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	81
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 28x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TQFP
Supplier Device Package	100-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsame51n20a-au

14.5.3 Clocks

The GCLK bus clock (CLK_GCLK_APB) can be enabled and disabled in the Main Clock Controller.

Related Links

15.6.2.6 Peripheral Clock Masking

29. OSC32KCTRL – 32KHz Oscillators Controller

14.5.4 DMA

Not applicable.

14.5.5 Interrupts

Not applicable.

14.5.6 Events

Not applicable.

14.5.7 Debug Operation

When the CPU is halted in debug mode the GCLK continues normal operation. If the GCLK is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

14.5.8 Register Access Protection

All registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC).

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

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

Related Links

27. PAC - Peripheral Access Controller

14.5.9 Analog Connections

Not applicable.

14.6 Functional Description

14.6.1 Principle of Operation

The GCLK module is comprised of twelve Generic Clock Generators (Generators) sourcing up to 64 Peripheral Channels and the Main Clock signal GCLK_MAIN.

A clock source selected as input to a Generator can either be used directly, or it can be prescaled in the Generator. A generator output is used by one or more Peripheral Channels to provide a peripheral generic clock signal (GCLK_PERIPH) to the peripherals.

14.6.2 Basic Operation

14.6.2.1 Initialization

Before a Generator is enabled, the corresponding clock source should be enabled. The Peripheral clock must be configured as outlined by the following steps:

17.8.6 Debug Control

Name: DBGCTRL
Offset: 0x0F
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
							ECCELOG	ECCDIS
Access							R/W	R/W
Reset							0	0

Bit 1 – ECCELOG ECC Error Log
When DBGCTRL.ECCDIS=0, This bit controls whether ECC errors are logged in the INTFLAG register.
When DBGCTRL.ECCDIS=1, this bit has no meaning.

Value	Description
0	ECC errors for debugger reads are not logged.
1	ECC errors for debugger reads are logged if DBGCTRL.ECCDIS=0.

Bit 0 – ECCDIS ECC Disable
By default, ECC errors during debugger reads are corrected and logged based on DBGCTRL.ECCELOG.
Setting this bit will disable ECC correction and logging.

Value	Description
0	ECC errors are are corrected for debugger reads and logged based on DBGCTRL.ECCELOG.
1	ECC errors are masked for debugger reads.

20. WDT – Watchdog Timer

20.1 Overview

The Watchdog Timer (WDT) is a system function for monitoring correct program operation. It makes it possible to recover from error situations such as runaway or deadlocked code. The WDT is configured to a predefined time-out period, and is constantly running when enabled. If the WDT is not cleared within the time-out period, it will issue a system reset. An early-warning interrupt is available to indicate an upcoming watchdog time-out condition.

The window mode makes it possible to define a time slot (or window) inside the total time-out period during which the WDT must be cleared. If the WDT is cleared outside this window, either too early or too late, a system reset will be issued. Compared to the normal mode, this can also catch situations where a code error causes the WDT to be cleared frequently.

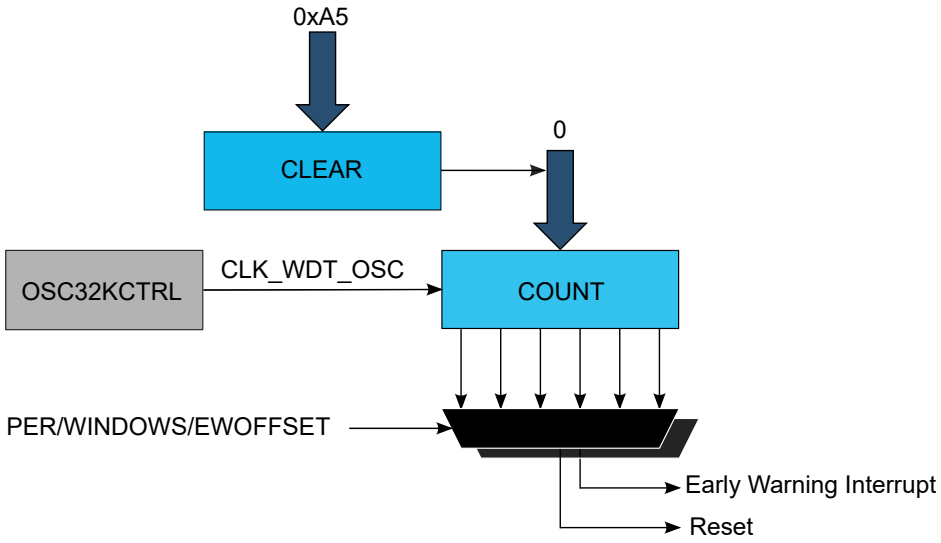
When enabled, the WDT will run in active mode and all sleep modes. It is asynchronous and runs from a CPU-independent clock source. The WDT will continue operation and issue a system reset or interrupt even if the main clocks fail.

