**Welcome to E-XFL.COM**

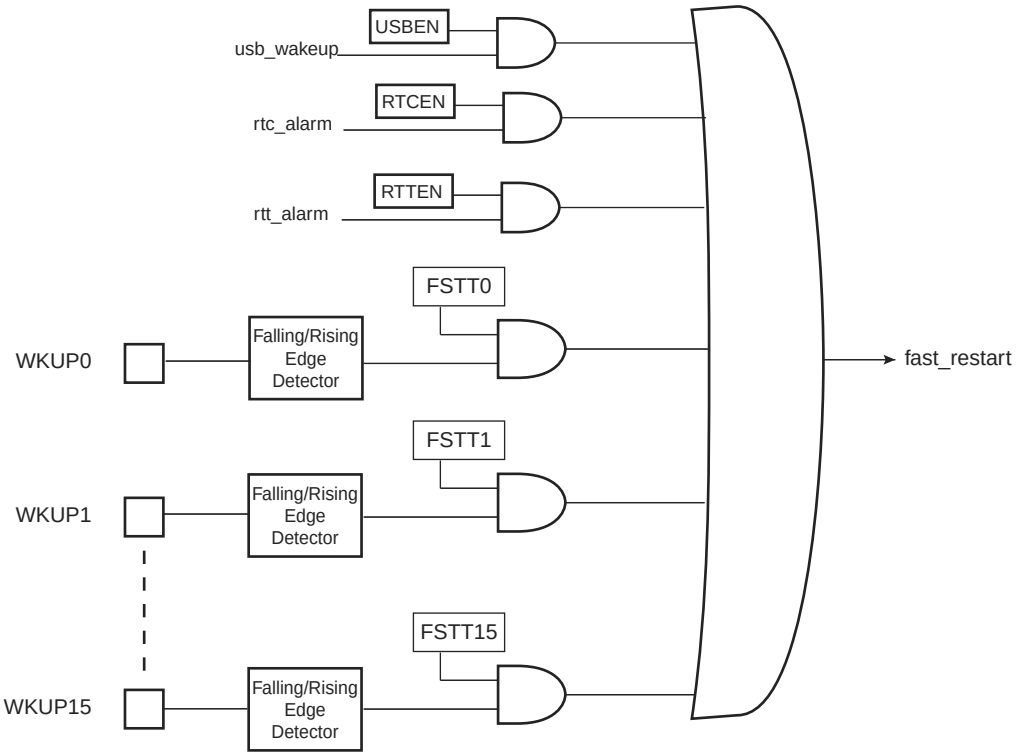
## What is "Embedded - Microcontrollers"?

"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®-M3 |
| Core Size | 32-Bit Single-Core |
| Speed | 64MHz |
| Connectivity | I²C, MMC, SPI, SSC, UART/USART, USB |
| Peripherals | Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT |
| Number of I/O | 47 |
| Program Memory Size | 128KB (128K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 32K x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.62V ~ 3.6V |
| Data Converters | A/D 10x10/12b; D/A 2x12b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 64-LQFP |
| Supplier Device Package | 64-LQFP (10x10) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/atsam3s2ba-aur |

**Figure 5-6. Fast Start-Up Sources**



The processor sets EXC_RETURN bits[31:4] to 0xFFFFFFF. When this value is loaded into the PC it indicates to the processor that the exception is complete, and the processor initiates the exception return sequence.

**Table 11-10. Exception return behavior**

| EXC_RETURN[3:0] | Description |
|---|---|
| bXXX0 | Reserved. |
| b0001 | Return to Handler mode.<br>Exception return gets state from MSP.<br>Execution uses MSP after return. |
| b0011 | Reserved. |
| b01X1 | Reserved. |
| b1001 | Return to Thread mode.<br>Exception return gets state from MSP.<br>Execution uses MSP after return. |
| b1101 | Return to Thread mode.<br>Exception return gets state from PSP.<br>Execution uses PSP after return. |
| b1X11 | Reserved. |

## 11.6 Fault handling

Faults are a subset of the exceptions, see “Exception model” on page 61. The following generate a fault:

- a bus error on:
  - an instruction fetch or vector table load
  - a data access
- an internally-detected error such as an undefined instruction or an attempt to change state with a BX instruction
- attempting to execute an instruction from a memory region marked as *Non-Executable* (XN).
- an MPU fault because of a privilege violation or an attempt to access an unmanaged region.

