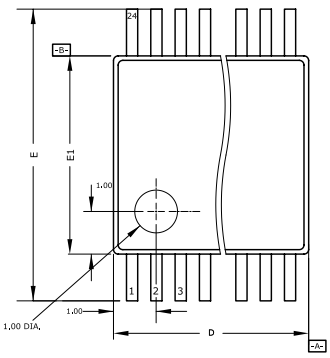
Table 49-9. Package Reference

JEDEC Drawing Reference	MO-220
JESD97 Classification	E3

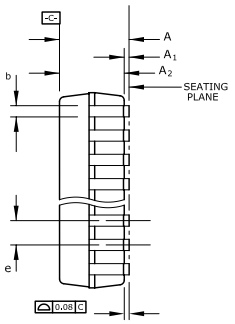
Downloaded from Arrow.com.

49.2.4 24-pin SSOP

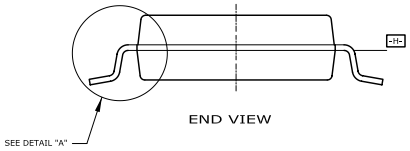
DRAWINGS NOT SCALED



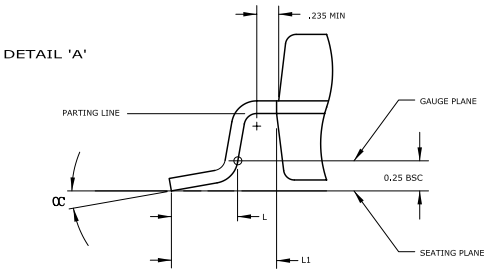
TOP VIEW



SIDE VIEW



END VIEW



COMMON DIMENSIONS

(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	-----	-----	2.00	
A1	0.05	-----	-----	
A2	1.65	1.75	1.85	
D	7.90	8.20	8.50	2
E	7.40	7.80	8.20	
E1	5.00	5.30	5.60	2
L	0.55	0.75	0.95	
L1	1.25 REF			
b	0.22	-----	0.38	
c	0.09	-----	0.25	
e	0.65 bsc			
θC	0°	4°	8°	
n	24			

Notes :
1. This drawing is for general information only. Refer to JEDEC Drawing MO-150, Variation AG.
2. Dimensions D and E1 do not include mold protrusion. Allowable protrusion shall not exceed 0.15mm per side. Dimensions D and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.08mm maximum.

04/12/2017

Atmel	TITLE	GPC	DRAWING NO.	REV.
Package Drawing Contact: **packagedrawings@atmel.com**	IC, 24 Lds - 0.65mm Pitch, 8.2x5.3x1.75mm Body Size Shrink Small Outline Package (SSOP .209")	TZY	IC	D

Downloaded from Arrow.com.

Table 49-10. Device and Package Maximum Weight

187.322	mg

Table 49-11. Package Characteristics

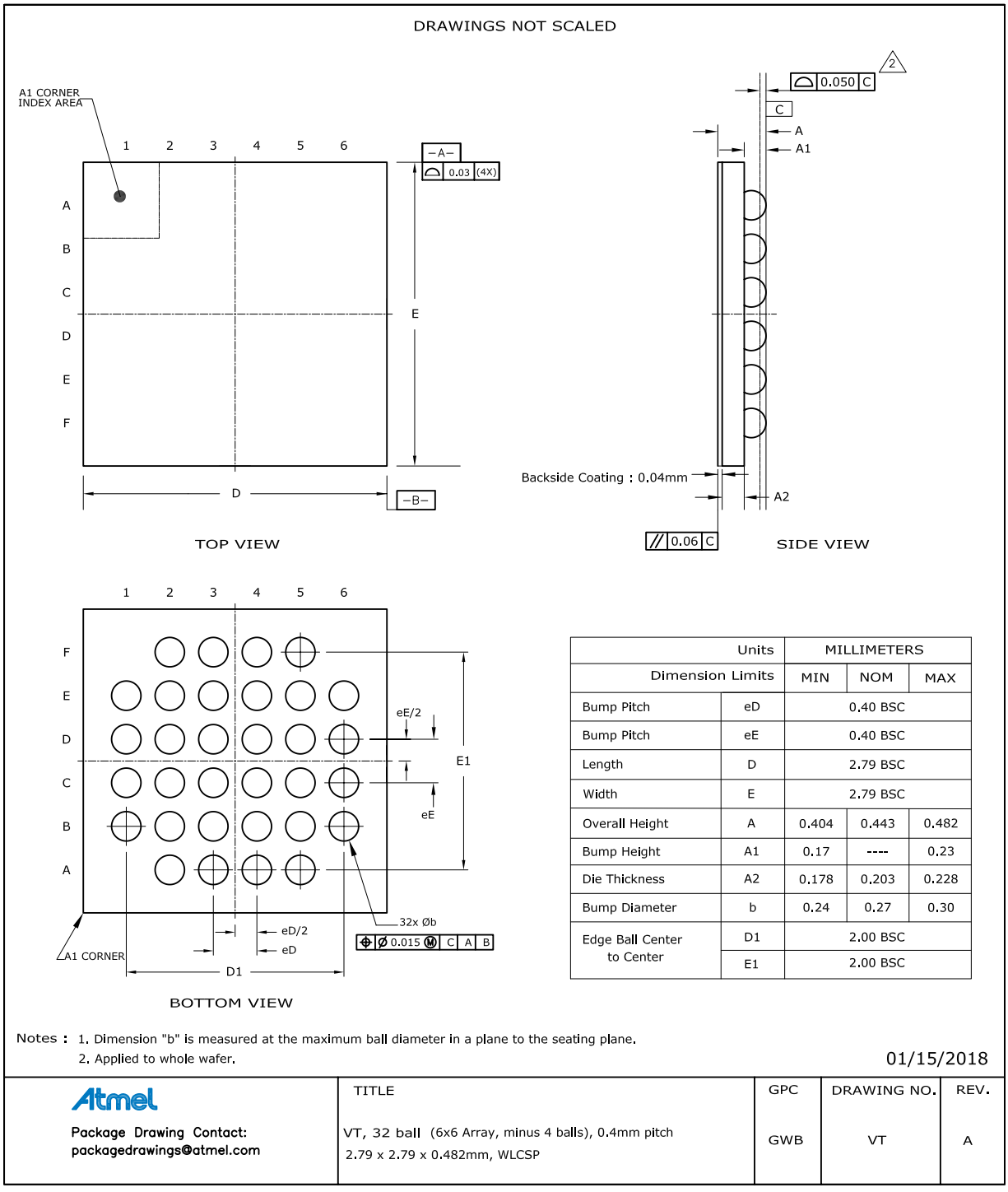
Moisture Sensitivity Level	MSL3

Table 49-12. Package Reference

JEDEC Drawing Reference	MO-150
JESD97 Classification	E3

Downloaded from Arrow.com.

49.2.5 32-pin WLCSP



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Bump Pitch	eD		0.40 BSC	
Bump Pitch	eE		0.40 BSC	
Length	D		2.79 BSC	
Width	E		2.79 BSC	
Overall Height	A	0.404	0.443	0.482
Bump Height	A1	0.17	----	0.23
Die Thickness	A2	0.178	0.203	0.228
Bump Diameter	b	0.24	0.27	0.30
Edge Ball Center to Center	D1		2.00 BSC	
	E1		2.00 BSC	

Notes : 1. Dimension "b" is measured at the maximum ball diameter in a plane to the seating plane.
2. Applied to whole wafer.