20.2 Features

- Issues a system reset if the Watchdog Timer is not cleared before its time-out period
- Early Warning interrupt generation
- Asynchronous operation from dedicated oscillator
- Two types of operation
  - Normal
  - Window mode
- Selectable time-out periods
  - From 8 cycles to 16,384 cycles in Normal mode
  - From 16 cycles to 32,768 cycles in Window mode
- Always-On capability

20.3 Block Diagram

Figure 20-1. WDT Block Diagram



20.4 Signal Description

Not applicable.

20.5 Product Dependencies

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

20.5.1 I/O Lines

Not applicable.

20.5.2 Power Management

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

Related Links

18. PM – Power Manager

20.5.3 Clocks

The WDT bus clock (CLK_WDT_APB) can be enabled and disabled (masked) in the Main Clock module (MCLK).

A 1 kHz oscillator clock (CLK_WDT_OSC) is required to clock the WDT internal counter.

CLK_WDT_OSC is sourced from the clock of the internal ultra-low-power oscillator, OSCULP32K. Due to the ultra-low-power design, the oscillator is not very accurate, and so the exact time-out period may vary from device to device. This variation must be kept in mind when designing software that uses the WDT to ensure that the time-out periods used are valid for all devices.

24.9.49 GMAC 1024 to 1518 Byte Frames Transmitted Register

Name: TBFT1518
Offset: 0x12C
Reset: 0x00000000
Property: Read-Only

Bit	31	30	29	28	27	26	25	24
	NFTX[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	NFTX[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NFTX[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
	NFTX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NFTX[31:0] 1024 to 1518 Byte Frames Transmitted without Error
This register counts the number of 1024 to 1518 byte frames successfully transmitted without error, i.e., no underrun and not too many retries.

24.9.84 GMAC 1588 Timer Seconds High Register

Name: TSH
Offset: 0x1C0
Reset: 0x00000000
Property: Read/Write

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
	TCS[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
	TCS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – TCS[15:0] Timer Count in Seconds
This register is writable. It increments by 1 when the IEEE 1588 nanoseconds counter counts to one second. It may also be incremented when the Timer Adjust Register is written.

24.9.95 GMAC PTP Peer Event Frame Transmitted Seconds Low Register

Name: PEFTSL
Offset: 0x1F0
Reset: 0x00000000
Property: Read-Only

Bit	31	30	29	28	27	26	25	24
	RUD[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RUD[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	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 31:0 – RUD[31:0] Register Update
The register is updated with the value that the IEEE 1588 Timer Seconds Register holds when the SFD of a PTP transmit peer event crosses the MII interface. An interrupt is issued when the register is updated.

25.8.14 SmartEEPROM Status

Name: SEESTAT
Offset: 0x2C
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
						PSZ[2:0]		
Access						R	R	R
Reset						0	0	x

Bit	15	14	13	12	11	10	9	8
					SBLK[3:0]			
Access					R	R	R	R
Reset					0	0	0	x

Bit	7	6	5	4	3	2	1	0
				RLOCK	LOCK	BUSY	LOAD	ASEES
Access				R	R	R	R	R
Reset				0	x	0	0	x

Bits 18:16 – PSZ[2:0] SmartEEPROM Page Size
This bit field is automatically loaded from the user page during startup.

Indicates the page size. Not all device families will provide all the page sizes indicated in the table.

Bits 11:8 – SBLK[3:0] Blocks Number In a Sector
This bit field is automatically loaded from the user page during startup.