### 11.6.1 Fault types

Table 11-11 shows the types of fault, the handler used for the fault, the corresponding fault status register, and the register bit that indicates that the fault has occurred. See “Configurable Fault Status Register” on page 177 for more information about the fault status registers.

**Table 11-11. Faults**

| Fault | Handler | Bit name | Fault status register |
|---|---|---|---|
| Bus error on a vector read | Hard fault | VECTTBL | “Hard Fault Status Register” on page 183 |
| Fault escalated to a hard fault | | FORCED | |
| MPU mismatch: | Memory management fault | - | - |
| on instruction access | | IACCVIOL (1) | “Memory Management Fault Address Register” on page 184 |
| on data access | | DACCVIOL | |
| during exception stacking | | MSTKERR | |
| during exception unstacking | | MUNSKERR | |

**Table 11-37. TEX, C, B, and S encoding (Continued)**

| TEX | C | B | S | Memory type | Shareability | Other attributes |
|---|---|---|---|---|---|---|
| b001 | 0 | 0 | 0 | Normal | Not shareable | |
| | | | 1 | | Shareable | |
| | | 1 | x[1] | Reserved encoding | | - |
| | 1 | 0 | x[1] | Implementation defined attributes. | | - |
| | | 1 | 0 | Normal | Not shareable | Outer and inner write-back. Write and read allocate. |
| | | | 1 | | Shareable | |
| b010 | 0 | 0 | x[1] | Device | Not shareable | Nonshared Device. |
| | | 1 | x[1] | Reserved encoding | | - |
| | 1 | x[1] | x[1] | Reserved encoding | | - |
| b1BB | A | A | 0 | Normal | Not shareable | |
| | | | 1 | | Shareable | |

1. The MPU ignores the value of this bit.

Table 11-38 shows the cache policy for memory attribute encodings with a TEX value is in the range 4-7.

**Table 11-38. Cache policy for memory attribute encoding**

| Encoding, AA or BB | Corresponding cache policy |
|---|---|
| 00 | Non-cacheable |
| 01 | Write back, write and read allocate |
| 10 | Write through, no write allocate |
| 11 | Write back, no write allocate |

Table 11-39 shows the AP encodings that define the access permissions for privileged and unprivileged software.

**Table 11-39. AP encoding**

| AP[2:0] | Privileged permissions | Unprivileged permissions | Description |
|---|---|---|---|
| 000 | No access | No access | All accesses generate a permission fault |
| 001 | RW | No access | Access from privileged software only |
| 010 | RW | RO | Writes by unprivileged software generate a permission fault |
| 011 | RW | RW | Full access |
| 100 | Unpredictable | Unpredictable | Reserved |

**Table 20-9. Write Command (Continued)**

| Step | Handshake Sequence | MODE[3:0] | DATA[7:0] |
|---|---|---|---|
| n | Write handshaking | ADDR0 | Memory Address LSB |
| n+1 | Write handshaking | ADDR1 | Memory Address |
| n+2 | Write handshaking | ADDR2 | Memory Address |
| n+3 | Write handshaking | ADDR3 | Memory Address |
| n+4 | Write handshaking | DATA | *Memory Address++ |
| n+5 | Write handshaking | DATA | *Memory Address++ |
| ... | ... | ... | ... |

The Flash command **Write Page and Lock (WPL)** is equivalent to the Flash Write Command. However, the lock bit is automatically set at the end of the Flash write operation. As a lock region is composed of several pages, the programmer writes to the first pages of the lock region using Flash write commands and writes to the last page of the lock region using a Flash write and lock command.

The Flash command **Erase Page and Write (EWP)** is equivalent to the Flash Write Command. However, before programming the load buffer, the page is erased.

The Flash command **Erase Page and Write the Lock (EWPL)** combines EWP and WPL commands.

#### 20.2.5.3 Flash Full Erase Command

This command is used to erase the Flash memory planes.

