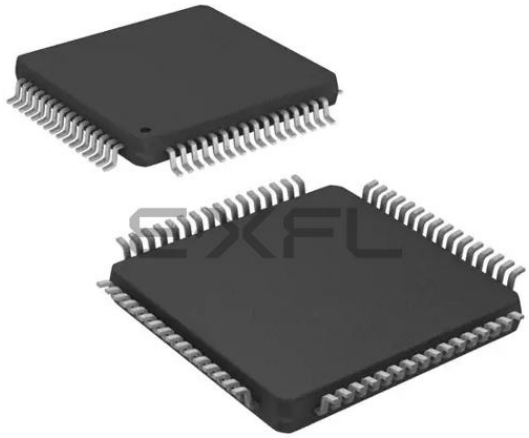


Welcome to E-XFL.COM

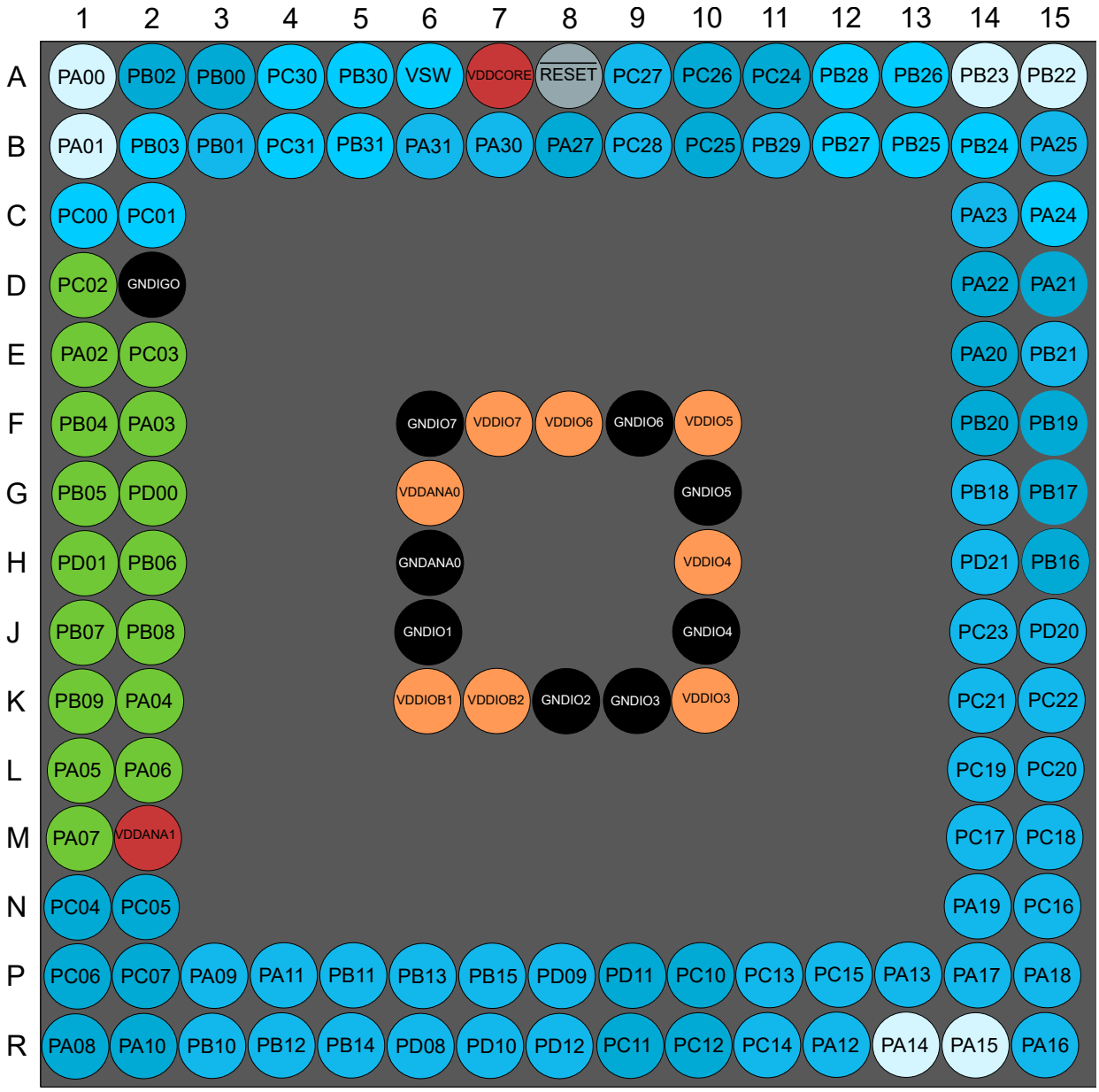
What is "Embedded - Microcontrollers"?

