**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 | 256KB (256K 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/at32uc3l0256-aut |

## 3.3 Signal Descriptions

The following table gives details on signal names classified by peripheral.

**Table 3-7.** Signal Descriptions List

| Signal Name | Function | Type | Active Level | Comments |
|---|---|---|---|---|
| **Analog Comparator Interface - ACIFB** | | | | |
| ACAN3 - ACAN0 | Negative inputs for comparators "A" | Analog | | |
| ACAP3 - ACAP0 | Positive inputs for comparators "A" | Analog | | |
| ACBN3 - ACBN0 | Negative inputs for comparators "B" | Analog | | |
| ACBP3 - ACBP0 | Positive inputs for comparators "B" | Analog | | |
| ACREFN | Common negative reference | Analog | | |
| **ADC Interface - ADCIFB** | | | | |
| AD8 - AD0 | Analog Signal | Analog | | |
| ADP1 - ADP0 | Drive Pin for resistive touch screen | Output | | |
| TRIGGER | External trigger | Input | | |
| **aWire - AW** | | | | |
| DATA | aWire data | I/O | | |
| DATAOUT | aWire data output for 2-pin mode | I/O | | |
| **Capacitive Touch Module - CAT** | | | | |
| CSA16 - CSA0 | Capacitive Sense A | I/O | | |
| CSB16 - CSB0 | Capacitive Sense B | I/O | | |
| DIS | Discharge current control | Analog | | |
| SMP | SMP signal | Output | | |
| SYNC | Synchronize signal | Input | | |
| VDIVEN | Voltage divider enable | Output | | |
| **External Interrupt Controller - EIC** | | | | |
| NMI (EXTINT0) | Non-Maskable Interrupt | Input | | |
| EXTINT5 - EXTINT1 | External interrupt | Input | | |
| **Glue Logic Controller - GLOC** | | | | |
| IN7 - IN0 | Inputs to lookup tables | Input | | |
| OUT1 - OUT0 | Outputs from lookup tables | Output | | |
| **JTAG module - JTAG** | | | | |
| TCK | Test Clock | Input | | |
| TDI | Test Data In | Input | | |
| TDO | Test Data Out | Output | | |
| TMS | Test Mode Select | Input | | |

### 7.7.3 Performance Monitor Memory Map

**Table 7-3.** PDCA Performance Monitor Registers[1]

| Offset | Register | Register Name | Access | Reset |
|---|---|---|---|---|
| 0x800 | Performance Control Register | PCONTROL | Read/Write | 0x00000000 |
| 0x804 | Channel0 Read Data Cycles | PRDATA0 | Read-only | 0x00000000 |
| 0x808 | Channel0 Read Stall Cycles | PRSTALL0 | Read-only | 0x00000000 |
| 0x80C | Channel0 Read Max Latency | PRLAT0 | Read-only | 0x00000000 |
| 0x810 | Channel0 Write Data Cycles | PWDATA0 | Read-only | 0x00000000 |
| 0x814 | Channel0 Write Stall Cycles | PWSTALL0 | Read-only | 0x00000000 |
| 0x818 | Channel0 Write Max Latency | PWLAT0 | Read-only | 0x00000000 |
| 0x81C | Channel1 Read Data Cycles | PRDATA1 | Read-only | 0x00000000 |
| 0x820 | Channel1 Read Stall Cycles | PRSTALL1 | Read-only | 0x00000000 |
| 0x824 | Channel1 Read Max Latency | PRLAT1 | Read-only | 0x00000000 |
| 0x828 | Channel1 Write Data Cycles | PWDATA1 | Read-only | 0x00000000 |
| 0x82C | Channel1 Write Stall Cycles | PWSTALL1 | Read-only | 0x00000000 |
| 0x830 | Channel1 Write Max Latency | PWLAT1 | Read-only | 0x00000000 |

Note: 1. The number of performance monitors is device specific. If the device has only one performance monitor, the Channel1 registers are not available. Please refer to the Module Configuration section at the end of this chapter for the number of performance monitors on this device.