Indicates the number of blocks allocated to a SEES.

Bit 4 – RLOCK RLOCK
SmartEEPROM Write Access To Register Address Space Is Locked

Bit 3 – LOCK SmartEEPROM Section Locked
This bit field is automatically loaded from the user page during startup.

Access to the SmartEEPROM data is locked. Writes to AHB2 throws hardfault exceptions.

0: SmartEEPROM access is not locked

1: SmartEEPROM access is locked

Bit 2 – BUSY Busy
0: SmartEEPROM is ready.

Bit 30 – WRPINCFG Write PINCFG
This bit determines whether the atomic write operation will update the Pin Configuration register (PINCFGy) or not for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits.

Writing '0' to this bit has no effect.

Writing '1' to this bit updates the configuration of the selected pins with the written WRCONFIG.DRVSTR, WRCONFIG.PULLEN, WRCONFIG.INEN, WRCONFIG.PMUXEN, and WRCONFIG.PINMASK values.

This bit will always read as zero.

Value	Description
0	The PINCFGy registers of the selected pins will not be updated.
1	The PINCFGy registers of the selected pins will be updated.

Bit 28 – WRPMUX Write PMUX
This bit determines whether the atomic write operation will update the Peripheral Multiplexing register (PMUXn) or not for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits.

Writing '0' to this bit has no effect.

Writing '1' to this bit updates the pin multiplexer configuration of the selected pins with the written WRCONFIG. PMUX value.

This bit will always read as zero.

Value	Description
0	The PMUXn registers of the selected pins will not be updated.
1	The PMUXn registers of the selected pins will be updated.

Bits 27:24 – PMUX[3:0] Peripheral Multiplexing
These bits determine the new value written to the Peripheral Multiplexing register (PMUXn) for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPMUX bit is set.

These bits will always read as zero.

Bit 22 – DRVSTR Output Driver Strength Selection
This bit determines the new value written to PINCFGy.DRVSTR for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPINCFG bit is set.

This bit will always read as zero.

Bit 18 – PULLEN Pull Enable
This bit determines the new value written to PINCFGy.PULLEN for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPINCFG bit is set.

This bit will always read as zero.

Bit 17 – INEN Input Enable
This bit determines the new value written to PINCFGy.INEN for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPINCFG bit is set.

This bit will always read as zero.

Figure 33-3. Baud Rate Generator

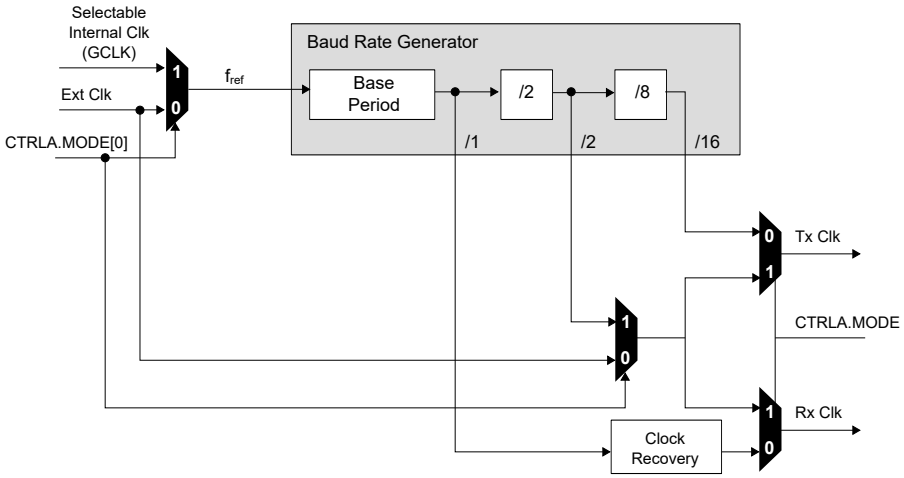


Table 33-2 contains equations for the baud rate (in bits per second) and the BAUD register value for each operating mode.

For asynchronous operation, there is one mode: *arithmetic mode*, the BAUD register value is 16 bits (0 to 65,535).*fractional mode*, the BAUD register value is 13 bits, while the fractional adjustment is 3 bits. In this mode the BAUD setting must be greater than or equal to 1.

