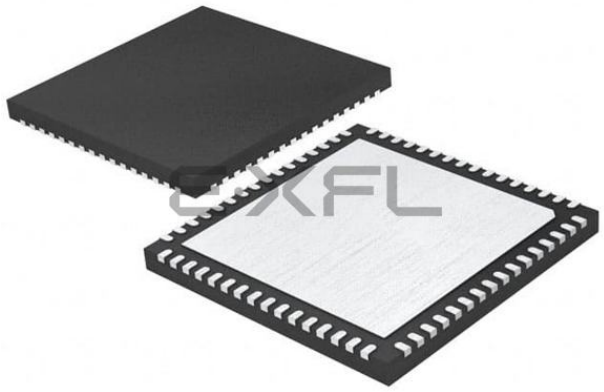


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

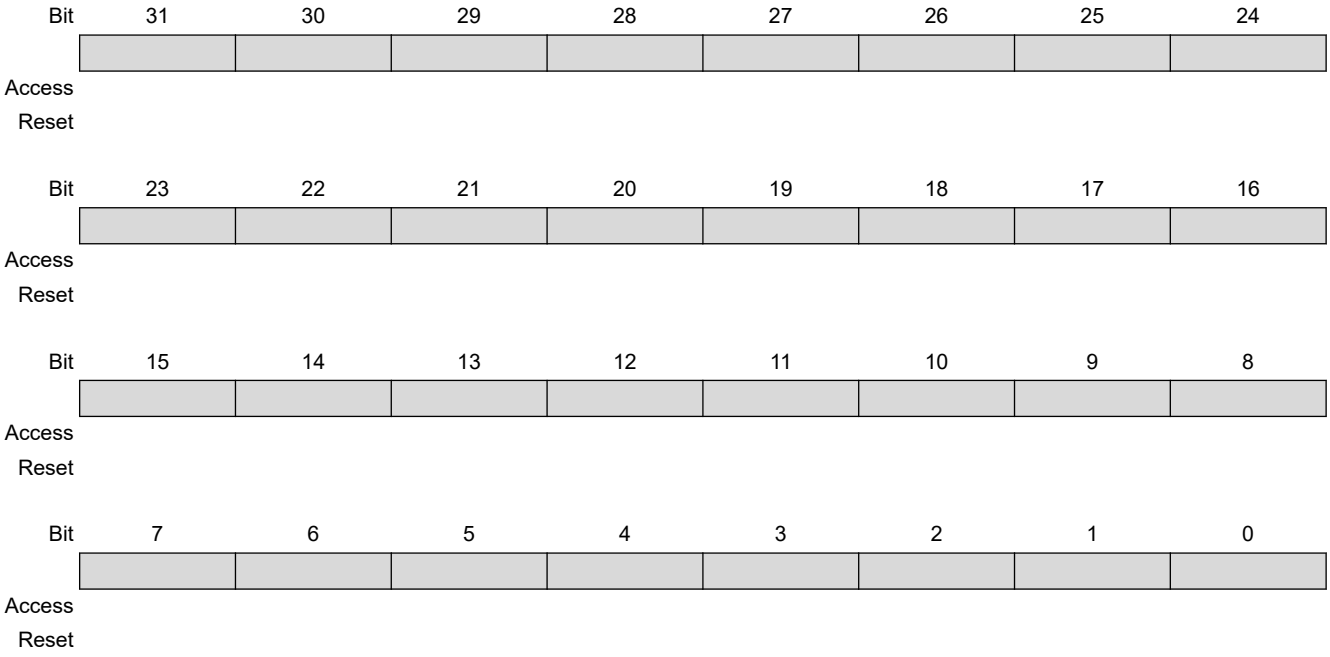
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	ARM® Cortex®-M4F
Core Size	32-Bit Single-Core
Speed	120MHz
Connectivity	CANbus, EBI/EMI, I²C, IrDA, LINbus, MMC/SD, QSPI, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM
Number of I/O	51
Program Memory Size	1MB (1M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256K x 8
Voltage - Supply (Vcc/Vdd)	1.71V ~ 3.63V
Data Converters	A/D 24x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-VFQFN Exposed Pad
Supplier Device Package	64-VQFN (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsame51j20a-mu