01/15/2018

Atmel	TITLE	GPC	DRAWING NO.	REV.
Package Drawing Contact: packagedrawings@atmel.com	VT, 32 ball (6x6 Array, minus 4 balls), 0.4mm pitch 2.79 x 2.79 x 0.482mm, WLCSP	GWB	VT	A

Table 49-13. Device and Package Maximum Weight

6.04	mg

Downloaded from Arrow.com.

Table 49-14. Package Characteristics

Moisture Sensitivity Level	MSL1

Table 49-15. Package Reference

JEDEC Drawing Reference	N/A
JESD97 Classification	E1

49.3 Soldering Profile

The following table gives the recommended soldering profile from J-STD-20.

Table 49-16. Recommended Soldering Profile

Profile Feature	Green Package
Average Ramp-up Rate (217°C to peak)	3°C/s max.
Preheat Temperature 175°C ±25°C	150-200°C
Time Maintained Above 217°C	60-150s
Time within 5°C of Actual Peak Temperature	30s
Peak Temperature Range	260°C
Ramp-down Rate	6°C/s max.
Time 25°C to Peak Temperature	8 minutes max.

A maximum of three reflow passes is allowed per component.

Downloaded from Arrow.com.

50. Schematic Checklist

50.1 Introduction

This chapter describes a common checklist which should be used when starting and reviewing the schematics for a SAM L10/L11 design. This chapter illustrates the recommended power supply connections, how to connect external analog references, programmer, debugger, oscillator and crystal.

50.2 Power Supply

The SAM L10/L11 supports a single or dual power supply from 1.62V to 3.63V. The same voltage must be applied to both VDDIO and VDDANA.

The internal voltage regulator has four different modes:

- Linear mode: this mode does not require any external inductor. This is the default mode when CPU and peripherals are running
- Switching mode (Buck): the most efficient mode when the CPU and peripherals are running
- Low Power (LP) mode: This is the default mode used when the device is in Standby mode

Selecting between switching mode and linear mode can be done by software on the fly, but the power supply must be designed according to which mode is to be used.

50.2.1 Power Supply Connections

The following figures shows the recommended power supply connections for Switched/Linear mode and Linear mode only.

Downloaded from Arrow.com.

Figure 50-1. Power Supply Connection for Switching/Linear Mode

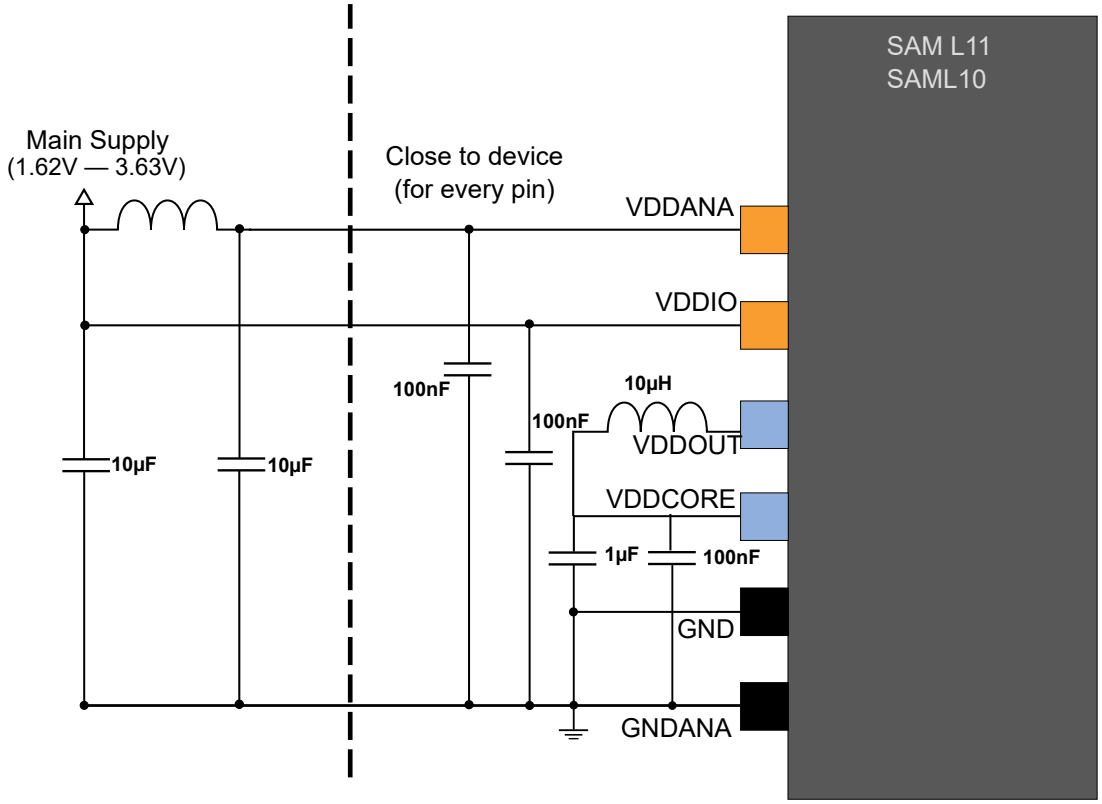
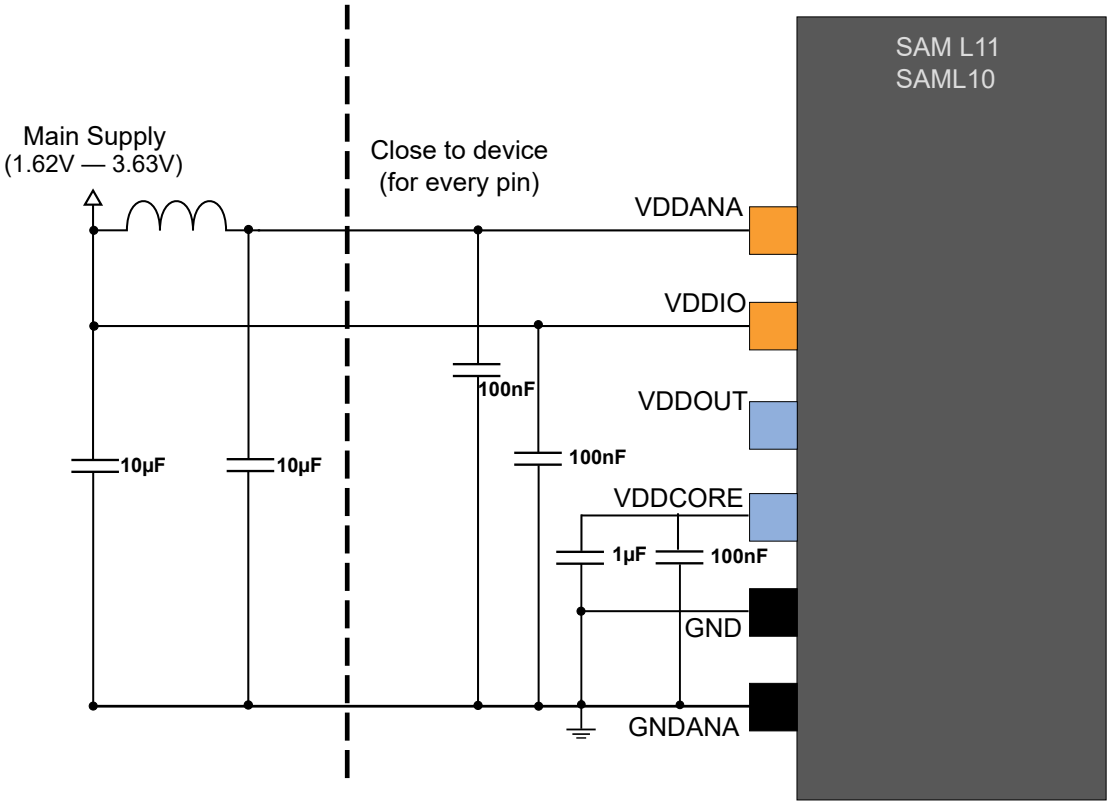


Figure 50-2. Power Supply Connection for Linear Mode Only



Downloaded from Arrow.com.