"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, Ethernet, I²C, IrDA, LINbus, MMC/SD, QSPI, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	51
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	192K 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/atsame53j19a-aut

4.4 Pin Count 120 (P)



Top View

- DIGITAL PIN
- ANALOG PIN
- OSCILLATOR / DIGITAL PIN
- GROUND
- INPUT SUPPLY
- REGULATED INPUT/OUPUT SUPPLY
- RESET PIN

14.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0								SWRST
0x01 ... 0x03	Reserved									
0x04	SYNCBUSY	7:0	GENCTRL5	GENCTRL4	GENCTRL3	GENCTRL2	GENCTRL1	GENCTRL0		SWRST
		15:8				GENCTRL10	GENCTRL9	GENCTRL8	GENCTRL7	GENCTRL6
		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	GENCTRL5	7:0				SRC[4:0]				
		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
		23:16	DIV[7:0]							
		31:24	DIV[15:8]							
0x38	GENCTRL6	7:0				SRC[4:0]				
		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
		23:16	DIV[7:0]							
		31:24	DIV[15:8]							
0x3C	GENCTRL7	7:0				SRC[4:0]				
		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
		23:16	DIV[7:0]							
		31:24	DIV[15:8]							

Power Supply Reset. When VDD crosses above the brown-out threshold level (BOD33.LEVEL), the device will leave battery backup mode and will wakeup from backup mode if the BBPS.WAKEEN bit is set.

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.

Related Links

9.4 NVM User Page Mapping

19.6.5.3.1 BOD33 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 (VDD or VBAT) 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 Backup or Hibernate mode by writing to the BOD33 bits (BOD33.BKUPCFG = 1 or BOD33.HIBCFG = 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 32 kHz clock, output by the 32 kHz Ultra Low-Power Oscillator OSCULP32K.

Note: If (BOD33.PSEL) is 0, sampling mode is disabled.

As the sampling clock is different from the APB clock domain, synchronization among the clocks is necessary. See 19.6.7 Synchronization for additional information.

Related Links

9.4 NVM User Page Mapping

19.6.5.3.2 BOD33 Low Power Mode

BOD33 Low Power mode is automatically enabled in Backup or Hibernate sleep mode.

BOD33 Low Power mode can be enabled in Standby sleep mode by writting to '1' the BOD33.STDBYCFG bit.

Related Links

9.4 NVM User Page Mapping

19.6.5.3.3 BOD33 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 19-2. BOD Hysteresis Principle

Hysteresis OFF:

21.10.14 Tamper Control

Name: TAMPCTRL
Offset: 0x60
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
				DEBNC4	DEBNC3	DEBNC2	DEBNC1	DEBNC0
Access								
Reset				0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
				TAMLVL4	TAMLVL3	TAMLVL2	TAMLVL1	TAMLVL0
Access								
Reset				0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
							IN4ACT[1:0]	
Access								
Reset							0	0

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, 28 – 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, 20 – 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, 8:9 – 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

- 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)

DMA Channel Initialization

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 Configuration
  - The channel number of the DMA channel to configure must be written to the Channel Control A register (CHCTRLA) register
  - Trigger action must be selected by writing the Trigger Action bit field in the Channel Control A register (CHCTRLA.TRIGACT)
  - Trigger source must be selected by writing the Trigger Source bit field in the Channel Control A register (CHCTRLA.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)
  - 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

CRC Calculation

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)

Register Properties

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)

Bits 11:8 – RWS[3:0] NVM Read Wait States
These bits give the number of wait states for a read operation when AUTOWS=0. Zero indicates zero wait states, one 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.

