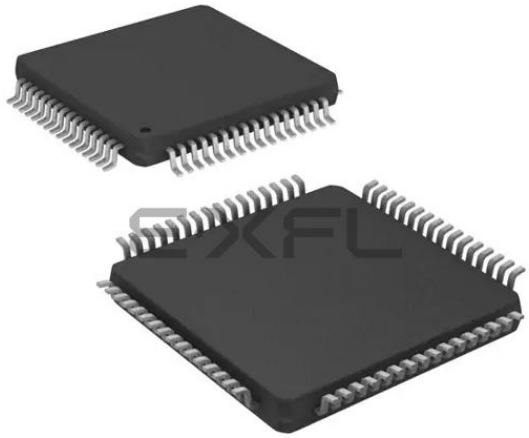


Welcome to E-XFL.COM

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	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/atsame53j20a-au

19.8.5 3.3V Brown-Out Detector (BOD33) Control

Name: BOD33
Offset: 0x10
Reset: Determined from NVM User Row
Property: Write-Synchronized, PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	VBATLEVEL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	LEVEL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	x	x	x	x	x	x

Bit	15	14	13	12	11	10	9	8
		PSEL[2:0]			HYST[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	x	x	x	x

Bit	7	6	5	4	3	2	1	0
	RUNBKUP	RUNHIB	RUNSTDBY	STDBYCFG	ACTION[1:0]		ENABLE	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	y	y	z	

Bits 31:24 – VBATLEVEL[7:0] BOD33 Threshold Level on VBAT
This field sets the triggering voltage threshold for the BOD33 when the BOD33 monitors VBAT in battery backup sleep mode.

This field is not synchronized.

Bits 23:16 – LEVEL[7:0] BOD33 Threshold Level on VDD
This field sets the triggering voltage threshold for the BOD33 when the BOD33 monitors VDD. If an hysteresis value is programmed (BOD33.HYST), this field corresponds to the lower threshold (V_{BOD-}).

These bits are loaded from NVM User Row at start-up.

This field is not synchronized.

The VBOD- input voltage can be calculated as follows: VBOD- = 1.5 + LEVEL[7:0) x Level_Step

And the upper threshold (VBOD+) is then: VBOD+ = VBOD- + N x HYST_STEP, With N=0 to 15 according to HYST[3:0] value and HYST_STEP = Level_Step, (refer to Bits 11:8 – HYST[3:0]: BOD33 Hysteresis voltage value on VDD).

At the upper side of Level[7:0] values depending on the Hysteresis value chosen with HYST[3:0], the VBOD+ level reaches an overflow, e.g., for HYST[3:0] = 0d2 the hysteresis is 2 x Level_Step = 12 mV up to position 253 and position 254 to 255 above must not be used.

19.8.10 Backup Output (BKOUT) Control

Name: BKOUT
Offset: 0x24
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
							RTCTGL[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
							SET[1:0]	
Access							W	W
Reset							0	0

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

Bit	7	6	5	4	3	2	1	0
							EN[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 25:24 – RTCTGL[1:0] RTC Toggle Output

Value	Description
0	The output will not toggle on RTC event.
1	The output will toggle on RTC event.

Bits 17:16 – SET[1:0] Set Output
Writing a '0' to a bit has no effect.

Writing a '1' to a bit will set the corresponding output.

Reading this bit returns '0'.

Bits 9:8 – CLR[1:0] Clear Output
Writing a '0' to a bit has no effect.

Writing a '1' to a bit will clear the corresponding output.

Reading this bit returns '0'.

Bits 1:0 – EN[1:0] Enable Output

Writing a '1' to this bit will clear the Periodic Interval n Interrupt Enable bit, which disables the Periodic Interval n interrupt.

Value	Description
0	Periodic Interval n interrupt is disabled.
1	Periodic Interval n interrupt is enabled.

22.6.3.10 Memory CRC Monitor

When enabled, it is possible to continuously check a a memory block data integrity by calculating and checking the CRC checksum. The expected CRC checksum value must be located in the last memory block location, as shown in the table below:

CRCCTRL.CRCPOLY	CRCCTRL.CRCBEATSIZE	Last Memory Block Byte Locations Value (MSB Byte First)	CHECKSUM Result
CRC-16	Byte	Expected CRC[7:0]	0x00000000
	Half-word	Expected CRC[15:8]	
	Word	0x00 0x00 Expected CRC[7:0] Expected CRC[15:8]	
CRC-32	Byte	Expected CRC[31:24]	CRC Magic Number (0x2144DF1C)
	Half-word	Expected CRC[23:16]	
	Word	Expected CRC[15:8] Expected CRC[7:0]	

When the channel is enabled and the descriptor is fetched, the CRC Checksum register (CRCCHKSUM) is reloaded with the initial checksum value (CHKINIT), stored in the DSTADDR location of the first descriptor. The DMA read and calculate the checksum over the entire data from the source address.When the checksum calculation is completed the DMA read the last beat from the memory, the calculated CRC value from the CRC Checksum register is compared to zero or CRC magic number, depending on CRC polynomial selection.