12.13.17 Peripheral Identification 5

Name: PID5
Offset: 0x1FD4
Reset: 0x00000000
Property: Read-Only



Offset	Name	Bit Pos.								
		31:24								
0x9C	PCHCTRL7	7:0	WRTLOCK	CHEN			GEN[3:0]			
		15:8								
		23:16								
		31:24								
0xA0	PCHCTRL8	7:0	WRTLOCK	CHEN			GEN[3:0]			
		15:8								
		23:16								
		31:24								
0xA4	PCHCTRL9	7:0	WRTLOCK	CHEN			GEN[3:0]			
		15:8								
		23:16								
		31:24								
0xA8	PCHCTRL10	7:0	WRTLOCK	CHEN			GEN[3:0]			
		15:8								
		23:16								
		31:24								
0xAC	PCHCTRL11	7:0	WRTLOCK	CHEN			GEN[3:0]			
		15:8								
		23:16								
		31:24								
0xB0	PCHCTRL12	7:0	WRTLOCK	CHEN			GEN[3:0]			
		15:8								
		23:16								
		31:24								
0xB4	PCHCTRL13	7:0	WRTLOCK	CHEN			GEN[3:0]			
		15:8								
		23:16								
		31:24								
0xB8	PCHCTRL14	7:0	WRTLOCK	CHEN			GEN[3:0]			
		15:8								
		23:16								
		31:24								
0xBC	PCHCTRL15	7:0	WRTLOCK	CHEN			GEN[3:0]			
		15:8								
		23:16								
		31:24								
0xC0	PCHCTRL16	7:0	WRTLOCK	CHEN			GEN[3:0]			
		15:8								
		23:16								
		31:24								
0xC4	PCHCTRL17	7:0	WRTLOCK	CHEN			GEN[3:0]			
		15:8								
		23:16								
		31:24								
0xC8	PCHCTRL18	7:0	WRTLOCK	CHEN			GEN[3:0]			

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.

20.8.8 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.

Offset	Name	Bit Pos.								
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]		BURSTLEN[3:0]			
0x01D4	CHCTRLB25	7:0							CMD[1:0]	
0x01D5	CHPRILVL25	7:0							PRILVL[1:0]	
0x01D6	CHEVCTRL25	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0x01D7 ... 0x01DB	Reserved									
0x01DC	CHINTENCLR25	7:0						SUSP	TCMPL	TERR
0x01DD	CHINTENSET25	7:0						SUSP	TCMPL	TERR
0x01DE	CHINTFLAG25	7:0						SUSP	TCMPL	TERR
0x01DF	CHSTATUS25	7:0					CRCERR	FERR	BUSY	PEND
0x01E0	CHCTRLA26	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]		BURSTLEN[3:0]			
0x01E4	CHCTRLB26	7:0							CMD[1:0]	
0x01E5	CHPRILVL26	7:0							PRILVL[1:0]	
0x01E6	CHEVCTRL26	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0x01E7 ... 0x01EB	Reserved									
0x01EC	CHINTENCLR26	7:0						SUSP	TCMPL	TERR
0x01ED	CHINTENSET26	7:0						SUSP	TCMPL	TERR
0x01EE	CHINTFLAG26	7:0						SUSP	TCMPL	TERR
0x01EF	CHSTATUS26	7:0					CRCERR	FERR	BUSY	PEND
0x01F0	CHCTRLA27	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]		BURSTLEN[3:0]			
0x01F4	CHCTRLB27	7:0							CMD[1:0]	
0x01F5	CHPRILVL27	7:0							PRILVL[1:0]	
0x01F6	CHEVCTRL27	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0x01F7 ... 0x01FB	Reserved									
0x01FC	CHINTENCLR27	7:0						SUSP	TCMPL	TERR
0x01FD	CHINTENSET27	7:0						SUSP	TCMPL	TERR
0x01FE	CHINTFLAG27	7:0						SUSP	TCMPL	TERR
0x01FF	CHSTATUS27	7:0					CRCERR	FERR	BUSY	PEND
0x0200	CHCTRLA28	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]		BURSTLEN[3:0]			
0x0204	CHCTRLB28	7:0							CMD[1:0]	