Table 50-1. Power Supply Connections

Signal Name	Recommended Pin Connection	Description
V_{DDIO}	1.62V to 3.63V Decoupling/filtering capacitors 100 nF(1,2) and 10 µF(1) Decoupling/filtering inductor 10 µH(1,3)	Digital supply voltage
V_{DDANA}	1.62V to 3.63V Decoupling/filtering capacitors 100 nF(1,2) and 10 µF(1) Ferrite bead(4) prevents the V_{DD} noise interfering with V_{DDANA}	Analog supply voltage
V_{DDCORE}	0.9V to 1.2V typical Decoupling/filtering capacitors 100 nF(1,2) and 1µF(1)	Linear regulator mode: Core supply voltage output/external decoupling pin Switched regulator mode: Core supply voltage input, must be connected to V_{DDOUT} via inductor
V_{DDOUT}	Switching regulator mode: 10 µH inductor with saturation current above 150mA and DCR<1Ω Linear regulator mode: Not connected	On-chip Switching mode regulator output
GND		Ground
GND_{ANA}		Ground for the analog power domain

1. These values are only given as a typical example.

2. Decoupling capacitors should be placed close to the device for each supply pin pair in the signal group, low ESR capacitors should be used for better decoupling.

3. An inductor should be added between the external power and the V_{DD} for power filtering.

4. A ferrite bead has better filtering performance compared to standard inductor at high frequencies. A ferrite bead can be added between the main power supply and V_{DDANA} to prevent digital noise from entering the analog power domain. The bead should provide enough impedance (for example, 50Ω at 20MHz and 220Ω at 100MHz) to separate the digital and analog power domains. Make sure to select a ferrite bead designed for filtering applications with a low DC resistance to avoid a large voltage drop across the ferrite bead.

50.2.2 Special Considerations for QFN Packages

The QFN package has an exposed paddle that must be connected to GND.

50.3 External Analog Reference Connections

The following schematic checklist is only necessary if the application is using one or more of the external analog references. If the internal references are used instead, the circuits in the following two figures are not necessary.

Downloaded from Arrow.com.

Figure 50-3. External Analog Reference Schematic With Two References

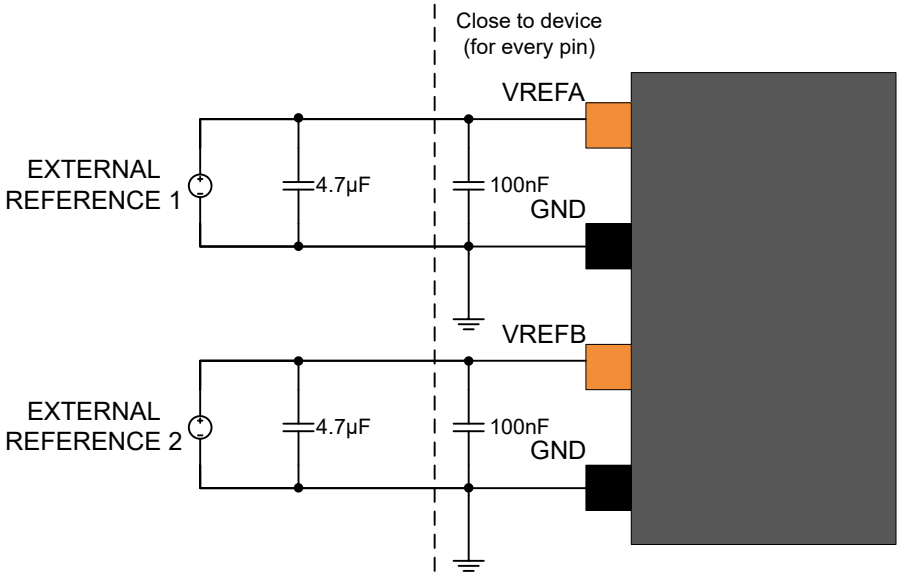


Figure 50-4. External Analog Reference Schematic With One Reference

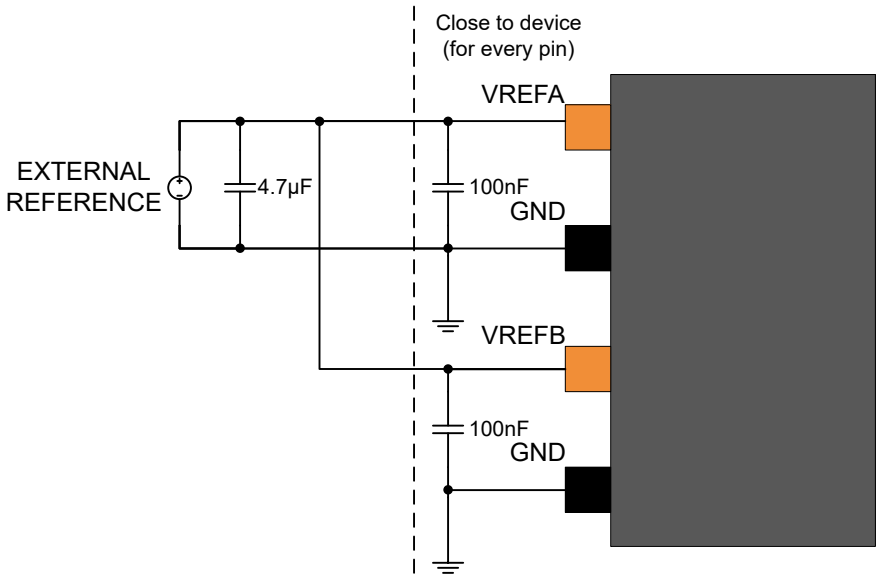


Table 50-2. External Analog Reference Connections

Signal Name	Recommended Pin Connection	Description
VREFx	1.0V to (V_{DDANA} - 0.6V) for ADC 1.0V to (V_{DDANA} - 0.15V) for DAC Decoupling/filtering capacitors 100nF(1)(2) and 4.7µF(1)	External reference VREFx for the analog port
GND		Ground

1. These values are only given as a typical example.

Downloaded from Arrow.com.