If the CHECKSUM does not match the comparison value the DMA channel is disabled, and both and the CRC Error bit in the Channel n Status register (CHSTATUSn.CRCERR) and Transfer Error interrupt flag (CHINTFLAGn.TERR) are set. If enabled, the Transfer Error interrupt is generated.

If the calculated checksum value matches the compare value, the Transfer Complete interrupt flag (CHINTFLAGn.TCMPL) is set, optional interrupt is generated and the DMA will perform the following actions, depending on the descriptor list settings:

- If the list has only one descriptor, the DMA will re-fetch the descriptor
- If the current descriptor is the last descriptor from the list, the DMA will fetch the first descriptor from the list

When the fetch is completed, the DMA restarts the operations described above when new triggers are detected.

In order to enable the memory CRC monitor, the following actions must be performed:

1. The CRC module must be set to be used with a DMA channel (CRCCTRL.CRCSRC)
2. Reserve memory space addresses to configure a descriptor or a list of descriptors
3. Configure each descriptor
   – Set the next descriptor address (DESCADDR)

24.9.2 GMAC Network Configuration Register

Name: NCFGR
Offset: 0x004
Reset: 0x00080000
Property: R/W

Bit	31	30	29	28	27	26	25	24
		IRXER	RXBP	IPGSEN		IRXFCS	EFRHD	RXCOEN
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
	DCPF			CLK[2:0]			RFCS	LFERD
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	1	0	0	0

Bit	15	14	13	12	11	10	9	8
	RXBUFO[1:0]		PEN	RTY				MAXFS
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bit	7	6	5	4	3	2	1	0
	UNIHEN	MTIHEN	NBC	CAF	JFRAME	DNVLAN	FD	SPD
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 30 – IRXER Ignore IPG GRXER
When this bit is written to '1', the Receive Error signal (GRXER) has no effect on the GMAC operation when Receive Data Valid signal (GRXDV) is low.

Bit 29 – RXBP Receive Bad Preamble
When written to '1', frames with non-standard preamble are not rejected.

Bit 28 – IPGSEN IP Stretch Enable
Writing a '1' to this bit allows the transmit IPG to increase above 96 bit times, depending on the previous frame length using the IPG Stretch Register.

Bit 26 – IRXFCS Ignore RX FCS
For normal operation this bit must be written to zero.

When this bit is written to '1', frames with FCS/CRC errors will not be rejected. FCS error statistics will still be collected for frames with bad FCS, and FCS status will be recorded in the DMA descriptor of the frame.

Bit 25 – EFRHD Enable Frames Received in half-duplex
Writing a '1' to this bit enables frames to be received in half-duplex mode while transmitting.

33. SERCOM – Serial Communication Interface

33.1 Overview

There are up to eight instances of the serial communication interface (SERCOM) peripheral.

A SERCOM can be configured to support a number of modes: I^2C, SPI, and USART. When an instance of SERCOM is configured and enabled, all of the resources of that SERCOM instance will be dedicated to the selected mode.

The SERCOM serial engine consists of a transmitter and receiver, baud-rate generator and address matching functionality. It can use the internal generic clock or an external clock. Using an external clock allows the SERCOM to be operated in all Sleep modes.

Related Links

34. SERCOM USART - SERCOM Synchronous and Asynchronous Receiver and Transmitter
35. SERCOM SPI – SERCOM Serial Peripheral Interface
36. SERCOM I2C – Inter-Integrated Circuit
6.2.6 SERCOM I2C Configurations

