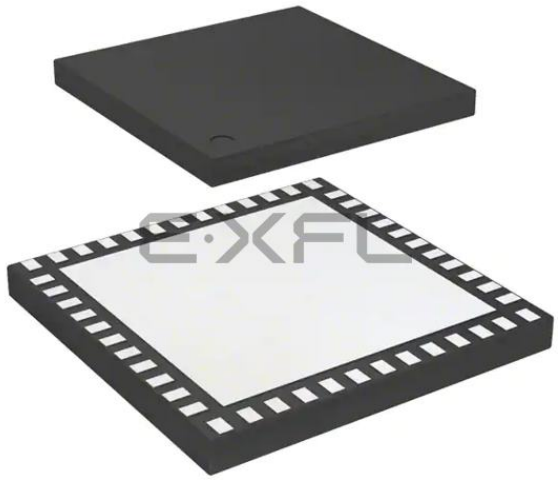


Welcome to E-XFL.COM

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	Obsolete
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	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16K 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-UFLGA Exposed Pad
Supplier Device Package	48-TLLGA (5.5x5.5)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at32uc3l032-d3ur

7.7.11 Mode Register

Name: MR

Access Type: Read/Write

Offset: 0x018 + n*0x040

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
-	-	-	-	RING	ETRIG	SIZE	

- **RING: Ring Buffer**
  0:The Ring buffer functionality is disabled.
  1:The Ring buffer functionality is enabled. When enabled, the reload registers, MARR and TCRR will not be cleared after reload.
- **ETRIG: Event Trigger**
  0:Start transfer when the peripheral selected in Peripheral Select Register (PSR) requests a transfer.
  1:Start transfer only when or after a peripheral event is received.
- **SIZE: Size of Transfer**

Table 7-5. Size of Transfer

SIZE	Size of Transfer
0	Byte
1	Halfword
2	Word
3	Reserved

8.8.3 Flash Status Register

Name: FSR

Access Type: Read-only

Offset: 0x08

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
LOCK15	LOCK14	LOCK13	LOCK12	LOCK11	LOCK10	LOCK9	LOCK8
23	22	21	20	19	18	17	16
LOCK7	LOCK6	LOCK5	LOCK4	LOCK3	LOCK2	LOCK1	LOCK0
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	HSMODE	QPRR	SECURITY	PROGE	LOCKE	-	FRDY

- **LOCKx: Lock Region x Lock Status**
  0: The corresponding lock region is not locked.
  1: The corresponding lock region is locked.
- **HSMODE: High-Speed Mode**
  0: High-speed mode disabled.
  1: High-speed mode enabled.
- **QPRR: Quick Page Read Result**
  0: The result is zero, i.e. the page is not erased.
  1: The result is one, i.e. the page is erased.
- **SECURITY: Security Bit Status**
  0: The security bit is inactive.
  1: The security bit is active.
- **PROGE: Programming Error Status**
  Automatically cleared when FSR is read.
  0: No invalid commands and no bad keywords were written in the Flash Command Register FCMD.
  1: An invalid command and/or a bad keyword was/were written in the Flash Command Register FCMD.
- **LOCKE: Lock Error Status**
  Automatically cleared when FSR is read.
  0: No programming of at least one locked lock region has happened since the last read of FSR.
  1: Programming of at least one locked lock region has happened since the last read of FSR.
- **FRDY: Flash Ready Status**
  0: The Flash Controller is busy and the application must wait before running a new command.
  1: The Flash Controller is ready to run a new command.

10.5.5 Special Function Registers

Name: SFR0...SFR15

Access Type: Read/Write

Offset: 0x110 - 0x14C

Reset Value: -

31	30	29	28	27	26	25	24
SFR							

23	22	21	20	19	18	17	16
SFR							

15	14	13	12	11	10	9	8
SFR							

7	6	5	4	3	2	1	0
SFR							

- **SFR: Special Function Register Fields**

  Those registers are not a HMATRIX specific register. The field of those will be defined where they are used.

