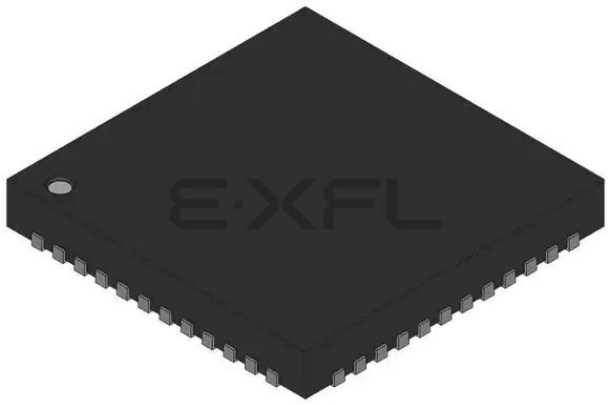


**Welcome to E-XFL.COM**

## 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 | AVR |
| Core Size | 32-Bit Single-Core |
| Speed | 48MHz |
| Connectivity | I²C, SPI, UART/USART, USB |
| Peripherals | Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT |
| Number of I/O | 35 |
| Program Memory Size | 64KB (64K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 16K x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.65V ~ 3.6V |
| Data Converters | A/D 6x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 48-VFQFN Exposed Pad |
| Supplier Device Package | 48-QFN (7x7) |
| Purchase URL | https://www.e-xfl.com/product-detail/atmel/atuc64d4-z1ut |

buffer value. The only way to change a bit from zero to one is to erase the entire page buffer with the Clear Page Buffer command.

The page buffer is not automatically reset after a page write. The programmer should do this manually by issuing the Clear Page Buffer flash command. This can be done after a page write, or before the page buffer is loaded with data to be stored to the flash page.

## 8.5 Flash Commands

The FLASHCDW offers a command set to manage programming of the flash memory, locking and unlocking of regions, and full flash erasing. See Section 8.8.2 for a complete list of commands.

To run a command, the CMD field in the Flash Command Register (FCMD) has to be written with the command number. As soon as the FCMD register is written, the FRDY bit in the Flash Status Register (FSR) is automatically cleared. Once the current command is complete, the FSR.FRDY bit is automatically set. If an interrupt has been enabled by writing a one to FCR.FRDY, the interrupt request line of the Flash Controller is activated. All flash commands except for Quick Page Read (QPR) and Quick User Page Read (QPRUP) will generate an interrupt request upon completion if FCR.FRDY is one.

Any HSB bus transfers attempting to read flash memory when the FLASHCDW is busy executing a flash command will be stalled, and allowed to continue when the flash command is complete.

After a command has been written to FCMD, the programming algorithm should wait until the command has been executed before attempting to read instructions or data from the flash or writing to the page buffer, as the flash will be busy. The waiting can be performed either by polling the Flash Status Register (FSR) or by waiting for the flash ready interrupt. The command written to FCMD is initiated on the first clock cycle where the HSB bus interface in FLASHCDW is IDLE. The user must make sure that the access pattern to the FLASHCDW HSB interface contains an IDLE cycle so that the command is allowed to start. Make sure that no bus masters such as DMA controllers are performing endless burst transfers from the flash. Also, make sure that the CPU does not perform endless burst transfers from flash. This is done by letting the CPU enter sleep mode after writing to FCMD, or by polling FSR for command completion. This polling will result in an access pattern with IDLE HSB cycles.

All the commands are protected by the same keyword, which has to be written in the eight highest bits of the FCMD register. Writing FCMD with data that does not contain the correct key and/or with an invalid command has no effect on the flash memory; however, the PROGE bit is set in the Flash Status Register (FSR). This bit is automatically cleared by a read access to the FSR register.

Writing a command to FCMD while another command is being executed has no effect on the flash memory; however, the PROGE bit is set in the Flash Status Register (FSR). This bit is automatically cleared by a read access to the FSR register.