33.2 Features

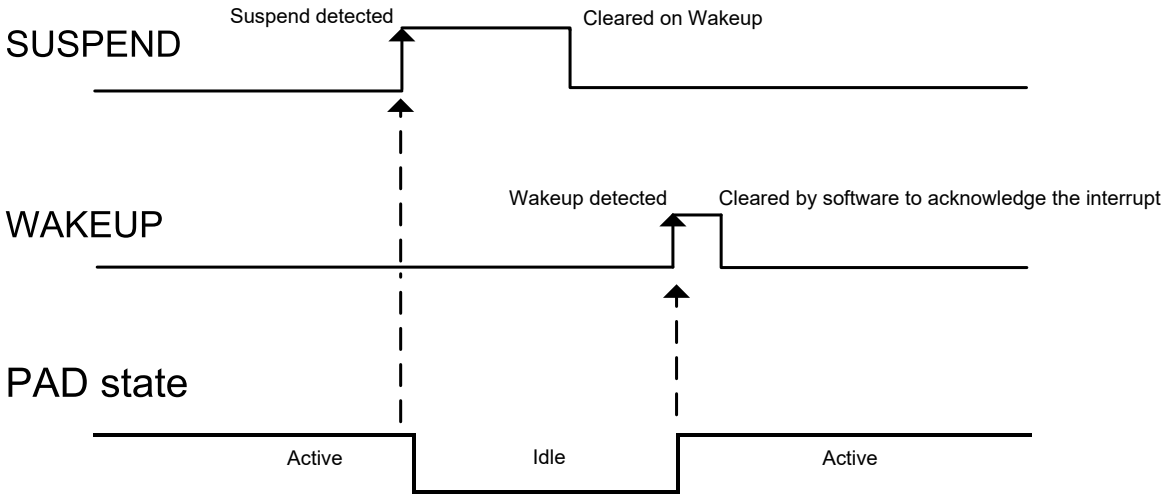
- Interface for configuring into one of the following:
  - Inter-Integrated Circuit (I^2C) Two-wire Serial Interface
  - System Management Bus (SMBus™) compatible
  - Serial Peripheral Interface (SPI)
  - Universal Synchronous/Asynchronous Receiver/Transmitter (USART)
- Single transmit buffer and double receive buffer
- Baud-rate generator
- Address match/mask logic
- Operational in all Sleep modes with an external clock source
- Can be used with DMA
- 32-bit extension for better system bus utilization

See the Related Links for full feature lists of the interface configurations.

Related Links

34. SERCOM USART - SERCOM Synchronous and Asynchronous Receiver and Transmitter
35. SERCOM SPI – SERCOM Serial Peripheral Interface
36. SERCOM I2C – Inter-Integrated Circuit
6.2.6 SERCOM I2C Configurations

Figure 38-8. Pad Events



The Suspend Interrupt bit in the Device Interrupt Flag register (INTFLAG.SUSPEND) is set when a USB Suspend state has been detected on the USB bus. The USB pad is then automatically put in the Idle state. The detection of a non-idle state sets the Wake Up Interrupt bit (INTFLAG.WAKEUP) and wakes the USB pad.

The pad goes to the Idle state if the USB module is disabled or if CTRLB.DETACH is written to one. It returns to the Active state when CTRLA.ENABLE is written to one and CTRLB.DETACH is written to zero.

38.6.2.14 Remote Wakeup

The remote wakeup request (also known as upstream resume) is the only request the device may send on its own initiative. This should be preceded by a DEVICE_REMOTE_WAKEUP request from the host.

First, the USB must have detected a “Suspend” state on the bus, i.e. the remote wakeup request can only be sent after INTFLAG.SUSPEND has been set.

The user may then write a one to the Remote Wakeup bit (CTRLB.UPRSM) to send an Upstream Resume to the host initiating the wakeup. This will automatically be done by the controller after 5 ms of inactivity on the USB bus.

When the controller sends the Upstream Resume INTFLAG.WAKEUP is set and INTFLAG.SUSPEND is cleared.

The CTRLB.UPRSM is cleared at the end of the transmitting Upstream Resume.

In case of a rebroadcast resume initiated by the host, the End of Resume bit (INTFLAG.EORSM) flag is set when the rebroadcast resume is completed.

38.8.7.5 Host Status Bank

Name: STATUS_BK
Offset: 0x0A & 0x1A
Reset: 0xxxxxxxx
Property: NA

Bit	7	6	5	4	3	2	1	0
							ERRORFLOW	CRCERR
Access							R/W	R/W
Reset							x	x

Bit 1 – ERRORFLOW Error Flow Status
This bit defines the Error Flow Status.

This bit is set when a Error Flow has been detected during transfer from/towards this bank.

For IN transfer, a NAK handshake has been received. For OUT transfer, a NAK handshake has been received. For Isochronous IN transfer, an overrun condition has occurred. For Isochronous OUT transfer, an underflow condition has occurred.

Value	Description
0	No Error Flow detected.
1	A Error Flow has been detected.

Bit 0 – CRCERR CRC Error
This bit defines the CRC Error Status.

This bit is set when a CRC error has been detected in an isochronous IN endpoint bank.

Value	Description
0	No CRC Error.
1	CRC Error detected.

