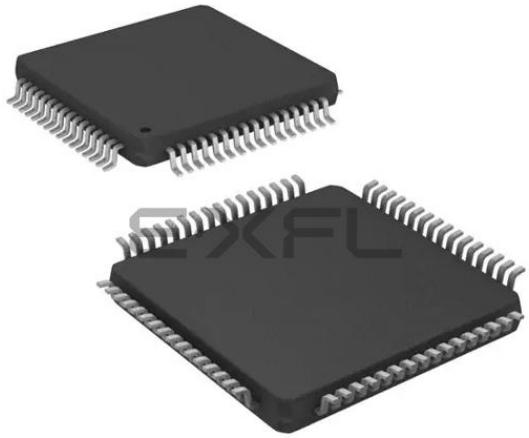


Welcome to E-XFL.COM

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	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-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsamd51j20a-aut

11.8 RAM Properties

The following table shows the different access properties of the three RAM blocks, according the different modes described in the previous chapters.

Table 11-2. Access to RAM

Access Condition	DATA RAM	TAG RAM	METADATARAM
CPU access when CMCC DISABLED	R/W	no R/W - hardfault	no R/W - hardfault
CPU access when CMCC ENABLED	CACHE section configured: R/W(1) TCM section configured: R/W	no R/W - hardfault	no R/W - hardfault
Debugger access when CMCC DISABLED	R/W	R/W	R/W
Debugger access when CMCC ENABLED	CACHE section configured: R/W(1) TCM section configured: R/W	no R/W	no R/W

Note:

1. A write operation in this zone can corrupt the coherency of the cache. An invalidate operation may be needed.

Related Links

11.7 DEBUG Mode

CPU Clock Domain	
Peripheral Clock	**Default State**
CLK_PORT_APB	Enabled
CLK_PTC_APB	Enabled
CLK_PUKCC_AHB	Enabled
CLK_QSPI_AHB	Enabled
CLK_QSPI2X_AHB	Enabled
CLK_SDHC0_AHB	Enabled
CLK_SDHC1_AHB	Enabled
CLK_SERCOM0_APB	Disabled
CLK_SERCOM1_APB	Disabled
CLK_SERCOM2_APB	Disabled
CLK_SERCOM3_APB	Disabled
CLK_SERCOM4_APB	Disabled
CLK_SERCOM5_APB	Disabled
CLK_SERCOM6_APB	Disabled
CLK_SERCOM7_APB	Disabled
CLK_TC0_APB	Disabled
CLK_TC1_APB	Disabled
CLK_TC2_APB	Disabled
CLK_TC3_APB	Disabled
CLK_TC4_APB	Disabled
CLK_TC5_APB	Disabled
CLK_TC6_APB	Disabled
CLK_TC7_APB	Disabled
CLK_TCC0_APB	Disabled
CLK_TCC1_APB	Disabled
CLK_TCC2_APB	Disabled
CLK_TCC3_APB	Disabled
CLK_TCC4_APB	Disabled
CLK_USB_AHB	Enabled
CLK_USB_APB	Disabled
CLK_WDT_APB	Enabled

21.12.8 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[3:0]			
Access					R	R	R	R
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CLOCKSYNC			MASKn[1:0]				
Access	R			R	R			
Reset	0			0	0			

Bit	7	6	5	4	3	2	1	0
		ALARMn[1:0]			CLOCK	FREQCORR	ENABLE	SWRST
Access		R	R		R	R	R	R
Reset		0	0		0	0	0	0

Bits 19:16 – GPn[3:0] 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.

Bits 12:11 – MASKn[1:0] Mask n Synchronization Busy Status [n = 1..0]

Value	Description
0	Write synchronization for MASKx register is complete.
1	Write synchronization for MASKx register is ongoing.

Bits 6:5 – ALARMn[1:0] Alarm n Synchronization Busy Status [n = 1..0]

Value	Description
0	Write synchronization for ALARMx register is complete.
1	Write synchronization for ALARMx register is ongoing.

