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 | 50MHz |
| Connectivity | I²C, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, DMA, POR, PWM, WDT |
| Number of I/O | 36 |
| 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 8x12b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 48-TQFP |
| Supplier Device Package | 48-TQFP (7x7) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/at32uc3l0128-aur |

used in Shutdown mode, PINSEL must be written to one, and XIN32_2 and XOUT32_2 must be used.

### 13.5.3 PLL Operation

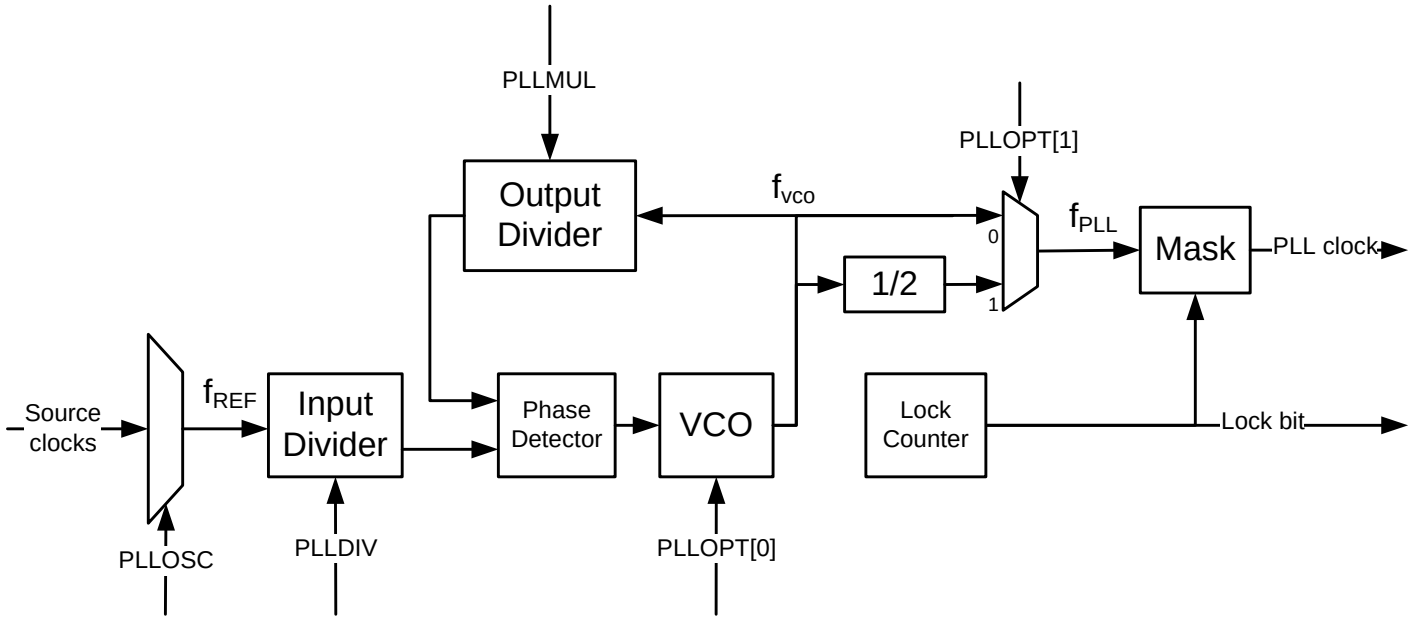
Rev: 1.1.0.0

The device contains one Phase Locked Loop (PLL), which is controlled by the Phase Locked Loop Interface (PLLIF). The PLL is disabled by default, but can be enabled to provide high frequency source clocks for synchronous or generic clocks. The PLL can use different clock sources as reference clock, please refer to the “PLL Clock Sources” table in the SCIF Module Configuration section for details. The PLL output is divided by a multiplication factor, and the PLL compares the phase of the resulting clock to the reference clock. The PLL will adjust its output frequency until the two compared clocks phases are equal, thus locking the output frequency to a multiple of the reference clock frequency.

When the PLL is switched on, or when changing the clock source or multiplication factor for the PLL, the PLL is unlocked and the output frequency is undefined. The PLL clock for the digital logic is automatically masked when the PLL is unlocked, to prevent the connected digital logic from receiving a too high frequency and thus become unstable.

The PLL can be configured by writing the PLL Control Register (PLLn). To prevent unexpected writes due to software bugs, write access to the PLLn register is protected by a locking mechanism, for details please refer to the UNLOCK register description.

**Figure 13-1.** PLL with Control Logic and Filters




#### 13.5.3.1 *Enabling the PLL*

Before the PLL is enabled it must be set up correctly. The PLL Oscillator Select field (PLLOSC) selects a source for the reference clock. The PLL Multiply Factor (PLLMUL) and PLL Division