40.8.4 Argument 1 Register

Name: ARG1R
Offset: 0x08
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ARG1[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
	ARG1[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

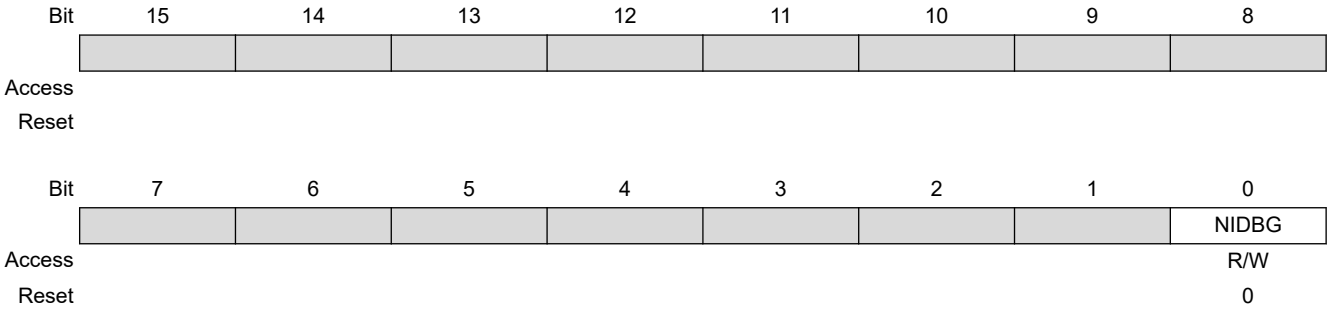
Bit	15	14	13	12	11	10	9	8
	ARG1[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ARG1[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 – ARG1[31:0] Argument 1
This register contains the SD command argument which is specified as the bit 39-8 of Command-Format in the “Physical Layer Simplified Specification V3.01” or “Embedded MultiMedia Card (e.MMC) Electrical Standard 4.51”.

40.8.42 Debug Register

Name: DBGR
Offset: 0x234
Reset: 0x00
Property: -



Bit 0 – NIDBG Non-Intrusive Debug

Value	Name	Description
0	DISABLED	Reading the BDPR via debugger increments the dual port RAM read pointer.
1	ENABLED	Reading the BDPR via debugger does not increment the dual port RAM read pointer.

- {nu1XBase, u2XLength}, {nu1ZBase, 2*u2XLength} or {nu1RBase, 2*u2XLength} are not in Crypto RAM
- u2XLength is either: < 4, > 0xffc or not a 32-bit length
- {nu1RBase, 2*u2XLength} overlaps {nu1XBase,u2XLength}
- {nu1RBase, 2*u2XLength} overlaps {nu1ZBase, 2*u2XLength} and nu1RBase >nu1ZBase

If a modular reduction is specified, the relevant parameters must be defined according to the chosen reduction and follow the description in 43.3.5.1 Modular Reduction. Additional constraints to be respected and error codes are described in this section and in Table 43-49.

Multiplication with Accumulation or Subtraction

Where the options bits specify that either an Accumulation or a subtraction should be performed, this command performs the following operation:

$R = (Z \pm (X^2 + CarryOperand)) mod\ B^{2 \times XLength}$

Table 43-32. Multiplication with Accumulation or Subtraction

Option AND CARRYOPTIONS	CarryOperand	Resulting Operation
SET_CARRYOPTION(ADD_CARRY)	CarryIn	R = Z ± (X^2 + CarryIn)
SET_CARRYOPTION(SUB_CARRY)	- CarryIn	R = Z ± (X^2 - CarryIn)
SET_CARRYOPTION(ADD_1_PLUS_CARRY)	1 + CarryIn	R = Z ± (X^2 + 1 + CarryIn)
SET_CARRYOPTION(ADD_1_MINUS_CARRY)	1 - CarryIn	R = Z ± (X^2 + 1 - CarryIn)
SET_CARRYOPTION(CARRY_NONE)	0	R = Z ± (X^2)
SET_CARRYOPTION(ADD_1)	1	R = Z ± (X^2 + 1)
SET_CARRYOPTION(SUB_1)	- 1	R = Z ± (X^2 - 1)
SET_CARRYOPTION(ADD_2)	2	R = Z ± (X^2 + 2)

43.3.4.10.9 Multiplication without Accumulation or Subtraction

Where the options bits specify that either an accumulation or a subtraction should be performed, this command performs the following operation:

$R = (X^2 + CarryOperand) mod\ B^{2 \times XLength}$

Table 43-33. Square Service Carry Settings

Option AND CARRYOPTIONS	CarryOperand	Resulting Operation
SET_CARRYOPTION(ADD_CARRY)	CarryIn	R = X^2 + CarryIn
SET_CARRYOPTION(SUB_CARRY)	- CarryIn	R = X^2 - CarryIn
SET_CARRYOPTION(ADD_1_PLUS_CARRY)	1 + CarryIn	R = X^2 + 1 + CarryIn
SET_CARRYOPTION(ADD_1_MINUS_CARRY)	1 - CarryIn	R = X^2 + 1 - CarryIn
SET_CARRYOPTION(CARRY_NONE)	0	R = X^2
SET_CARRYOPTION(ADD_1)	1	R = X^2 + 1
SET_CARRYOPTION(SUB_1)	- 1	R = X^2 - 1
SET_CARRYOPTION(ADD_2)	2	R = X^2 + 2

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})

43.3.6.4.4 Parameters Definition

Table 43-72. ZpEccDblFastService

Parameter	Type	Direction	Location	Data Length	Before Executing the Service	After Executing the Service
nu1ModBase	nu1	I	Crypto RAM	u2ModLength + 4	Base of modulus P	Base of modulus P
nu1CnsBase	nu1	I	Crypto RAM	u2ModLength + 8	Base of Cns	Base of Cns
u2ModLength	u2	I	–	–	Length of modulus P	Length of modulus P
nu1ABase	u2	I	Crypto RAM	u2ModLength + 4	Parameter a of the elliptic curve	Parameter a of the elliptic curve
nu1PointABase	nu1	I/O	Crypto RAM	3*u2ModLength + 12	Input point A (projective coordinates)	Resulting point C (projective coordinates)
nu1Workspace	nu1	I	Crypto RAM	4*u2ModLength + 28	–	Corrupted workspace

43.3.6.4.5 Code Example

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


PUKCL (u2Option) = 0;

PUKCL _ZpEccDbl(nu1ModBase) = <Base of the ram location of P>;
PUKCL _ZpEccDbl(u2ModLength) = <Byte length of P>;
PUKCL _ZpEccDbl(nu1CnsBase) = <Base of the ram location of Cns>;
PUKCL _ZpEccDbl(nu1PointABase) = <Base of the ram location of the A point>;
PUKCL _ZpEccDbl(nu1ABase) = <Base of the a parameter of the elliptic curve>;
PUKCL _ZpEccDbl(nu1Workspace) = <Base of the ram location of the workspace>;
...