If the current command writes or erases a page in a locked region, or a page protected by the BOOTPROT fuses, the command has no effect on the flash memory; however, the LOCKE bit is set in the FSR register. This bit is automatically cleared by a read access to the FSR register.

### 8.8.4 Flash Parameter Register

**Name:** FPR

**Access Type:** Read-only

**Offset**: 0x0C

**Reset Value:** -

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

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

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

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

- **PSZ: Page Size**
  The size of each flash page.

**Table 8-8.** Flash Page Size

| PSZ | Page Size |
|---|---|
| 0 | 32 Byte |
| 1 | 64 Byte |
| 2 | 128 Byte |
| 3 | 256 Byte |
| 4 | 512 Byte |
| 5 | 1024 Byte |
| 6 | 2048 Byte |
| 7 | 4096 Byte |

### 12.7.13 Interrupt Clear Register

**Name:** ICR

**Access Type:** Write-only

**Offset:** 0x0D0

**Reset Value:** -

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

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

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

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| - | - | CKRDY | - | - | - | - | CFD |

Writing a zero to a bit in this register has no effect.
Writing a one to a bit in this register will clear the corresponding bit in ISR.

**Table 13-4.** PLLOPT Fields Description

| | **Description** |
|---|---|
| PLLOPT[0]: VCO frequency | |
| 0 | 80MHz<$f_{vco}$<180MHz |
| 1 | 160MHz<$f_{vco}$<240MHz |
| PLLOPT[1]: Output divider | |
| 0 | $f_{PLL} = f_{vco}$ |
| 1 | $f_{PLL} = f_{vco}/2$ |
| PLLOPT[2] | |
| 0 | Wide Bandwidth Mode enabled |
| 1 | Wide Bandwidth Mode disabled |

- **PLLOSC: PLL Oscillator Select**
  0: Oscillator 0 is the source for the PLL.
  others: Reserved.
- **PLLEN: PLL Enable**
  0: PLL is disabled.
  1: PLL is enabled.

Note that this register is protected by a lock. To write to this register the UNLOCK register has to be written first. Please refer to the UNLOCK register description for details.

### 16.7.7 Edge Register

**Name:** EDGE

**Access Type:** Read/Write

**Offset:** 0x018

**Reset Value:** 0x00000000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| - | INT30 | INT29 | INT28 | INT27 | INT26 | INT25 | INT24 |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| INT23 | INT22 | INT21 | INT20 | INT19 | INT18 | INT17 | INT16 |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| INT15 | INT14 | INT13 | INT12 | INT11 | INT10 | INT9 | INT8 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| INT7 | INT6 | INT5 | INT4 | INT3 | INT2 | INT1 | NMI |

- **INTn: External Interrupt n**
  0: The external interrupt triggers on falling edge.
  1: The external interrupt triggers on rising edge.
  Please refer to the Module Configuration section for the number of external interrupts.
- **NMI: Non-Maskable Interrupt**
  0: The Non-Maskable Interrupt triggers on falling edge.
  1: The Non-Maskable Interrupt triggers on rising edge.

### 18.7.21 Version Register

**Name:** VERSION

**Access Type:** Read-only

**Offset**: 0x1FC

**Reset Value:** -

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

- **VARIANT: Variant Number**
  Reserved. No functionality associated.
- **VERSION: Version Number**
  Version number of the module. No functionality associated.

- The Start of Frame (SOF) interrupt with a frame number CRC error (FNCERR is one)

- *Endpoint interrupts*

  The processing device endpoint interrupts are:

  - The Transmitted IN Data Interrupt (TXINI)
  - The Received OUT Data Interrupt (RXOUTI)
  - The Received SETUP Interrupt (RXSTPI)
  - The Number of Busy Banks (NBUSYBK) interrupt

  The exception device endpoint interrupts are:

  - The Errorflow Interrupt (ERRORFI)
  - The NAKed OUT Interrupt (NAKOUTI)
  - The NAKed IN Interrupt (NAKINI)
  - The STALLed Interrupt (STALLEDI)
  - The CRC Error Interrupt (CRCERRI)