The user must configure the FP frequency by writing to the FPMUL and FPDIV fields of the FPMUL and FPDIV registers. FPMUL and FPDIV must not be equal to zero and FPDIV must be greater or equal to FPMUL. This results in the output frequency:

$$f_{FPCLK} = f_{SRC} * FPMUL/ (2*FPDIV)$$

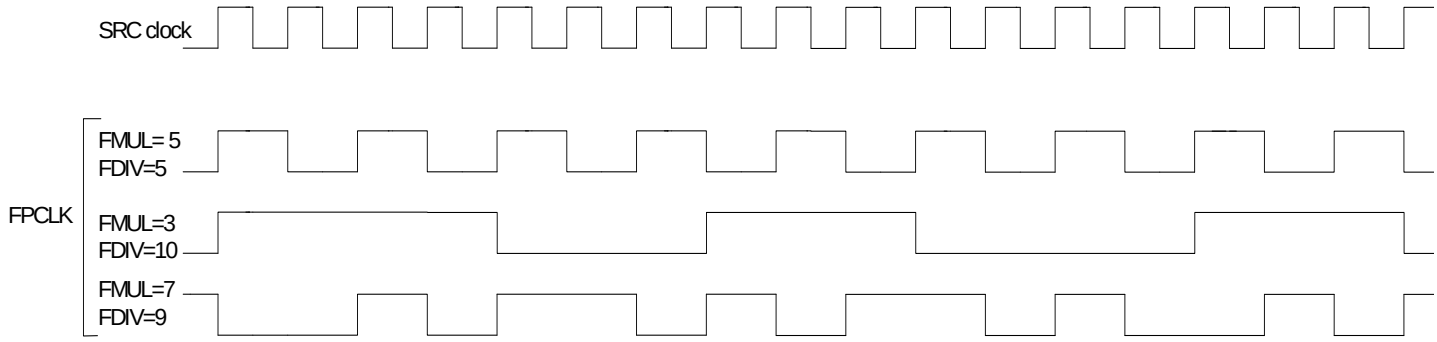
The CKSEL field can not be changed dynamically but the FPMUL and FPDIV fields can be changed on-the-fly.

- Jitter description

As described in Figure 13-9, the CLKFP half period lengths are integer multiples of the source clock period but are not always equals. However the difference between the low level half period length and the high level half period length is at the most one source clock period.

This induces when FPDIV is not an integer multiple of FPMUL a jitter on the FPCLK. The more the FPCLK frequency is low, the more the jitter incidence is reduced.

**Figure 13-9.** Fractional Prescaler Jitter Examples




### 13.5.15 Generic Clocks

Rev: 1.1.0.0

Timers, communication modules, and other modules connected to external circuitry may require specific clock frequencies to operate correctly. The SCIF defines a number of generic clocks that can provide a wide range of accurate clock frequencies.

Each generic clock runs from either clock source listed in the “Generic Clock Sources” table in the SCIF Module Configuration section. The selected source can optionally be divided by any even integer up to 512. Each clock can be independently enabled and disabled, and is also automatically disabled along with peripheral clocks by the Sleep Controller in the Power Manager.

### 13.6.19 Supply Monitor 33 Calibration Register

**Name:** SM33

**Access Type:** Read/Write

**Reset Value:** -

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

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| - | - | - | - | - | ONSM | SFV | FCD |

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

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| FS | - | - | - | CTRL | | | |

- **SAMPFREQ: Sampling Frequency**
  Selects the sampling mode frequency of the 3.3V supply monitor. In sampling mode, the SM33 performs a measurement every $2^{(SAMPFREQ+5)}$ cycles of the internal 32kHz RC oscillator.
- **ONSM: Supply Monitor On Indicator**
  0: The supply monitor is disabled.
  1: The supply monitor is enabled.
  This bit is read-only. Writing to this bit has no effect.
- **SFV: Store Final Value**
  0: The register is read/write
  1: The register is read-only, to protect against further accidental writes.
  This bit is cleared after a reset.
- **FCD: Flash Calibration Done**
  This bit is cleared after a reset.
  This bit is set when CALIB field has been updated after a reset.
- **CALIB: Calibration Value**
  Calibration Value for the SM33.
- **FS: Force Sampling Mode**
  0: Sampling mode is enabled in DeepStop and Static mode only.
  1: Sampling mode is always enabled.
- **CTRL: Supply Monitor Control**

### 13.6.25 PLL Control Register

**Name:** PLLn

**Access Type:** Read/Write

**Reset Value:** 0x00000000

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

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

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

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| - | - | PLLOPT | | | PLLOSC | | PLLEN |

- **PLLCOUNT: PLL Count**
  Specifies the number of RCSYS clock cycles before ISR.PLLLOCKn will be set after PLLn has been written, or after PLLn has been automatically re-enabled after exiting a sleep mode.
- **PLLMUL: PLL Multiply Factor**
- **PLLDIV: PLL Division Factor**
  These fields determine the ratio of the PLL output frequency to the source oscillator frequency:
  $f_{vco}$ = (PLLMUL+1)/PLLDIV • $f_{REF}$ if PLLDIV >0
  $f_{vco}$ = 2•(PLLMUL+1) • $f_{REF}$ if PLLDIV = 0
  Note that the PLLMUL field should always be greater than 1 or the behavior of the PLL will be undefined.
- **PLLOPT: PLL Option**
  **PLLOPT[0]: Selects the VCO frequency range ($f_{vco}$).**
  0: 80MHz<$f_{vco}$<180MHz
  1: 160MHz<$f_{vco}$<240MHz
  **PLLOPT[1]: Divides the output frequency by 2.**
  0: $f_{PLL}$= $f_{vco}$
  1: $f_{PLL}$ = $f_{vco}$/2
  **PLLOPT[2]:Wide-Bandwidth mode.**
  0: Wide Bandwidth Mode enabled
  1: Wide Bandwidth Mode disabled
- **PLLOSC: PLL Oscillator Select**
  Reference clock source select for the reference clock, please refer to the “PLL Clock Sources” table in the SCIF Module Configuration section for details.