### 7.7.4 Version Register Memory Map

**Table 7-4.** PDCA Version Register Memory Map

| Offset | Register | Register Name | Access | Reset |
|---|---|---|---|---|
| 0x834 | Version Register | VERSION | Read-only | -[1] |

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

#### 12.6.3.3 Waking from sleep modes

There are two types of wake-up sources from sleep mode, synchronous and asynchronous. Synchronous wake-up sources are all non-masked interrupts. Asynchronous wake-up sources are AST, WDT, external interrupts from EIC, external reset, external wake pin (WAKE_N), and all asynchronous wake-ups enabled in the Asynchronous Wake Up Enable (AWEN) register. The valid wake-up sources for each sleep mode are detailed in Table 12-3 on page 156.

In Shutdown the only wake-up sources are external reset, external wake-up pin or AST. See Section 12.6.4.3 on page 158.

**Table 12-3.** Wake-up Sources

| Index[1] | Sleep Mode | Wake-up Sources |
|---|---|---|
| 0 | **Idle** | Synchronous, Asynchronous |
| 1 | **Frozen** | Synchronous[2], Asynchronous |
| 2 | **Standby** | Asynchronous |
| 3 | **Stop** | Asynchronous |
| 4 | **DeepStop** | Asynchronous |
| 5 | **Static** | Asynchronous[3] |
| 6 | **Shutdown** | External reset, External wake-up pin |

Notes:
1. The sleep mode index is used as argument for the sleep instruction.
2. Only PB modules operational, as HSB module clocks are stopped.
3. WDT only available if clocked from pre-enabled OSC32K.

#### 12.6.3.4 SleepWalking

In all sleep modes where the PBx clocks are stopped, except for Shutdown mode, the device can partially wake up if a PBx module asynchronously discovers that it needs its clock. Only the requested clocks and clock sources needed will be started, all other clocks will remain masked to zero. E.g. if the main clock source is OSC0, only OSC0 will be started even if other clock sources were enabled in normal mode. Generic clocks can also be started in a similar way. The state where only requested clocks are running is referred to as SleepWalking.

The time spent to start the requested clock is mostly limited by the startup time of the given clock source. This allows PBx modules to handle incoming requests, while still keeping the power consumption at a minimum.

When the device is SleepWalking any asynchronous wake-up can wake the device up at any time without stopping the requested PBx clock.

All requests to start clocks can be masked by writing to the Peripheral Power Control Register (PPCR), all requests are enabled at reset.

During SleepWalking the interrupt controller clock will be running. If an interrupt is pending when entering SleepWalking, it will wake the whole device up.

#### 12.6.3.5 Precautions when entering sleep mode

Modules communicating with external circuits should normally be disabled before entering a sleep mode that will stop the module operation. This will prevent erratic behavior caused by entering or exiting sleep modes. Please refer to the relevant module documentation for recommended actions.

the BRIFARDY bit in the Interrupt Mask Register (IMR.BRIFARDY) is set. This bit is set by writing a one to the corresponding bit in the Interrupt Enable Register (IER.BRIFARDY).

After powering up the device the Backup Register Interface Valid bit in PCLKSR (PCLKSR.BRIFAVALID) is cleared, indicating that the content of the backup registers has not been written and contains the reset value. After writing to one of the backup registers the PCLKSR.BRIFAVALID bit is set. During writes to the backup registers (when BRIFARDY is zero) BRIFAVALID will be zero. If a reset occurs when BRIFARDY is zero, BRIFAVALID will be cleared after the reset, indicating that the content of the backup registers is not valid. If BRIFARDY is one when a reset occurs, BRIFAVALID will be one and the content is the same as before the reset.

The user must ensure that BRIFAVALID and BRIFARDY are both set before reading the backup register values.

### 13.5.13 32kHz RC Oscillator (RC32K)

Rev: 1.1.0.0

The RC32K can be used as source for the generic clocks, as described in The Generic Clocks section.