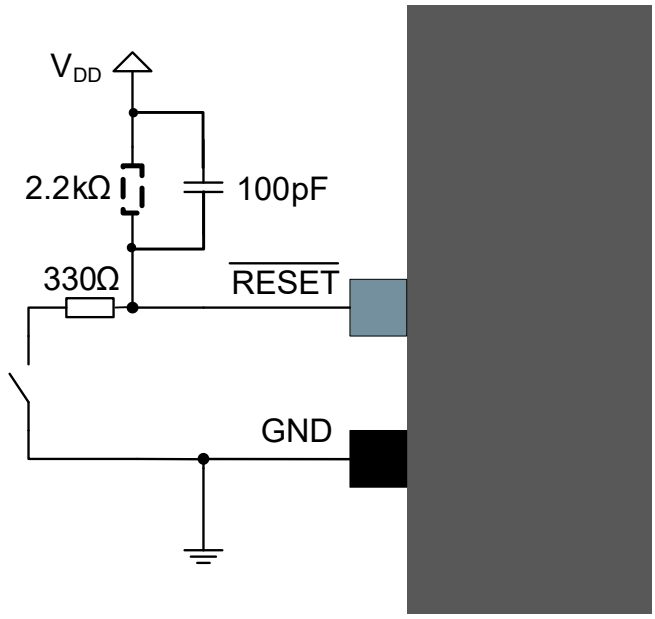
2. Decoupling capacitor should be placed close to the device for each supply pin pair in the signal group.

50.4 External Reset Circuit

The external Reset circuit is connected to the $\overline{\text{RESET}}$ pin when the external Reset function is used. The circuit is not necessary when the $\overline{\text{RESET}}$ pin is not driven LOW externally by the application circuitry.

The reset switch can also be removed, if a manual reset is not desired. The $\overline{\text{RESET}}$ pin itself has an internal pull-up resistor, hence it is optional to add any external pull-up resistor.

Figure 50-5. External Reset Circuit Schematic



A pull-up resistor makes sure that the reset does not go low and unintentionally causing a device reset. An additional resistor has been added in series with the switch to safely discharge the filtering capacitor, i.e. preventing a current surge when shorting the filtering capacitor which again can cause a noise spike that can have a negative effect on the system.

Table 50-3. Reset Circuit Connections

Signal Name	Recommended Pin Connection	Description
$\overline{\text{RESET}}$	Reset low level threshold voltage V_{DDIO} = 1.62V - 2.0V: Below 0.33 * V_{DDIO} V_{DDIO} = 2.7V - 3.63V: Below 0.36 * V_{DDIO} Decoupling/filter capacitor 100pF(1) Pull-up resistor 2.2kΩ(1,2) Resistor in series with the switch 330Ω(1)	Reset pin

1. These values are only given as a typical example.

2. The SAM L10/L11 features an internal pull-up resistor on the $\overline{\text{RESET}}$ pin, hence an external pull-up is optional.

Downloaded from Arrow.com.

50.5 Unused or Unconnected Pins

For unused pins the default state of the pins will give the lowest current leakage. Thus there is no need to do any configuration of the unused pins in order to lower the power consumption.

50.6 Clocks and Crystal Oscillators

The SAM L10/L11 can be run from internal or external clock sources, or a mix of internal and external sources. An example of usage can be to use the internal 16MHz oscillator as source for the system clock and an external 32.768kHz watch crystal as clock source for the Real-Time counter (RTC).

50.6.1 External Clock Source

Figure 50-6. External Clock Source Schematic

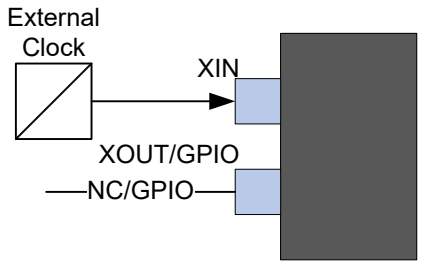
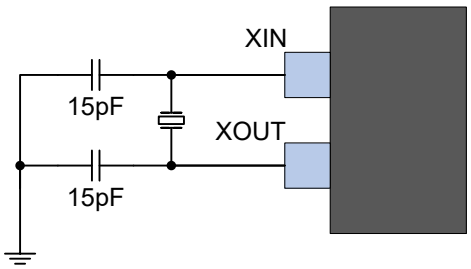


Table 50-4. External Clock Source Connections

Signal Name	Recommended Pin Connection	Description
XIN	XIN is used as input for an external clock signal	Input for inverting oscillator pin
XOUT/GPIO	Can be left unconnected or used as normal GPIO	NC/GPIO

50.6.2 Crystal Oscillator

Figure 50-7. Crystal Oscillator Schematic



The crystal should be located as close to the device as possible. Long signal lines may cause too high load to operate the crystal, and cause crosstalk to other parts of the system.

Table 50-5. Crystal Oscillator Checklist

Signal Name	Recommended Pin Connection	Description
XIN	Load capacitor 15pF(1)(2)	External crystal between 0.4 to 32MHz
XOUT	Load capacitor 15pF(1)(2)	

Downloaded from Arrow.com.

1. These values are only given as a typical example.

2. The capacitors should be placed close to the device for each supply pin pair in the signal group.

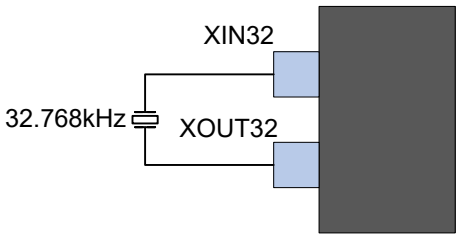
50.6.3 External Real Time Oscillator

The low frequency crystal oscillator is optimized for use with a 32.768 kHz watch crystal. When selecting crystals, load capacitance and the crystal's Equivalent Series Resistance (ESR) must be taken into consideration. Both values are specified by the crystal vendor.

SAM L10/L11 oscillator is optimized for very low power consumption, hence close attention should be made when selecting crystals.

The typical parasitic load capacitance values are available in the Electrical Characteristics chapters. This capacitance and PCB capacitance can allow using a crystal inferior to 12.5 pF load capacitance without external capacitors as shown in the following figure.

Figure 50-8. External Real Time Oscillator without Load Capacitor



To improve accuracy and Safety Factor, the crystal data sheet can recommend adding external capacitors, as shown in the following figure.

To find suitable load capacitance for a 32.768 kHz crystal, consult the crystal data sheet.

Figure 50-9. External Real Time Oscillator with Load Capacitor

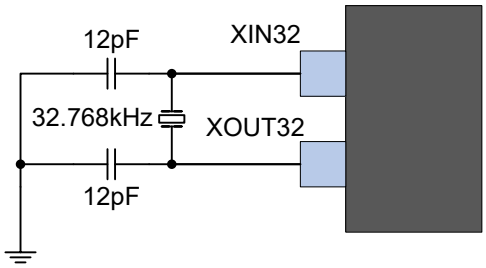


Table 50-6. External Real Time Oscillator Checklist

Signal Name	Recommended Pin Connection	Description
XIN32	Load capacitor 12 pF(1,2)	Timer oscillator input
XOUT32	Load capacitor 12 pF(1,2)	Timer oscillator output