### 13.6.34 32KHz Oscillator Version Register

**Name:** OSC32VERSION

**Access Type:** Read-only

**Offset:** 0x03CC

**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.

### 15.6.1 Control Register

**Name:** CTRL

**Access Type:** Read/Write

**Offset:** 0x000

**Reset Value:** 0x00010080

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

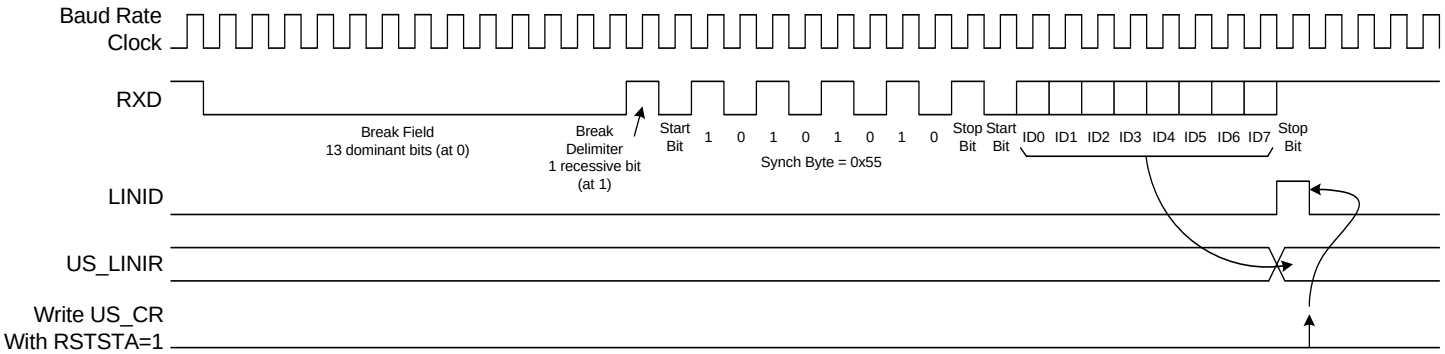
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| - | TBAN | | | | | CSSEL | CEN |

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

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| FCD | - | - | - | SFV | MODE | DAR | EN |

- **KEY**
  This field must be written twice, first with key value 0x55, then 0xAA, for a write operation to be effective. This field always reads as zero.
- **TBAN: Time Ban Prescale Select**
  Counter bit TBAN is used as watchdog “banned” time frame. In this time frame clearing the WDT timer is forbidden, otherwise a watchdog reset is generated and the WDT timer is cleared.
- **CSSEL: Clock Source Select**
  0: Select the system RC oscillator (RCSYS) as clock source.
  1: Select the 32KHz crystal oscillator (OSC32K) as clock source.
- **CEN: Clock Enable**
  0: The WDT clock is disabled.
  1: The WDT clock is enabled.
- **PSEL: Time Out Prescale Select**
  Counter bit PSEL is used as watchdog timeout period.
- **FCD: Flash Calibration Done**
  This bit is set after any reset.
  0: The flash calibration will be redone after a watchdog reset.
  1: The flash calibration will not be redone after a watchdog reset.
- **SFV: WDT Control Register Store Final Value**
  0: WDT Control Register is not locked.
  1: WDT Control Register is locked.
  Once locked, the Control Register can not be re-written, only a reset unlocks the SFV bit.
- **MODE: WDT Mode**
  0: The WDT is in basic mode, only PSEL time is used.
  1: The WDT is in window mode. Total timeout period is now TBAN+PSEL.
  Writing to this bit when the WDT is enabled has no effect.

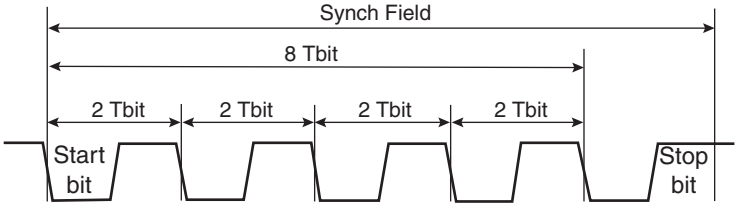
**Figure 19-22.** Header Reception




### *19.6.5.7 Slave Node Synchronization*

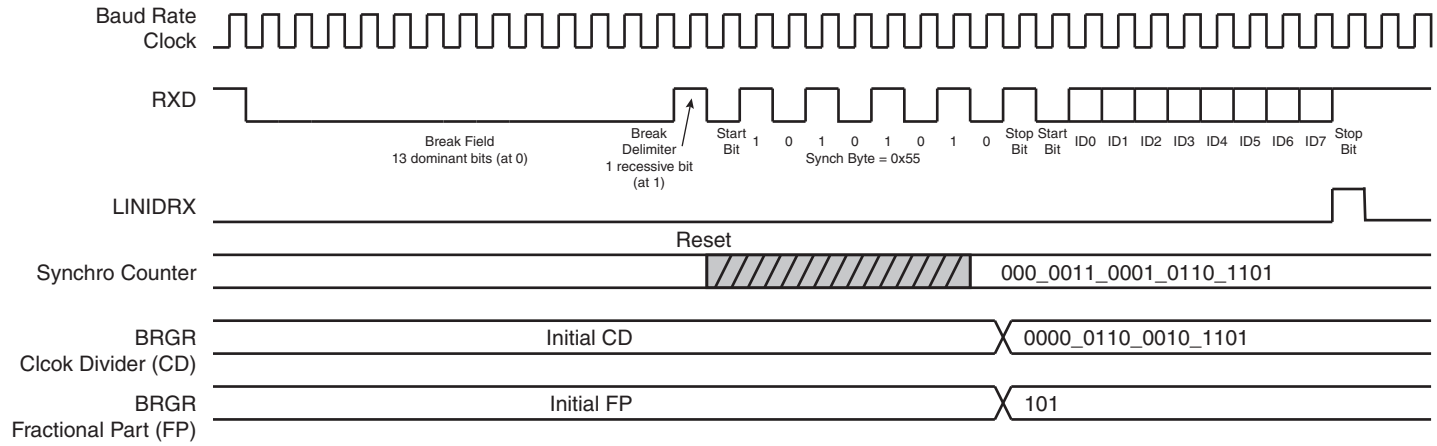
Synchronization is only done by the slave. If the Sync field is not 0x55, an Inconsistent Sync Field error (CSR.LINISFE) is generated. The time between falling edges is measured by a 19-bit counter, driven by the sampling clock (see Section 19.6.1).