*19.7.2.15 Endpoint n Control Clear Register*

**Register Name:** UECONnCLR, n in [0..6]

**Access Type:** Write-Only

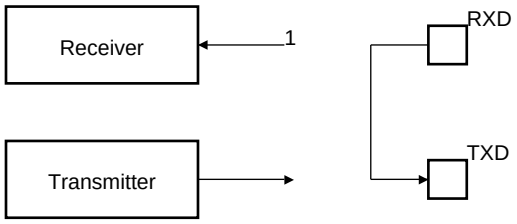
**Offset:** 0x0220 + (n * 0x04)

**Reset Value:** 0x00000000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| - | - | - | - | - | - | BUSY1EC | BUSY0EC |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| - | - | - | - | STALLRQC | - | - | - |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| - | FIFOCONC | - | NBUSYBKEC | RAMACEREC | - | - | - |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | STALLEDEC /CRCERREC | - | NAKINEC | NAKOUTEC | RXSTPEC/ ERRORFEC | RXOUTEC | TXINEC |

Writing a zero to a bit in this register has no effect.
Writing a one to a bit in this register will clear the corresponding bit in UECONn.
These bits always read as zero.

**Figure 20-23.** Remote Loopback Mode Configuration




### 20.6.7 Write Protection Registers

To prevent single software errors from corrupting USART behavior, certain address spaces can be write-protected by writing the correct Write Protect KEY and a one to the Write Protect Enable bit in the Write Protect Mode Register (WPMR.WPKEY, and WPMR.WPEN). Disabling the write protection is done by writing the correct key, and a zero to WPEN.

Write attempts to a write protected register are detected and the Write Protect Violation Status bit in the Write Protect Status Register (WPSR.WPVS) is set, while the Write Protect Violation Source field (WPSR.WPVSRC) indicates the targeted register. Writing the correct key to the Write Protect KEY bit (WPMR.WPKEY) clears WPVSRC and WPVS.

The protected registers are:

- ”Mode Register” on page 358
- ”Baud Rate Generator Register” on page 368
- ”Receiver Time-out Register” on page 369
- ”Transmitter Timeguard Register” on page 370

# 23. Two-wire Master Interface (TWIM)

Rev.: 1.1.0.1

## 23.1 Features

- **Compatible with I²C standard**
  - **Multi-master support**
  - **Transfer speeds of 100 and 400 kbit/s**
  - **7- and 10-bit and General Call addressing**
- **Compatible with SMBus standard**
  - **Hardware Packet Error Checking (CRC) generation and verification with ACK control**
  - **25 ms clock low timeout delay**
  - **10 ms master cumulative clock low extend time**
  - **25 ms slave cumulative clock low extend time**
- **Compatible with PMBus**
- **Compatible with Atmel Two-wire Interface Serial Memories**
- **DMA interface for reducing CPU load**
- **Arbitrary transfer lengths, including 0 data bytes**
- **Optional clock stretching if transmit or receive buffers not ready for data transfer**

## 23.2 Overview

The Atmel Two-wire Master Interface (TWIM) interconnects components on a unique two-wire bus, made up of one clock line and one data line with speeds of up to 400 kbit/s, based on a byte-oriented transfer format. It can be used with any Atmel Two-wire Interface bus serial EEPROM and I²C compatible device such as a real time clock (RTC), dot matrix/graphic LCD controller, and temperature sensor, to name a few. The TWIM is always a bus master and can transfer sequential or single bytes. Multiple master capability is supported. Arbitration of the bus is performed internally and relinquishes the bus automatically if the bus arbitration is lost.

A configurable baud rate generator permits the output data rate to be adapted to a wide range of core clock frequencies.Table 23-1 lists the compatibility level of the Atmel Two-wire Interface in Master Mode and a full I²C compatible device.