24.9.37 GMAC PTP Peer Event Frame Received Seconds High Register

Name: PEFRSH
Offset: 0x0F4
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
	RUD[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
	RUD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RUD[15:0] Register Update
The register is updated with the value that the 1588 timer seconds register held when the SFD of a PTP transmit peer event crosses the MII interface. An interrupt is issued when the register is updated.

The lower blocks in the NVM main address space can be allocated as a boot loader section by using the BOOTPROT fuses, and the upper rows can be allocated to EEPROM.

The NVM memory is separated into six parts:

1. CB space
   Contains factory calibration and system configuration information.
   - Address; 0x00800000
   - Size: 1 page
   - Property: Read-Only
2. FS space
   Contains the factory signature information.
   - Address; 0x00806000
   - Size: 4 pages
   - Property: Read-Only.
3. USER space
   Contains user defined startup configuration. The first word is reserved, and used during the NVMCTRL start-up to automatically configure the device.
   - Address: 0x00804000
   - Size: 1 page
   - Property: Read-Write
4. Main address space
   The main address space is divided into 32 equally sized regions. Each region can be protected against write or erase operation. The 32-bit RUNLOCK register reflects the protection of each region. This register is automatically updated after power-up with the region lock user fuse data; To lock or unlock a region, the LR or UR commmands can be issued.
   - Address: 0x00000000
   - Size: PARAM.NVMP pages.
   - Property: Read-Write
5. Bootloader space
   The bootloader section starts at the beginning of the main address space; Its size is defined by the BOOTPROT[3:0] fuse. It is protected against write or erase operations, except if STATUS.BPDIS is set. Issuing a write or erase command at an address inside the BOOTPROT section sets STATUS.PROGE and STATUS.LOCKE. STATUS.BPDIS can be set by issuing the Set BOOTPROT Disable command (SBPDIS). It is cleared by issuing the Clear BOOTPROT Disable command (CBPDIS). This allows to program an new bootloader without changing the user page and issuing a new NVMCTRL startup sequence to reload the user configuration. The BOOTPROT section is not erased during a Chip-Erase operation even if STATUS.BPDIS is high.
   - Address: 0x00000000
   - Size: (15 - STATUS.BOOTPROT) × 8192
   - Property: Read-Only.
6. SmartEEPROM raw data space
   The SmartEEPROM algorithm emulates an EEPROM with a portion of the NVM main. Smart-EEPROM raw data is mapped at the end of the main address space. SmartEEPROM allocated space in the main address space is not accessible from AHB0/1. Any AHB access throws a

26. ICM - Integrity Check Monitor

26.1 Overview

The Integrity Check Monitor (ICM) is a DMA controller that performs hash calculation over multiple memory regions through the use of transfer descriptors located in memory (ICM Descriptor Area). The Hash function is based on the Secure Hash Algorithm (SHA). The ICM controller integrates two modes of operation. The first one is used to hash a list of memory regions and save the digests to memory (ICM Hash Area). The second operation mode is an active monitoring of the memory. In that mode, the hash function is evaluated and compared to the digest located at a predefined memory address (ICM Hash Area). If a mismatch occurs, an interrupt is raised.