For synchronous operation, the BAUD register value is 8 bits (0 to 255).

Table 33-2. Baud Rate Equations

Operating Mode	Condition	Baud Rate (Bits Per Second)	BAUD Register Value Calculation
Asynchronous Arithmetic	$f_{BAUD} \leq \frac{f_{ref}}{16}$	$f_{BAUD} = \frac{f_{ref}}{16}\left(1 - \frac{BAUD}{65536}\right)$	$BAUD = 65536 \cdot \left(1 - 16 \cdot \frac{f_{BAUD}}{f_{ref}}\right)$
Asynchronous Fractional	$f_{BAUD} \leq \frac{f_{ref}}{S}$	$f_{BAUD} = \frac{f_{ref}}{S \cdot \left(BAUD + \frac{FP}{8}\right)}$	$BAUD = \frac{f_{ref}}{S \cdot f_{BAUD}} - \frac{FP}{8}$
Synchronous	$f_{BAUD} \leq \frac{f_{ref}}{2}$	$f_{BAUD} = \frac{f_{ref}}{2 \cdot (BAUD + 1)}$	$BAUD = \frac{f_{ref}}{2 \cdot f_{BAUD}} - 1$

S - Number of samples per bit, which can be 16, 8, or 3.

The Asynchronous Fractional option is used for auto-baud detection.

The baud rate error is represented by the following formula:

$$\text{Error} = 1 - \left(\frac{\text{ExpectedBaudRate}}{\text{ActualBaudRate}}\right)$$

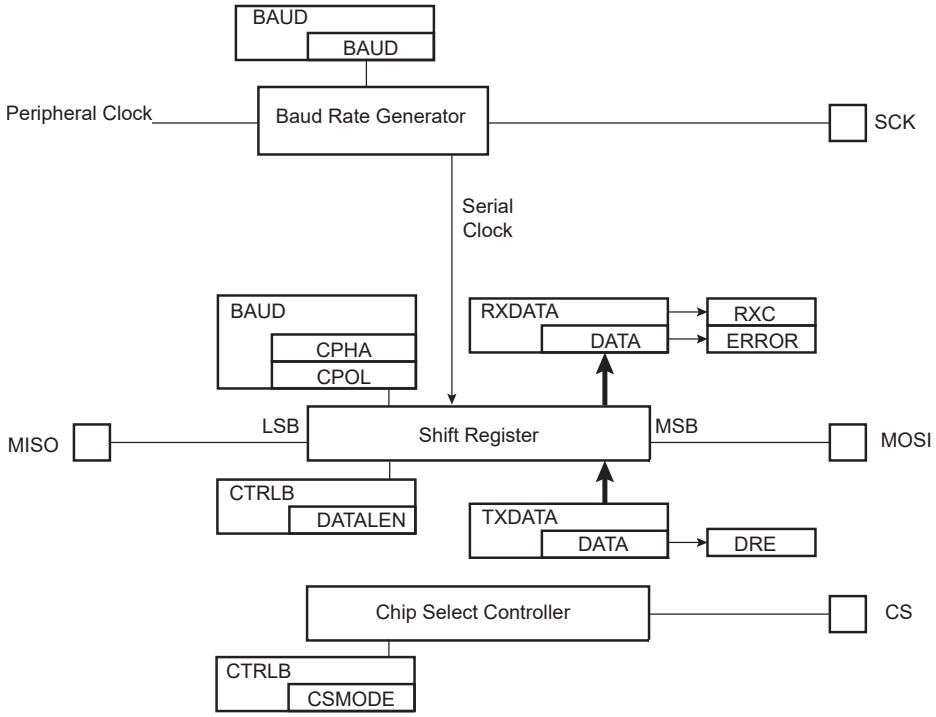
33.6.2.3.1 Asynchronous Arithmetic Mode BAUD Value Selection

The formula given for f_{BAUD} calculates the average frequency over 65536 f_{ref} cycles. Although the BAUD register can be set to any value between 0 and 65536, the actual average frequency of f_{BAUD} over a single frame is more granular. The BAUD register values that will affect the average frequency over a single frame lead to an integer increase in the cycles per frame (CPF)

$$CPF = \frac{f_{ref}}{f_{BAUD}}(D + S)$$

37.6.7.2 SPI Mode Block Diagram

Figure 37-6. SPI Mode Block Diagram



In case of a successful frame transmission, and if storage of Tx events is enabled, a Tx Event FIFO element is written with Event Type ET = “10” (transmission in spite of cancellation).