The 32kHz RC oscillator (RC32K) is forced on after reset, and output on PA20. The clock is available on the pad until the PPCR.FRC32 bit in the Power Manager has been cleared or a different peripheral function has been chosen on PA20 (PA20 will start with peripheral function F by default). Note that the forcing will only enable the clock output. To be able to use the RC32K normally the oscillator must be enabled as described below.

The oscillator is enabled by writing a one to the Enable bit in the 32kHz RC Oscillator Configuration Register (RC32KCR.EN) and disabled by writing a zero to RC32KCR.EN. The oscillator is also automatically enabled when the sampling mode is requested for the SM33. In this case, writing a zero to RC32KCR.EN will not disable the RC32K until the sampling mode is no longer requested.

### 13.5.14 Generic Clock Prescalers

Rev: 1.0.0.0

The generic clocks can be sourced by two special prescalers to increase the generic clock frequency precision.

These prescalers are named the High Resolution Prescaler (HRP) and the Fractional Prescaler (FP).

#### *13.5.14.1 High resolution prescaler*

The HRP is a 24-bit counter that can generate a very accurate clock waveform. The clock obtained has 50% duty cycle.

### 16.7.13 Disable Register

**Name:** DIS

**Access Type:** Write-only

**Offset:** 0x034

**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**
  Writing a zero to this bit has no effect.
  Writing a one to this bit will disable the corresponding external interrupt.
  Please refer to the Module Configuration section for the number of external interrupts.
- **NMI: Non-Maskable Interrupt**
  Writing a zero to this bit has no effect.
  Writing a one to this bit will disable the Non-Maskable Interrupt.

# 17. Frequency Meter (FREQM)

Rev: 3.1.0.1

## 17.1 Features

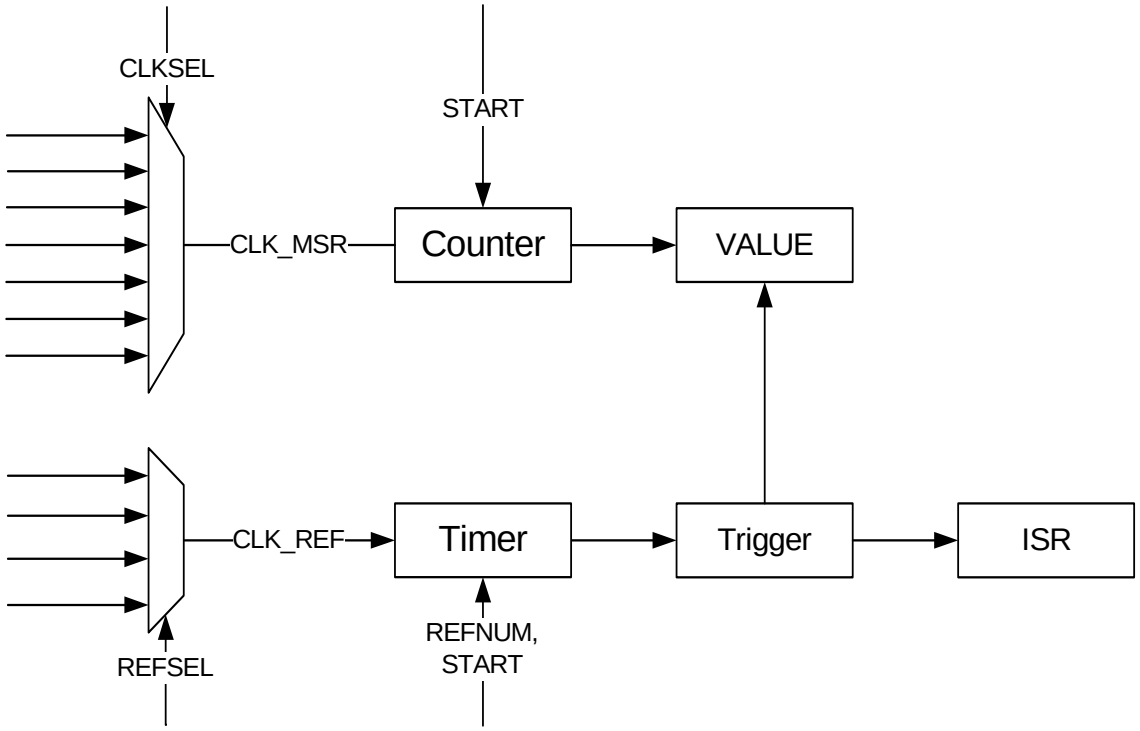
- **Accurately measures a clock frequency**
- **Selectable reference clock**
- **A selectable clock can be measured**
- **Ratio can be measured with 24-bit accuracy**