22.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRL	7:0							DMAENABLE	SWRST
		15:8					LVLENx3	LVLENx2	LVLENx1	LVLENx0
0x02	CRCCTRL	7:0					CRCPOLY[1:0]		CRCBEATSIZE[1:0]	
		15:8	CRCMODE[1:0]		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						CRCERR	CRCZERO	CRCBUSY
0x0D	DBGCTRL	7:0								DBGRUN
0x0E ... 0x0F	Reserved									
0x10	SWTRIGCTRL	7:0	SWTRIGn[7:0]							
		15:8	SWTRIGn[15:8]							
		23:16	SWTRIGn[23:16]							
		31:24	SWTRIGn[31:24]							
0x14	PRICTRL0	7:0	RRLVLEN0	QOS00[1:0]		LVLPRI0[4:0]				
		15:8	RRLVLEN1	QOS01[1:0]		LVLPRI1[4:0]				
		23:16	RRLVLEN2	QOS02[1:0]		LVLPRI2[4:0]				
		31:24	RRLVLEN3	QOS03[1:0]		LVLPRI3[4:0]				
0x18 ... 0x1F	Reserved									
0x20	INTPEND	7:0				ID[4:0]				
		15:8	PEND	BUSY	FERR	CRCERR		SUSP	TCMPL	TERR
0x22 ... 0x23	Reserved									
0x24	INTSTATUS	7:0	CHINTn[7:0]							
		15:8	CHINTn[15:8]							
		23:16	CHINTn[23:16]							
		31:24	CHINTn[31:24]							
0x28	BUSYCH	7:0	BUSYCHn[7:0]							
		15:8	BUSYCHn[15:8]							
		23:16	BUSYCHn[23:16]							
		31:24	BUSYCHn[31:24]							
0x2C	PENDCH	7:0	PENDCH7	PENDCH6	PENDCH5	PENDCH4	PENDCH3	PENDCH2	PENDCH1	PENDCH0
		15:8	PENDCH15	PENDCH14	PENDCH13	PENDCH12	PENDCH11	PENDCH10	PENDCH9	PENDCH8

24.6.16.2 802.3 Pause Frame Transmission

Automatic transmission of pause frames is supported through the transmit pause frame bits of the Network Control register. If either bit 11 or bit 12 of the Network Control register is written with logic 1, an 802.3 pause frame will be transmitted, providing full duplex is selected in the Network Configuration register and the transmit block is enabled in the Network Control register.

Pause frame transmission will happen immediately if transmit is inactive or if transmit is active between the current frame and the next frame due to be transmitted.

Transmitted pause frames comprise the following:

- A destination address of 01-80-C2-00-00-01
- A source address taken from Specific Address register 1
- A type ID of 88-08 (MAC control frame)
- A pause opcode of 00-01
- A pause quantum register
- Fill of 00 to take the frame to minimum frame length
- Valid FCS

The pause quantum used in the generated frame will depend on the trigger source for the frame as follows:

- If bit 11 is written with a '1', the pause quantum will be taken from the Transmit Pause Quantum register. The Transmit Pause Quantum register resets to a value of 0xFFFF giving maximum pause quantum as default.
- If bit 12 is written with a '1', the pause quantum will be zero.

After transmission, a pause frame transmitted interrupt will be generated (bit 14 of the Interrupt Status register) and the only statistics register that will be incremented will be the Pause Frames Transmitted register.

Pause frames can also be transmitted by the MAC using normal frame transmission methods.

24.6.17 MAC PFC Priority-based Pause Frame Support

Note: Refer to the 802.1Qbb standard for a full description of priority-based pause operation.

The following table shows the start of a Priority-based Flow Control (PFC) pause frame.

Table 24-14. Start of a PFC Pause Frame

Address		Type (Mac Control Frame)	Pause Opcode	Priority Enable Vector	Pause Time
Destination	Source				
0x0180C2000001	6 bytes	0x8808	0x1001	2 bytes	8 × 2 bytes

The GMAC supports PFC priority-based pause transmission and reception. Before PFC pause frames can be received, bit 16 of the Network Control register must be set.

24.6.17.1 PFC Pause Frame Reception

The ability to receive and decode priority-based pause frames is enabled by setting bit 16 of the Network Control register. When this bit is set, the GMAC will match either classic 802.3 pause frames or PFC priority-based pause frames. Once a priority-based pause frame has been received and matched, then from that moment on the GMAC will only match on priority-based pause frames (this is an 802.1Qbb

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.