All lock regions must be unlocked before the Full Erase command by using the CLB command. Otherwise, the erase command is aborted and no page is erased.

**Table 20-10. Full Erase Command**

| Step | Handshake Sequence | MODE[3:0] | DATA[15:0] or DATA[7:0] |
|---|---|---|---|
| 1 | Write handshaking | CMDE | EA |
| 2 | Write handshaking | DATA | 0 |

#### 20.2.5.4 Flash Lock Commands

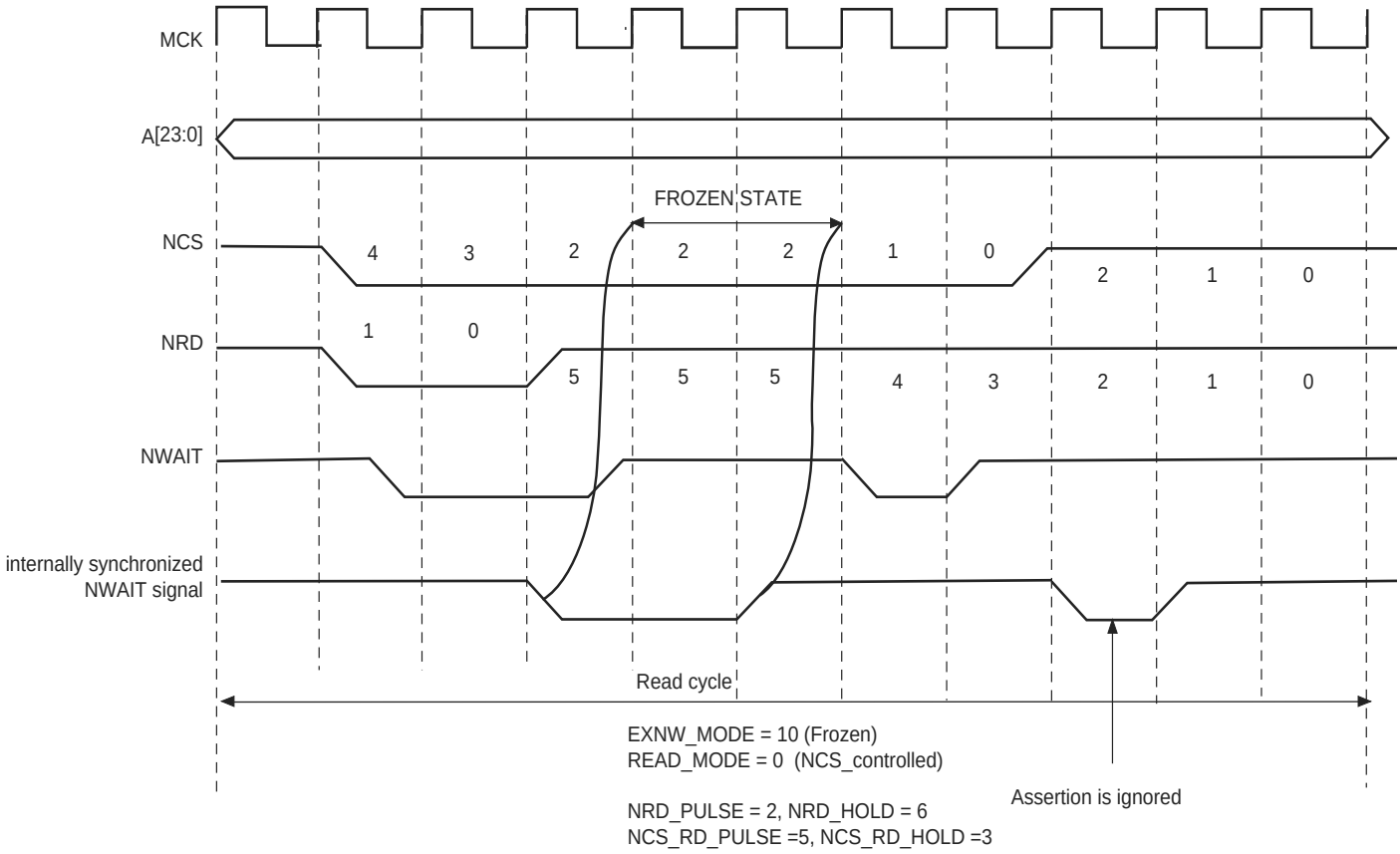
Lock bits can be set using WPL or EWPL commands. They can also be set by using the **Set Lock** command **(SLB)**. With this command, several lock bits can be activated. A Bit Mask is provided as argument to the command. When bit 0 of the bit mask is set, then the first lock bit is activated.