## 17.2 Overview

The Frequency Meter (FREQM) can be used to accurately measure the frequency of a clock by comparing it to a known reference clock.

## 17.3 Block Diagram

**Figure 17-1.** Frequency Meter Block Diagram




## 17.4 Product Dependencies

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

### 17.4.1 Power Management

The device can enter a sleep mode while a measurement is ongoing. However, make sure that neither CLK_MSR nor CLK_REF is stopped in the actual sleep mode. FREQM interrupts can wake up the device from sleep modes when the measurement is done, but only from sleep modes where CLK_FREQM is running. Please refer to the Power Manager chapter for details.

### 18.7.8 Output Driver Enable Register

**Name:** ODER

**Access:** Read/Write, Set, Clear, Toggle

**Offset**: 0x040, 0x044, 0x048, 0x04C

**Reset Value:** -

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| P31 | P30 | P29 | P28 | P27 | P26 | P25 | P24 |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| P23 | P22 | P21 | P20 | P19 | P18 | P17 | P16 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| P15 | P14 | P13 | P12 | P11 | P10 | P9 | P8 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |

- **P0-31: Output Driver Enable**
  0: The output driver is disabled for the corresponding pin.
  1: The output driver is enabled for the corresponding pin.

**Figure 19-16.** Receiver Behavior when Operating with Hardware Handshaking

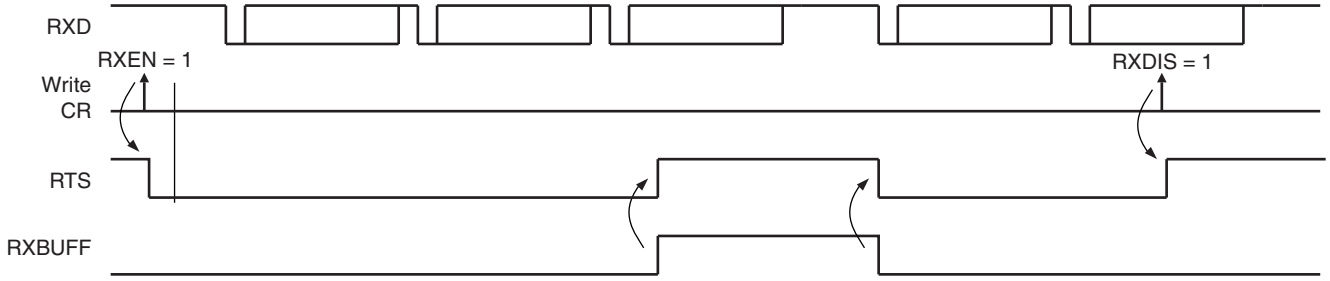



**Figure 19-17.** Transmitter Behavior when Operating with Hardware Handshaking

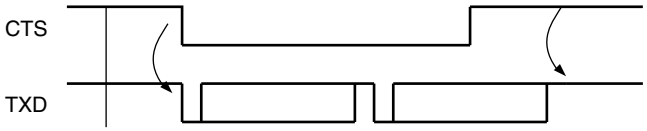



**Figure 19-18.**

### 19.6.4 SPI Mode

The USART features a Serial Peripheral Interface (SPI) link compliant mode, supporting synchronous, full-duplex communication, in both master and slave mode. Writing 0xE (master) or 0xF (slave) to MR.MODE will enable this mode. A SPI in master mode controls the data flow to and from the other SPI devices, who are in slave mode. It is possible to let devices take turns being masters (aka multi-master protocol), and one master may shift data simultaneously into several slaves, but only one slave may respond at a time. A slave is selected when its slave select (NSS) signal has been raised by the master. The USART can only generate one NSS signal, and it is possible to use standard I/O lines to address more than one slave.