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(ZpEccDblFast,&PUKCLParam);
if (PUKCL (u2Status) == PUKCL_OK)
            {
            ...
            }
else // Manage the error
```

43.3.6.4.6 Constraints

No overlapping between either input and output are allowed. The following conditions must be avoided to ensure that the service works correctly:

- nu1ModBase, nu1CnsBase, nu1PointABase, nu1ABase, nu1Workspace are not aligned on 32-bit boundaries
- {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength + 8}, {nu1PointABase, 3*u2ModLength+ 12}, {nu1ABase, u2ModLength + 4}, {nu1Workspace, <WorkspaceLength>} are not in Crypto RAM
- u2ModLength is either: < 12, > 0xffc or not a 32-bit length

43.3.8.3 Service Timing

The values in the following tables are estimated performances for CPU clock of 120 MHz. The CPU and PUKCC are operated at the same frequency. Due to possible change in the parameters values, the measurements show approximated values.

Other test conditions:

- PUKCL library data in Crypto RAM
- Test code and test data in SRAM
- ICache and DCache are disabled

43.3.8.3.1 Service Timing for RSA

RSA uses the ExpMod service for encryption and decryption. Following tables show service timing, where 'W' indicates window size.

Table 43-114. RSA1024

Operation	Clock Cycles	Timing one block
RSA 1024 decryption / signature generation. No CRT, Regular implementation, W=4	3.05 MCycles	25.42 ms
RSA 1024 decryption / signature generation. With CRT, Regular implementation, W=4	1.04 MCycles	8.67 ms
RSA 1024 encryption / signature verification. No CRT, Fast implementation, W=1 Exponent=3	0.07 MCycles	0.58 ms
RSA 1024 encryption / signature verification. No CRT, Fast implementation, W=1 Exponent=0x10001	0.07 MCycles	0.58 ms

Table 43-115. RSA2048

Operation	Clock Cycles	Timing One block
RSA 2048 decryption / signature generation. No CRT, Regular implementation, W=4	21.9 MCycles	182 ms
RSA 2048 decryption / signature generation. With CRT, Regular implementation, W=4	6.19 MCycles	51.6 ms
RSA 2048 encryption / signature verification. No CRT, Fast implementation, W=1 Exponent=3	0.24 MCycles	2 ms
RSA 2048 encryption / signature verification. No CRT, Fast implementation, W=1 Exponent=0x10001	0.24 MCycles	2 ms

47.8.14 Data Buffer DAC1

Name: DATABUF1
Offset: 0x16
Reset: 0x0000
Property: Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	DATABUF[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATABUF[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – DATABUF[15:0] DAC1 Data Buffer
DATABUF1 contains the value to be transferred into DATA1 when a START1 event occurs.

48.7.2 Register Summary - 16-bit Mode