Bits 7:6 – PRM[1:0] Power Reduction Mode during Sleep
Indicates the power reduction mode during sleep.

Value	Name	Description
0x0	SEMIAUTO	NVM block enters low-power mode when entering standby mode. NVM block enters low-power mode when SPRM command is issued. NVM block exits low-power mode upon first access.
0x1	FULLAUTO	NVM block enters low-power mode when entering standby mode. NVM block enters low-power mode when SPRM command is issued. NVM block exits low-power mode when system is not in standby mode.
0x2		Reserved
0x3	MANUAL	NVM block does not enter low-power mode when entering standby mode. NVM block enters low-power mode when SPRM command is issued. NVM block exits low-power mode upon first access.

Bits 5:4 – WMODE[1:0] Write Mode
Write commands can be generated automatically when crossing address boundaries while writing to the NVM. Boundaries depend on the settings below.

Value	Name	Description
0x0	MAN	Manual Write
0x1	ADW	Automatic Double Word Write
0x2	AQW	Automatic Quad Word
0x3	AP	Automatic Page Write

Bit 3 – SUSPEN Suspend Enable
0: The write and erase suspend resume feature is disabled.

1: A write or erase operation can be suspended in case of a read in the same bank.

Bit 2 – AUTOWS Auto Wait State Enable
0: Automatic wait state generation is disabled. The number of wait states used is given by RWS.

1: Automatic wait state generation is enabled. The number of wait states used is automatically detected therefore the module can operate at any frequency up to the device maximum frequency. A minimum of one cycle latency is induced.

Bit 3 – TCC2 TCC2 APB Protection Enable

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

Bit 2 – GMAC GMAC APB Protection Enable

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

Bit 1 – CAN1 CAN1 APB Protection Enable

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

Bit 0 – CAN0 CAN0 APB Protection Enable

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

29.8.4 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
					XOSC32KSW	XOSC32KFAIL		XOSC32KRDY
Access					R	R		R
Reset					0	0		0

Bit 3 – XOSC32KSW 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.

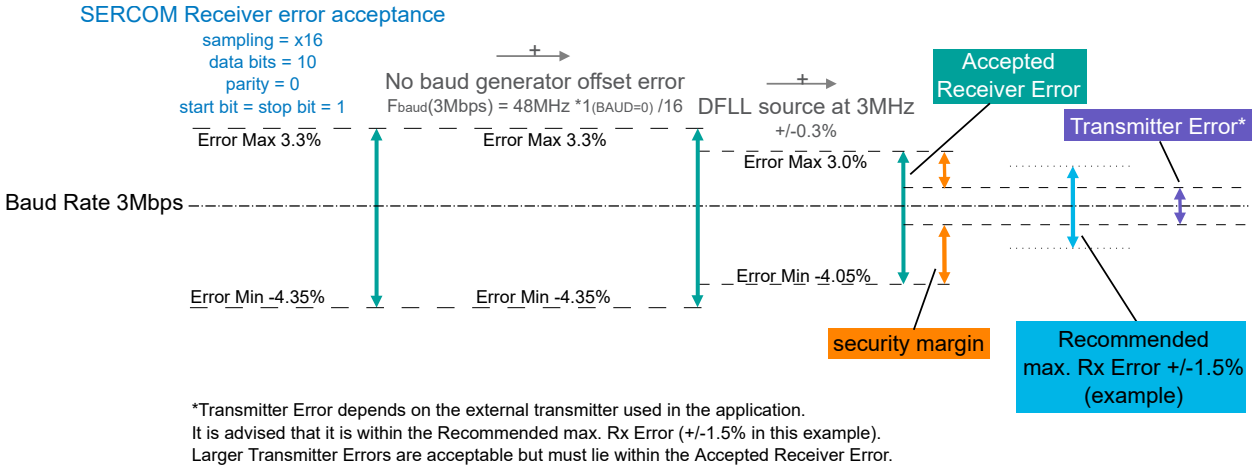
Bit 2 – XOSC32KFAIL XOSC32K Clock Failure Detector

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.