26.2 Features

- DMA AHB master interface
- Supports monitoring of up to four non-contiguous memory regions
- Supports block gathering through the use of linked list
- Supports Secure Hash Algorithm (SHA1, SHA224, SHA256)
- Compliant with FIPS Publication 180-2
- Configurable processing period:
  - When SHA1 algorithm is processed, the run-time period is either 85 or 209 clock cycles.
  - When SHA256 or SHA224 algorithm is processed, the run-time period is either 72 or 194 clock cycles.
- Programmable bus burden

28.8.10 DFLL48M Multiplier

Name: DFLLMUL
Offset: 0x28
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
	CSTEP[5:0]							
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
	FSTEP[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
	MUL[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
	MUL[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 31:26 – CSTEP[5:0] Coarse Maximum Step
This bit group indicates the maximum step size allowed during coarse adjustment in closed-loop mode. When adjusting to a new frequency, the expected output frequency overshoot depends on this step size.

Bits 23:16 – FSTEP[7:0] Fine Maximum Step
This bit group indicates the maximum step size allowed during fine adjustment in closed-loop mode. When adjusting to a new frequency, the expected output frequency overshoot depends on this step size.

Bits 15:0 – MUL[15:0] DFLL Multiply Factor
This field determines the ratio of the CLK_DFLL output frequency to the CLK_DFLL_REF input frequency. Writing to the MUL bits will cause locks to be lost and the fine calibration value to be reset to its midpoint.

29.8.3 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
						XOSC32KFAIL		XOSC32KRDY
Access						R/W		R/W
Reset						0		0

Bit 2 – XOSC32KFAIL XOSC32K Clock Failure Detector
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.XOSC32KFAIL) and will generate an interrupt request if INTENSET.XOSC32KFAIL 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.

36.10.11 Data

Name: DATA
Offset: 0x28
Reset: 0x00000000
Property: Write-Synchronized, Read-Synchronized

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
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).

When CTRLC.DATA32B=1, read and write transactions from/to the DATA register are 32 bit in size. Otherwise, reads and writes are 8 bit.

Writing or reading DATA.DATA when not in smart mode does not require synchronization.

37.8.8 Interrupt Flag Status and Clear

Name: INTFLAG
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
						INSTREND		CSRISE
Access						R/W		R/W
Reset						0		0

Bit	7	6	5	4	3	2	1	0
					ERROR	TXC	DRE	RXC
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 10 – INSTREND Instruction End
This bit is set when an Instruction End has been detected.

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

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

Bit 8 – CSRISE Chip Select Rise
The bit is set when a Chip Select Rise has been detected.

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

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

Bit 3 – ERROR Overrun Error
This bit is set when an ERROR has occurred.

An ERROR occurs when RXDATA is loaded at least twice from the serializer.

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

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

Bit 2 – TXC Transmission Complete
0: As soon as data is written in TXDATA.

42.8.2 Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
					GFMUL	EOM	NEWMSG	START
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 3 – GFMUL GF Multiplication
This bit is applicable only to GCM mode.

Value	Description
0	No action
1	Setting this bit calculates GF multiplication with data buffer content and hashkey register content.

Bit 2 – EOM End of Message
This bit is applicable only to GCM mode.

Value	Description
0	No action
1	Setting this bit generates final GHASH value for the message.

Bit 1 – NEWMSG New Message
This bit is used in cipher block chaining (CBC), cipher feedback (CFB) and output feedback (OFB), counter (CTR) modes to indicate the hardware to use Initialization vector for encrypting the first block of message.

Value	Description
0	No action
1	Setting this bit indicates start of new message to the module.

Bit 0 – START Start Encryption/Decryption

Value	Description
0	No action
1	Start encryption / decryption in manual mode.

- Operation: Fast ModularReduction.
  `PUKCL(u2Options) = PUKCL_REDMOD_REDUCTION | PUKCL_REDMOD_USING_FASTRED;`

For this command three exclusive options can be specified. The following table lists the operations that can be performed.

Table 43-47. RedMod Service Options

Option	Purpose	Required Parameters
PUKCL_REDMOD_SETUP	Perform the Cns value computation	nu1ModBase, u2ModLength, nu1CnsBase, nu1XBase
PUKCL_REDMOD_REDUCTION	Perform R ≡ X Mod N, see sub-option for details	nu1ModBase, u2ModLength, nu1CndBase, nu1XBase, nu1RBase
PUKCL_REDMOD_NORMALIZE	Perform R = X Mod N	nu1ModBase, u2ModLength, nu1CndBase, nu1XBase, nu1RBase

When selecting the PUKCL_REDMOD_REDUCTION option, one of the two sub-options listed in the following table must be selected.