**Figure 19-23.** Sync Field




The counter starts when the Sync field start bit is detected, and continues for eight bit periods. The 16 most significant bits (counter value divided by 8) becomes the new clock divider (BRGR.CD), and the three least significant bits (the remainder) becomes the new fractional part (BRGR.FP).

**Figure 19-24.** Slave Node Synchronization




The synchronization accuracy depends on:

- The theoretical slave node clock frequency; nominal clock frequency ($F_{Nom}$)
- The baud rate

### 19.7.5 Interrupt Mask Register

**Name:** IMR

**Access Type:** Read-only

**Offset:** 0x10

**Reset Value:** 0x00000000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| – | – | LINSNRE | LINCE | LINIPE | LINISFE | LINBE | – |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| – | – | – | – | CTSIC | – | – | – |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| LINTC | LINID | NACK/LINBK | RXBUFF | – | ITER/UNRE | TXEMPTY | TIMEOUT |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PARE | FRAME | OVRE | – | – | RXBRK | TXRDY | RXRDY |

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.

### 19.7.13 LIN Identifier Register

**Name:** LINIR

**Access Type:** Read-write or Read-only

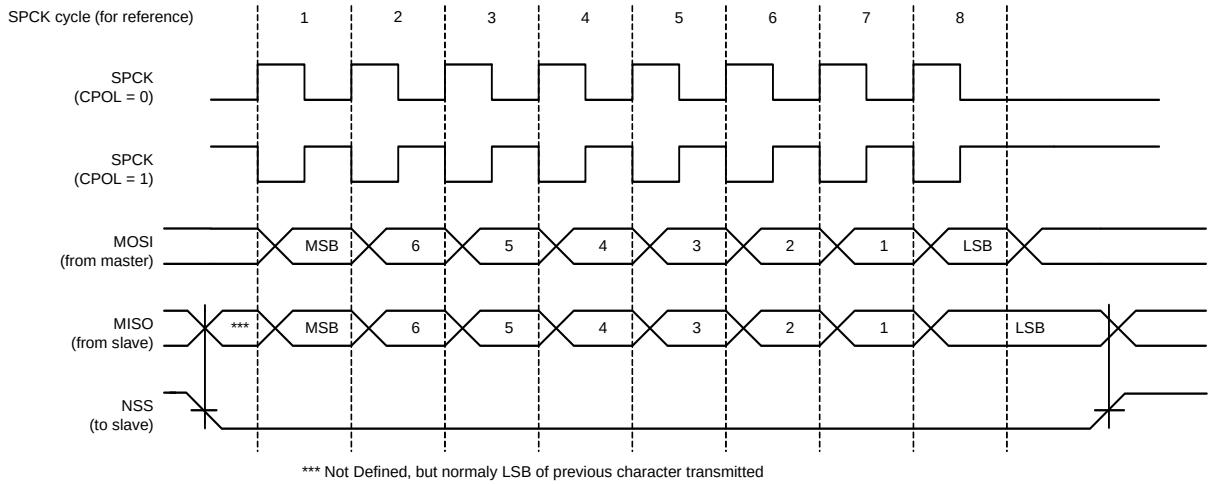
**Offset:** 0x58

**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 |
| IDCHR | | | | | | | |

- **IDCHR: Identifier Character**
  If USART is in LIN master mode, the IDCHR field is read-write, and its value is the Identifier character to be transmitted.
  If USART is in LIN slave mode, the IDCHR field is read-only, and its value is the last received Identifier character.

**Figure 20-4.** SPI Transfer Format (NCPHA = 0, 8 bits per transfer)




### 20.7.3 Master Mode Operations

When configured in master mode, the SPI uses the internal programmable baud rate generator as clock source. It fully controls the data transfers to and from the slave(s) connected to the SPI bus. The SPI drives the chip select line to the slave and the serial clock signal (SPCK).

The SPI features two holding registers, the Transmit Data Register (TDR) and the Receive Data Register (RDR), and a single Shift Register. The holding registers maintain the data flow at a constant rate.

After enabling the SPI, a data transfer begins when the processor writes to the TDR register. The written data is immediately transferred in the Shift Register and transfer on the SPI bus starts. While the data in the Shift Register is shifted on the MOSI line, the MISO line is sampled and shifted in the Shift Register. Transmission cannot occur without reception.

Before writing to the TDR, the Peripheral Chip Select field in TDR (TDR.PCS) must be written in order to select a slave.