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST
		15:8	DMAOS				ALOCK	PRESCALER[2:0]		
		23:16			COPEN1	COPEN0			CAPTEN1	CAPTEN0
		31:24				CAPTMODE1[1:0]			CAPTMODE0[1:0]	
0x04	CTRLBCLR	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x05	CTRLBSET	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x06	EVCTRL	7:0			TCEI	TCINV		EVACT[2:0]		
		15:8			MCEO1	MCEO0				OVFEO
0x08	INTENCLR	7:0			MC1	MC0			ERR	OVF
0x09	INTENSET	7:0			MC1	MC0			ERR	OVF
0x0A	INTFLAG	7:0			MC1	MC0			ERR	OVF
0x0B	STATUS	7:0				CCBUFVx	PERBUFV		SLAVE	STOP
0x0C	WAVE	7:0							WAVEGEN[1:0]	
0x0D	DRVCTRL	7:0								INVENx
0x0E	Reserved									
0x0F	DBGCTRL	7:0								DBGRUN
0x10	SYNCBUSY	7:0		CCx		COUNT	STATUS	CTRLB	ENABLE	SWRST
		15:8								
		23:16								
		31:24								
0x14	COUNT	7:0	COUNT[7:0]							
		15:8	COUNT[15:8]							
0x16 ... 0x1B	Reserved									
0x1C	CC0	7:0	CC[7:0]							
		15:8	CC[15:8]							
0x1E	CC1	7:0	CC[7:0]							
		15:8	CC[15:8]							
0x20 ... 0x2F	Reserved									
0x30	CCBUF0	7:0	CCBUF[7:0]							
		15:8	CCBUF[15:8]							
0x32	CCBUF1	7:0	CCBUF[7:0]							
		15:8	CCBUF[15:8]							

Figure 49-26. Waveform Generation in RAMP1 mode with Restart Action

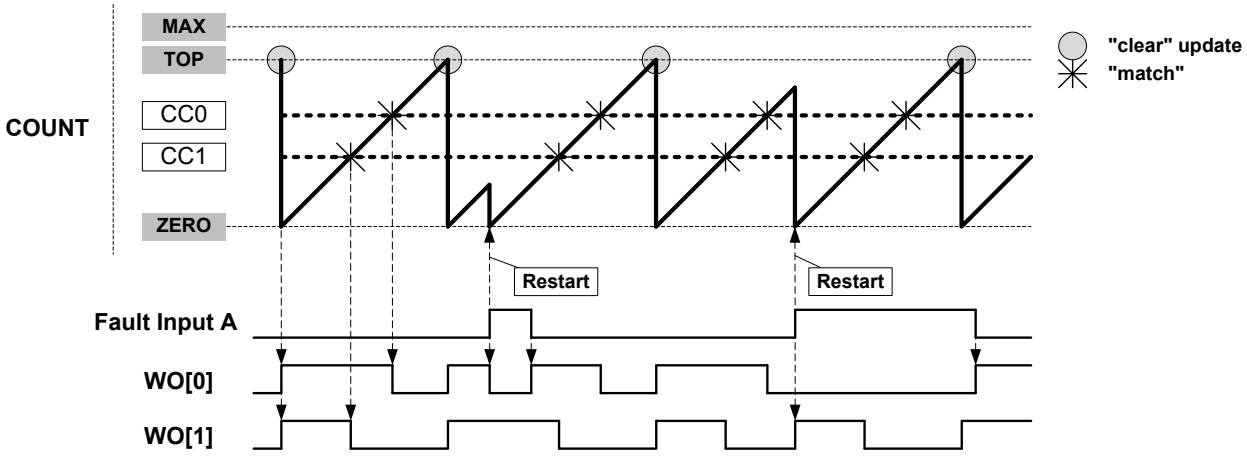
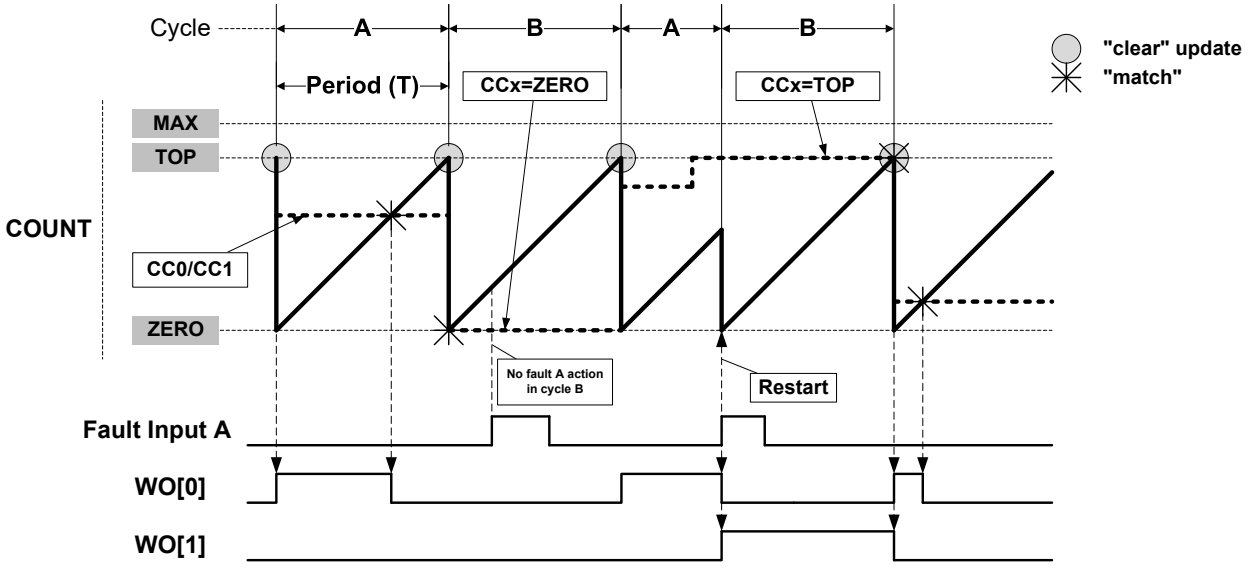


Figure 49-27. Waveform Generation in RAMP2 mode with Restart Action



Capture Action

Several capture actions can be selected by writing the Fault n Capture Action bits in the Fault n Control register (FCTRLn.CAPTURE). When one of the capture operations is selected, the counter value is captured when the fault occurs. These capture operations are available:

- CAPT - the equivalent to a standard capture operation, for further details refer to 49.6.2.7 Capture Operations
- CAPTMIN - gets the minimum time stamped value: on each new local minimum captured value, an event or interrupt is issued.
- CAPTMAX - gets the maximum time stamped value: on each new local maximum captured value, an event or interrupt (IT) is issued, see Figure 49-28.
- LOCMIN - notifies by event or interrupt when a local minimum captured value is detected.
- LOCMAX - notifies by event or interrupt when a local maximum captured value is detected.
- DERIV0 - notifies by event or interrupt when a local extreme captured value is detected, see Figure 49-29.

CTRLA.RESOLUTION	Bits [n:0]
0x0 - NONE	-
0x1 - DITH4	3:0
0x2 - DITH5	4:0
0x3 - DITH6	5:0 (depicted)

Refer to *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

Related Links

6. I/O Multiplexing and Considerations

50.5 System Dependencies

In order to use this Peripheral, configure the other components of the system as described in the following sections.

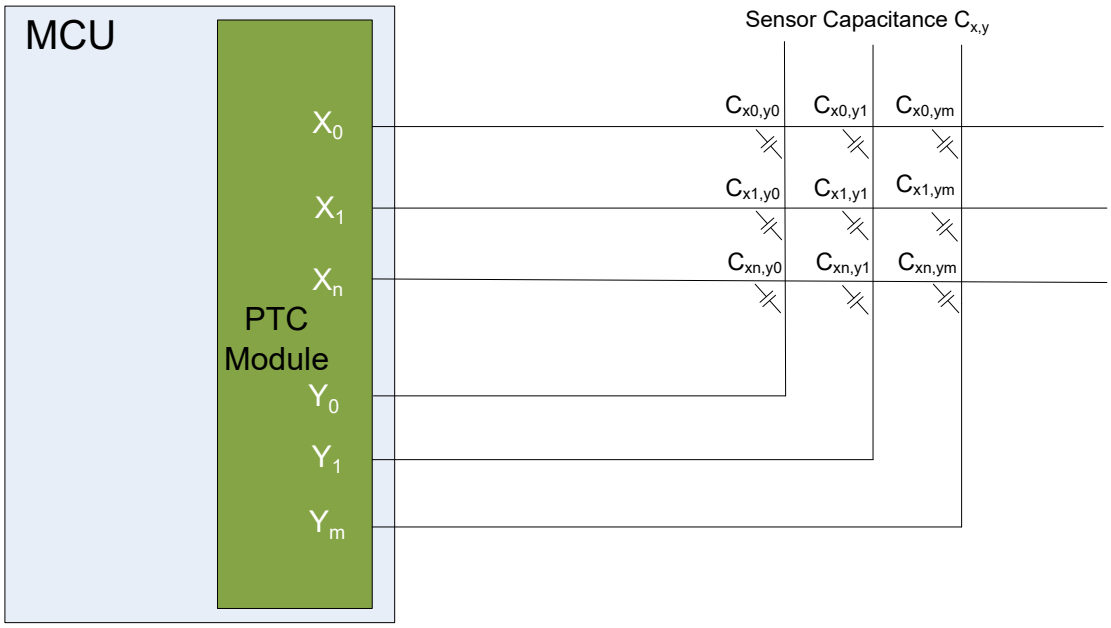