Figure 34-6. 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

33.6.2.3 Clock Generation – Baud-Rate Generator

33.6.2.3.1 Asynchronous Arithmetic Mode BAUD Value Selection

34.6.3 Additional Features

34.6.3.1 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.

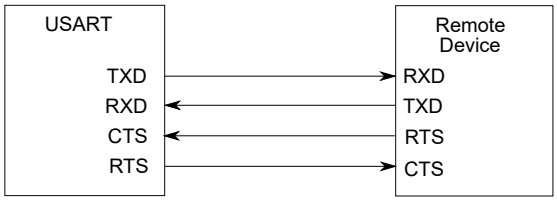
If *odd parity* is selected (CTRLB.PMODE=1), the parity bit of an outgoing frame is '1' if the data contains an even number of bits that are '0', making the total number of '1' odd.

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.

34.6.3.2 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 34-7. 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).

Value	Description
0	The instruction is not sent to the serial flash memory.
1	The instruction is sent to the serial flash memory.

Bits 2:0 – WIDTH[2:0] Instruction Code, Address, Option Code and Data Width
This field defines the width of the instruction code, the address, the option and the data.

Value	Name	Description
0x0	SINGLE_BIT_SPI	Instruction: Single-bit SPI / Address-Option: Single-bit SPI / Data: Single-bit SPI
0x1	DUAL_OUTPUT	Instruction: Single-bit SPI / Address-Option: Single-bit SPI / Data: Dual SPI
0x2	QUAD_OUTPUT	Instruction: Single-bit SPI / Address-Option: Single-bit SPI / Data: Quad SPI
0x3	DUAL_IO	Instruction: Single-bit SPI / Address-Option: Dual SPI / Data: Dual SPI
0x4	QUAD_IO	Instruction: Single-bit SPI / Address-Option: Quad SPI / Data: Quad SPI
0x5	DUAL_CMD	Instruction: Dual SPI / Address-Option: Dual SPI / Data: Dual SPI
0x6	QUAD_CMD	Instruction: Quad SPI / Address-Option: Quad SPI / Data: Quad SPI
0x7		Reserved

Bit 4 – DATTEO Data Timeout error
This bit is set to 1 when detecting one of following timeout conditions:

- Busy timeout for R1b, R5b response type (see “Physical Layer Simplified Specification V3.01” and “SDIO Simplified Specification V3.00”).
- Busy timeout after Write CRC Status.
- Write CRC Status timeout.
- Read data timeout.

This bit can only be set to 1 if EISTER.DATTEO is set to 1. An interrupt can only be generated if EISIER.DATTEO is set to 1.

Writing this bit to 1 clears this bit.

Value	Description
0	No error
1	Error

Bit 3 – CMDIDX Command Index Error
This bit is set to 1 if a Command Index error occurs in the command response.

This bit can only be set to 1 if EISTER.CMDIDX is set to 1. An interrupt can only be generated if EISIER.CMDIDX is set to 1.

Writing this bit to 1 clears this bit.

Value	Description
0	No error
1	Error

Bit 2 – CMDEND Command End Bit Error
This bit is set to 1 when detecting that the end bit of a command response is 0.

This bit can only be set to 1 if EISTER.CMDEND is set to 1. An interrupt can only be generated if EISIER.CMDEND is set to 1.

Writing this bit to 1 clears this bit.

Value	Description
0	No error
1	Error

Bit 1 – CMDCRC Command CRC Error
The Command CRC Error is generated in two cases.

If a response is returned and Command Timeout Error (CMDTEO) is set to 0 (indicating no command timeout), this bit is set to 1 when detecting a CRC error in the command response.

The peripheral detects a CMD line conflict by monitoring the CMD line when a command is issued. If the peripheral drives the CMD line to 1 level, but detects 0 level on the CMD line at the next SDCLK edge, then the peripheral aborts the command (stops driving the CMD line) and sets this bit to 1. CMDTEO is also set to 1 to indicate a CMD line conflict (refer to Table 40-2).