#### *19.6.4.1 Modes of Operation*

The SPI system consists of two data lines and two control lines:

- Master Out Slave In (MOSI): This line supplies the data shifted from master to slave. In master mode this is connected to TXD, and in slave mode to RXD.
- Master In Slave Out (MISO): This line supplies the data shifted from slave to master. In master mode this is connected to RXD, and in slave mode to TXD.
- Serial Clock (CLK): This is controlled by the master. One period per bit transmission. In both modes this is connected to CLK.
- Slave Select (NSS): This control line allows the master to select or deselect a slave. In master mode this is connected to RTS, and in slave mode to CTS.

Changing SPI mode after initial configuration has to be followed by a transceiver software reset in order to avoid unpredictable behavior.

#### *19.6.4.2 Baud Rate*

The baud rate generator operates as described in ”Baud Rate in Synchronous and SPI Mode” on page 381, with the following requirements:

In SPI Master Mode:

CPOL is used to determine the inactive state value of the serial clock (SPCK). It is used with NCPHA to produce the required clock/data relationship between master and slave devices.

### 20.8.16 Version Register

**Register Name:** VERSION

**Access Type:** Read-only

**Offset:** 0xFC

**Reset Value:** –

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

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| | | | | VERSION[11:8] | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| VERSION[7:0] | | | | | | | |

- **MFN**

  Reserved. No functionality associated.

- **VERSION**

  Version number of the module. No functionality associated.

### 22.9.7 Status Register

**Name:** SR

**Access Type:** Read-only

**Offset**: 0x18

**Reset Value:** 0x000000002

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

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| BTF | REP | STO | SMBDAM | SMBHHM | SMBALERTM | GCM | SAM |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| - | BUSERR | SMBPECERR | SMBTOUT | - | - | - | NAK |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| ORUN | URUN | TRA | - | TCOMP | SEN | TXRDY | RXRDY |

- **BTF: Byte Transfer Finished**
  This bit is cleared when the corresponding bit in SCR is written to one.
  This bit is set when byte transfer has completed.
- **REP: Repeated Start Received**
  This bit is cleared when the corresponding bit in SCR is written to one.
  This bit is set when a REPEATED START condition is received.
- **STO: Stop Received**
  This bit is cleared when the corresponding bit in SCR is written to one.
  This bit is set when the STOP condition is received.
- **SMBDAM: SMBus Default Address Match**
  This bit is cleared when the corresponding bit in SCR is written to one.
  This bit is set when the received address matched the SMBus Default Address.
- **SMBHHM: SMBus Host Header Address Match**
  This bit is cleared when the corresponding bit in SCR is written to one.
  This bit is set when the received address matched the SMBus Host Header Address.
- **SMBALERTM: SMBus Alert Response Address Match**
  This bit is cleared when the corresponding bit in SCR is written to one.
  This bit is set when the received address matched the SMBus Alert Response Address.
- **GCM: General Call Match**
  This bit is cleared when the corresponding bit in SCR is written to one.
  This bit is set when the received address matched the General Call Address.
- **SAM: Slave Address Match**
  This bit is cleared when the corresponding bit in SCR is written to one.
  This bit is set when the received address matched the Slave Address.
- **BUSERR: Bus Error**
  This bit is cleared when the corresponding bit in SCR is written to one.
  This bit is set when a misplaced START or STOP condition has occurred.

### 23.6.7 Open Drain Mode

Some pins can be used in open drain mode, allowing the PWMA waveform to toggle between 0V and up to 5V on these pins. In this mode the PWMA will drive the pin to zero or leave the output open. An external pullup can be used to pull the pin up to the desired voltage.

To enable open drain mode on a pin the PWMAOD function must be selected instead of the PWMA function in the I/O Controller. Please refer to the Module Configuration chapter for information about which pins are available in open drain mode.