If new data is written to TDR during the transfer, it stays in it until the current transfer is completed. Then, the received data is transferred from the Shift Register to RDR, the data in TDR is loaded in the Shift Register and a new transfer starts.

The transfer of a data written in TDR in the Shift Register is indicated by the Transmit Data Register Empty bit in the Status Register (SR.TDRE). When new data is written in TDR, this bit is cleared. The SR.TDRE bit is used to trigger the Transmit Peripheral DMA Controller channel.

The end of transfer is indicated by the Transmission Registers Empty bit in the SR register (SR.TXEMPTY). If a transfer delay (CSRn.DLYBCT) is greater than zero for the last transfer, SR.TXEMPTY is set after the completion of said delay. The CLK_SPI can be switched off at this time.

During reception, received data are transferred from the Shift Register to the reception FIFO. The FIFO can contain up to 4 characters (both Receive Data and Peripheral Chip Select fields). While a character of the FIFO is unread, the Receive Data Register Full bit in SR remains high (SR.RDRF). Characters are read through the RDR register. If the four characters stored in the FIFO are not read and if a new character is stored, this sets the Overrun Error Status bit in the SR register (SR.OVRES). The procedure to follow in such a case is described in Section 20.7.3.8.

### 20.8.14 Write Protection Status Register

**Register Name:** WPSR

**Access Type:** Read-only

**Offset:** 0xE8

**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 |
|---|---|---|---|---|---|---|---|
| SPIWPVSRC | | | | | | | |

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

- **SPIWPVSRC: SPI Write Protection Violation Source**
  This Field indicates the Peripheral Bus Offset of the register concerned by the violation (MR or CSRx)
- **SPIWPVS: SPI Write Protection Violation Status**

| SPIWPVS value | Violation Type |
|---|---|
| 1 | The Write Protection has blocked a Write access to a protected register (since the last read). |
| 2 | Software Reset has been performed while Write Protection was enabled (since the last read or since the last write access on MR, IER, IDR or CSRx). |
| 3 | Both Write Protection violation and software reset with Write Protection enabled have occurred since the last read. |
| 4 | Write accesses have been detected on MR (while a chip select was active) or on CSRi (while the Chip Select “i” was active) since the last read. |
| 5 | The Write Protection has blocked a Write access to a protected register and write accesses have been detected on MR (while a chip select was active) or on CSRi (while the Chip Select “i” was active) since the last read. |
| 6 | Software Reset has been performed while Write Protection was enabled (since the last read or since the last write access on MR, IER, IDR or CSRx) and some write accesses have been detected on MR (while a chip select was active) or on CSRi (while the Chip Select “i” was active) since the last read. |
| 7 | - The Write Protection has blocked a Write access to a protected register.<br>and<br>- Software Reset has been performed while Write Protection was enabled.<br>and<br>- Write accesses have been detected on MR (while a chip select was active) or on CSRi (while the Chip Select “i” was active) since the last read. |

In Waveform mode, TIOA is always configured to be an output and TIOB is an output if it is not selected to be the external trigger.

#### 24.6.1.6 *Trigger*

A trigger resets the counter and starts the counter clock. Three types of triggers are common to both modes, and a fourth external trigger is available to each mode.

The following triggers are common to both modes:

- Software Trigger: each channel has a software trigger, available by writing a one to the Software Trigger Command bit in CCRn (CCRn.SWTRG).
- SYNC: each channel has a synchronization signal SYNC. When asserted, this signal has the same effect as a software trigger. The SYNC signals of all channels are asserted simultaneously by writing a one to the Synchro Command bit in the BCR register (BCR.SYNC).
- Compare RC Trigger: RC is implemented in each channel and can provide a trigger when the counter value matches the RC value if the RC Compare Trigger Enable bit in CMRn (CMRn.CPCTRG) is written to one.

The channel can also be configured to have an external trigger. In Capture mode, the external trigger signal can be selected between TIOA and TIOB. In Waveform mode, an external event can be programmed to be one of the following signals: TIOB, XC0, XC1, or XC2. This external event can then be programmed to perform a trigger by writing a one to the External Event Trigger Enable bit in CMRn (CMRn.ENETRG).

If an external trigger is used, the duration of the pulses must be longer than the CLK_TC period in order to be detected.

Regardless of the trigger used, it will be taken into account at the following active edge of the selected clock. This means that the counter value can be read differently from zero just after a trigger, especially when a low frequency signal is selected as the clock.

#### 24.6.1.7 *Peripheral events on TIOA inputs*

The TIOA input lines are ored internally with peripheral events from the Peripheral Event System. To capture using events the user must ensure that the corresponding pin functions for the TIOA line are disabled. When capturing on the external TIOA pin the user must ensure that no peripheral events are generated on this pin.

### 24.6.2 Capture Operating Mode

This mode is entered by writing a zero to the CMRn.WAVE bit.

Capture mode allows the TC channel to perform measurements such as pulse timing, frequency, period, duty cycle and phase on TIOA and TIOB signals which are considered as inputs.

Figure 24-4 on page 566 shows the configuration of the TC channel when programmed in Capture mode.

#### 24.6.2.1 *Capture registers A and B*

Registers A and B (RA and RB) are used as capture registers. This means that they can be loaded with the counter value when a programmable event occurs on the signal TIOA.

### 24.6.3 Waveform Operating Mode

Waveform operating mode is entered by writing a one to the CMRn.WAVE bit.