39.6.2.8 Test Modes

To enable write access to register TEST, bit CCCR.TEST has to be set to ‘1’. This allows the configuration of the test modes and test functions.

Four output functions are available for the CAN transmit pin CAN_TX by programming TEST.TX. Additionally to its default function – the serial data output – it can drive the CAN Sample Point signal to monitor the CAN’s bit timing and it can drive constant dominant or recessive values. The actual value at pin CAN_RX can be read from TEST.RX. Both functions can be used to check the CAN bus’ physical layer.

Due to the synchronization mechanism between GCLK_CAN and GCLK_CAN_APB domains, there may be a delay of several GCLK_CAN_APB periods between writing to TEST.TX until the new configuration is visible at output pin CAN_TX. This applies also when reading input pin CAN_RX via TEST.RX.

Note: Test modes should be used for production tests or self test only. The software control for pin CAN_TX interferes with all CAN protocol functions. It is not recommended to use test modes for application.

External Loop Back Mode

The CAN can be set in External Loop Back Mode by programming TEST.LBCK to ‘1’. In Loop Back Mode, the CAN treats its own transmitted messages as received messages and stores them (if they pass acceptance filtering) into an Rx Buffer or an Rx FIFO. The figure below shows the connection of signals CAN_TX and CAN_RX to the CAN in External Loop Back Mode.

This mode is provided for hardware self-test. To be independent from external stimulation, the CAN ignores acknowledge errors (recessive bit sampled in the acknowledge slot of a data/remote frame) in Loop Back Mode. In this mode the CAN performs an internal feedback from its Tx output to its Rx input. The actual value of the CAN_RX input pin is disregarded by the CAN. The transmitted messages can be monitored at the CAN_TX pin.

Internal Loop Back Mode

Internal Loop Back Mode is entered by programming bits TEST.LBCK and CCCR.MON to ‘1’. This mode can be used for a “Hot Selftest”, meaning the CAN can be tested without affecting a running CAN system connected to the pins CAN_TX and CAN_RX. In this mode pin CAN_RX is disconnected from the CAN and pin CAN_TX is held recessive. The figure below shows the connection of CAN_TX and CAN_RX to the CAN in case of Internal Loop Back Mode.

39.8.23 Extended ID AND Mask

Name: XIDAM
Offset: 0x90
Reset: 0x1FFFFFFF
Property: Write-restricted

This register is write-restricted and only writable if bit fields CCCR.CCE = 1 and CCCR.INIT = 1.

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

Bit	23	22	21	20	19	18	17	16
	EIDM[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
	EIDM[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

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

Bits 28:0 – EIDM[28:0] Extended ID Mask
For acceptance filtering of extended frames the Extended ID AND Mask is ANDed with the Message ID of a received frame. Intended for masking of 29-bit IDs in SAE J1939. With the reset value of all bits set to one the mask is not active.

Table 43-116. RSA4096

Operation	Clock Cycles	Timing One block
RSA 4096 Decryption / signature generation. No CRT, Regular implementation, W=1	208 MCycles	1.73s
RSA 4096 Decryption / signature generation. With CRT, Regular implementation, W=3	45.5 MCycles	379 ms
RSA 4096 encryption / signature verification. No CRT, Fast implementation, W=1 Exponent=3	0.92 MCycles	7.67 ms
RSA 4096 encryption / signature verification. No CRT, Fast implementation, W=1 Exponent=0x10001	0.92 MCycles	7.67 ms

43.3.8.3.2 Service Timing for Prime Generation

Prime generation uses the PrimeGen service.