**Table 23-1.** Atmel TWIM Compatibility with I²C Standard

| I²C Standard | Atmel TWIM |
|---|---|
| Standard-mode (100 kbit/s) | Supported |
| Fast-mode (400 kbit/s) | Supported |
| Fast-mode Plus (1 Mbit/s) | Supported |
| 7- or 10-bits Slave Addressing | Supported |
| START BYTE(1) | Not Supported |
| Repeated Start (Sr) Condition | Supported |
| ACK and NACK Management | Supported |
| Slope Control and Input Filtering (Fast mode) | Supported |
| Clock Stretching | Supported |

Note: 1. START + b000000001 + Ack + Sr

### 23.8.2.1 Clock Generation

The Clock Waveform Generator Register (CWGR) is used to control the waveform of the TWCK clock. CWGR must be written so that the desired TWI bus timings are generated. CWGR describes bus timings as a function of cycles of a prescaled clock. The clock prescaling can be selected through the Clock Prescaler field in CWGR (CWGR.EXP).

$$f_{PRESCALER} = \frac{f_{CLK\_TWIM}}{2^{(EXP+1)}}$$

CWGR has the following fields:

LOW: Prescaled clock cycles in clock low count. Used to time $T_{LOW}$ and $T_{BUF}$.

HIGH: Prescaled clock cycles in clock high count. Used to time $T_{HIGH}$.

STASTO: Prescaled clock cycles in clock high count. Used to time $T_{HD\_STA}$, $T_{SU\_STA}$, $T_{SU\_STO}$.

DATA: Prescaled clock cycles for data setup and hold count. Used to time $T_{HD\_DAT}$, $T_{SU\_DAT}$.

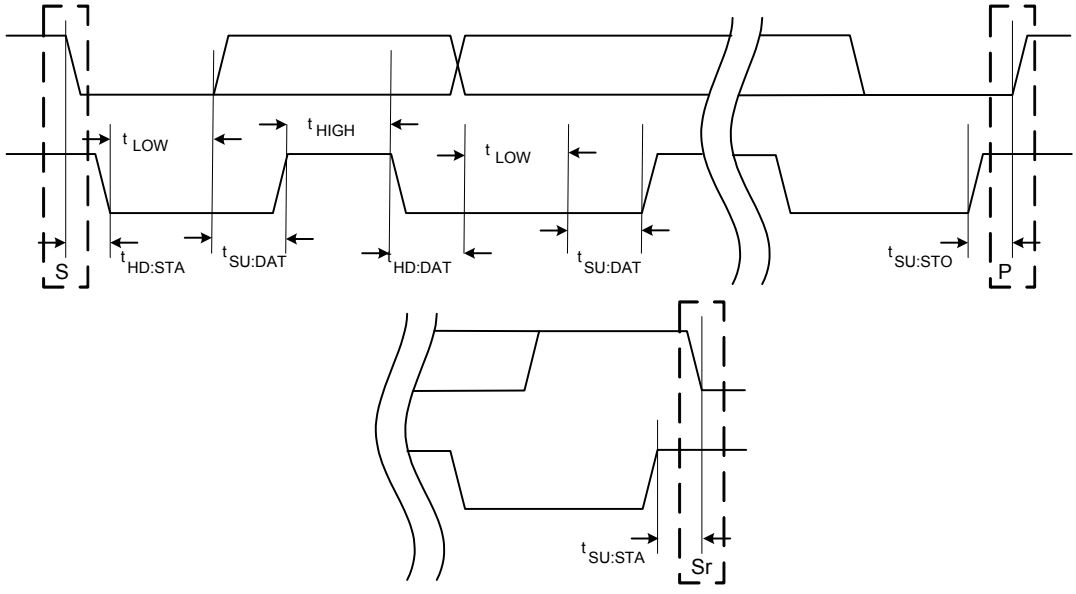
EXP: Specifies the clock prescaler setting.