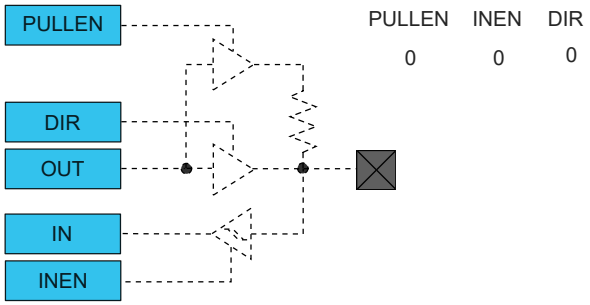
26.2 Features

- 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

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 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.

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

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'.

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.

Bit 3 – RF0L Rx FIFO 0 Message Lost

Value	Description
0	No Rx FIFO 0 message lost.
1	Rx FIFO 0 message lost. also set after write attempt to Rx FIFO 0 of size zero.

Bit 2 – RF0F Rx FIFO 0 Full

Value	Description
0	Rx FIFO 0 not full.
1	Rx FIFO 0 full.

Bit 1 – RF0W Rx FIFO 0 Watermark Reached

Value	Description
0	Rx FIFO 0 fill level below watermark.
1	Rx FIFO 0 fill level reached watermark.

Bit 0 – RF0N Rx FIFO 0 New Message

Value	Description
0	No new message written to Rx FIFO 0.
1	New message written to Rx FIFO 0.

42.8.9 Data

Name: DATA
Offset: 0x38
Reset: 0x00000000

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 Value
A write to or read from this register corresponds to a write to or read from one of the four data registers. The four 32-bit Data registers set the 128-bit data block used for encryption/decryption. The data register that is written to or read from is given by the `DATABUFPTR.DATPTR` field.
Note: Both input and output shares the same data buffer. Reading DATA register will return 0's when AES is performing encryption or decryption operation.

Table 43-31. Square Service Options

Option	Purpose	Required Parameters
SET_MULTIPLIEROPTION(PUKCL_SQUARE_ONLY)	Perform R = X^2 + CarryOperand	nu1RBase, nu1ZBase, nu1XBase, u2XLength
SET_MULTIPLIEROPTION(PUKCL_SQUARE_ADD)	Perform R = Z + X^2 + CarryOperand	nu1RBase, nu1ZBase, nu1XBase, u2XLength
SET_MULTIPLIEROPTION(PUKCL_SQUARE_SUB)	Perform R = Z - (X^2 + CarryOperand)	nu1RBase, nu1ZBase, nu1Xlength, u2XLength

43.3.4.10.6 Code Example

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


// Gf2n and CarryIn shall be beforehand filled (with zero or one)
PUKCL(Specific).Gf2n = ...;
PUKCL(Specific).CarryIn = ...;

PUKCL(u2Option) =...;
// Depending on the option specified, not all fields should be filled
PUKCL_Fmult(nu1XBase) = <Base of the ram location of X>;
PUKCL_Fmult(u2XLength) = <Length of X>;
PUKCL_Fmult(nu1ZBase) = <Base of the ram location of Z>;