50.5.1 I/O Lines

The I/O lines used for analog X-lines and Y-lines must be connected to external capacitive touch sensor electrodes. External components are not required for normal operation. However, to improve the EMC performance, a series resistor of 1 kΩ or more can be used on X-lines and Y-lines.

50.5.1.1 Mutual-Capacitance Sensor Arrangement

A mutual-capacitance sensor is formed between two I/O lines - an X electrode for transmitting and Y electrode for sensing. The mutual capacitance between the X and Y electrode is measured by the Peripheral Touch Controller.

Figure 50-3. Mutual Capacitance Sensor Arrangement



50.5.1.2 Self-Capacitance Sensor Arrangement

A self-capacitance sensor is connected to a single pin on the Peripheral Touch Controller through the Y electrode for sensing the signal. The sense electrode capacitance is measured by the Peripheral Touch Controller.

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
V_{OUTmin}	Min Output Voltage	-	-	-	0.15	V
V_{OUTmax}	Max Output Voltage	-	V_{DDANA}-0.15	-	-	
V_{REF}	External Reference input	CTRLB.REFSEL[1:0]=0x2 (VREFAB)	1	-	V_{DDANA}-0.15	V
		CTRLB.REFSEL[1:0]=0x0 (VREFAU)	1	-	V_{DDANA}	
C_{VREF}	External decoupling capacitor	-	-	220	-	nF
C_{LOAD}	Output capacitor load	-	-	-	50	pF
R_{LOAD}	Output resistance load	-	5	-	-	kΩ
t_S	Settling time	For reaching ±1LSB of the final value. Step size < 500 LSB - C_{load} = 50pF	-	-	1	µs
t_{S_FS}	Settling time 0x080 to 0xF7F	For reaching ±1LSB of the final value. Step size from 0% to 100% - C_{load} = 50pF	-	5	7	µs

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.

Table 54-29. Differential Mode

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
INL	Integral Non Linearity, Best-fit curve from 0x080 to 0xF7F	i12clk = 12 MHz, V_{DDANA} = 3.0V, External Ref. = 2.0V, C_{LOAD} = 50 pF	-	±2.4	±3.4	LSB
		i12clk = 12 MHz, V_{DDANA} = 3.0V - 1V Internal Ref (1) = 2.0V, C_{LOAD} = 50 pF	-	±3.2	±4.2	
DNL	Differential Non Linearity, Best-fit curve from 0x080 to 0xF7F	i12clk = 12 MHz, V_{DDANA} = 3.0V,External Ref. = 2.0V, C_{LOAD} = 50 pF	-	±2.4	±3.6	LSB
		i12clk = 12 MHz, V_{DDANA} = 3.0V - 1V Internal Ref (1) = 2.0V, C_{LOAD} = 50 pF	-	±3.5	±4.4	
Gerr	Gain Error	External Reference voltage	-	±0.4	±1.7	% FSR
		1.0V Internal Reference voltage	-	±0.8	±8.0	
Offerr	Offset Error	External Reference voltage	-	±13	±40	mV
		1.0V Internal Reference voltage	-	±8	±74	
ENOB	Effective Number Of Bits	Fs = 1 Ms/s - External Ref - CCTRL = 0x2	9.9	10.7	10.9	Bits
SNR	Signal to Noise ratio		63.5	68.6	72.6	dB
THD	Total Harmonic Distortion		-79.1	-72.5	-61.0	dB

Note: Specified only at Temp > 0°C when 1V internal reference is used.