Note that the total clock low time generated is the sum of $T_{HD\_DAT}$ + $T_{SU\_DAT}$ + $T_{LOW}$.

Any slave or other bus master taking part in the transfer may extend the TWCK low period at any time.

The TWIM hardware monitors the state of the TWCK line as required by the I²C specification. The clock generation counters are started when a high/low level is detected on the TWCK line, not when the TWIM hardware releases/drives the TWCK line. This means that the CWGR settings alone do not determine the TWCK frequency. The CWGR settings determine the clock low time and the clock high time, but the TWCK rise and fall times are determined by the external circuitry (capacitive load, etc.).

**Figure 23-5.** Bus Timing Diagram



## 24.9 User Interface

**Table 24-6.** TWIS Register Memory Map

| Offset | Register | Register Name | Access | Reset |
|---|---|---|---|---|
| 0x00 | Control Register | CR | Read/Write | 0x00000000 |
| 0x04 | NBYTES Register | NBYTES | Read/Write | 0x00000000 |
| 0x08 | Timing Register | TR | Read/Write | 0x00000000 |
| 0x0C | Receive Holding Register | RHR | Read-only | 0x00000000 |
| 0x10 | Transmit Holding Register | THR | Write-only | 0x00000000 |
| 0x14 | Packet Error Check Register | PECR | Read-only | 0x00000000 |
| 0x18 | Status Register | SR | Read-only | 0x00000002 |
| 0x1C | Interrupt Enable Register | IER | Write-only | 0x00000000 |
| 0x20 | Interrupt Disable Register | IDR | Write-only | 0x00000000 |
| 0x24 | Interrupt Mask Register | IMR | Read-only | 0x00000000 |
| 0x28 | Status Clear Register | SCR | Write-only | 0x00000000 |
| 0x2C | Parameter Register | PR | Read-only | -(1) |
| 0x30 | Version Register | VR | Read-only | -(1) |

Note: 1. The reset values for these registers are device specific. Please refer to the Module Configuration section at the end of this chapter.

- **SMBPECERR: SMBus PEC Error**
  This bit is cleared when the corresponding bit in SCR is written to one.
  This bit is set when a SMBus PEC error has occurred.
- **SMBTOUT: SMBus Timeout**
  This bit is cleared when the corresponding bit in SCR is written to one.
  This bit is set when a SMBus timeout has occurred.
- **NAK: NAK Received**
  This bit is cleared when the corresponding bit in SCR is written to one.
  This bit is set when a NAK was received from the master during slave transmitter operation.
- **ORUN: Overrun**
  This bit is cleared when the corresponding bit in SCR is written to one.
  This bit is set when an overrun has occurred in slave receiver mode. Can only occur if CR.STREN is zero.
- **URUN: Underrun**
  This bit is cleared when the corresponding bit in SCR is written to one.
  This bit is set when an underrun has occurred in slave transmitter mode. Can only occur if CR.STREN is zero.
- **TRA: Transmitter Mode**
  0: The slave is in slave receiver mode.
  1: The slave is in slave transmitter mode.
- **TCOMP: Transmission Complete**
  This bit is cleared when the corresponding bit in SCR is written to one.
  This bit is set when transmission is complete. Set after receiving a STOP after being addressed.
- **SEN: Slave Enabled**
  0: The slave interface is disabled.
  1: The slave interface is enabled.
- **TXRDY: TX Buffer Ready**
  0: The TX buffer is full and should not be written to.
  1: The TX buffer is empty, and can accept new data.
- **RXRDY: RX Buffer Ready**
  0: No RX data ready in RHR.
  1: RX data is ready to be read from RHR.

# 26. Timer/Counter (TC)

Rev: 2.2.3.3

## 26.1 Features

- **Three 16-bit Timer Counter channels**
- **A wide range of functions including:**
  - **Frequency measurement**
  - **Event counting**
  - **Interval measurement**
  - **Pulse generation**
  - **Delay timing**
  - **Pulse width modulation**
  - **Up/down capabilities**
- **Each channel is user-configurable and contains:**
  - **Three external clock inputs**
  - **Five internal clock inputs**
  - **Two multi-purpose input/output signals**
- **Internal  interrupt signal**
- **Two global registers that act on all three TC channels**

## 26.2 Overview

The Timer Counter (TC) includes three identical 16-bit Timer Counter channels.