12.7.7 Clock Failure Detector Control Register

Name: CFDCTRL

Access Type: Read/Write

Offset: 0x054

Reset Value: 0x00000000

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

- **SFV: Store Final Value**
  - 0: The register is read/write
  - 1: The register is read-only, to protect against further accidental writes.
- **CFDEN: Clock Failure Detection Enable**
  - 0: Clock Failure Detector is disabled
  - 1: Clock Failure Detector 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.

The peripheral event will be generated if the corresponding bit in the Event Mask (EVM) register is set. Bits in EVM register are set by writing a one to the corresponding bit in the Event Enable (EVE) register, and cleared by writing a one to the corresponding bit in the Event Disable (EVD) register.

14.5.5 AST wakeup

The AST can wake up the CPU directly, without the need to trigger an interrupt. A wakeup can be generated when the counter overflows, when the counter reaches the selected alarm value, or when the selected prescaler bit has a 0-to-1 transition. In this case, the CPU will continue executing from the instruction following the sleep instruction.

The AST wakeup is enabled by writing a one to the corresponding bit in the Wake Enable Register (WER). When the CPU wakes from sleep, the wake signal must be cleared by writing a one to the corresponding bit in SCR to clear the internal wake signal to the sleep controller. If the wake signal is not cleared after waking from sleep, the next sleep instruction will have no effect because the CPU will wake immediately after this sleep instruction.

The AST wakeup can wake the CPU from any sleep mode where the source clock is active. The AST wakeup can be configured independently of the interrupt masking.

14.5.6 Shutdown Mode

If the AST is configured to use a clock that is available in Shutdown mode, the AST can be used to wake up the system from shutdown. Both the alarm wakeup, periodic wakeup, and overflow wakeup mechanisms can be used in this mode.

When waking up from Shutdown mode all control registers will have the same value as before the shutdown was entered, except the Interrupt Mask Register (IMR). IMR will be reset with all interrupts turned off. The software must first reconfigure the interrupt controller and then enable the interrupts in the AST to again receive interrupts from the AST.

The CV register will be updated with the current counter value directly after wakeup from shutdown. The SR will show the status of the AST, including the status bits set during shutdown operation.

When waking up the system from shutdown the CPU will start executing code from the reset start address.

14.5.7 Digital Tuner

The digital tuner adds the possibility to compensate for a too-slow or a too-fast input clock. The ADD bit in the Digital Tuner Register (DTR.ADD) selects if the prescaler frequency should be reduced or increased. If ADD is ‘0’, the prescaler frequency is reduced:

$$f_{TUNED} = f_0\left(1 - \frac{1}{\text{roundup}\left(\frac{256}{VALUE}\right)\cdot(2^{EXP})+1}\right)$$

where f_{TUNED} is the tuned frequency, f_0 is the original prescaler frequency, and VALUE and EXP are the corresponding fields to be programmed in DTR. Note that DTR.EXP must be greater than zero. Frequency tuning is disabled by programming DTR.VALUE as zero.

18.7.8 Output Driver Enable Register

Name: ODER

Access: Read/Write, Set, Clear, Toggle

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

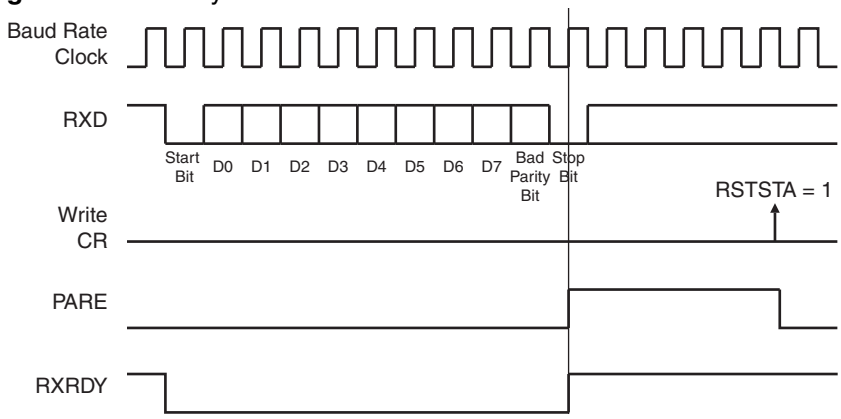
Reset Value: -

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.