Table 43-117. Prime Generation

Operation	Clock Cycles	Timing One Block
Regular Generation of two primes, Prime_Length=512 bits, W=4, Rabin Miller Iterations Number = 3, (average of 200 samples)	Mean = 47.4 MCycles	Mean = 0.40s
Regular Generation of two primes, Prime_Length=512 bits, W=4, Rabin Miller Iterations Number = 3, (Standard Deviation for 200 samples)	Std Dev = 30.3 Mcycles	Std Dev = 0.25s
Regular Generation of two primes, Prime_Length=1024 bits, W=4, Rabin Miller Iterations Number = 3, (average of 200 samples)	Mean = 448 MCycles	Mean = 3.73s
Regular Generation of two primes, Prime_Length=1024 bits, W=4, Rabin Miller Iterations Number = 3, (Standard Deviation for 200 samples)	Std Dev = 294 Mcycles	Std Dev = 2.45s
Regular Generation of two primes, Prime_Length=2048 bits, W=4, Rabin Miller Iterations Number = 3, (average of 200 samples)	Mean = 4.78 GCycles	Mean = 39.8s
Regular Generation of two primes, Prime_Length=2048 bits, W=4, Rabin Miller Iterations Number = 3, (Standard Deviation for 200 samples)	Std Dev = 3,05 GCycles	Std Dev = 25.4s

43.3.8.3.3 Service Timing for ECDSA on Prime Field

In the following table, ECDSA signature generation uses the ZpEcDsaGenerateFast service and signature verification uses ZpEcDsaQuickVerify

Condition	Value	Action when DMA writes to DSEQDATA
		• INPUTCTRL ← DSEQDATA[15:0] if DSEQSTAT.INPUTCTRL = 1 • CTRLB ← DSEQDATA[31:16] if DSEQSTAT.CTRLB = 1
DSEQSTAT.REFCTRL or DSEQSTAT.AVGCTRL or DSEQSTAT.SAMPCTRL	0	• No DMA trigger is generated • No data in the memory must be reserved
	1	• A DMA trigger is generated • One word (32-bit) must be reserved in the memory • REFCTRL ← DSEQDATA[7:0] if DSEQSTAT.REFCTRL = 1 • AVGCTRL ← DSEQDATA[23:16] if DSEQSTAT.AVGCTRL = 1 • SAMPCTRL ← DSEQDATA[31:24] if DSEQSTAT.SAMPCTRL = 1
DSEQSTAT.WINLT or DSEQSTAT.WINUT	0	• No DMA trigger is generated • No data in the memory must be reserved
	1	• A DMA trigger is generated • One word (32-bit) must be reserved in the memory • WINLT ← DSEQDATA[15:0] if DSEQSTAT.WINLT = 1 • WINUT ← DSEQDATA[31:16] if DSEQSTAT.WINUT = 1
DSEQSTAT.GAINCORR or DSEQSTAT.OFFSETCORR	0	• No DMA trigger is generated • No data in the memory must be reserved
	1	• A DMA trigger is generated • One word (32-bit) must be reserved in the memory • GAINCORR ← DSEQDATA[15:0] if DSEQSTAT.GAINCORR = 1 • OFFSETCORR ← DSEQDATA[31:16] if DSEQSTAT.OFFSETCORR = 1

The DMA Sequential Status register (DSEQSTAT) stores the remaining registers to be updated by the DMA. During a sequence and when a write access to the DSEQDATA register is detected, the DSEQSTAT bits which were source of the corresponding DSEQ trigger will be cleared. When all DSEQSTAT bits are zero (except BUSY bit), the DSEQCTRL register bits (except AUTOSTART) are copied into the DSEQSTAT register and a new DMA sequence is started when a new ADC conversion starts.

DMA Descriptor Setup and Data Memory Organization

When DMA sequencing is enabled, the DMA Controller (DMAC) must be configured in the following way:

- Select 32-bit beat size transfer (DMAC.BTCTRL.BEATSIZE=WORD).