In the same way, the **Clear Lock** command **(CLB)** is used to clear lock bits.

**Table 20-11. Set and Clear Lock Bit Command**

| Step | Handshake Sequence | MODE[3:0] | DATA[15:0] or DATA[7:0] |
|---|---|---|---|
| 1 | Write handshaking | CMDE | SLB or CLB |
| 2 | Write handshaking | DATA | Bit Mask |

**Figure 24-24. Read Access with NWAIT Assertion in Frozen Mode (EXNW_MODE = 10)**



### 25.5.6 Receive Next Counter Register

**Name:** PERIPH_RNCR

**Access:** Read-write

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | – | – |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| – | – | – | – | – | – | – | – |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RXNCTR | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RXNCTR | | | | | | | |

- **RXNCTR: Receive Next Counter**

RXNCTR contains next receive buffer size.

When a half duplex peripheral is connected to the PDC, RXNCTR = TXNCTR.

### 25.5.10 Transfer Status Register

**Name:** PERIPH_PTSR

**Access:** Read-only

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | – | – |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| – | – | – | – | – | – | – | – |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| – | – | – | – | – | – | – | TXTEN |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| – | – | – | – | – | – | – | RXTEN |

- **RXTEN: Receiver Transfer Enable**

0 = PDC Receiver channel requests are disabled.

1 = PDC Receiver channel requests are enabled.

- **TXTEN: Transmitter Transfer Enable**

0 = PDC Transmitter channel requests are disabled.

1 = PDC Transmitter channel requests are enabled.

### 31.8.5 SPI Status Register

**Name:** SPI_SR

**Access:** Read-only

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | – | – |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| – | – | – | – | – | – | – | SPIENS |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| – | – | – | – | – | UNDES | TXEMPTY | NSSR |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TXBUFE | RXBUFF | ENDTX | ENDRX | OVRES | MODF | TDRE | RDRF |

- **RDRF: Receive Data Register Full**

0 = No data has been received since the last read of SPI_RDR

1 = Data has been received and the received data has been transferred from the serializer to SPI_RDR since the last read of SPI_RDR.

- **TDRE: Transmit Data Register Empty**

0 = Data has been written to SPI_TDR and not yet transferred to the serializer.

1 = The last data written in the Transmit Data Register has been transferred to the serializer.

TDRE equals zero when the SPI is disabled or at reset. The SPI enable command sets this bit to one.

- **MODF: Mode Fault Error**

0 = No Mode Fault has been detected since the last read of SPI_SR.

1 = A Mode Fault occurred since the last read of the SPI_SR.

- **OVRES: Overrun Error Status**

0 = No overrun has been detected since the last read of SPI_SR.

1 = An overrun has occurred since the last read of SPI_SR.

An overrun occurs when SPI_RDR is loaded at least twice from the serializer since the last read of the SPI_RDR.

- **ENDRX: End of RX buffer**

0 = The Receive Counter Register has not reached 0 since the last write in SPI_RCR(1) or SPI_RNCR(1).

1 = The Receive Counter Register has reached 0 since the last write in SPI_RCR(1) or SPI_RNCR(1).

- **ENDTX: End of TX buffer**

0 = The Transmit Counter Register has not reached 0 since the last write in SPI_TCR(1) or SPI_TNCR(1).

1 = The Transmit Counter Register has reached 0 since the last write in SPI_TCR(1) or SPI_TNCR(1).

- **RXBUFF: RX Buffer Full**

0 = SPI_RCR(1) or SPI_RNCR(1) has a value other than 0.

1 = Both SPI_RCR(1) and SPI_RNCR(1) have a value of 0.

- **TXBUFE: TX Buffer Empty**

0 = SPI_TCR(1) or SPI_TNCR(1) has a value other than 0.