The receiver will report parity errors in CSR.PARE, unless parity is disabled. Writing a one to CR.RSTSTA will clear PARE. See Figure 19-10

Figure 19-10. Parity Error



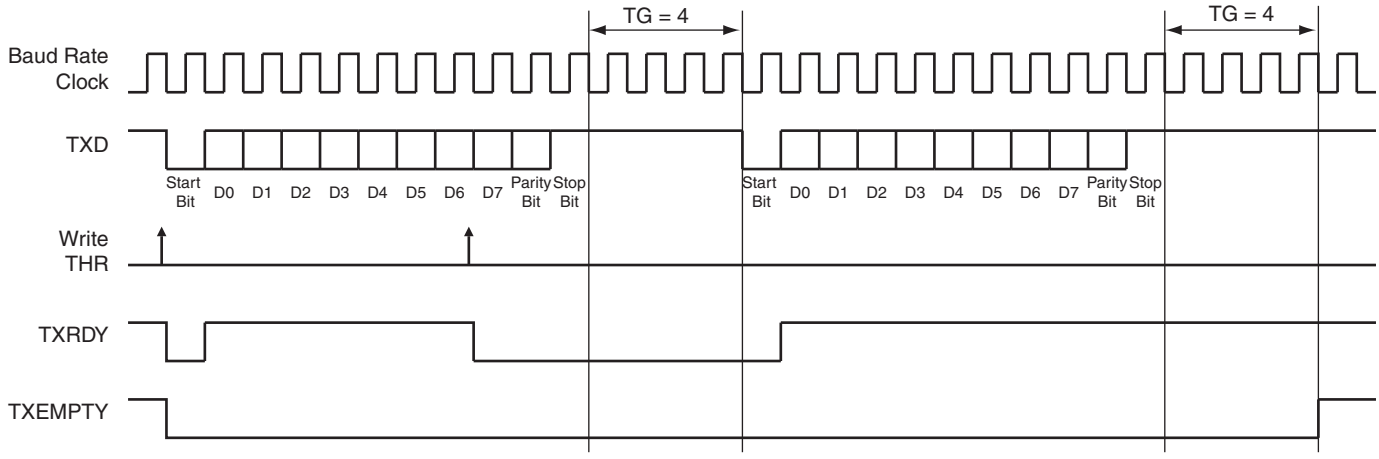
19.6.3.6 *Multidrop Mode*

If PAR is either 0x6 or 0x7, the USART runs in Multidrop mode. This mode differentiates data and address characters. Data has the parity bit zero and addresses have a one. By writing a one to the Send Address bit (CR.SENDA) the user will cause the next character written to THR to be transmitted as an address. Receiving a character with a one as parity bit will set PARE.

19.6.3.7 *Transmitter Timeguard*

The timeguard feature enables the USART to interface slow devices by inserting an idle state on the TXD line in between two characters. This idle state corresponds to a long stop bit, whose duration is selected by the Timeguard Value field in the Transmitter Timeguard Register (TTGR.TG). The transmitter will hold the TXD line high for TG bit periods, in addition to the number of stop bits. As illustrated in Figure 19-11, the behavior of TXRDY and TXEMPTY is modified when TG has a non-zero value. If a pending character has been written to THR, the TXRDY bit will not be set until this characters start bit has been sent. TXEMPTY will remain low until the timeguard transmission has completed.

Figure 19-11. Timeguard Operation



to an interrupt, and thus might lead to difficulties for interfacing with some serial peripherals requiring the chip select line to remain active during a full set of transfers.

To facilitate interfacing with such devices, the CSRn registers can be configured with the Chip Select Active After Transfer bit written to one (CSRn.CSAAT) . This allows the chip select lines to remain in their current state (low = active) until transfer to another peripheral is required.

When the CSRn.CSAAT bit is written to qero, the NPCS does not rise in all cases between two transfers on the same peripheral. During a transfer on a Chip Select, the SR.TDRE bit rises as soon as the content of the TDR is transferred into the internal shifter. When this bit is detected the TDR can be reloaded. If this reload occurs before the end of the current transfer and if the next transfer is performed on the same chip select as the current transfer, the Chip Select is not de-asserted between the two transfers. This might lead to difficulties for interfacing with some serial peripherals requiring the chip select to be de-asserted after each transfer. To facilitate interfacing with such devices, the CSRn registers can be configured with the Chip Select Not Active After Transfer bit (CSRn.CSNAAT) written to one. This allows to de-assert systematically the chip select lines during a time DLYBCS. (The value of the CSRn.CSNAAT bit is taken into account only if the CSRn.CSAAT bit is written to zero for the same Chip Select).