1. These values are only given as typical examples.

2. The capacitors should be placed close to the device for each supply pin pair in the signal group.

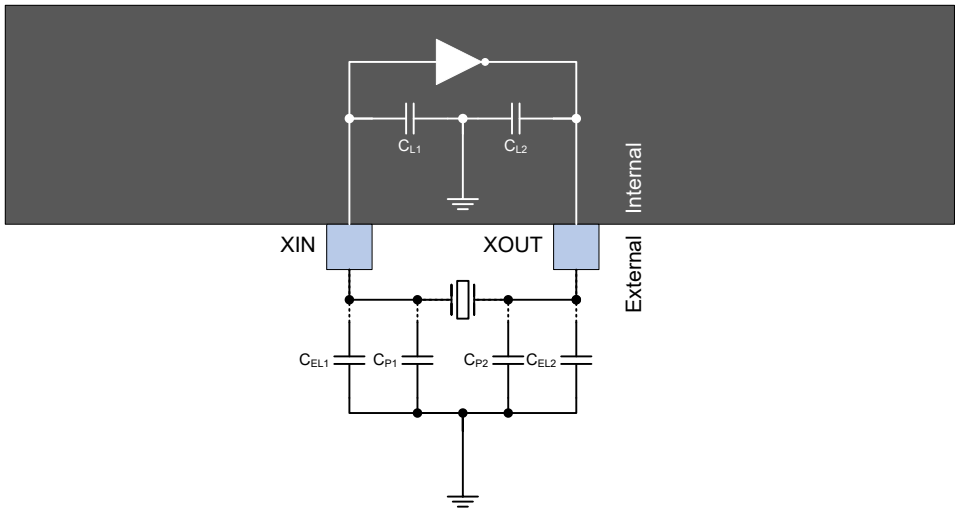
Note: To minimize the cycle-to-cycle jitter of the external oscillator, keep the neighboring pins as steady as possible. For neighboring pin details, refer to 4.2 Oscillators Pinout.

Downloaded from Arrow.com.

50.6.4 Calculating the Correct Crystal Decoupling Capacitor

The model shown in the following figure can be used to calculate correct load capacitor for a given crystal. This model includes internal capacitors C_{Ln}, external parasitic capacitance C_{ELn} and external load capacitance C_{Pn}.

Figure 50-10. Crystal Circuit With Internal, External and Parasitic Capacitance



Using this model the total capacitive load for the crystal can be calculated as shown in the equation below:

$$\sum C_{\text{tot}} = \frac{(C_{L1} + C_{P1} + C_{\text{EL1}})(C_{L2} + C_{P2} + C_{\text{EL2}})}{C_{L1} + C_{P1} + C_{\text{EL1}} + C_{L2} + C_{P2} + C_{\text{EL2}}}$$

where C_{tot} is the total load capacitance seen by the crystal. This value should be equal to the load capacitance value found in the crystal manufacturer datasheet.

The parasitic capacitance C_{ELn} can in most applications be disregarded as these are usually very small. If accounted for, these values are dependent on the PCB material and PCB layout.

For some crystal the internal capacitive load provided by the device itself can be enough. To calculate the total load capacitance in this case. C_{ELn} and C_{Pn} are both zero, $C_{L1} = C_{L2} = C_L$, and the equation reduces to the following:

$$\sum C_{\text{tot}} = \frac{C_L}{2}$$

See the related links for equivalent internal pin capacitance values.

50.7 Programming and Debug Ports

For programming and/or debugging the SAM L10/L11, the device should be connected using the Serial Wire Debug, SWD, interface. Currently the SWD interface is supported by several Microchip and third party programmers and debuggers, like the Atmel-ICE, SAM-ICE or SAM L10/L11 Xplained Pro (SAM L10/L11 evaluation kit) Embedded Debugger.

Refer to the Atmel-ICE, SAM-ICE or SAM L10/L11 Xplained Pro user guides for details on debugging and programming connections and options. For connecting to any other programming or debugging tool, refer to that specific programmer or debugger's user guide.

Downloaded from Arrow.com.

The SAM L10/L11 Xplained Pro evaluation board supports programming and debugging through the on-board embedded debugger so no external programmer or debugger is needed.

The SWDIO pin should be pulled up on the target.

It is recommended that the SWCLK pin is pulled to a defined state of the target board.

50.7.1 Cortex Debug Connector (10-pin)

For debuggers and/or programmers that support the Cortex Debug Connector (10-pin) interface the signals should be connected as shown in the following figure, with details described in the following table.

Figure 50-11. Cortex Debug Connector (10-pin)

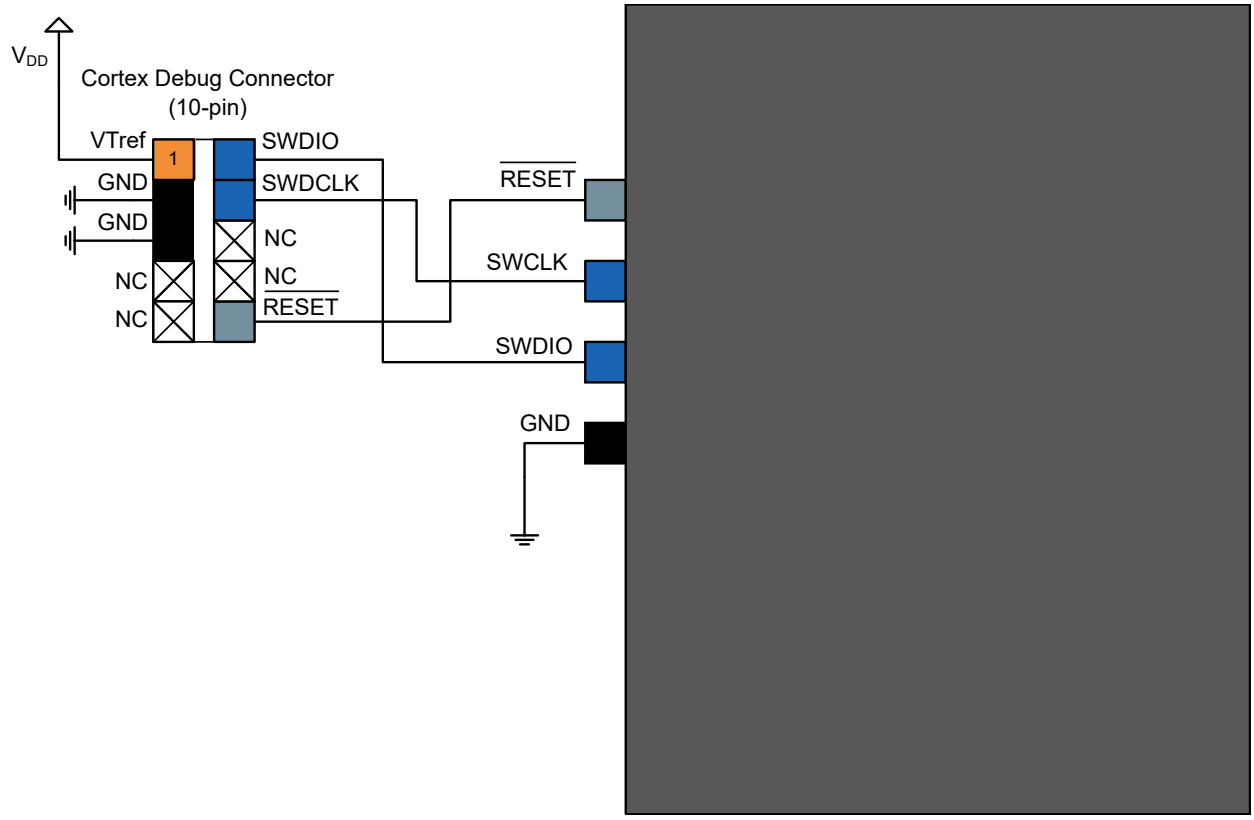


Table 50-7. Cortex Debug Connector (10-pin)

Header Signal Name	Description
SWDCLK	Serial wire clock pin
SWDIO	Serial wire bidirectional data pin
$\overline{\text{RESET}}$	Target device reset pin, active low
VTref	Target voltage sense, should be connected to the device V_{DD}
GND	Ground

50.7.2 10-pin JTAGICE3 Compatible Serial Wire Debug Interface

The JTAGICE3 debugger and programmer does not support the Cortex Debug Connector (10-pin) directly, hence a special pinout is needed to directly connect the SAM L10/L11 to the JTAGICE3, alternatively one can use the JTAGICE3 squid cable and manually match the signals between the

Downloaded from Arrow.com.