46.8.7 Status A

Name: STATUSA
Offset: 0x07
Reset: 0x00
Property: Read-Only

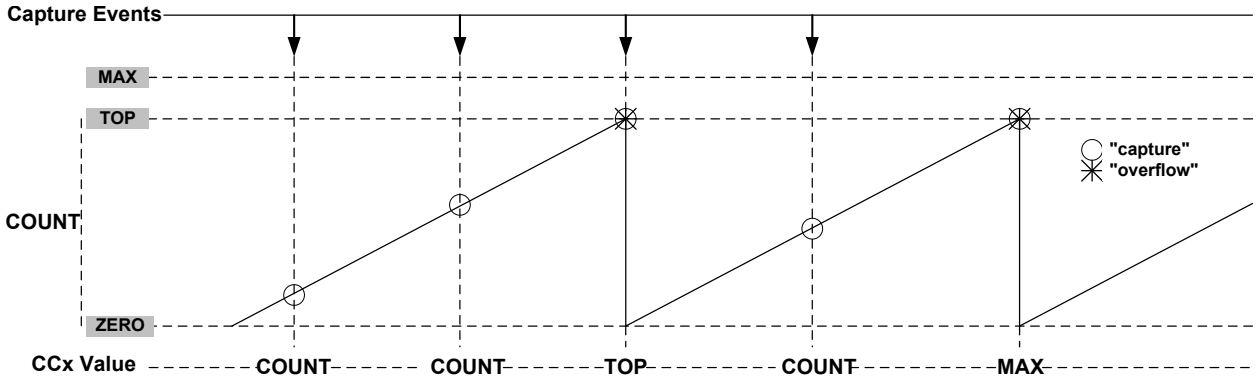
Bit	7	6	5	4	3	2	1	0
			WSTATE0[1:0]				STATEx	STATEx
Access			R	R			R	R
Reset			0	0			0	0

Bits 5:4 – WSTATE0[1:0] Window 0 Current State
These bits show the current state of the signal if the window 0 mode is enabled.

Value	Name	Description
0x0	ABOVE	Signal is above window
0x1	INSIDE	Signal is inside window
0x2	BELOW	Signal is below window
0x3		Reserved

Bits 1,0 – STATEx Comparator x Current State
This bit shows the current state of the output signal from COMPx. STATEx is valid only when STATUSB.READYx is one.

Figure 48-15. Time-Stamp



48.6.3.3 Minimum Capture

The minimum capture is enabled by writing the CAPTMIN mode in the Channel n Capture Mode bits in the Control A register (CTRLA.CAPTMODEn = CAPTMIN).

CCx Content:

In CAPTMIN operations, CCx keeps the Minimum captured values. Before enabling this mode of capture, the user must initialize the corresponding CCx register value to a value different from zero. If the CCx register initial value is zero, no captures will be performed using the corresponding channel.

MCx Behaviour:

In CAPTMIN operation, capture is performed only when on capture event time, the counter value is lower than the last captured value. The MCx interrupt flag is set only when on capture event time, the counter value is upper or equal to the value captured on the previous event. So interrupt flag is set when a new absolute local Minimum value has been detected.

48.6.3.4 Maximum Capture

The maximum capture is enabled by writing the CAPTMAX mode in the Channel n Capture Mode bits in the Control A register (CTRLA.CAPTMODEn = CAPTMAX).

CCx Content:

In CAPTMAX operations, CCx keeps the Maximum captured values. Before enabling this mode of capture, the user must initialize the corresponding CCx register value to a value different from TOP. If the CCx register initial value is TOP, no captures will be performed using the corresponding channel.

MCx Behaviour:

In CAPTMAX operation, capture is performed only when on capture event time, the counter value is upper than the last captured value. The MCx interrupt flag is set only when on capture event time, the counter value is lower or equal to the value captured on the previous event. So interrupt flag is set when a new absolute local Maximum value has been detected.

Bit 11 – DFS Non-Recoverable Debug Fault State Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after an Debug Fault State occurs.

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

Writing a '1' to this bit clears the Debug Fault State interrupt flag.

Bit 3 – ERR Error Interrupt Flag
This flag is set if a new capture occurs on a channel when the corresponding Match or Capture Channel x interrupt flag is one. In which case there is nowhere to store the new capture.

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

Writing a '1' to this bit clears the error interrupt flag.

Bit 2 – CNT Counter Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a counter event occurs.

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

Writing a '1' to this bit clears the CNT interrupt flag.

Bit 1 – TRG Retrigger Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a counter retrigger occurs.

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

Writing a '1' to this bit clears the re-trigger interrupt flag.

Bit 0 – OVF Overflow Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after an overflow condition occurs.