Figure 20-8 on page 424 shows different peripheral deselection cases and the effect of the CSRn.CSAAT and CSRn.CSNAAT bits.

20.7.3.8 FIFO management

A FIFO has been implemented in Reception FIFO (both in master and in slave mode), in order to be able to store up to 4 characters without causing an overrun error. If an attempt is made to store a fifth character, an overrun error rises. If such an event occurs, the FIFO must be flushed. There are two ways to Flush the FIFO:

- By performing four read accesses of the RDR (the data read must be ignored)
- By writing a one to the Flush Fifo Command bit in the CR register (CR.FLUSHFIFO).

After that, the SPI is able to receive new data.

20.8 User Interface

Table 20-3. SPI Register Memory Map

Offset	Register	Register Name	Access	Reset
0x00	Control Register	CR	Write-only	0x00000000
0x04	Mode Register	MR	Read/Write	0x00000000
0x08	Receive Data Register	RDR	Read-only	0x00000000
0x0C	Transmit Data Register	TDR	Write-only	0x00000000
0x10	Status Register	SR	Read-only	0x00000000
0x14	Interrupt Enable Register	IER	Write-only	0x00000000
0x18	Interrupt Disable Register	IDR	Write-only	0x00000000
0x1C	Interrupt Mask Register	IMR	Read-only	0x00000000
0x30	Chip Select Register 0	CSR0	Read/Write	0x00000000
0x34	Chip Select Register 1	CSR1	Read/Write	0x00000000
0x38	Chip Select Register 2	CSR2	Read/Write	0x00000000
0x3C	Chip Select Register 3	CSR3	Read/Write	0x00000000
0x E4	Write Protection Control Register	WPCR	Read/Write	0X00000000
0xE8	Write Protection Status Register	WPSR	Read-only	0x00000000
0xF8	Features Register	FEATURES	Read-only	- (1)
0xFC	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.

- **BITS: Bits Per Transfer**
  The BITS field determines the number of data bits transferred. Reserved values should not be used.

BITS	Bits Per Transfer
0000	8
0001	9
0010	10
0011	11
0100	12
0101	13
0110	14
0111	15
1000	16
1001	4
1010	5
1011	6
1100	7
1101	Reserved
1110	Reserved
1111	Reserved

- **CSAAT: Chip Select Active After Transfer**
  1: The Peripheral Chip Select does not rise after the last transfer is achieved. It remains active until a new transfer is requested on a different chip select.
  0: The Peripheral Chip Select Line rises as soon as the last transfer is achieved.
- **CSNAAT: Chip Select Not Active After Transfer (Ignored if CSAAT = 1)**
  0: The Peripheral Chip Select does not rise between two transfers if the TDR is reloaded before the end of the first transfer and if the two transfers occur on the same Chip Select.
  1: The Peripheral Chip Select rises systematically between each transfer performed on the same slave for a minimal duration of:

  $\frac{DLYBCS}{CLKSPI}$ (if DLYBCT field is different from 0)

  $\frac{DLYBCS + 1}{CLK\overline{SPI}}$ (if DLYBCT field equals 0)
- **NCPHA: Clock Phase**
  1: Data is captured after the leading (inactive-to-active) edge of SPCK and changed on the trailing (active-to-inactive) edge of SPCK.
  0: Data is changed on the leading (inactive-to-active) edge of SPCK and captured after the trailing (active-to-inactive) edge of SPCK.
  NCPHA determines which edge of SPCK causes data to change and which edge causes data to be captured. NCPHA is used with CPOL to produce the required clock/data relationship between master and slave devices.
- **CPOL: Clock Polarity**
  1: The inactive state value of SPCK is logic level one.
  0: The inactive state value of SPCK is logic level zero.