In Waveform operating mode the TC channel generates one or two PWM signals with the same frequency and independently programmable duty cycles, or generates different types of one-shot or repetitive pulses.

In this mode, TIOA is configured as an output and TIOB is defined as an output if it is not used as an external event.

Figure 24-5 on page 568 shows the configuration of the TC channel when programmed in Waveform operating mode.

#### *24.6.3.1 Waveform selection*

Depending on the Waveform Selection field in CMRn (CMRn.WAVSEL), the behavior of CVn varies.

With any selection, RA, RB and RC can all be used as compare registers.

RA Compare is used to control the TIOA output, RB Compare is used to control the TIOB output (if correctly configured) and RC Compare is used to control TIOA and/or TIOB outputs.

#### 27.6.1.2 User Triggered Single Measurement Mode

In the UT mode, the user starts a single comparison by writing a one to the User Start Single Comparison bit (CTRL.USTART). This mode is enabled by writing CONFn.MODE to 2. After the startup time has passed, a single comparison is done and SR is updated. Appropriate peripheral events and interrupts are also generated. No new comparisons will be performed. CTRL.USTART is cleared automatically by hardware when the single comparison has been done.

#### 27.6.1.3 Event Triggered Single Measurement Mode

This mode is enabled by writing CONFn.MODE to 3 and Peripheral Event Trigger Enable (CTRL.EVENTEN) to one. The ET mode is similar to the UT mode, the difference is that a peripheral event from another hardware module causes the hardware to automatically set the Peripheral Event Start Single Comparison bit (CTRL.ESTART). After the startup time has passed, a single comparison is done and SR is updated. Appropriate peripheral events and interrupts are also generated. No new comparisons will be performed. CTRL.ESTART is cleared automatically by hardware when the single comparison has been done.

#### 27.6.1.4 Selecting Comparator Inputs

Each Analog Comparator has one positive (INP) and one negative (INN) input. The positive input is fed from an external input pin (ACPn). The negative input can either be fed from an external input pin (ACNn) or from a reference voltage common to all ACs (ACREFN).

The user selects the input source as follows:

- In normal mode with the Negative Input Select and Positive Input Select fields (CONFn.INSELN and CONFn.INSELP).
- In window mode with CONFn.INSELN, CONFn.INSELP and CONFn+1.INSELN, CONFn+1,INSELP. The user must configure CONFn.INSELN and CONFn+1.INSELP to the same source.

### 27.6.2 Interrupt Generation

The interrupt request will be generated if the corresponding bit in the Interrupt Mask Register (IMR) is set. Bits in IMR are set by writing a one to the corresponding bit in the Interrupt Enable Register (IER), and cleared by writing a one to the corresponding bit in the Interrupt Disable Register (IDR). The interrupt request remains active until the corresponding bit in ISR is cleared by writing a one to the corresponding bit in the Interrupt Status Clear Register (ICR).

### 27.6.3 Peripheral Event Generation

The ACIFB can be set up so that certain comparison results notify other parts of the device via the Peripheral Event system. Refer to Section 27.6.4.3 and Section 27.6.5.3 for information on which comparison results can generate events, and how to configure the ACIFB to achieve this.

Zero or one event will be generated per comparison.

### 27.6.4 Normal Mode

In normal mode all Analog Comparators are operating independently.

#### 27.6.4.1 Normal Mode Output

Each Analog Comparator generates one output ACOUT according to the input voltages on INP (AC positive input) and INN (AC negative input):

**Table 28-1.** I/O Lines Description

| Name | Description | Type |
|---|---|---|
| SMP | SMP line (only used for QMatrix) | Output |
| SYNC | Synchronize signal | Input |
| VDIVEN | Voltage divider enable (only used for QMatrix) | Output |

## 28.5 Product Dependencies

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

### 28.5.1 I/O Lines

The CAT pins may be multiplexed with other peripherals. The user must first program the I/O Controller to give control of the pins to the CAT module. In QMatrix mode, the Y lines must be driven by the CAT and analog comparators sense the voltage on the Y lines. Thus, the CAT (not the Analog Comparator Interface) must be the selected function for the Y lines in the I/O Controller.

By writing ones and zeros to bits in the Pin Mode Registers (PINMODEx), most of the CAT pins can be individually selected to implement the QTouch method or the QMatrix method. Each pin has a different name and function depending on whether it is implementing the QTouch method or the QMatrix method. The following table shows the pin names for each method and the bits in the PINMODEx registers which control the selection of the QTouch or QMatrix method.

**Table 28-2.** Pin Selection Guide

| CAT Module Pin Name | QTouch Method Pin Name | QMatrix Method Pin Name | Selection Bit in PINMODEx Register |
|---|---|---|---|
| CSA0 | SNS0 | X0 | SP0 |
| CSB0 | SNSK0 | X1 | SP0 |
| CSA1 | SNS1 | Y0 | SP1 |
| CSB1 | SNSK1 | YK0 | SP1 |
| CSA2 | SNS2 | X2 | SP2 |
| CSB2 | SNSK2 | X3 | SP2 |
| CSA3 | SNS3 | Y1 | SP3 |
| CSB3 | SNSK3 | YK1 | SP3 |
| CSA4 | SNS4 | X4 | SP4 |
| CSB4 | SNSK4 | X5 | SP4 |
| CSA5 | SNS5 | Y2 | SP5 |
| CSB5 | SNSK5 | YK2 | SP5 |
| CSA6 | SNS6 | X6 | SP6 |
| CSB6 | SNSK6 | X7 | SP6 |
| CSA7 | SNS7 | Y3 | SP7 |
| CSB7 | SNSK7 | YK3 | SP7 |
| CSA8 | SNS8 | X8 | SP8 |