### 31.8.6 SPI Interrupt Enable Register

**Name:** SPI_IER

**Access:** Write-only

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | – | – |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| – | – | – | – | – | – | – | – |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| – | – | – | – | – | UNDES | TXEMPTY | NSSR |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TXBUFE | RXBUFF | ENDTX | ENDRX | OVRES | MODF | TDRE | RDRF |

0 = No effect.

1 = Enables the corresponding interrupt.

- **RDRF: Receive Data Register Full Interrupt Enable**
- **TDRE: SPI Transmit Data Register Empty Interrupt Enable**
- **MODF: Mode Fault Error Interrupt Enable**
- **OVRES: Overrun Error Interrupt Enable**
- **ENDRX: End of Receive Buffer Interrupt Enable**
- **ENDTX: End of Transmit Buffer Interrupt Enable**
- **RXBUFF: Receive Buffer Full Interrupt Enable**
- **TXBUFE: Transmit Buffer Empty Interrupt Enable**
- **NSSR: NSS Rising Interrupt Enable**
- **TXEMPTY: Transmission Registers Empty Enable**
- **UNDES: Underrun Error Interrupt Enable**

### 32.11.3 TWI Slave Mode Register

Name: TWI_SMR

Access: Read-write

**Reset:** 0x00000000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | – | – |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| – | SADR | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | – | – |

- **SADR: Slave Address**

The slave device address is used in Slave mode in order to be accessed by master devices in read or write mode.

SADR must be programmed before enabling the Slave mode or after a general call. Writes at other times have no effect.

- **CTSIC: Clear to Send Input Change Flag**

0: No input change has been detected on the CTS pin since the last read of US_CSR.

1: At least one input change has been detected on the CTS pin since the last read of US_CSR.

- **RI: Image of RI Input**

0: RI is set to 0.

1: RI is set to 1.

- **DSR: Image of DSR Input**

0: DSR is set to 0

1: DSR is set to 1.

- **DCD: Image of DCD Input**

0: DCD is set to 0.

1: DCD is set to 1.

- **CTS: Image of CTS Input**

0: CTS is set to 0.

1: CTS is set to 1.

- **MANERR: Manchester Error**

0: No Manchester error has been detected since the last RSTSTA.

1: At least one Manchester error has been detected since the last RSTSTA.

### 34.8.16 USART Write Protect Mode Register

**Name:** US_WPMR

**Access:** Read-write

**Reset:** See Table 34-16

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| WPKEY | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| WPKEY | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| WPKEY | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| — | — | — | — | — | — | — | WPEN |

- **WPEN: Write Protect Enable**

0 = Disables the Write Protect if WPKEY corresponds to 0x555341 (“USA” in ASCII).

1 = Enables the Write Protect if WPKEY corresponds to 0x555341 (“USA” in ASCII).

Protects the registers:

- “USART Mode Register” on page 719
- “USART Baud Rate Generator Register” on page 734
- “USART Receiver Time-out Register” on page 735
- “USART Transmitter Timeguard Register” on page 736
- “USART FI DI RATIO Register” on page 737
- “USART IrDA FILTER Register” on page 739
- “USART Manchester Configuration Register” on page 740

- **WPKEY: Write Protect KEY**

Should be written at value 0x555341 (“USA” in ASCII). Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Each channel can independently select an internal or external clock source for its counter:

- Internal clock signals: TIMER_CLOCK1, TIMER_CLOCK2, TIMER_CLOCK3, TIMER_CLOCK4, TIMER_CLOCK5
- External clock signals: XC0, XC1 or XC2