Table 43-48. RedMode Service Options with PUKCL_RED_MOD_REDUCTION

Option	Purpose	Required Parameters
PUKCL_REDMOD _USING_DIVISION	Perform R = X Mod N	nu1ModBase, u2ModLength, nu1CndBase, nu1XBase
PUKCL_REDMOD _USING_FASTRED	Perform R ≡ X Mod N The entropy is minimized (~2 bits)	nu1ModBase, u2ModLength, nu1CndBase, nu1XBase, nu1RBase

43.3.5.1.11 Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;

PUKCL(Specific).CarryIn = 0;
PUKCL(Specific).GF2n = ...;

PUKCL(u2Option) =...;

// Depending on the option specified, not all fields should be filled
PUKCL_RedMod(nu1ModBase) = <Base of the ram location of N>;
PUKCL_RedMod(u2ModLength) = <Length of N>;
PUKCL_RedMod(nu1CnsBase) = <Base of the ram location of Cns>;
...

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(RedMod,pvPUKCLParam);
if (PUKCL_Param.Status == PUKCL_OK)
            {
            // operation has correctly been performed
            ...
            }
else // Manage the error
```

43.3.5.1.12 Constraints

Depending on the options chosen the lengths of the R area and Cns area differ:

Parameter	Type	Direction	Location	Data Length	Before Executing the Service	After Executing the Service
pfu1ExpBase (see **Note 2**)	pfu1	I	Any place (see **Note 3**)	u2ExpLength + 4	Base of the Exponent	Base of the Exponent untouched
u2ExpLength (see **Note 4**)	u2	I	–	–	Significant length of Exponent	Significant length of Exponent
u1Blinding (see **Note 5**)	u1	I	–	–	Exponent unblinding value	Exponent unblinding value untouched

Note:

1. This zone contains the number to be exponentiated (u2ModLength bytes) and is used during the computations as a workspace (four 32-bit words longer than the number to be exponentiated). At the end of the computation, it contains the correct result of the operation.
2. The exponent must be given with a supplemental word on the LSB side (low addresses). This word shall be set to zero.
3. If the PUKCL_EXPMOD_EXPINPUKCCRAM option is not set, the location of the exponent MUST NOT be the Crypto RAM, even partially.
4. The u2ExpLength parameter does not take into account the supplemental word needed on the LSB side of the exponent.
5. It is possible to mask the exponent in memory using an 8-bits XOR mask value. Be aware that not only the exponent, but also the supplemental word has to be masked. If masking is not desired, then this parameter should be set to 0.

43.3.5.2.5 Options

The options are set by the u2Options input parameter, which is composed of:

- the mandatory Calculus Mode Option described in Table 43-51
- the mandatory Window Size Option described in Table 43-52
- the indication of the presence of the exponent in Crypto RAM

Note: Please check precisely if one part of the exponent is in Crypto RAM. If this is the case the PUKCL_EXPMOD_EXPINPUKCCRAM must be used.

The u2Options number is calculated by an “Inclusive OR” of the options. Some examples in C language are:

- Operation:Fast Modular Exponentiation with the window size equal to 1 and with no part of the Exponent in the Crypto RAM
  ```
  PUKCL(u2Options) = PUKCL_EXPMOD_FASTRSA | PUKCL_EXPMOD_WINDOWSIZE_1;
  ```
- Operation: Regular Modular Exponentiation with the window size equal to 2 and with one part of the Exponent in the Crypto RAM
  ```
  PUKCL(u2Options) = PUKCL_EXPMOD_REGULARRSA | PUKCL_EXPMOD_WINDOWSIZE_2 |
  PUKCL_EXPMOD_EXPINPUKCCRAM;
  ```

- All overlapping between {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength +8}, {nu1PointABase, 3*u2ModLength + 12}, {nu1ABase, u2ModLength + 4}, {nu1ScalarNumber, u2ScalarLength} and {nu1Workspace, 8*u2ModLength + 44}

43.3.6.5.7 Status Returned Values

Returned Status	Importance	Meaning
PUKCL_OK	–	The computation passed without problem.