## 30.7 User Interface

**Table 30-2. aWire UART user interface Register Memory Map**

| Offset | Register | Register Name | Access | Reset |
|---|---|---|---|---|
| 0x00 | Control Register | CTRL | Read/Write | 0x00000000 |
| 0x04 | Status Register | SR | Read-only | 0x00000000 |
| 0x08 | Status Clear Register | SCR | Write-only | - |
| 0x0C | Interrupt Enable Register | IER | Write-only | - |
| 0x10 | Interrupt Disable Register | IDR | Write-only | - |
| 0x14 | Interrupt Mask Register | IMR | Read-only | 0x00000000 |
| 0x18 | Receive Holding Register | RHR | Read-only | 0x00000000 |
| 0x1C | Transmit Holding Register | THR | Read/Write | 0x00000000 |
| 0x20 | Baud Rate Register | BRR | Read/Write | 0x00000000 |
| 0x24 | Version Register | VERSION | Read-only | -(1) |
| 0x28 | Clock Request Register | CLKR | Read/Write | 0x00000000 |

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

**Table 31-11.** Instruction Description (Continued)

| Instruction | Description |
|---|---|
| DR Size | Shows the number of bits in the data register chain when this instruction is active.<br>Example: 34 bits |
| DR input value | Shows which bit pattern to shift into the data register in the Shift-DR state when this instruction is active. Multiple such lines may exist, e.g., to distinguish between reads and writes.<br>Example: aaaaaaar xxxxxxxx xxxxxxxx xxxxxxxx xx |
| DR output value | Shows the bit pattern shifted out of the data register in the Shift-DR state when this instruction is active. Multiple such lines may exist, e.g., to distinguish between reads and writes.<br>Example: xx **xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxeb** |

### 31.5.2 Public JTAG Instructions

The JTAG standard defines a number of public JTAG instructions. These instructions are described in the sections below.

#### *31.5.2.1 IDCODE*

This instruction selects the 32 bit Device Identification register (DID) as Data Register. The DID register consists of a version number, a device number, and the manufacturer code chosen by JEDEC. This is the default instruction after a JTAG reset. Details about the DID register can be found in the module configuration section at the end of this chapter.

Starting in Run-Test/Idle, the Device Identification register is accessed in the following way:

1. Select the IR Scan path.
2. In Capture-IR: The IR output value is latched into the shift register.
3. In Shift-IR: The instruction register is shifted by the TCK input.
4. Return to Run-Test/Idle.
5. Select the DR Scan path.
6. In Capture-DR: The IDCODE value is latched into the shift register.
7. In Shift-DR: The IDCODE scan chain is shifted by the TCK input.
8. Return to Run-Test/Idle.

**Table 31-12.** IDCODE Details

| Instructions | Details |
|---|---|
| IR input value | **00001** (0x01) |
| IR output value | p0001 |
| DR Size | 32 |
| DR input value | xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx |
| DR output value | Device Identification Register |

#### *31.5.2.2 SAMPLE_PRELOAD*

This instruction takes a snap-shot of the input/output pins without affecting the system operation, and pre-loading the scan chain without updating the DR-latch. The boundary-scan chain is selected as Data Register.

Starting in Run-Test/Idle, the Device Identification register is accessed in the following way:

**Table 31-21.** MEMORY_SIZED_ACCESS Details (Continued)

| Instructions | Details |
|---|---|
| DR output value (Address phase) | xxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxeb |
| DR output value (Data read phase) | xxxxxeb **dddddddd dddddddd dddddddd dddddddd** |
| DR output value (Data write phase) | xxxxxxx **xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxeb** |

### *31.5.3.4 MEMORY_WORD_ACCESS*

This instruction allows access to the entire Service Access Bus data area. Data is accessed through the 34 MSB of the SAB address, a direction bit, and 32 bits of data. This instruction is identical to MEMORY_SIZED_ACCESS except that it always does word sized accesses. The size field is implied, and the two lowest address bits are removed and not scanned in.

Note: This instruction was previously known as MEMORY_ACCESS, and is provided for backwards compatibility.

The data register is alternately interpreted by the SAB as an address register and a data register. The SAB starts in address mode after the MEMORY_WORD_ACCESS instruction is selected, and toggles between address and data mode each time a data scan completes with the busy bit cleared.

Starting in Run-Test/Idle, SAB data is accessed in the following way:

1. Select the IR Scan path.
2. In Capture-IR: The IR output value is latched into the shift register.
3. In Shift-IR: The instruction register is shifted by the TCK input.
4. Return to Run-Test/Idle.
5. Select the DR Scan path.
6. In Shift-DR: Scan in the direction bit (1=read, 0=write) and the 34-bit address of the data to access.
7. Go to Update-DR and re-enter Select-DR Scan.
8. In Shift-DR: For a read operation, scan out the contents of the addressed area. For a write operation, scan in the new contents of the area.
9. Return to Run-Test/Idle.

For any operation, the full 34 bits of the address must be provided. For write operations, 32 data bits must be provided, or the result will be undefined. For read operations, shifting may be terminated once the required number of bits have been acquired.