JTAGICE3 and SAM L10/L11. The following figure describes how to connect a 10-pin header that support connecting the JTAGICE3 directly to the SAM L10/L11 without the need for a squid cable. This can also be used for the Atmel-ICE AVR connector port.

The JTAGICE3 squid cable or the JTACICE3 50mil cable can be used to connect the JTAGICE3 programmer and debugger to the SAM L10/L11. The figure illustrates the correct pinout for the JTAGICE3 50 mil, and details are given in the following table.

Figure 50-12. 10-pin JTAGICE3 Compatible Serial Wire Debug Interface

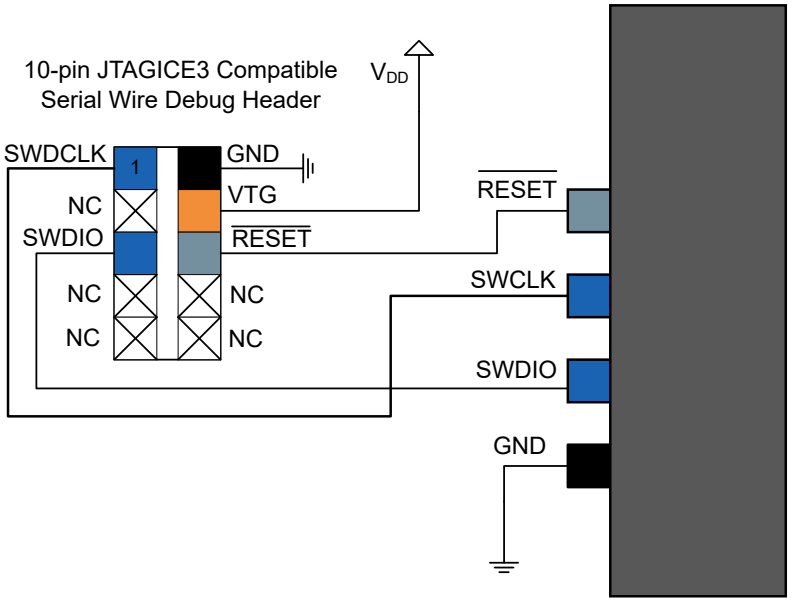


Table 50-8. 10-pin JTAGICE3 Compatible Serial Wire Debug Interface

Header Signal Name	Description
SWDCLK	Serial wire clock pin
SWDIO	Serial wire bidirectional data pin
$\overline{\text{RESET}}$	Target device reset pin, active low
VTG	Target voltage sense, should be connected to the device V_{DD}
GND	Ground

50.7.3 20-pin IDC JTAG Connector

For debuggers and/or programmers that support the 20-pin IDC JTAG Connector, e.g., the SAM-ICE, the signals should be connected, as shown in the following figure, with details described in the following table.

Downloaded from Arrow.com.

Figure 50-13. 20-pin IDC JTAG Connector

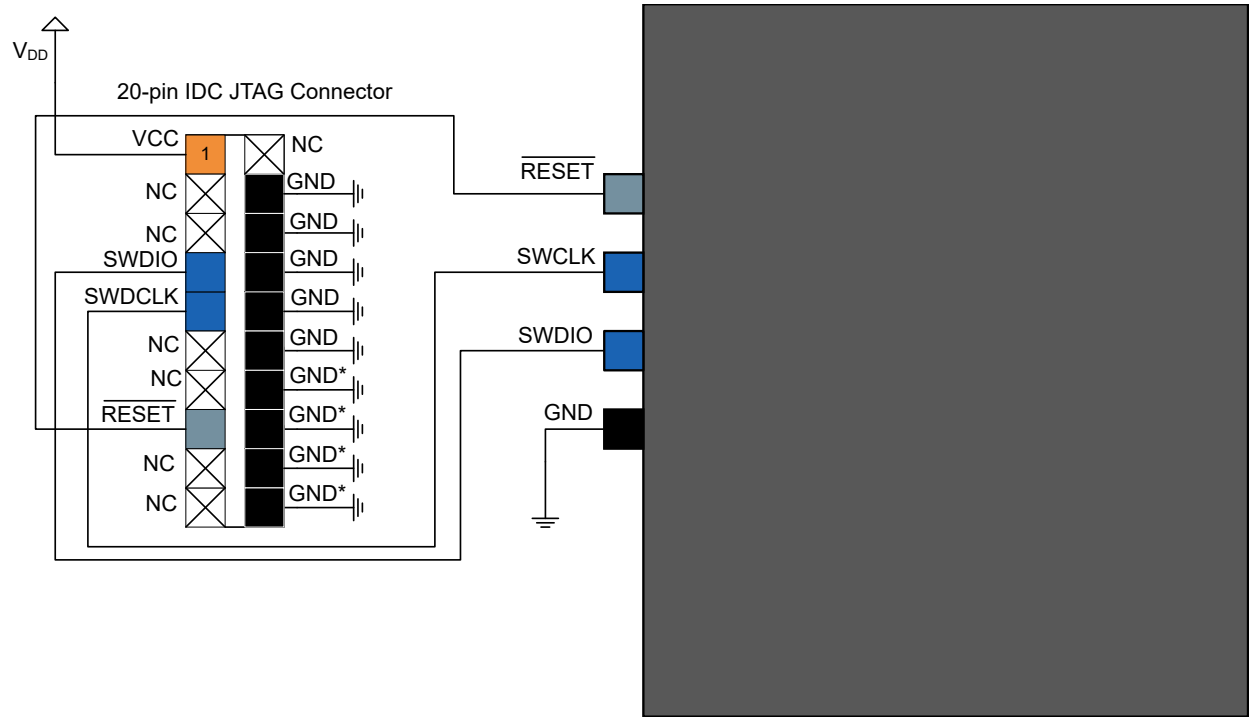


Table 50-9. 20-pin IDC JTAG Connector

Header Signal Name	Description
SWDCLK	Serial wire clock pin
SWDIO	Serial wire bidirectional data pin
$\overline{\text{RESET}}$	Target device reset pin, active low
VCC	Target voltage sense, should be connected to the device V_{DD}
GND	Ground
GND*	These pins are reserved for firmware extension purposes. They can be left unconnected or connected to GND in normal debug environment. They are not essential for SWD in general.

50.8 Peripherals Considerations

ADC Accuracy

The ADC accuracy may depend on different parameters, such as its input sources, as well as its conversion speed.