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(Square,pvPUKCLParam);
if (PUKCL(u2Status) == PUKCL_OK)
            {
            // The Squaring has been executed correctly
            ...
            }
else // Manage the error
```

43.3.4.10.7 Important Considerations for Modular Reduction of a Square Computation

Note:
Additional options are available through the use of a modular reduction to be executed at the end of this operation. Some important considerations have to be taken into account concerning the length of resulting operands to get a mathematically correct result.

The output of this operation is not obviously compatible with the modular reduction as it may be either smaller or bigger. In the case (most of the time) the result (pointed by nu1RBase) is smaller in size than “twice the modulus plus one word” by one word, a padding word must be added to zero. Otherwise, the reduced value will be taken considering the high order words (potentially uninitialized) as part of the number, thus resulting in getting a mathematically correct but unexpected result.

In the case that the result is greater than twice the modulus plus one word, the modular reduction feature has to be executed as a separate operation, using an Euclidean division.

43.3.4.10.8 Constraints

When the options only indicate a square, the constraints involving nu1ZBase are not checked. The following conditions must be avoided to ensure that the service works correctly:

- nu1XBase, nu1RBase or nu1ZBase are not aligned on 32-bit boundaries

43.3.4.12 GCD, Modular Inverse

43.3.4.12.1 Purpose

The purpose of this command is to compute the Greatest Common Divisor (GCD) and the Modular Inverse. The algorithm used is the Extended Euclidean Algorithm for the GCD.

This command accepts as input two multiple precision numbers in GF(p) or two polynomials in GF(2^n) X and Y and computes their GCD (D), if D equals one, the command also supplies the inverse of X modulo Y.

The available options are as follows:

- Work in the GF(2^n) field or in the standard integer arithmetic field GF(p)

43.3.4.12.2 How to Use the Service

43.3.4.12.3 Description

This command calculates:

$D = GCD(X,Y)$.

and parameter A in the Bezout equation:

$A \times X + B \times Y = D$.

The first input, or input to inverse is X.

The second input, or modulus is Y.

The GCD is output in D.

The modular inverse if X and Y are co-primes is output A:

$A = X^{-1} mod(Y)$

The command calculates the GCD and the value A. The value A is the multiplicative inverse of X, only if X and Y are co-prime. As a supplemental result, Z is given back, being the quotient of Y divided by D only if D is different from zero:

$$Z = \left[\frac{Y}{D}\right]$$

At the end of the command: X is overwritten by D.

Y is cleared.

The value of A is calculated and stored.

The value of Z is calculated and stored if D is different from zero.

The service name for this operation is `GCD`.

In this computation, the following areas have to be provided:

- X (pointed by {nu1XBase,u2Length}) filled with X (with MSB word to zero)
- Y (pointed by {nu1YBase,u2Length}) filled with Y (with MSB word to zero)
- A (pointed by {nu1ABase,u2Length}) to contain calculated A
- Z (pointed by {nu1ZBase,u2Length}) to contain calculated Z
- The workspace (pointed by {nu1WorkSpace,32})

PUKCL Service	STACK Usage (Bytes)
GF2NEcConvAffineToProjective	56
GF2NEccDblFast	136
GF2NEccMulFast	208
GF2NEcDsaGenerateFast	376
GF2NEcDsaVerifyFast	440
GF2NEcRandomiseCoordinate	56

43.3.8.2 Parameter Size Limits for Different Services

The following table lists parameter size limits for different services.

For the services ModExp, PrimeGen, and CRT, additional details are available in the service description.

Table 43-113. Parameter Size Limits

API	Min/Max Sizes	Comments
SelfTest	–	–
ClearFlags	–	–
Swap	4 bytes to 2044 bytes	Per block to be swapped
Fill	4 bytes to 4088 bytes	–
Fast Copy/Clear	4 bytes to 2044 bytes	Supposing Length(R) = Length(X)
Conditional Copy/Clear	4 bytes to 2044 bytes	Supposing Length(R) = Length(X)
Smult	4 bytes to 2040 bytes	Supposing Length(R) = Length(X) + 4 Bytes, No Z Parameter, No Reduction
Compare	4 bytes to 2044 bytes	Supposing Length(X) = Length(Y)
FMult	Input: 4 bytes to 1020 bytes Output: 4bytes to 2040 bytes	Supposing Length(Y) = Length(X), No Z Parameter, No Reduction
Square	Input: 4 bytes to 1020 bytes Output: 4 bytes to 2040 bytes	Supposing No Z Parameter, No Reduction
Euclidean Division	Divider: 8 to 1016 bytes Num.: 8 to 2032 bytes	Supposing Length(Num) = 2*Length(Divider)
Mod. inv. / GCD	8 to 1012 bytes	–
ModRed	Modulus: 12 to 1016 bytes Input: 24 to 2032 bytes	Supposing RBase = XBase
Fast ModExp Exp in Crypto RAM	12 to 576 bytes (96 to 4608 bits)	Supposing Length(Exponent) = Length(Modulus), Window Size = 1 With the Exponent in Crypto RAM

49.8.9 Event Control