Each channel can be independently programmed to perform a wide range of functions including frequency measurement, event counting, interval measurement, pulse generation, delay timing, and pulse width modulation.

Each channel has three external clock inputs, five internal clock inputs, and two multi-purpose input/output signals which can be configured by the user. Each channel drives an internal interrupt signal which can be programmed to generate processor interrupts.

The TC block has two global registers which act upon all three TC channels.

The Block Control Register (BCR) allows the three channels to be started simultaneously with the same instruction.

The Block Mode Register (BMR) defines the external clock inputs for each channel, allowing them to be chained.

0: TIOB is used as an external trigger.

- **ETRGEDG: External Trigger Edge Selection**

| ETRGEDG | Edge |
|---|---|
| 0 | none |
| 1 | rising edge |
| 2 | falling edge |
| 3 | each edge |

- **LDBDIS: Counter Clock Disable with RB Loading**
  1: Counter clock is disabled when RB loading occurs.
  0: Counter clock is not disabled when RB loading occurs.
- **LDBSTOP: Counter Clock Stopped with RB Loading**
  1: Counter clock is stopped when RB loading occurs.
  0: Counter clock is not stopped when RB loading occurs.
- **BURST: Burst Signal Selection**

| BURST | Burst Signal Selection |
|---|---|
| 0 | The clock is not gated by an external signal |
| 1 | XC0 is ANDed with the selected clock |
| 2 | XC1 is ANDed with the selected clock |
| 3 | XC2 is ANDed with the selected clock |

- **CLKI: Clock Invert**
  1: The counter is incremented on falling edge of the clock.
  0: The counter is incremented on rising edge of the clock.
- **TCCLKS: Clock Selection**

| TCCLKS | Clock Selected |
|---|---|
| 0 | TIMER_CLOCK1 |
| 1 | TIMER_CLOCK2 |
| 2 | TIMER_CLOCK3 |
| 3 | TIMER_CLOCK4 |
| 4 | TIMER_CLOCK5 |
| 5 | XC0 |
| 6 | XC1 |
| 7 | XC2 |

### 26.7.8 Channel Status Register

**Name:** SR

**Access Type:** Read-only

**Offset:** 0x20 + n * 0x40

**Reset Value:** 0x00000000

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

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| - | - | - | - | - | MTIOB | MTIOA | CLKSTA |

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

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| ETRGS | LDRBS | LDRAS | CPCS | CPBS | CPAS | LOVRS | COVFS |

Note: Reading the Status Register will also clear the interrupt bit for the corresponding interrupts.

- **MTIOB: TIOB Mirror**
  1: TIOB is high. If CMRn.WAVE is zero, this means that TIOB pin is high. If CMRn.WAVE is one, this means that TIOB is driven high.
  0: TIOB is low. If CMRn.WAVE is zero, this means that TIOB pin is low. If CMRn.WAVE is one, this means that TIOB is driven low.
- **MTIOA: TIOA Mirror**
  1: TIOA is high. If CMRn.WAVE is zero, this means that TIOA pin is high. If CMRn.WAVE is one, this means that TIOA is driven high.
  0: TIOA is low. If CMRn.WAVE is zero, this means that TIOA pin is low. If CMRn.WAVE is one, this means that TIOA is driven low.
- **CLKSTA: Clock Enabling Status**
  1: This bit is set when the clock is enabled.
  0: This bit is cleared when the clock is disabled.
- **ETRGS: External Trigger Status**
  1: This bit is set when an external trigger has occurred.
  0: This bit is cleared when the SR register is read.
- **LDRBS: RB Loading Status**
  1: This bit is set when an RB Load has occurred and CMRn.WAVE is zero.
  0: This bit is cleared when the SR register is read.
- **LDRAS: RA Loading Status**
  1: This bit is set when an RA Load has occurred and CMRn.WAVE is zero.
  0: This bit is cleared when the SR register is read.
- **CPCS: RC Compare Status**
  1: This bit is set when an RC Compare has occurred.
  0: This bit is cleared when the SR register is read.