20.8.13 Write Protection Control Register

Register Name: WPCR

Access Type: Read-write

Offset: 0xE4

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
SPIWPKEY[23:16]							

23	22	21	20	19	18	17	16
SPIWPKEY[15:8]							

15	14	13	12	11	10	9	8
SPIWPKEY[7:0]							

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	SPIWPEN

- **SPIWPKEY: SPI Write Protection Key Password**

  If a value is written in SPIWPEN, the value is taken into account only if SPIWPKEY is written with “SPI” (SPI written in ASCII Code, i.e. 0x535049 in hexadecimal).

- **SPIWPEN: SPI Write Protection Enable**

  1: The Write Protection is Enabled

  0: The Write Protection is Disabled

Writing a one to this bit resets the TWIS.

- **STREN: Clock Stretch Enable**
  0: Disables clock stretching if RHR/THR buffer full/empty. May cause over/underrun.
  1: Enables clock stretching if RHR/THR buffer full/empty.
- **GCMATCH: General Call Address Match**
  0: Causes the TWIS not to acknowledge the General Call Address.
  1: Causes the TWIS to acknowledge the General Call Address.
- **SMATCH: Slave Address Match**
  0: Causes the TWIS not to acknowledge the Slave Address.
  1: Causes the TWIS to acknowledge the Slave Address.
- **SMEN: SMBus Mode Enable**
  0: Disables SMBus mode.
  1: Enables SMBus mode.
- **SEN: Slave Enable**
  0: Disables the slave interface.
  1: Enables the slave interface.

23.7.11 Version Register

Name: VERSION

Access Type: Read-only

Offset: 0x28

Reset Value: -

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

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

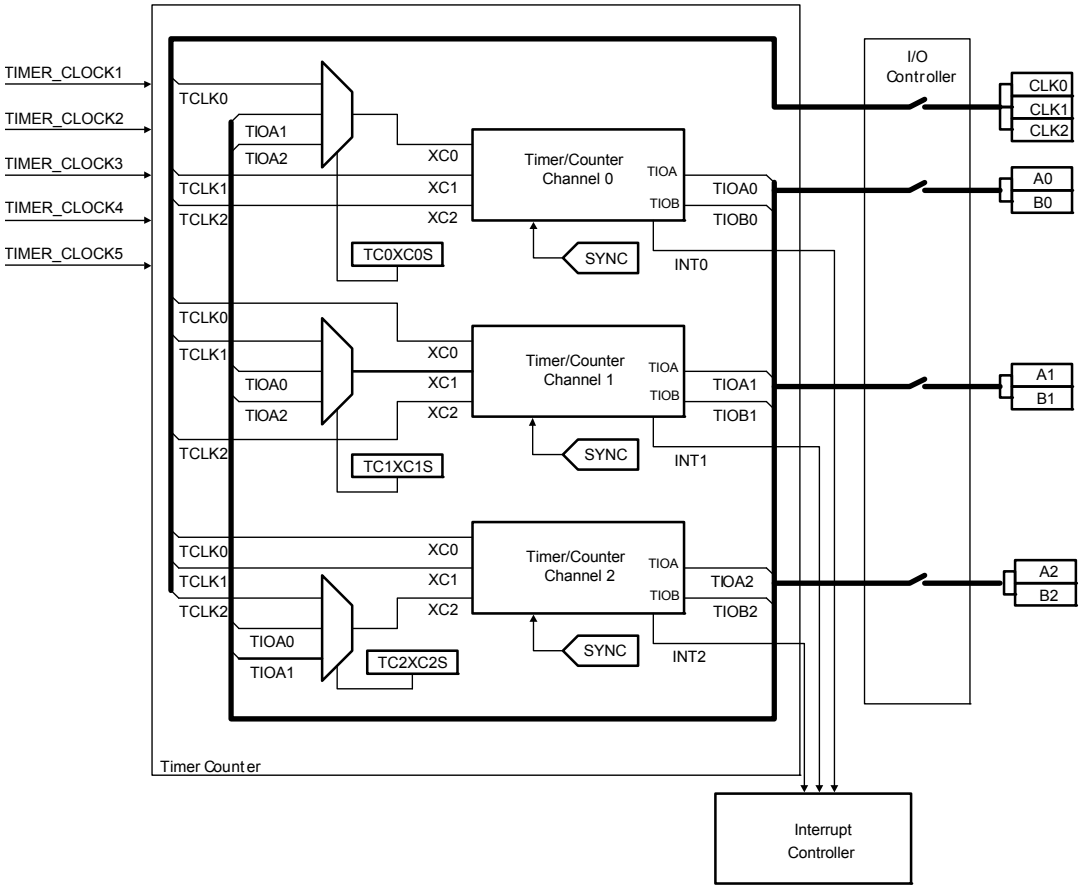
15	14	13	12	11	10	9	8
-	-	-	-	VERSION[11:8]			