This bit can only be set to 1 if EISTER.CMDCRC is set to 1. An interrupt can only be generated if EISIER.CMDCRC is set to 1.

42.8.13 Galois Hash x (GCM mode only)

Name: CIPLEN
Offset: 0X80
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	CIPLEN[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
	CIPLEN[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
	CIPLEN[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
	CIPLEN[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 – CIPLEN[31:0] Cipher Length
This register contains the length in bytes of the Cipher text that is to be processed. This is programmed by the user in GCM mode for Tag generation.

To check whether the version returned by the PUKCL is correct, the following code can be used.

```
while (pvPUKCLParam->P.PUKCL_SelfTest.u4Version != PUKCL_VERSION);
```

In a similar way, other returns can also be accessed.

43.3.3.3 Parameter Passing (Special Considerations)

Most of the PUKCL services work with memory area and accept pointers and lengths as parameters to define input and output areas. Most of the time, the pointers and lengths are untouched by the services, while the defined areas are read, filled, or overwritten. These memory areas are defined with an initial pointer and a byte length. For most of the commands, the memory area location must be in the PUKCC Cryptographic RAM. The Cryptographic RAM is the memory area for parameter exchange with the PUKCL and is 4 Kbytes large. Sometimes memory areas can be located in Embedded SRAM, which is detailed in the Location column of the parameters description tables.

When working with binary fields, polynomials in GF(2^n) need no transformation to be written in an area:

- Each bit represents a polynomial coefficient 0 or 1
- The polynomials must be written Low Significant Byte First
- A zero padding on the Most Significant Bytes may be added if the area is larger than the real size of the polynomial

Important: The Cryptographic RAM is 4 Kbytes in size and is dedicated to PUKCC. However, to ensure correct library operation, the two last 32-bit words must not be used. Unless otherwise specified, these memory areas contain integers in GF(p) or polynomials in GF(2^n) with the Less Significant Byte first.

Unless otherwise specified, the length must be a multiple of four and the pointers must be four bytes aligned. This is because most of the services work with 32-bit words.

43.3.3.4 Aligned Significant Length

Parameters in memory areas can have any Significant Length in bytes. As the lengths in PUKCL must be a multiple of four, a padding is processed on the Most Significant Side with zero to three bytes cleared to zero. Now the parameter can be considered to meet the Aligned Significant Length requirement for PUKCL.

43.3.3.5 Processing Field GF(p) and GF(2^n)

The library can process arithmetic functions over GF(p) (or Zp integers) and GF(2^n), when applicable. The choice of these processing fields is made using the following rules:

- If a processing field is not applicable to the function, it is not mentioned and the Specific.GF2n bit has no effect.
- If the function can support both processing fields, the choice is mentioned and the Specific.GF2n bit must be filled according to the choice.
- If the function supports only one of the processing fields, the processing field is mentioned and the Specific.GF2n bit has no effect.

43.3.3.6 Return Codes

Each call to one of the PUKCL services returns a status code indicating whether or not the execution is correct, which can be decoded, as shown in the following figure.

The minimum value for u2ModLength is 8 bytes, so the significant length of Num must be at least 8 bytes. To divide by a 32-bit value, the divider and numerator shall be multiplied by 232. The resulting remainder will have to be divided by 2^{32}, the quotient will be exact.

43.3.4.11.5 Code Example

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

// Fill all the fields
// In that case, the quotient will be computed
// If it was not needed, set nu1QuoBase to NULL
PUKCL_Div(nu1NumBase) = <Base of the ram location of Num>;
PUKCL_Div(nu1ModBase) = <Base of the ram location of Mod>;
PUKCL_Div(nu1QuoBase) = <Base of the ram location of Quo>;
PUKCL_Div(nu1WorkSpace) = <Base of the workspace>;
PUKCL_Div(nu1RBase) = <Base of the ram location of R>;
PUKCL_Div(u2NumLength) = <Length of Num>;
PUKCL_Div(u2ModLength) = <Length of Mod>;

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

43.3.4.11.6 Constraints

The following conditions must be avoided to ensure the service works correctly:

- nu1ModBase, nu1RBase, nu1QuoBase, nu1WorkSpace or nu1NumBase are not aligned on 32-bit boundaries
- {nu1ModBase, u2ModLength}, {nu1RBase, u2ModLength}, {nu1WorkSpace, 64} or{nu1NumBase, u2NumLength} are not in Crypto RAM
- u2ModLength, u2NumLength is either: < 4, > 0xffc or not a 32-bit length
- One or more overlaps exist between two of the areas: {nu1ModBase,u2ModLength},{nu1RBase, u2ModLength} {nu1NumBase, u2NumLength}(1) or {nu1WorkSpace,64}
- If nu1QuoBase is different from zero and: {nu1QuoBase, u2NumLength - u2ModLength + 4} are not in Crypto RAM
- If nu1QuoBase is different from zero and one or more overlaps exist between two of the areas: {nu1QuoBase, u2NumLength - u2ModLength + 4}, {nu1ModBase, u2ModLength}, {nu1RBase, u2ModLength}, {nu1NumBase, u2NumLength} or {nu1WorkSpace, 64}

Overlaps between {nu1RBase, u2ModLength} and {nu1NumBase, u2NumLength} are forbidden, but the equality between nu1RBase and nu1NumBase is authorized

43.3.4.11.7 Status Returned Values

Table 43-36. Div Service Return Codes

Returned Status	Importance	Meaning
PUKCL_OK	–	Service functioned correctly.
PUKCL_DIVISION_BY_ZERO	Severe	The operation was not performed because the Denominator value is zero.

Table 45-2. Accumulation

Number of Accumulated Samples	AVGCTRL. SAMPLENUM	Intermediate Result Precision	Number of Automatic Right Shifts	Final Result Precision	Automatic Division Factor
1	0x0	12 bits	0	12 bits	0
2	0x1	13 bits	0	13 bits	0
4	0x2	14 bits	0	14 bits	0
8	0x3	15 bits	0	15 bits	0
16	0x4	16 bits	0	16 bits	0
32	0x5	17 bits	1	16 bits	2
64	0x6	18 bits	2	16 bits	4
128	0x7	19 bits	3	16 bits	8
256	0x8	20 bits	4	16 bits	16
512	0x9	21 bits	5	16 bits	32
1024	0xA	22 bits	6	16 bits	64
Reserved	0xB –0xF	12 bits		12 bits	0

45.6.2.10 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 45.6.2.9 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 45-3.

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 45-3.

Note: To perform the averaging of two or more samples, the Conversion Result Resolution field in the Control B register (CTRLB.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 45-3. 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

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

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.

48.7.3.11 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.

Figure 52-6. PCC Waveforms (DSIZE=4_DATA, ALWYS = 1, HALFS = 0)

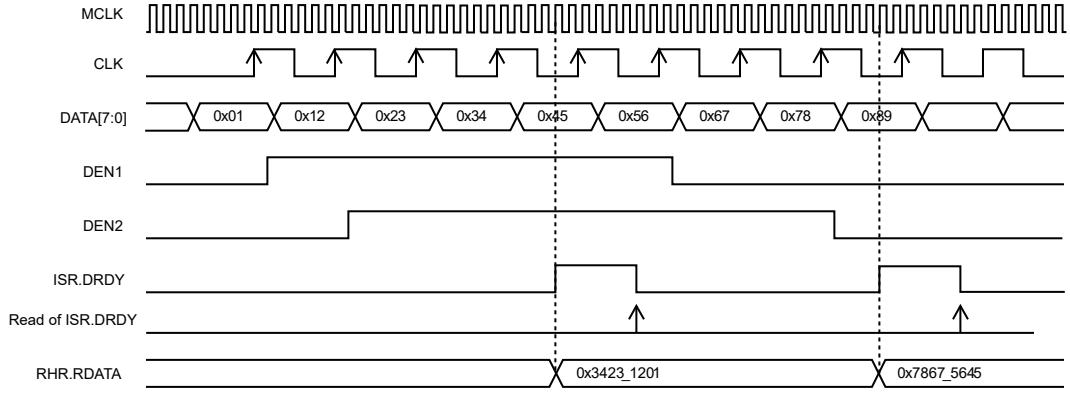


Figure 52-7. PCC Waveforms (ISIZE=10_BITS, DSIZE=2_DATA, ALWYS = 1, HALFS = 0, SCALE = 0)