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

Writing a '1' to this bit clears the Overflow interrupt flag.

51.9.8 Rx Serializer Control

Name: RXCTRL
Offset: 0x24
Reset: 0x00000000
Property: Enable-Protected, PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
						RXLOOP	DMA	MONO
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	SLOTDISx	SLOTDISx	SLOTDISx	SLOTDISx	SLOTDISx	SLOTDISx	SLOTDISx	SLOTDISx
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
	BITREV	EXTEND[1:0]		WORDADJ		DATASIZE[2:0]		
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

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

Bit 26 – RXLOOP Loop-back Test Mode
This bit enables a loop-back test mode:

Value	Description
0	Each Receiver uses its SDn pin as input (default mode).
1	Receiver uses as input the transmitter output of the other Serializer in the pair: e.g. SD1 for SD0 or SD0 for SD1.

Bit 25 – DMA Single or Multiple DMA Channels
This bit selects whether even- and odd-numbered slots use separate DMA channels or the same DMA channel.

DMA	Name	Description
0x0	SINGLE	Single DMA channel
0x1	MULTIPLE	One DMA channel per data channel

53.6.3.3 Register Lock Update

Prescaler (PRESC), FILTER, and CCx registers are buffered (PRESCBUF, FILTERBUF, CCBUFx registers, respectively). When a new value is written in a buffer register, the corresponding Buffer Valid bit is set in the Buffer Status register (STATUS.FILTERBUFV, STATUS.PRESCBUFV, STATUS.CCBUFVx).

By default, a register is updated with the its buffer register's value on UPDATE condition, which represents:

- The next filter transition in QDEC and HALL mode of operation
- The overflow/underflow or re-trigger event detection in COUNT mode of operation

The buffer valid flags in the STATUS register are automatically cleared by hardware when the data is copied from the buffer to the corresponding register.

It is possible to lock the updates by writing a '1' to the Lock Update bit in Control B Set register (CTRLBSET.LUPD).

The lock feature is disabled by writing a '1' to the Lock Update bit in Control B Clear register (CTRLBCLR.LUPD). When a buffer valid status flag is '1' and updating is not locked, the data from the buffer register will be copied into the corresponding register on UPDATE condition.

It is also possible to modify the LUPD bit behavior by hardware, by writing a '1' to the Auto-lock bit in Control A register (CTRLA.ALOCK). When the bit is '1', the Lock Update bit in Control B register (CTRLBSET.LUPD) is set when the UPDATE condition is detected.

53.6.3.4 Software Command and Event Actions

The PDEC peripheral supports software commands and event actions. The software commands are applied by the Software Command bit field in the Control B register (CTRLBSET.CMD, CTRLBCLR.CMD). The event actions are available in the Event Action bit-field in Event Control register (EVCTRL.EVACT).

53.6.3.4.1 Re-trigger Software Command or Event Action

A re-trigger command can be issued from software by using PDEC Command bits in Control B Set register (CTRLBSET.CMD = RETRIGGER) or when the re-trigger event action is configured in the Input Event Action bits in Event Control register (EVCTRL.EVACT = RETRIGGER) and an event is detected by hardware.

When the re-trigger command is detected during counting operation, the counter will be reloaded or cleared, depending on the counting direction (DIR). If the re-trigger command is detected when the counter is stopped, the counter will resume counting operation from the value in the COUNT register.

Note: When re-trigger event action is enabled, enabling the counter will not start the counter. The counter will start on the next incoming event and restart on any following event.

53.6.3.4.2 Count Event Action

The count action can be selected in the Event Control register (EVCTRL.EVACT) and can be used to count external events. When an event is received, the counter increments the value.

53.6.3.4.3 Force Update Software Command

A Force Update command can be issued by writing the PDEC Command bits in Control B Set register (CTRLBSET.CMD = UPDATE). When the command is issued, the buffered registers will be updated.

53.6.3.4.4 Force Read Synchronization Software Command

A Force Read Synchronization command can be issued writing the PDEC Command bits in Control B Set register (CTRLBSET.CMD = READSYNC). When the command is issued, a COUNT register read synchronization is forced.

Note: This command should be used to read the most updated COUNT internal value.