Please refer to the *Analog-to-Digital Converter (ADC) Characteristics* section in the *Electrical Characteristics* chapters for more details.

Downloaded from Arrow.com.

51. Conventions

51.1 Numerical Notation

Table 51-1. Numerical Notation

Symbol	Description
165	Decimal number
0b0101	Binary number (example 0b0101 = 5 decimal)
'0101'	Binary numbers are given without prefix if unambiguous.
0x3B24	Hexadecimal number
X	Represents an unknown or don't care value
Z	Represents a high-impedance (floating) state for either a signal or a bus

51.2 Memory Size and Type

Table 51-2. Memory Size and Bit Rate

Symbol	Description
KB (kbyte)	kilobyte (2^{10} = 1024)
MB (Mbyte)	megabyte (2^{20} = 1024*1024)
GB (Gbyte)	gigabyte (2^{30} = 1024*1024*1024)
b	bit (binary '0' or '1')
B	byte (8 bits)
1kbit/s	1,000 bit/s rate (not 1,024 bit/s)
1Mbit/s	1,000,000 bit/s rate
1Gbit/s	1,000,000,000 bit/s rate
word	32 bit
half-word	16 bit

51.3 Frequency and Time

Table 51-3. Frequency and Time

Symbol	Description
kHz	1kHz = 10^3Hz = 1,000Hz
KHz	1KHz = 1,024Hz, 32KHz = 32,768Hz

Downloaded from Arrow.com.

Symbol	Description
MHz	10^6 = 1,000,000Hz
GHz	10^9 = 1,000,000,000Hz
s	second
ms	millisecond
µs	microsecond
ns	nanosecond

51.4 Registers and Bits

Table 51-4. Register and Bit Mnemonics

Symbol	Description
R/W	Read/Write accessible register bit. The user can read from and write to this bit.
R	Read-only accessible register bit. The user can only read this bit. Writes will be ignored.
W	Write-only accessible register bit. The user can only write this bit. Reading this bit will return an undefined value.
BIT	Bit names are shown in uppercase. (Example ENABLE)
FIELD[n:m]	A set of bits from bit n down to m. (Example: PINA[3:0] = {PINA3, PINA2, PINA1, PINA0}
Reserved	Reserved bits are unused and reserved for future use. For compatibility with future devices, always write reserved bits to zero when the register is written. Reserved bits will always return zero when read. Reserved bit field values must not be written to a bit field. A reserved value won't be read from a read-only bit field.
PERIPHERAL*i*	If several instances of a peripheral exist, the peripheral name is followed by a number to indicate the number of the instance in the range 0-n. PERIPHERAL0 denotes one specific instance.
Reset	Value of a register after a power Reset. This is also the value of registers in a peripheral after performing a software Reset of the peripheral, except for the Debug Control registers.
SET/CLR	Registers with SET/CLR suffix allows the user to clear and set bits in a register without doing a read-modify-write operation. These registers always come in pairs. Writing a one to a bit in the CLR register will clear the corresponding bit in both registers, while writing a one to a bit in the SET register will set the corresponding bit in both registers. Both registers will return the same value when read. If both registers are written simultaneously, the write to the CLR register will take precedence.

Downloaded from Arrow.com.

52. Acronyms and Abbreviations

The below table contains acronyms and abbreviations used in this document.

Table 52-1. Acronyms and Abbreviations

Abbreviation	Description
AC	Analog Comparator
ADC	Analog-to-Digital Converter
ADDR	Address
AES	Advanced Encryption Standard
AHB	Advanced High-performance Bus
AMBA	Advanced Microcontroller Bus Architecture
APB	AMBA Advanced Peripheral Bus
AREF	Analog Reference Voltage
BOD	Brown-out Detector
CAL	Calibration
CC	Compare/Capture
CCL	Configurable Custom Logic
CLK	Clock
CRC	Cyclic Redundancy Check
CTRL	Control
DAC	Digital-to-Analog Converter
DAP	Debug Access Port
DFLL	Digital Frequency Locked Loop
DPLL	Digital Phase Locked Loop
DMAC	DMA (Direct Memory Access) Controller
DSU	Device Service Unit
EEPROM	Electrically Erasable Programmable Read-Only Memory
EIC	External Interrupt Controller
EVSYS	Event System
FDPLL	Fractional Digital Phase Locked Loop, also DPLL
FREQM	Frequency Meter
GCLK	Generic Clock Controller
GND	Ground

Downloaded from Arrow.com.

Abbreviation	Description
GPIO	General Purpose Input/Output
I^2C	Inter-Integrated Circuit
IF	Interrupt Flag
INT	Interrupt
MBIST	Memory Built-In Self-Test
MEM-AP	Memory Access Port
MTB	Micro Trace Buffer
NMI	Non-maskable Interrupt
NVIC	Nested Vector Interrupt Controller
NVM	Nonvolatile Memory
NVMCTRL	Nonvolatile Memory Controller
OPAMP	Operation Amplifier
OSC	Oscillator
PAC	Peripheral Access Controller
PC	Program Counter
PER	Period
PM	Power Manager
POR	Power-on Reset
PORT	I/O Pin Controller
PTC	Peripheral Touch Controller
PWM	Pulse-Width Modulation
RAM	Random-Access Memory
REF	Reference
RTC	Real-Time Counter
RX	Receiver/Receive
SEEP	SmartEEPROM Page
SERCOM	Serial Communication Interface
SMBus	System Management Bus
SP	Stack Pointer
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
SUPC	Supply Controller

Downloaded from Arrow.com.

Abbreviation	Description
SWD	Serial Wire Debug
TC	Timer/Counter
TRNG	True Random Number Generator
TX	Transmitter/Transmit
ULP	Ultra Low-Power
USART	Universal Synchronous and Asynchronous Serial Receiver and Transmitter
V_{DD}	Common voltage to be applied to VDDIO and VDDANA
V_{DDIO}	Digital Supply Voltage
V_{DDANA}	Analog Supply Voltage
VREF	Voltage Reference
WDT	Watchdog Timer
XOSC	Crystal Oscillator

Downloaded from Arrow.com.

53. Datasheet Revision History

Note: The datasheet revision is independent of the die revision (Revision bit in the Device Identification register of the Device Service Unit, DSU.DID.REVISION) and the device variant (last letter of the ordering number).

53.1 Rev A - 09/2017

This is the initial released version of the document.

53.2 Rev B - 6/2018

Added new documentation for Electrical Characteristics -125°C.

Downloaded from Arrow.com.