### 23.6.8 Synchronization

Both the timebase counter and the spread spectrum counter can be reset and the duty cycle registers can be written through the user interface of the module. This requires a synchronization between the PB and GCLK clock domains, which takes a few clock cycles of each clock domain. The BUSY bit in SR indicates when the synchronization is ongoing. Writing to the module while the BUSY bit is set will result in discarding the new value.

Note that the duty cycle registers will not be updated with the new values until the timebase counter reaches its top value, in order to avoid glitches. The BUSY bit in SR will always be set during this updating and synchronization period.

### 23.6.9 Interrupts

When the timebase counter overflows, the Timebase Overflow bit in the Status Register (SR.TOFL) is set. If the corresponding bit in the Interrupt Mask Register (IMR) is set, an interrupt request will be generated.

Since the user needs to wait until the user interface is available between each write due to synchronization, a READY bit is provided in SR, which can be used to generate an interrupt request.

The interrupt request will be generated if the corresponding bit in 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 SR is cleared by writing a one to the corresponding bit in the Status Clear Register (SCR).

### 23.6.10 Peripheral Events

#### *23.6.10.1 Input Peripheral Events*

The pre-defined channels support input peripheral events from the Peripheral Event System. Input peripheral events must be enabled by writing a one to the corresponding bit in the Channel Event Enable Registers (CHEERs) before peripheral events can be used to control the duty cycle value. Each bit in the register corresponds to one channel, where bit 0 corresponds to channel 0 and so on. Both the increase and decrease events are enabled for the corresponding channel when a bit in the CHEER register is set.

An increase or decrease event (event_incr/event_decr) can either increase or decrease the duty cycle value by one in a PWM period. The events are taken into account when the common timebase counter reaches its top. The behavior is defined by the Channel Event Response Register (CHERR). Each bit in the register corresponds to one channel, where bit 0 corresponds to channel 0 and so on. If the bit in CHERR is set to 0 (default) for a channel, the increase event will increase the duty cycle value and the decrease event will decrease the duty cycle value for that channel. If the bit is set to 1, the increase and decrease event will have reverse function so that

the increase event will decrease the duty cycle value and decrease event will increase the duty cycle value. If both the increase event and the decrease event occur at the same time for a channel, the duty cycle value will not be changed.

The number of channels supporting input peripheral events is device specific. Please refer to the Module Configuration section at the end of this chapter for details.

#### *23.6.10.2 Output Peripheral Event*

The PWMA also supports one output peripheral event (event_ch0) to the Peripheral Event System. This output peripheral event is connected to channel 0 and will be asserted when the timebase counter reaches the duty cycle value for channel 0. This output event is always enabled.

**Table 25-3.** Peripheral Event Mapping from AST

| Generator | Generated Event | User | Effect | Asynchronous |
|---|---|---|---|---|
| AST | Overflow event | ACIFB | Comparison is triggered if the ACIFB.CONFn register is written to 11 (Event Triggered Single Measurement Mode) and the EVENTEN bit in the ACIFB.CTRL register is written to 1. | Yes |
| | Periodic event | | | |
| | Alarm event | | | |
| | Overflow event | ADCIFB | Conversion is triggered if the TRGMOD bit in the ADCIFB.TRGR register is written to 111 (Peripheral Event Trigger). | |
| | Periodic event | | | |
| | Alarm event | | | |
| | Overflow event | CAT | Trigger one iteration of autonomous touch detection. | |
| | Periodic event | | | |
| | Alarm event | | | |

**Table 25-4.** Peripheral Event Mapping from PWMA

| Generator | Generated Event | User | Effect | Asynchronous |
|---|---|---|---|---|
| PWMA channel 0 | Timebase counter reaches the duty cycle value. | ACIFB | Comparison is triggered if the ACIFB.CONFn register is written to 11 (Event Triggered Single Measurement Mode) and the EVENTEN bit in the ACIFB.CTRL register is written to 1. | No |
| | | ADCIFB | Conversion is triggered if the TRGMOD bit in the ADCIFB.TRGR register is written to 111 (Peripheral Event Trigger). | |

### 25.4.2 Peripheral Event Connections

Each generated peripheral event is connected to one or more users. If a peripheral event is connected to multiple users, the peripheral event can trigger actions in multiple modules.