Name: EVCTRL
Offset: 0x20
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
			MCEOx	MCEOx	MCEOx	MCEOx	MCEOx	MCEOx
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
			MCEIx	MCEIx	MCEIx	MCEIx	MCEIx	MCEIx
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
	TCEIx	TCEIx	TCINVx	TCINVx		CNTEO	TRGEO	OVFEO
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
	CNTSEL[1:0]		EVACT1[2:0]			EVACT0[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

Bits 29,28,27,26,25,24 – MCEOx Match or Capture Channel x Event Output Enable
These bits control if the Match/capture event on channel x is enabled and will be generated for every match or capture.

Value	Description
0	Match/capture x event is disabled and will not be generated.
1	Match/capture x event is enabled and will be generated for every compare/capture on channel x.

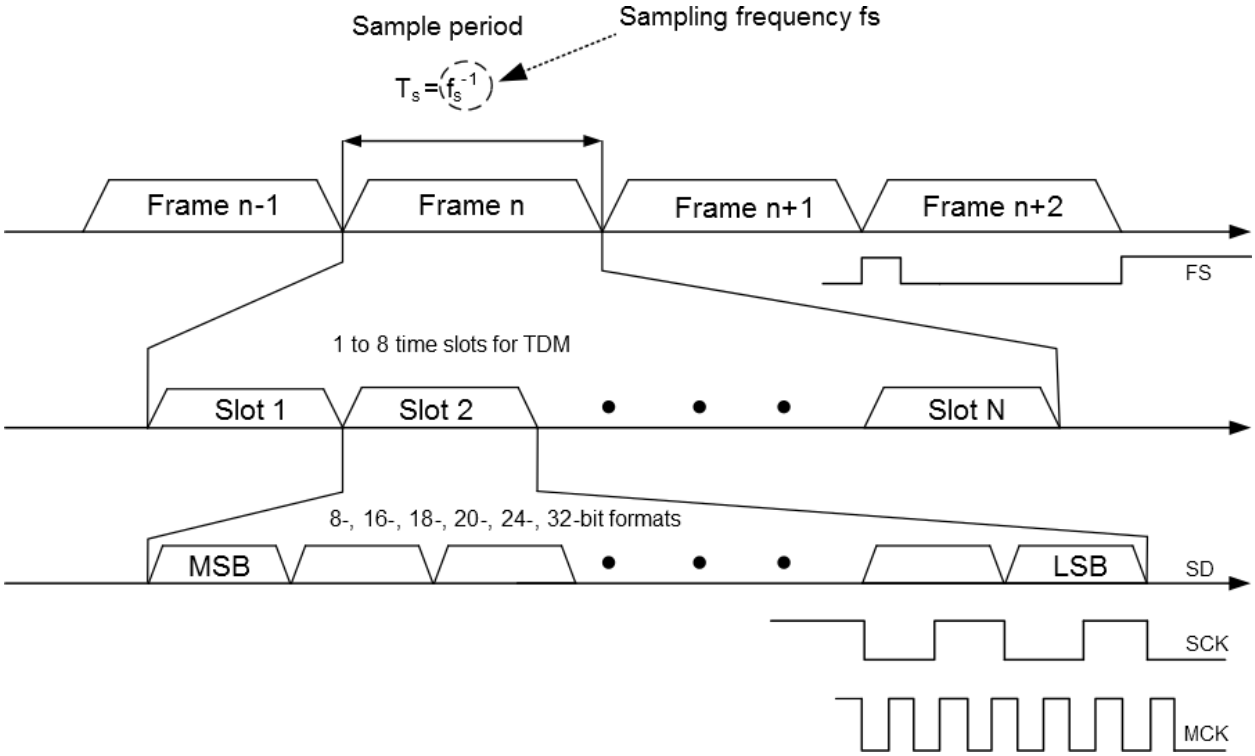
Bits 21,20,19,18,17,16 – MCEIx Match or Capture Channel x Event Input Enable
These bits indicate if the Match/capture x incoming event is enabled

These bits are used to enable match or capture input events to the CCx channel of TCC.

Value	Description
0	Incoming events are disabled.
1	Incoming events are enabled.

Bits 15,14 – TCEIx Timer/Counter Event Input x Enable
This bit is used to enable input event x to the TCC.

Figure 51-2. Data Format: Frames, Slot, Bits and Clocks



I²S supports multiple data formats such as:

- 32-, 24-, 20-, 18-, 16-, and 8-bit mono or stereo format
- 16- and 8-bit compact stereo format, with left and right samples packed in the same word to reduce data transfers

In mono format, Transmit mode, data written to the left channel is duplicated to the right output channel. In mono format, Receiver mode, data received from the right channel is ignored and data received from the left channel is duplicated in to the right channel.

In mono format, TDM Transmit mode with more than two slots, data written to the even-numbered slots is duplicated in to the following odd-numbered slot.

In mono format, TDM Receiver mode with more than two slots, data received from the even-numbered slots is duplicated in to the following odd-numbered slot.

Mono format can be enabled by writing a '1' to the MONO bit in the Serializer m Control register (SERCTRLm.MONO).

I²S support different data frame formats:

- 2-channel I²S with Word Select
- 1- to 8-slot Time Division Multiplexed (TDM) with Frame Sync and individually enabled slots