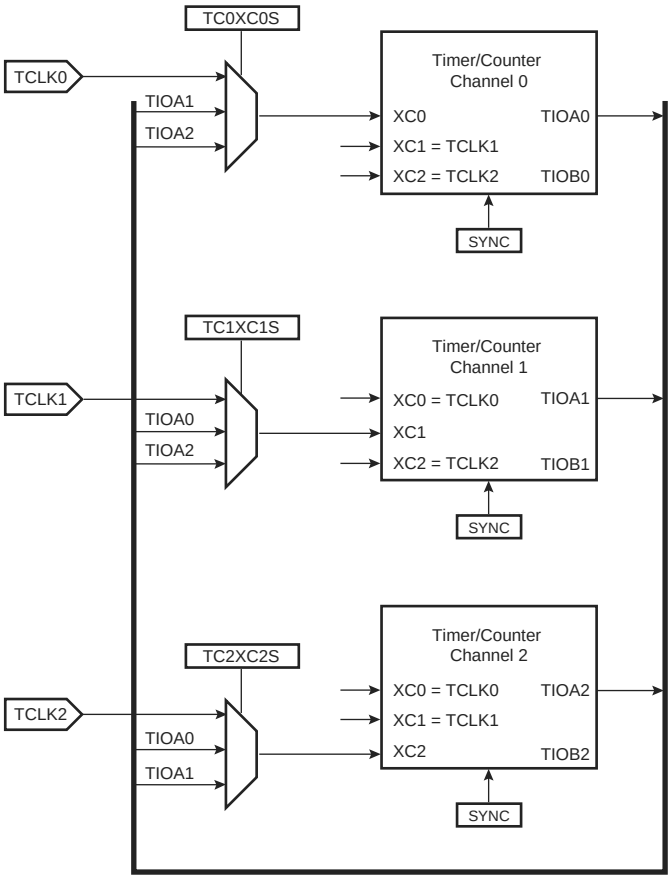
This selection is made by the TCCLKS bits in the TC Channel Mode Register.

The selected clock can be inverted with the CLKI bit in TC_CMR. This allows counting on the opposite edges of the clock.

The burst function allows the clock to be validated when an external signal is high. The BURST parameter in the Mode Register defines this signal (none, XC0, XC1, XC2). See Figure 35-3 ”Clock Selection”

Note: In all cases, if an external clock is used, the duration of each of its levels must be longer than the master clock period. The external clock frequency must be at least 2.5 times lower than the master clock

**Figure 35-2. Clock Chaining Selection**



### 37.7.10 PWM Sync Channels Update Control Register

**Name:** PWM_SCUC

**Access:** Read-write

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | – | – |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| – | – | – | – | – | – | – | – |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| – | – | – | – | – | – | – | – |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| – | – | – | – | – | – | – | UPDULOCK |

- **UPDULOCK: Synchronous Channels Update Unlock**

0 = No effect

1 = If the UPDM field is set to “0” in “PWM Sync Channels Mode Register” on page 882, writing the UPDULOCK bit to “1” triggers the update of the period value, the duty-cycle and the dead-time values of synchronous channels at the beginning of the next PWM period. If the field UPDM is set to “1” or “2”, writing the UPDULOCK bit to “1” triggers only the update of the period value and of the dead-time values of synchronous channels.

This bit is automatically reset when the update is done.

### 37.7.17 PWM Output Override Value Register

**Name:** PWM_OOV

**Access:** Read-write

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | – | – |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| – | – | – | – | OOVL3 | OOVL2 | OOVL1 | OOVL0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| – | – | – | – | – | – | – | – |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| – | – | – | – | OOVH3 | OOVH2 | OOVH1 | OOVH0 |

- **OOVHx: Output Override Value for PWMH output of the channel x**

0 = Override value is 0 for PWMH output of channel x.

1 = Override value is 1 for PWMH output of channel x.

- **OOVLx: Output Override Value for PWML output of the channel x**

0 = Override value is 0 for PWML output of channel x.

1 = Override value is 1 for PWML output of channel x.

### 37.7.34 PWM Comparison x Mode Register

**Name:** PWM_CMPxM
**Access:** Read-write

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | – | – |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| CUPRCNT | | | | CUPR | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| CPRCNT | | | | CPR | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CTR | | | | – | – | – | CEN |

- **CEN: Comparison x Enable**

0 = The comparison x is disabled and can not match.

1 = The comparison x is enabled and can match.

- **CTR: Comparison x Trigger**

The comparison x is performed when the value of the comparison x period counter (CPRCNT) reaches the value defined by CTR.

- **CPR: Comparison x Period**

CPR defines the maximum value of the comparison x period counter (CPRCNT). The comparison x value is performed periodically once every CPR+1 periods of the channel 0 counter.