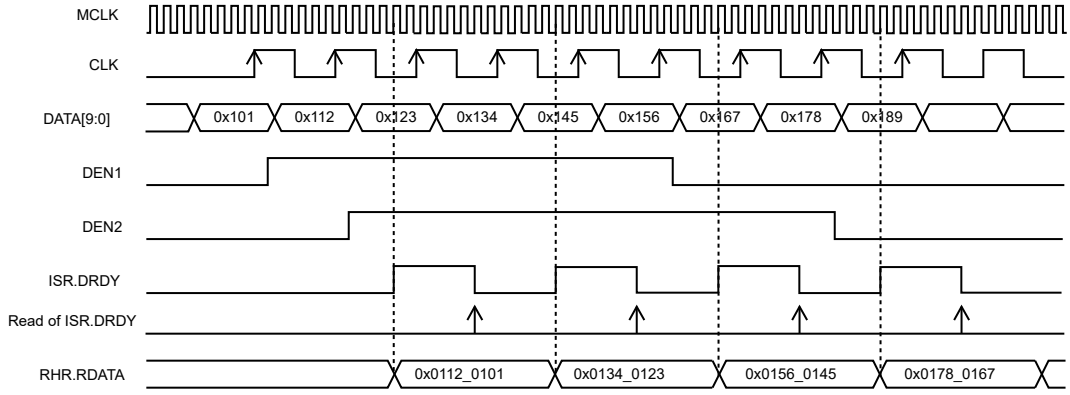


Figure 52-8. PCC Waveforms (ISIZE=10_BITS, DSIZE=2_DATA, ALWYS = 1, HALFS = 0, SCALE = 1)

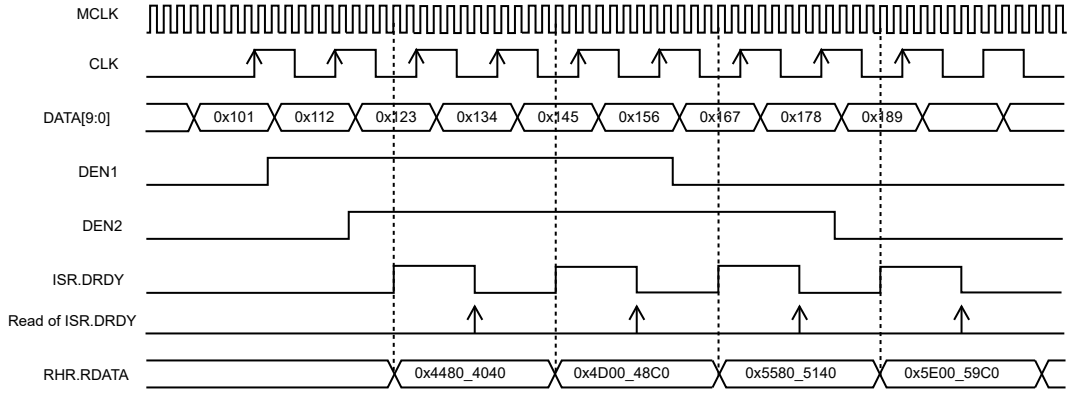


Table 55-1. Thermal Resistance Data

Package Type	θ_{JA}	θ_{JC}
64-pin TQFP	57.4°C/W	10.6°C/W
100-pin TQFP	55.0°C/W	11.1°C/W
128-pin TQFP	48.7°C/W	9.4°C/W
120-pin TFBGA	36.63°C/W	12.2°C/W
48-pin QFN	29.8°C/W	10.0°C/W
64-pin QFN	30.3°C/W	9.9°C/W
64-pin WLCSP	36.8°C/W	5.0°C/W

55.2.2 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 has to be fitted on the chip, the second equation should be used to compute the resulting average chip-junction temperature T_J in °C.

55.3 Package Drawings

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging.

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