The Microchip Web Site

Microchip provides online support via our web site at http://www.microchip.com/. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Customer Change Notification Service

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at http://www.microchip.com/. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

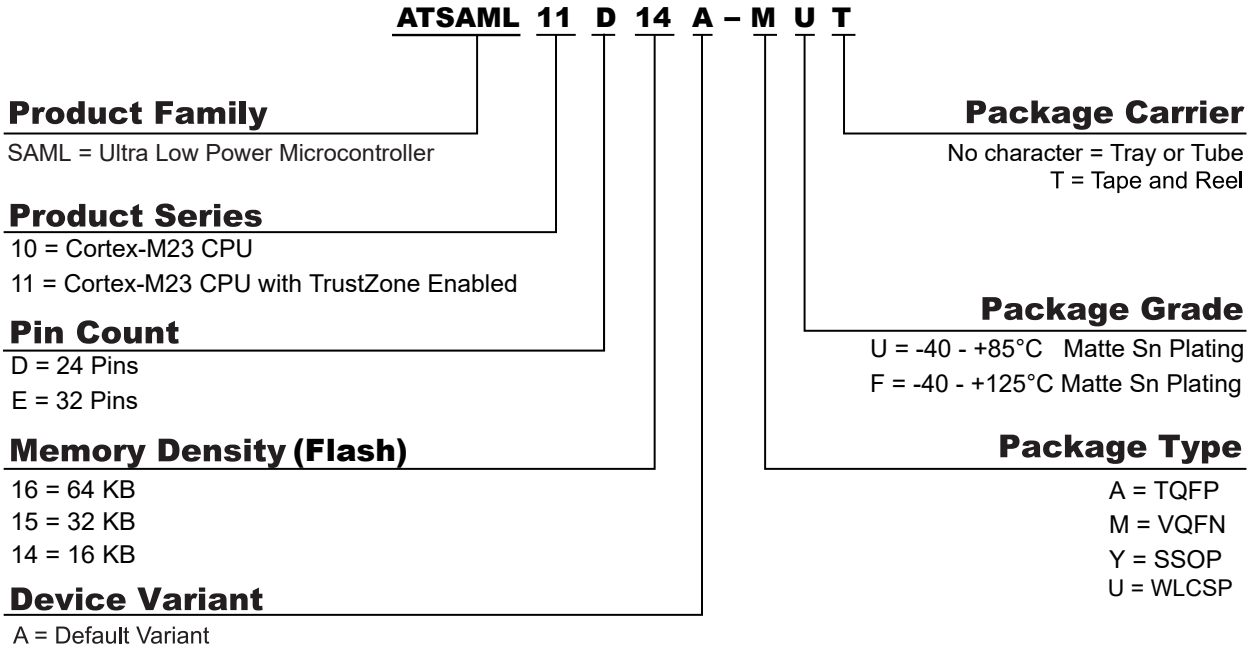
Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://www.microchip.com/support

Downloaded from Arrow.com.

Product Identification System

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.



Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR

Downloaded from Arrow.com.

WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, AnyRate, AVR, AVR logo, AVR Freaks, BeaconThings, BitCloud, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, Heldo, JukeBlox, KeeLoq, KeeLoq logo, Kleer, LANCheck, LINK MD, maXStylus, maXTouch, MediaLB, megaAVR, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, Prochip Designer, QTouch, RightTouch, SAM-BA, SpyNIC, SST, SST Logo, SuperFlash, tinyAVR, UNI/O, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, EtherSynch, Hyper Speed Control, HyperLight Load, IntelliMOS, mTouch, Precision Edge, and Quiet-Wire are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BodyCom, chipKIT, chipKIT logo, CodeGuard, CryptoAuthentication, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, JitterBlocker, KleerNet, KleerNet logo, Mindi, MiWi, motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PureSilicon, QMatrix, RightTouch logo, REAL ICE, Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2017, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-5224-3156-5

Quality Management System Certified by DNV

ISO/TS 16949

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and

Downloaded from Arrow.com.

analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Downloaded from Arrow.com.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: http://www.microchip.com/support Web Address: www.microchip.com **Atlanta** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **Austin, TX** Tel: 512-257-3370 **Boston** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **Chicago** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **Detroit** Novi, MI Tel: 248-848-4000 **Houston, TX** Tel: 281-894-5983 **Indianapolis** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **Los Angeles** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **Raleigh, NC** Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000 **San Jose, CA** Tel: 408-735-9110 Tel: 408-436-4270 **Canada - Toronto** Tel: 905-695-1980 Fax: 905-695-2078	**Asia Pacific Office** Suites 3707-14, 37th Floor Tower 6, The Gateway Harbour City, Kowloon **Hong Kong** Tel: 852-2943-5100 Fax: 852-2401-3431 **Australia - Sydney** Tel: 61-2-9868-6733 Fax: 61-2-9868-6755 **China - Beijing** Tel: 86-10-8569-7000 Fax: 86-10-8528-2104 **China - Chengdu** Tel: 86-28-8665-5511 Fax: 86-28-8665-7889 **China - Chongqing** Tel: 86-23-8980-9588 Fax: 86-23-8980-9500 **China - Dongguan** Tel: 86-769-8702-9880 **China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115 Fax: 86-571-8792-8116 **China - Hong Kong SAR** Tel: 852-2943-5100 Fax: 852-2401-3431 **China - Nanjing** Tel: 86-25-8473-2460 Fax: 86-25-8473-2470 **China - Qingdao** Tel: 86-532-8502-7355 Fax: 86-532-8502-7205 **China - Shanghai** Tel: 86-21-3326-8000 Fax: 86-21-3326-8021 **China - Shenyang** Tel: 86-24-2334-2829 Fax: 86-24-2334-2393 **China - Shenzhen** Tel: 86-755-8864-2200 Fax: 86-755-8203-1760 **China - Wuhan** Tel: 86-27-5980-5300 Fax: 86-27-5980-5118 **China - Xian** Tel: 86-29-8833-7252 Fax: 86-29-8833-7256	**China - Xiamen** Tel: 86-592-2388138 Fax: 86-592-2388130 **China - Zhuhai** Tel: 86-756-3210040 Fax: 86-756-3210049 **India - Bangalore** Tel: 91-80-3090-4444 Fax: 91-80-3090-4123 **India - New Delhi** Tel: 91-11-4160-8631 Fax: 91-11-4160-8632 **India - Pune** Tel: 91-20-3019-1500 **Japan - Osaka** Tel: 81-6-6152-7160 Fax: 81-6-6152-9310 **Japan - Tokyo** Tel: 81-3-6880- 3770 Fax: 81-3-6880-3771 **Korea - Daegu** Tel: 82-53-744-4301 Fax: 82-53-744-4302 **Korea - Seoul** Tel: 82-2-554-7200 Fax: 82-2-558-5932 or 82-2-558-5934 **Malaysia - Kuala Lumpur** Tel: 60-3-6201-9857 Fax: 60-3-6201-9859 **Malaysia - Penang** Tel: 60-4-227-8870 Fax: 60-4-227-4068 **Philippines - Manila** Tel: 63-2-634-9065 Fax: 63-2-634-9069 **Singapore** Tel: 65-6334-8870 Fax: 65-6334-8850 **Taiwan - Hsin Chu** Tel: 886-3-5778-366 Fax: 886-3-5770-955 **Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600 Fax: 886-2-2508-0102 **Thailand - Bangkok** Tel: 66-2-694-1351 Fax: 66-2-694-1350	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **Denmark - Copenhagen** Tel: 45-4450-2828 Fax: 45-4485-2829 **Finland - Espoo** Tel: 358-9-4520-820 **France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **France - Saint Cloud** Tel: 33-1-30-60-70-00 **Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400 **Germany - Heilbronn** Tel: 49-7131-67-3636 **Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Ra'anana** Tel: 972-9-744-7705 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781 **Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340 **Norway - Trondheim** Tel: 47-7289-7561 **Poland - Warsaw** Tel: 48-22-3325737 **Romania - Bucharest** Tel: 40-21-407-87-50 **Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654 **UK - Wokingham** Tel: 44-118-921-5800 Fax: 44-118-921-5820

Downloaded from Arrow.com.