A peripheral event user can likewise be connected to one or more peripheral event generators. If a peripheral event user is connected to multiple generators, the peripheral events are OR'ed together to a single peripheral event. This means that peripheral events from either one of the generators will result in a peripheral event to the user.

To configure a peripheral event, the peripheral event must be enabled at both the generator and user side. Even if a generator is connected to multiple users, only the users with the peripheral event enabled will trigger on the peripheral event.

### 25.4.3 Low Power Operation

As the peripheral events do not require CPU intervention, they are available in Idle mode. They are also available in deeper sleep modes if both the generator and user remain clocked in that mode.

Certain events are known as asynchronous peripheral events, as identified in Table 25-1 to Table 25-4. These can be issued even when the system clock is stopped, and revive unclocked user peripherals. The clock will be restarted for this module only, without waking the system from sleep mode. The clock remains active only as long as required by the triggered function, before being switched off again, and the system remains in the original sleep mode. The CPU and sys-

This bit is set when pen contact is detected and pen detect is enabled.

- **OVRE: Overrun Error Status**
  This bit is cleared when no Overrun Error has occurred since the start of a conversion sequence.
  This bit is set when one or more Overrun Error has occurred since the start of a conversion sequence.
- **DRDY: Data Ready Status**
  0: No data has been converted since the last reset.
  1: One or more conversions have completed since the last reset and data is available in LCDR.
  This bit is cleared when CR.SWRST is written to one.
  This bit is set when one or more conversions have completed and data is available in LCDR.

### 26.9.7 Interrupt Status Register

**Name:** ISR

**Access Type:** Read-only

**Offset:** 0x18

**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 |
| - | CELSE | CGT | CLT | - | - | BUSY | READY |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | - | NOCNT | PENCNT | - | - | OVRE | DRDY |

- **CELSE: Compare Else Status**
  This bit is cleared when the corresponding bit in ICR is written to one.
  This bit is set when the corresponding bit in SR has a zero-to-one transition.
- **CGT: Compare Greater Than Status**
  This bit is cleared when the corresponding bit in ICR is written to one.
  This bit is set when the corresponding bit in SR has a zero-to-one transition.
- **CLT: Compare Lesser Than Status**
  This bit is cleared when the corresponding bit in ICR is written to one.
  This bit is set when the corresponding bit in SR has a zero-to-one transition.
- **BUSY: Busy Status**
  This bit is cleared when the corresponding bit in ICR is written to one.
  This bit is set when the corresponding bit in SR has a zero-to-one transition.
- **READY: Ready Status**
  This bit is cleared when the corresponding bit in ICR is written to one.
  This bit is set when the corresponding bit in SR has a zero-to-one transition.
- **NOCNT: No Contact Status**
  This bit is cleared when the corresponding bit in ICR is written to one.
  This bit is set when the corresponding bit in SR has a zero-to-one transition.
- **PENCNT: Pen Contact Status**
  This bit is cleared when the corresponding bit in ICR is written to one.
  This bit is set when the corresponding bit in SR has a zero-to-one transition.
- **OVRE: Overrun Error Status**
  This bit is cleared when the corresponding bit in ICR is written to one.
  This bit is set when the corresponding bit in SR has a zero-to-one transition.
- **DRDY: Data Ready Status**
  This bit is cleared when the corresponding bit in ICR is written to one.
  This bit is set when a conversion has completed and new data is available in LCDR.