- **CPRCNT: Comparison x Period Counter**

Reports the value of the comparison x period counter.

Note: The field CPRCNT is read-only

- **CUPR: Comparison x Update Period**

Defines the time between each update of the comparison x mode and the comparison x value. This time is equal to CUPR+1 periods of the channel 0 counter.

- **CUPRCNT: Comparison x Update Period Counter**

Reports the value of the comparison x update period counter.

Note: The field CUPRCNT is read-only

### 39.7.7 ACC Analog Control Register

**Name:** ACC_ACR

**Access:** Read-write

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | – | – |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| – | – | – | – | – | – | – | – |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| – | – | – | – | – | – | – | – |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| – | – | – | – | – | HYST | | ISEL |

This register can only be written if the WPEN bit is cleared in ACC Write Protect Mode Register.

- **ISEL: Current SELection**

Refer to the product Electrical Characteristics.

0 (LOPW) = low power option.

1 (HISP) = high speed option.

- **HYST: HYSTeresis selection**

0 to 3: Refer to the product Electrical Characteristics.

## 41.7 Digital-to-Analog Converter (DACC) User Interface

**Table 41-3. Register Mapping**

| Offset | Register | Name | Access | Reset |
|---|---|---|---|---|
| 0x00 | Control Register | DACC_CR | Write-only | – |
| 0x04 | Mode Register | DACC_MR | Read-write | 0x00000000 |
| 0x08 | Reserved | – | – | – |
| 0x0C | Reserved | – | – | – |
| 0x10 | Channel Enable Register | DACC_CHER | Write-only | – |
| 0x14 | Channel Disable Register | DACC_CHDR | Write-only | – |
| 0x18 | Channel Status Register | DACC_CHSR | Read-only | 0x00000000 |
| 0x1C | Reserved | – | – | – |
| 0x20 | Conversion Data Register | DACC_CDR | Write-only | 0x00000000 |
| 0x24 | Interrupt Enable Register | DACC_IER | Write-only | – |
| 0x28 | Interrupt Disable Register | DACC_IDR | Write-only | – |
| 0x2C | Interrupt Mask Register | DACC_IMR | Read-only | 0x00000000 |
| 0x30 | Interrupt Status Register | DACC_ISR | Read-only | 0x00000000 |
| 0x94 | Analog Current Register | DACC_ACR | Read-write | 0x00000000 |
| 0xE4 | Write Protect Mode register | DACC_WPMR | Read-write | 0x00000000 |
| 0xE8 | Write Protect Status register | DACC_WPSR | Read-only | 0x00000000 |
| ... | ... | ... | ... | ... |
| 0xEC - 0xFC | Reserved | – | – | – |

**Table 42-2. DC Characteristics (Continued)**

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|---|
| $R_{PULLUP}$ | Pull-up Resistor | PA0-PA31, PB0-PB14, PC0-PC31 | 50 | 100 | 175 | kΩ |
| | | NRST | 50 | 100 | 175 | kΩ |
| $R_{PULLDOWN}$ | Pull-down Resistor | PA0-PA13, PA15-PA28, PA30-PA31, PB0-PB9, PB10-PB11, PB12-PB14, PC0-PC31 | 50 | 100 | 175 | kΩ |
| | | | 14.25 | 20 | 24.8 | |
| | | TST, JTAGSEL | 10 | | 20 | |
| $R_{ODT}$ | On-die Series Termination Resistor | PA4-PA31, PB0-PB9, PB12-PB14,PC0-PC31 | | 36 | | Ω |
| | | PA0-PA3 | | 18 | | |

Notes: 1. PA[4-13], PA[15-28], PB[0-14], PC[0-31]
2. Refer to Section 5.2.2 “VDDIO Versus VDDIN”