43.3.6.6 Quick Dual Multiplying by Two Scalar Numbers and Two Points

43.3.6.6.1 Purpose

This service is used to multiply two points by two integral constants K1 and K2, and then provide the addition of these multiplications results.

Important: This service has a quick implementation without additional security.

43.3.6.6.2 How to Use the Service

43.3.6.6.3 Description

This service processes the dual Multiplying by two scalar numbers:

$PtC = K_1 \times Pt_A + K_2 \times Pt_B$

In this computation, the following parameters need to be provided:

- A the first input point is filled in projective coordinates (X,Y,Z) (pointed by {pu1PointABase, (3*(u2ModLength + 4)) * (2(WA-2))}). This point can be the Infinite Point.
- B the 2nd input point is filled in projective coordinates (X,Y,Z) (pointed by {pu1PointBBase, (3*(u2ModLength + 4)) * (2(WB-2))}). This point can be the Infinite Point.
- P the modulus filled and Cns the Fast Modular Constant filled (pointed by {pu1ModCnsBase, 2*u2ModLength + 16})
- The a parameter filled and the workspace not initialized (pointed by {pu1AWorkBase, 9*u2ModLength +48}
- KAB the scalar numbers (pointed by {pu1KABBase, 2*u2KLength +8})
- The options are set by the u2Options input parameter, which is composed of:
  - wA: Size of window for Point A between 2 and15
  - wB: Size of window for Point B between 2 and15
  - PUKCL_ZPECCMUL_SCAL_IN_CLASSIC_RAM flag: to set only if the scalars are entirely in Classic RAM with no part in PUKCC RAM

The resulting C point is represented in projective coordinates (X,Y,Z) and is stored at (pu1AWorkBase + u2ModLength + 4). This point can be the Infinite Point.

Important: Before using this service, ensure that the constant Cns has been calculated with the setup of the Fast Modular Reduction service.

45. ADC – Analog-to-Digital Converter

45.1 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 bandgap voltage, 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.

The SAM D5x/E5x has two ADC instances, ADC0 and ADC1. The two inputs can be sampled simultaneously, as each ADC includes sample and hold circuits.

Note: When the Peripheral Touch Controller (PTC) is enabled, ADC0 is serving the PTC exclusively. In this case, ADC0 cannot be used by the user application.

45.2 Features

- Two Analog to Digital Converters (ADC) ADC0 and ADC1
- 8-, 10- or 12-bit resolution
- Up to 1,000,000 samples per second (1MSPS)
- Differential and single-ended inputs
  - Up to 32 analog inputs per ADC (20 unique channels total)
    32 positive and 10 negative, including internal and external
- Internal inputs:
  - Internal temperature sensor
  - Bandgap voltage
  - Scaled core supply
  - Scaled I/O supply
  - Scaled VBAT supply
  - DAC
- Single, continuous and sequencing options
- Windowing monitor with selectable channel
- Conversion range: V_{ref} = [1.0V to VDD_{ANA}]

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.

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.

49.8.5 Fault Control A and B

Name: FCTRL
Offset: 0x0C + n*0x04 [n=0..1]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
					FILTERVAL[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BLANKVAL[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
	BLANKPRESC	CAPTURE[2:0]			CHSEL[1:0]		HALT[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

Bit	7	6	5	4	3	2	1	0
	RESTART	BLANK[1:0]		QUAL	KEEP		SRC[1:0]	
Access	R/W	R/W	R/W	R/W	R/W		R/W	R/W
Reset	0	0	0	0	0		0	0

Bits 27:24 – FILTERVAL[3:0] Recoverable Fault n Filter Value
These bits define the filter value applied on MCEx (x=0,1) event input line. The value must be set to zero when MCEx event is used as synchronous event.

Bits 23:16 – BLANKVAL[7:0] Recoverable Fault n Blanking Value