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.

24.3 Block Diagram

Figure 24-1. TC Block Diagram



24.4 I/O Lines Description

Table 24-1. I/O Lines Description

Pin Name	Description	Type
CLK0-CLK2	External Clock Input	Input
A0-A2	I/O Line A	Input/Output
B0-B2	I/O Line B	Input/Output

24.5 Product Dependencies

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

24.5.1 I/O Lines

The pins used for interfacing the compliant external devices may be multiplexed with I/O lines. The user must first program the I/O Controller to assign the TC pins to their peripheral functions.

The current value of the counter is accessible in real time by reading the Channel n Counter Value Register (CVn). The counter can be reset by a trigger. In this case, the counter value passes to 0x0000 on the next valid edge of the selected clock.

24.6.1.3 *Clock selection*

At block level, input clock signals of each channel can either be connected to the external inputs TCLK0, TCLK1 or TCLK2, or be connected to the configurable I/O signals A0, A1 or A2 for chaining by writing to the BMR register. See Figure 24-2 on page 544.

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

- Internal clock signals: TIMER_CLOCK1, TIMER_CLOCK2, TIMER_CLOCK3, TIMER_CLOCK4, TIMER_CLOCK5. See the Module Configuration Chapter for details about the connection of these clock sources.
- External clock signals: XC0, XC1 or XC2. See the Module Configuration Chapter for details about the connection of these clock sources.

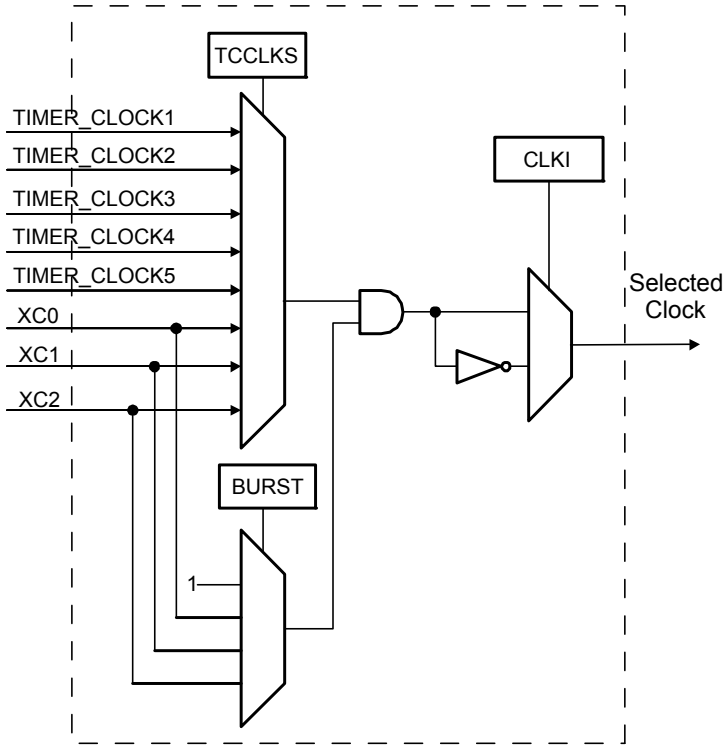
This selection is made by the Clock Selection field in the Channel n Mode Register (CMRn.TCCLKS).

The selected clock can be inverted with the Clock Invert bit in CMRn (CMRn.CLKI). This allows counting on the opposite edges of the clock.

The burst function allows the clock to be validated when an external signal is high. The Burst Signal Selection field in the CMRn register (CMRn.BURST) defines this signal.