**Table 31-22.** MEMORY_WORD_ACCESS Details

| Instructions | Details |
|---|---|
| IR input value | **10001** (0x11) |
| IR output value | peb01 |
| DR Size | 35 bits |
| DR input value (Address phase) | aaaaaaaa aaaaaaaa aaaaaaaa aaaaaaaa aar |
| DR input value (Data read phase) | **xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx** xxx |
| DR input value (Data write phase) | **dddddddd dddddddd dddddddd dddddddd** xxx |

**Table 31-23.** MEMORY_BLOCK_ACCESS Details (Continued)

| Instructions | Details |
|---|---|
| DR input value (Data write phase) | **dddddddd dddddddd dddddddd dddddddd** xx |
| DR output value (Data read phase) | eb **dddddddd dddddddd dddddddd dddddddd** |
| DR output value (Data write phase) | xx **xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxeb** |

The overhead using block word access is 4 cycles per 32 bits of data, resulting in an 88% transfer efficiency, or 2.1 MBytes per second with a 20 MHz TCK frequency.

#### *31.5.3.6 CANCEL_ACCESS*

If a very slow memory location is accessed during a SAB memory access, it could take a very long time until the busy bit is cleared, and the SAB becomes ready for the next operation. The CANCEL_ACCESS instruction provides a possibility to abort an ongoing transfer and report a timeout to the JTAG master.

When the CANCEL_ACCESS instruction is selected, the current access will be terminated as soon as possible. There are no guarantees about how long this will take, as the hardware may not always be able to cancel the access immediately. The SAB is ready to respond to a new command when the busy bit clears.

Starting in Run-Test/Idle, CANCEL_ACCESS is accessed in the following way:

1. Select the IR Scan path.
2. In Capture-IR: The IR output value is latched into the shift register.
3. In Shift-IR: The instruction register is shifted by the TCK input.
4. Return to Run-Test/Idle.

**Table 31-24.** CANCEL_ACCESS Details

| Instructions | Details |
|---|---|
| IR input value | **10011** (0x13) |
| IR output value | peb01 |
| DR Size | 1 |
| DR input value | x |
| DR output value | 0 |

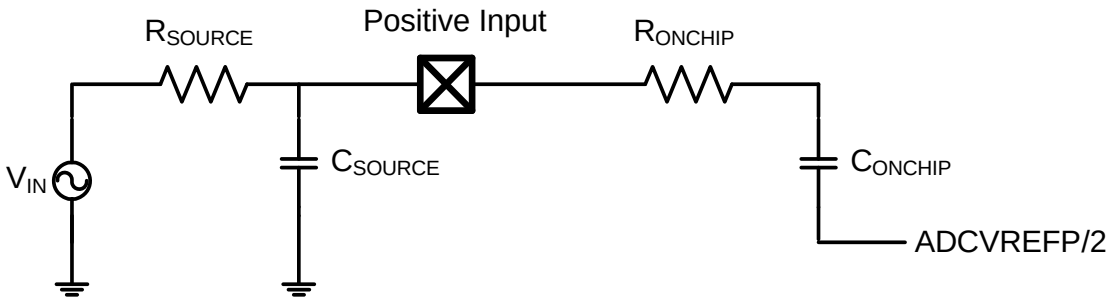
#### *31.5.3.7 SYNC*

This instruction allows external debuggers and testers to measure the ratio between the external JTAG clock and the internal system clock. The SYNC data register is a 16-bit counter that counts down to zero using the internal system clock. The busy bit stays high until the counter reaches zero.

Starting in Run-Test/Idle, SYNC instruction is used in the following way:

1. Select the IR Scan path.
2. In Capture-IR: The IR output value is latched into the shift register.
3. In Shift-IR: The instruction register is shifted by the TCK input.
4. Return to Run-Test/Idle.
5. Select the DR Scan path.

**Figure 32-7.** ADC Input




The minimum sample and hold time (in ns) can be found using this formula:

$$t_{SAMPLEHOLD} \geq (R_{ONCHIP} + R_{SOURCE}) \times (C_{ONCHIP} + C_{SOURCE}) \times \ln(2^{n+1})$$

Where n is the number of bits in the conversion. $t_{SAMPLEHOLD}$ is defined by the SHTIM field in the ADCIFB ACR register. Please refer to the ADCIFB chapter for more information.

### *32.8.6.2 Applicable Conditions and Derating Data*

**Table 32-30.** Transfer Characteristics 12-bit Resolution Mode[1]

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Resolution | | | 12 | | Bit |
| Integral non-linearity | ADC clock frequency = 6MHz, Input Voltage Range = 0 - $V_{ADVREFP}$ | | **+/-4** | | LSB |
| | ADC clock frequency = 6MHz, Input Voltage Range = (10% $V_{ADVREFP}$) - (90% $V_{ADVREFP}$) | | +/-2 | | |
| Differential non-linearity | ADC clock frequency = 6MHz | -1.5 | | 1.5 | |
| Offset error | | | +/-3 | | |
| Gain error | | | +/-5 | | |

Note: 1. These values are based on simulation and characterization of other AVR microcontrollers manufactured in the same process technology. These values are not covered by test limits in production.

**Table 32-31.** Transfer Characteristics, 10-bit Resolution Mode[1]

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Resolution | | | 10 | | Bit |
| Integral non-linearity | ADC clock frequency = 6MHz | | +/-1 | | LSB |
| Differential non-linearity | | -1 | | 1 | |
| Offset error | | | +/-1 | | |
| Gain error | | | +/-2 | | |