**Table 42-3. 1.8V Voltage Regulator Characteristics**

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|---|
| $V_{DDIN}$ | DC Input Voltage Range | (4) (5) | 1.8 | 3.3 | 3.6 | V |
| $V_{DDOUT}$ | DC Output Voltage | Normal Mode | | 1.8 | | V |
| | | Standby Mode | | 0 | | |
| | Output Voltage Accuracy | $I_{Load}$ = 0.5mA to 150 mA | -3 | | 3 | % |
| $I_{LOAD}$ | Maximum DC Output Current | $V_{DDIN}$ > 2V | | | 80 | mA |
| | | $V_{DDIN} \leq$ 2V | | | 40 | |
| $I_{LOAD-START}$ | Maximum Peak Current during startup | See Note(3). | | | 400 | mA |
| $V_{DROPOUT}$ | Dropout Voltage | $V_{DDIN}$ = 1.8V, $I_{Load}$ = 60 mA | | | 150 | mV |
| $V_{LINE}$ | Line Regulation | $V_{DDIN}$ from 2.7V to 3.6V; $I_{Load}$ MAX | | 20 | 50 | mV |
| $V_{LINE-TR}$ | Transient Line regulation | $V_{DDIN}$ from 2.7V to 3.6V; $t_r = t_f$ = 5µs; $I_{Load}$ Max $CD_{OUT}$ = 4.7µF | | 50 | 100 | |
| $V_{LOAD}$ | Load Regulation | $V_{DDIN} \geq$ 2.2V; $I_{Load}$ = 10% to 90% MAX | | 20 | 50 | mV |
| $V_{LOAD-TR}$ | Transient Load Regulation | $V_{DDIN} \geq$ 2.2V; $I_{Load}$ = 10% to 90% MAX $t_r = t_f$ = 5 µs $CD_{OUT}$ = 4.7 µF | | 50 | 100 | |
| $I_Q$ | Quiescent Current | Normal Mode; | | | | µA |
| | | @ $I_{Load}$ = 0 mA | | 7 | 10 | |
| | | @ $I_{Load}$ = 80 mA | | 700 | 1200 | |
| | | Standby Mode; | | | 1 | |
| $CD_{IN}$ | Input Decoupling Capacitor | See Note (1) | | 10 | | µF |

### 42.4.4 32.768 kHz Crystal Characteristics

**Table 42-25. Crystal Characteristics**

| Symbol | Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| ESR | Equivalent Series Resistor ($R_S$) | Crystal @ 32.768 kHz | | 50 | 100 | kΩ |
| $C_m$ | Motional capacitance | Crystal @ 32.768 kHz | 0.6 | | 3 | fF |
| $C_{SHUNT}$ | Shunt capacitance | Crystal @ 32.768 kHz | 0.6 | | 2 | pF |

### 42.4.5 32.768 kHz XIN32 Clock Input Characteristics in Bypass Mode

**Table 42-26. XIN32 Clock Electrical Characteristics (In Bypass Mode)**

| Symbol | Parameter | Conditions | Min | Max | Units |
|---|---|---|---|---|---|
| $1/(t_{CPXIN32})$ | XIN32 Clock Frequency | (1) | | 44 | kHz |
| $t_{CPXIN32}$ | XIN32 Clock Period | (1) | 22 | | µs |
| $t_{CHXIN32}$ | XIN32 Clock High Half-period | (1) | 11 | | µs |
| $t_{CLXIN32}$ | XIN32 Clock Low Half-period | (1) | 11 | | µs |
| $t_{CLCH}$ | Rise Time | | 400 | | ns |
| $t_{CHCL}$ | Fall Time | | 400 | | ns |
| $C_i$ | XIN32 Input Capacitance | | | 6 | pF |
| $R_{IN}$ | XIN32 Pull-down Resistor | | 3 | 5 | MΩ |
| $V_{XIN32\_IL}$ | $V_{XIN32}$ Input Low-level Voltage | | -0.3 | 0.3 x $V_{DDIO}$ | V |
| $V_{XIN32\_IH}$ | $V_{XIN32}$ Input High-level Voltage | | 0.7 x $V_{DDIO}$ | $V_{DDIO}$+0.3 | V |

Note: 1. These characteristics apply only when the 32.768 kHz crystal Oscillator is in bypass mode (i.e., when OSCBYPASS: = 1 in SUPC_MR and XTALSEL = 1 in the SUPC_CR registers.

**Figure 42-12. XIN32 Clock Timing**