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

Figure 24-2. Clock Selection



27.6.5.3 Window Mode Peripheral Events

When operating in window mode, each channel can generate the same peripheral events as in normal mode, see Section 27.6.4.3.

Additionally, when channels operate in window mode, programming Window Mode Event Selection Source (CONFWn.WEVSRC) can cause peripheral events to be generated when:

- As soon as the common input voltage is inside the window.
- As soon as the common input voltage is outside the window.
- On toggle of the window compare output (ACWOUT)
- Whenever a comparison is ready and the common input voltage is inside the window.
- Whenever a comparison is ready and the common input voltage is outside the window.
- When the comparison in both channels in the window pair is ready.

27.6.6 Filtering

The output of the comparator can be filtered to reduce noise. The filter length is determined by the Filter Length field in the CONFn register (CONFn.FLEN). The filter samples the Analog Comparator output at the GCLK frequency for $2^{CONFn.FLEN}$ samples. A separate counter (CNT) counts the number of cycles the AC output was one. This filter is deactivated if CONFn.FLEN equals 0.

If the filter is enabled, the Hysteresis Value field HYS in the CONFn register (CONFn.HYS) can be used to define a hysteresis value. The hysteresis value should be chosen so that:

$$\frac{2^{FLEN}}{2} > HYS$$

The filter function is defined by:

$$CNT \geq \left(\frac{2^{FLEN}}{2} + HYS\right) \Rightarrow comp = 1$$

$$\left(\frac{2^{FLEN}}{2} + HYS\right) > CNT \geq \left(\frac{2^{FLEN}}{2} - HYS\right) \Rightarrow comp \text{ unchanged}$$

$$CNT < \left(\frac{2^{FLEN}}{2} - HYS\right) \Rightarrow comp = 0$$

The filtering algorithm is explained in Figure 27-3. 2^{FLEN} measurements are sampled. If the number of measurements that are zero is less than ($2^{FLEN}/2$ - HYS), the filtered result is zero. If the number of measurements that are one is more than ($2^{FLEN}/2$ + HYS), the filtered result is one. Otherwise, the result is unchanged.

27.9.11 Window Configuration Register

Name: CONFWn

Access Type: Read/Write

Offset: 0x80,0x84,0x88,0x8C

Reset Value: 0x00000000

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

- **WFEN: Window Mode Enable**
  - 0: The window mode is disabled.
  - 1: The window mode is enabled.
- **WEVEN: Window Event Enable**
  - 0: Event from awout is disabled.
  - 1: Event from awout is enabled.
- **WEVSRC: Event Source Selection for Window Mode**
  - 000: Event on acwout rising edge.
  - 001: Event on acwout falling edge.
  - 010: Event on awout rising or falling edge.
  - 011: Inside window.
  - 100: Outside window.
  - 101: Measure done.
  - 110-111: Reserved.
- **WIS: Window Mode Interrupt Settings**
  - 00: Window interrupt as soon as the input voltage is inside the window.
  - 01: Window interrupt as soon as the input voltage is outside the window.
  - 10: Window interrupt on toggle of window compare output.
  - 11: Window interrupt when evaluation of input voltage is done.

29.6.2 Configuring the Lookup Table

The lookup table in each LUT unit can generate any logic expression OUT as a function of up to four inputs, IN[3:0]. The truth table for the expression is written to the TRUTH register for the LUT. Table 29-2 shows the truth table for LUT0. The truth table for LUTn is written to TRUTHn, and the corresponding input and outputs will be IN[4n] to IN[4n+3] and OUTn.

Table 29-2. Truth Table for the Lookup Table in LUT0

IN[3]	IN[2]	IN[1]	IN[0]	OUT[0]
0	0	0	0	TRUTH0[0]
0	0	0	1	TRUTH0[1]
0	0	1	0	TRUTH0[2]
0	0	1	1	TRUTH0[3]
0	1	0	0	TRUTH0[4]
0	1	0	1	TRUTH0[5]
0	1	1	0	TRUTH0[6]
0	1	1	1	TRUTH0[7]
1	0	0	0	TRUTH0[8]
1	0	0	1	TRUTH0[9]
1	0	1	0	TRUTH0[10]
1	0	1	1	TRUTH0[11]
1	1	0	0	TRUTH0[12]
1	1	0	1	TRUTH0[13]
1	1	1	0	TRUTH0[14]
1	1	1	1	TRUTH0[15]

29.6.3 Output Filter

By default, the output OUTn is a combinatorial function of the inputs IN[4n] to IN[4n+3]. This may cause some short glitches to occur when the inputs change value.