### 26.7.11 Channel Interrupt Mask Register

**Name:** IMR

**Access Type:** Read-only

**Offset:** 0x2C + n * 0x40

**Reset Value:** 0x00000000

| 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 |
|---|---|---|---|---|---|---|---|
| ETRGS | LDRBS | LDRAS | CPCS | CPBS | CPAS | LOVRS | COVFS |

0: The corresponding interrupt is disabled.
1: The corresponding interrupt is enabled.
A bit in this register is cleared when the corresponding bit in IDR is written to one.
A bit in this register is set when the corresponding bit in IER is written to one.

### 28.7.9 Timing Configuration Register

**Name :** TIM

**Access Type :** Read/Write

**Offset :** 0x20

**Reset Value :** 0x00000000

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

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| - | - | - | - | SHTIM | | | |

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

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

- **SHTIM: Sample and hold time**
  Sample and hold time in number of GCLK clock cycles : SHTIM+1
- **STARTUP: Startup time**
  Number of GCLK clock cycles to wait for : (STARTUP+1)*8

### 30.7.3 Status Clear Register

**Name:** SCR

**Access Type:** Write-only

**Offset:** 0x08

**Reset Value:** 0x00000000

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

Writing a zero to a bit in this register has no effect.
Writing a one to a bit in this register will clear the corresponding bit in SR and the corresponding interrupt request.

**Table 32-18.** Flash Characteristics

| Symbol | Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $T_{FPP}$ | Page programming time | $f_{CLK\_HSB}$= 48MHz | | 5 | | ms |
| $T_{FPE}$ | Page erase time | | | 5 | | |
| $T_{FFP}$ | Fuse programming time | | | 1 | | |
| $T_{FEA}$ | Full chip erase time (EA) | | | 6 | | |
| $T_{FCE}$ | JTAG chip erase time (CHIP_ERASE) | $f_{CLK\_HSB}$= 115kHz | | 310 | | |

**Table 32-19.** Flash Endurance and Data Retention

| Symbol | Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $N_{FARRAY}$ | Array endurance (write/page) | | 100k | | | cycles |
| $N_{FFUSE}$ | General Purpose fuses endurance (write/bit) | | 10k | | | cycles |
| $t_{RET}$ | Data retention | | 15 | | | years |

## 32.9 Analog Characteristics

### 32.9.1 Voltage Regulator Characteristics

#### *32.9.1.1 Electrical Characteristics*

**Table 32-20.** Electrical Characteristics

| Symbol | Parameter | Condition | Min | Typ | Max | Units |
|---|---|---|---|---|---|---|
| $V_{VDDIN}$ | Input voltage range | | 3 | 3.3 | 3.6 | V |
| $V_{VDDCORE}$ | Output voltage | $V_{VDDIN}$ >= 3V | 1.75 | 1.8 | 1.85 | V |
| | Output voltage accuracy | $I_{OUT}$ = 0.1mA to 100mA, $V_{VDDIN}$>3V | | 2 | | % |
| $I_{OUT}$ | DC output current | $V_{VDDIN}$=3.3V | | | 100 | mA |
| $I_{VREG}$ | Static current of internal regulator | Low power mode | | 10 | | µA |

#### *32.9.1.2 Decoupling Requirements*

**Table 32-21.** Decoupling Requirements

| Symbol | Parameter | Condition | Typ | Techno. | Units |
|---|---|---|---|---|---|
| $C_{IN1}$ | Input regulator capacitor 1 | | 1 | NPO | nF |