### 27.9.2 Status Register

**Name:** SR

**Access Type:** Read-only

**Offset:** 0x04

**Reset Value:** 0x00000000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| - | - | - | - | WFCS3 | WFCS2 | WFCS1 | WFCS0 |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| - | - | - | - | - | - | - | - |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ACRDY7 | ACCS7 | ACRDY6 | ACCS6 | ACRDY5 | ACCS5 | ACRDY4 | ACCS4 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ACRDY3 | ACCS3 | ACRDY2 | ACCS2 | ACRDY1 | ACCS1 | ACRDY0 | ACCS0 |

- **WFCSn: Window Mode Current Status**
  This bit is cleared when the common input voltage is outside the window.
  This bit is set when the common input voltage is inside the window.
- **ACRDYn: ACn Ready**
  This bit is cleared when the AC output (ACOUT) is not ready.
  This bit is set when the AC output (ACOUT) is ready, AC is enabled and its startup time is over.
- **ACCSn: ACn Current Comparison Status**
  This bit is cleared when $V_{INP}$ is currently lower than $V_{INN}$
  This bit is set when $V_{INP}$ is currently greater than $V_{INN}$.

### 28.7.8 Touch Group x Configuration Register 0

**Name:** TGxCFG0

**Access Type**: Read/Write

**Offset:** 0x20, 0x28

**Reset Value:** 0x00000000

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

- **DIV: Clock Divider**
  The prescaler is used to ensure that the CLK_CAT clock is divided to around 1 MHz to produce the sampling clock.The prescaler uses the following formula to generate the sampling clock:
  Sampling clock = CLK_CAT / (2(DIV+1))
- **CHLEN: Charge Length**
  For the QTouch method, specifies how many sample clock cycles should be used for transferring charge to the sense capacitor.
- **SELEN: Settle Length**
  For the QTouch method, specifies how many sample clock cycles should be used for settling after charge transfer.

### 28.7.24 CSB Resistor Control Register

**Name:** CSBRES

**Access Type:** Read/Write

**Offset:** 0x68

**Reset Value:** 0x00000000

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

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| - | | | | | | | RES[16] |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RES[15:8] | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RES[7:0] | | | | | | | |

- **RES: Resistive Drive Enable**
  When RES[n] is 0, CSB[n] has the same drive properties as normal I/O pads.
  When RES[n] is 1, CSB[n] has a nominal output resistance of 1kOhm during the burst phase.

## 31.3 On-Chip Debug

Rev: 2.1.2.0

### 31.3.1 Features

- **Debug interface in compliance with IEEE-ISTO 5001-2003 (Nexus 2.0) Class 2+**
- **JTAG or aWire access to all on-chip debug functions**
- **Advanced Program, Data, Ownership, and Watchpoint trace supported**
- **NanoTrace aWire- or JTAG-based trace access**
- **Auxiliary port for high-speed trace information**
- **Hardware support for 6 Program and 2 Data breakpoints**
- **Unlimited number of software breakpoints supported**
- **Automatic CRC check of memory regions**

### 31.3.2 Overview

Debugging on the AT32UC3L0128/256 is facilitated by a powerful On-Chip Debug (OCD) system. The user accesses this through an external debug tool which connects to the JTAG or aWire port and the Auxiliary (AUX) port if implemented. The AUX port is primarily used for trace functions, and an aWire- or JTAG-based debugger is sufficient for basic debugging.

The debug system is based on the Nexus 2.0 standard, class 2+, which includes:

- Basic run-time control
- Program breakpoints
- Data breakpoints
- Program trace
- Ownership trace
- Data trace

In addition to the mandatory Nexus debug features, the AT32UC3L0128/256 implements several useful OCD features, such as:

- Debug Communication Channel between CPU and debugger
- Run-time PC monitoring
- CRC checking
- NanoTrace
- Software Quality Assurance (SQA) support

The OCD features are controlled by OCD registers, which can be accessed by the debugger, for instance when the NEXUS_ACCESS JTAG instruction is loaded. The CPU can also access OCD registers directly using mtdr/mfdr instructions in any privileged mode. The OCD registers are implemented based on the recommendations in the Nexus 2.0 standard, and are detailed in the AVR32UC Technical Reference Manual.

### 31.3.3 I/O Lines Description

The OCD AUX trace port contains a number of pins, as shown in Table 31-6 on page 741. These are multiplexed with I/O Controller lines, and must explicitly be enabled by writing OCD registers before the debug session starts. The AUX port is mapped to two different locations,