- 1- or 2-channel Pulse Density Modulation (PDM) reception for MEMS microphones
- 1-channel burst transfer with non-periodic Frame Sync

In 2 channel I²S mode, number of slots configured is one or two and successive data words corresponds to left and right channel. Left and right channel are identified by polarity of Word Select signal (FSn signal). Each frame consists of one or two data word(s). In the case of compact stereo format, the number of slots can be one. When 32-bit slot size is used, the number of slots can be two.

Figure 52-9. PCC Waveforms (DSIZE=4_DATA, ALWYS = 0, HALFS = 1, FRSTS = 0)

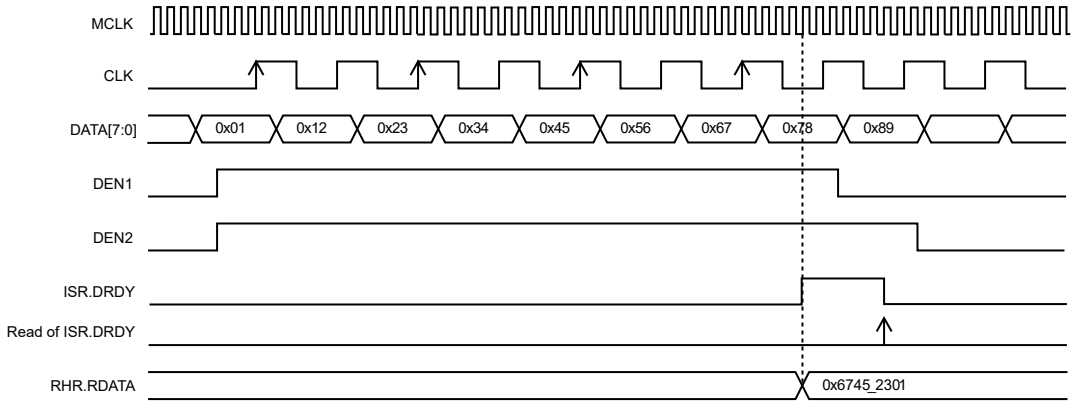


Figure 52-10. PCC Waveforms (ISIZE=10_BITS, DSIZE=2_DATA, ALWYS = 0, HALFS = 1, FRSTS = 0, SCALE = 0)

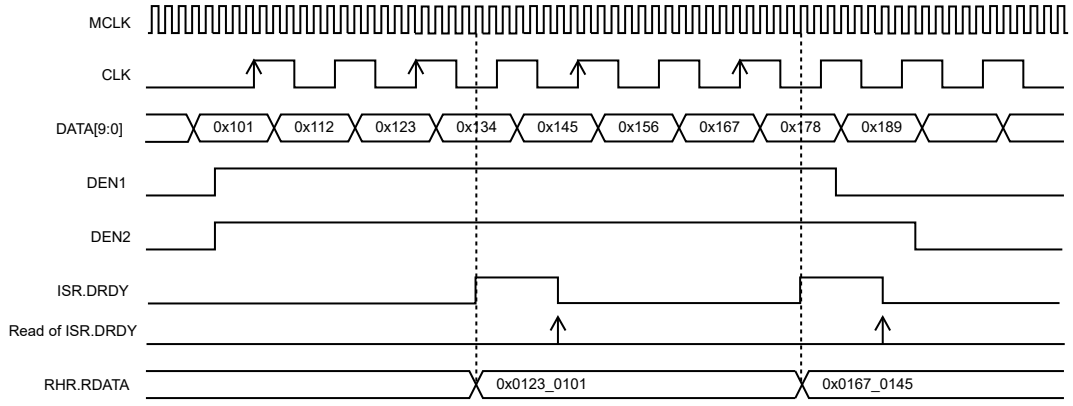
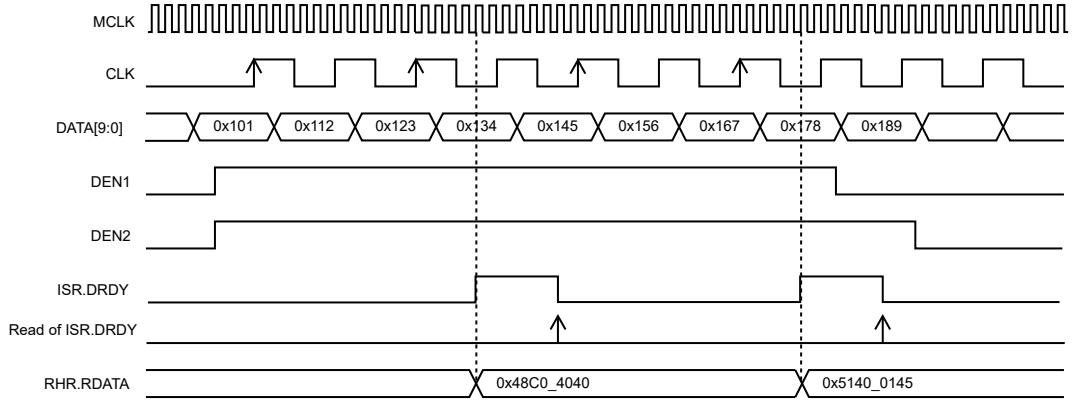


Figure 52-11. PCC Waveforms (ISIZE=10_BITS, DSIZE=2_DATA, ALWYS = 0, HALFS = 1, FRSTS = 0, SCALE = 1)



53.8.8 Status

Name: STATUS
Offset: 0x0C
Reset: 0x0040
Property: Read-Synchronized, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
			CCBUFV1	CCBUFV0			FILTERBUFV	PRESCBUFV
Access			R	R			R	R
Reset			0	0			0	0

Bit	7	6	5	4	3	2	1	0
	DIR	STOP	HERR	WINERR		MPERR	IDXERR	QERR
Access	R	R	RW	RW		RW	RW	RW
Reset	0	1	0	0		0	0	0

Bits 12, 13 – CCBUFV Compare Channel x Buffer Valid
The bit is set when a new value is written to the corresponding CCBUF register.

The bit is cleared by writing a '1' to the corresponding location or automatically cleared on an UPDATE condition.

Bit 9 – FILTERBUFV Filter Buffer Valid
This bit is set when a new value is written to the PRESCALERBUF register.

The bit is cleared by writing a '1' to the corresponding location or automatically cleared on an UPDATE condition.

This bit is always read '0' when COUNTER operation mode is selected.

Bit 8 – PRESCBUFV Prescaler Buffer Valid
This bit is set when a new value is written to the PRESC register.

The bit is cleared by writing a '1' to the corresponding location or automatically cleared on an UPDATE condition.

Bit 7 – DIR Direction Status Flag
This bit reflects the HALL/QDEC direction.

in COUNTER mode, this bits is always read '0'.

Value	Description
0	Clockwise direction.
1	Counter-clockwise direction.

Bit 6 – STOP Stop
This bit reflects the HALL/QDEC decoding status.

In COUNTER mode, this bits is always read '0'.

Figure 55-2. QFN 64 Pin

Figure 55-3. TQFP 64 Pin

Figure 55-4. TFBGA 120 Pin