It is also possible to clock the output through a filter to remove glitches. This requires that the corresponding generic clock (GCLK) has been enabled before use. The filter can then be enabled by writing a one to the Filter Enable (FILTEN) bit in CRn. The OUTn output will be delayed by three to four GCLK cycles when the filter is enabled.

29.7.2 Truth Table Register n

Name: TRUTHn

Access Type: Read/Write

Offset: 0x04+n*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
TRUTH[15:8]							

7	6	5	4	3	2	1	0
TRUTH[7:0]							

- **TRUTH: Truth Table Value**

  This value defines the output OUT as a function of inputs IN:
  OUT = TRUTH[IN]

Avoiding drive contention when changing direction

The aWire debug protocol uses one dataline in both directions. To avoid both the master and the slave to drive this line when changing direction the AW has a built in guard time before it starts to drive the line. At reset this guard time is set to maximum (128 bit cycles), but can be lowered by the master upon command.

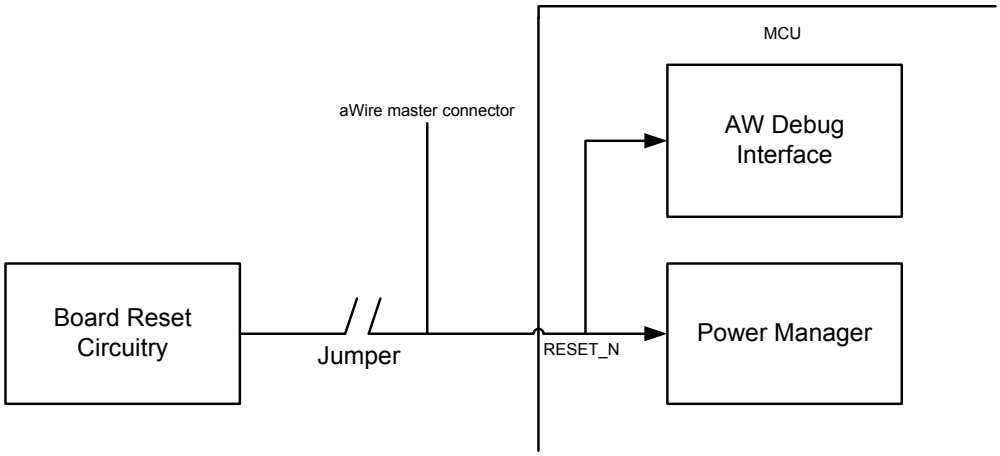
The AW will release the line immediately after the stop character has been transmitted.

During the direction change there can be a period when the line is not driven. An external pullup has to be added to RESET_N to keep the signal stable when neither master or slave is actively driving the line.

31.6.6.2 The RESET_N pin

Normal reset functionality on the RESET_N pin is disabled when using aWire. However, the user can reset the system through the RESET aWire command. During aWire operation the RESET_N pin should not be connected to an external reset circuitry, but disconnected via a switch or a jumper to avoid drive contention and speed problems.

Figure 31-11. Reset Circuitry and aWire.



31.6.6.3 Initializing the AW

To enable AW, the user has to send a 0x55 pattern with a baudrate of 1 kHz on the RESET_N pin. The AW is enabled after transmitting this pattern and the user can start transmitting commands. This pattern is not the sync pattern for the first command.

After enabling the aWire debug interface the halt bit is set automatically to prevent the system from running code after the interface is enabled. To make the CPU run again set halt to zero using the HALT command.

31.6.6.4 Disabling the AW

To disable AW, the user can keep the RESET_N pin low for 100 ms. This will disable the AW, return RESET_N to its normal function, and reset the device.

An aWire master can also disable aWire by sending the DISABLE command. After acking the command the AW will be disabled and RESET_N returns to its normal function.