These bits determine the duration of the blanking of the fault input source. Activation and edge selection of the blank filtering are done by the BLANK bits (FCTRLn.BLANK).

When enabled, the fault input source is internally disabled for BLANKVAL* prescaled GCLK_TCC periods after the detection of the waveform edge.

Bit 15 – BLANKPRESC Recoverable Fault n Blanking Value Prescaler
This bit enables a factor 64 prescaler factor on used as base frequency of the BLANKVAL value.

Value	Description
0	Blank time is BLANKVAL* prescaled GCLK_TCC.
1	Blank time is BLANKVAL* 64 * prescaled GCLK_TCC.

Bits 14:12 – CAPTURE[2:0] Recoverable Fault n Capture Action
These bits select the capture and Fault n interrupt/event conditions.

$$f_{MCKn} = \frac{8 \cdot (SLOTSIZE+1) \cdot (NBSLOTS+1) \cdot (MCKDIV+1)}{MCKOUTDIV+1}$$

If a Master Clock output is not required, the GCLK_I2S generic clock can be configured as SCKn by writing a '0'to CLKCTRLn.MCKDIV. Alternatively, if the frequency of the generic clock is a multiple of the required SCKn frequency, the MCKn-to-SCKn divider can be used with the ratio defined by writing the CLKCTRLn.MCKDIV field.

The FSn pin is used as Word Select in I²S format and as Frame Synchronization in TDM format, as described in 51.6.4 I2S Format - Reception and Transmission Sequence with Word Select and 51.6.5 TDM Format - Reception and Transmission Sequence, respectively.

51.6.2.2 Data Holding Registers

For both the Transmit and the Receive Serializer, the I²S user interface includes a Data register (TXDATA and RXDATA, respectively). They are used to access data samples for all data slots.

51.6.2.2.1 Data Reception Mode

In receiver mode, the RXDATA register stores the received data.

When a new data word is available in the RXDATA register, the Receive Ready bit (RXRDYm) in the Interrupt Flag Status and Clear register (INTFLAG) is set. Reading the RXDATA register will clear this bit.

A receive overrun condition occurs if a new data word becomes available before the previous data word has been read from the RXDATA register. Then, the Receive Overrun bit in INTFLAG will be set (INTFLAG.RXORm). This interrupt can be cleared by writing a '1' to it.

51.6.2.2.2 Data Transmission Mode

In Transmitter mode, the TXDATA register contains the data to be transmitted.

when TXDATA is empty, the Transmit Ready bit in the Interrupt Flag Status and Clear register is set (INTFLAG.TXRDYm). Writing to TXDATA will clear this bit.

A transmit underrun condition occurs if data present in TXDATA is sent and no new data is written to TXDATA register before the next time slot. Then, the Transmit Underrun bit in INTFLAG will be set (INTFLAG.TXURm). This interrupt can be cleared by writing a '1' to it. The Transmit Data when Underrun bit in the Tx Serializer Control register (TXCTRL.TXSAME) configures whether a zero data word is transmitted in case of underrun (TXCTRL.TXSAME=0), or the previous data word for the current transmit slot number is transmitted again (TXCTRL.TXSAME=1).

51.6.3 Master, Controller, and Slave Modes

In Master and Controller modes, the I²S provides the Serial Clock, a Word Select/Frame Sync signal and optionally a Master Clock.

In Controller mode, the I²S Serializers are disabled. Only the clocks are enabled and output for external receivers and/or transmitters.

In Slave mode, the I²S receives the Serial Clock and the Word Select/Frame Sync Signal from an external master. SCKn and FSn pins are inputs.

51.6.4 I²S Format - Reception and Transmission Sequence with Word Select

As specified in the I²S protocol, data bits are left-adjusted in the Word Select slot, with the MSB transmitted first, starting one clock period after the transition on the Word Select line.