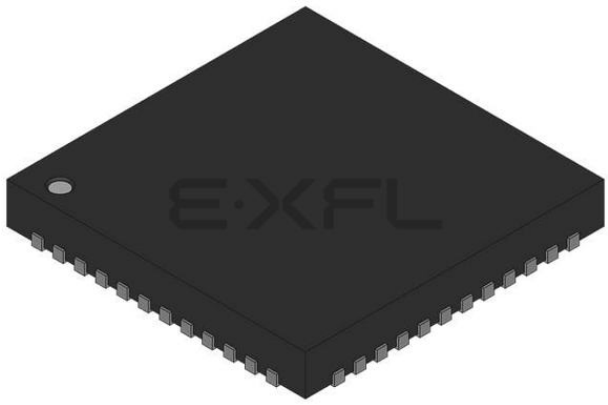
Atmel - ATUC64L4U-ZAUT Datasheet



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"

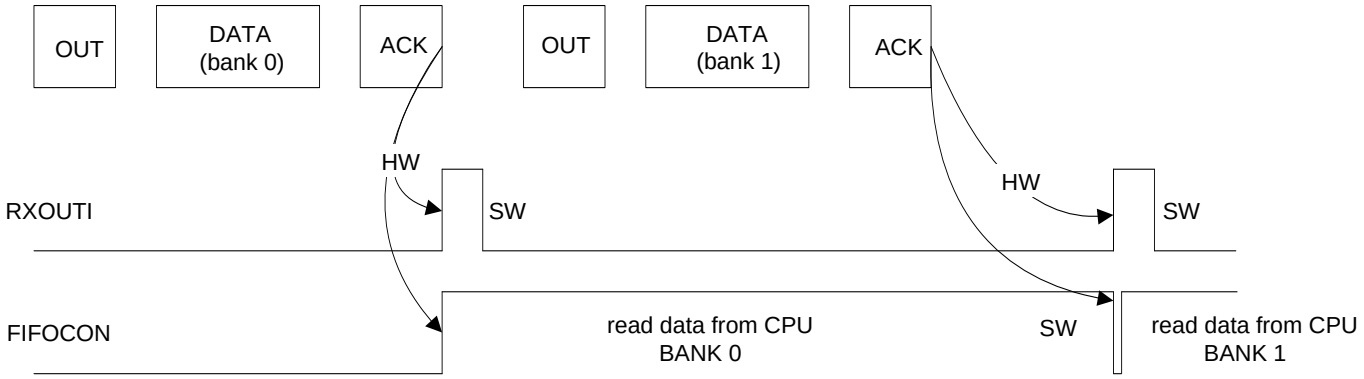
Details	
Product Status	Active
Core Processor	AVR
Core Size	32-Bit Single-Core
Speed	50MHz
Connectivity	I²C, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	36
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16K x 8
Voltage - 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-VFQFN Exposed Pad
Supplier Device Package	48-QFN (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/atmel/atuc64l4u-zaut

6.1.3 Regulator Connection

The ATUC64/128/256L3/4U supports three power supply configurations:

- 3.3V single supply mode
  - Shutdown mode is not available
- 1.8V single supply mode
  - Shutdown mode is not available
- 3.3V supply mode, with 1.8V regulated I/O lines
  - Shutdown mode is available

Figure 8-12. Example of an OUT endpoint with two data banks



• *Detailed description*

Before using the OUT endpoint, one should properly initialize its descriptor for each bank. See Figure 8-5 on page 91.

The data is read, according to this sequence:

- When the bank is full, RXOUTI and FIFOCON are set, which triggers an EPnINT interrupt if RXOUTE is one.
- The user acknowledges the interrupt by writing a one to RXOUTIC in order to clear RXOUTI.
- The user reads the UESTAX.CURRBK field to know the current bank number.
- The user reads the byte count of the current bank from the descriptor in RAM (EPn_PCKSIZE_BK0/1.BYTE_COUNT) to know how many bytes to read.
- The user reads the data in the current bank, located in RAM as described by its descriptor: EPn_ADDR_BK0/1.
- The user frees the bank and switches to the next bank (if any) by clearing FIFOCON.

If the endpoint uses several banks, the current one can be read while the next is being written by the host. When the user clears FIFOCON, the following bank may already be ready and RXOUTI will be immediately set.

• *Multi packet mode for OUT endpoints*

In multi packet mode, the user can extend the size of the bank allowing the storage of n USB packets in the bank.

To enable the multi packet mode, the user should configure the endpoint descriptor (EPn_PCKSIZE_BK0/1.MULTI_PACKET_SIZE) to match the size of the multi packet.This value should be a multiple of the endpoint size (UECFGn.EPSIZE).

Since the EPn_PCKSIZE_BK0/1.BYTE_COUNT is incremented (by the received packet size) after each successful transaction, it should be set to zero when setting up a new multi packet transfer.

As for single packet mode, the number of received data bytes is stored in the BYTE_CNT field.

The bank is considered as “valid” and the RX_OUT flag is set when:

8.7.1.3 General Status Clear Register

Register Name: USBSTACLR

Access Type: Write-Only

Offset: 0x0808

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
-	-	-	-	-	-	-	-

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

- **SUSP: Suspend Interrupt**

  This bit is cleared when the UDINTCLR.SUSPC bit is written to one to acknowledge the interrupt or when the Wakeup (WAKEUP) interrupt bit is set.

  This bit is set when a USB “Suspend” idle bus state has been detected for 3 frame periods (J state for 3 ms). This triggers a USB interrupt if SUSPE is one.

In ATUC64/128/256L3/4U, there are 10 generic clocks. These are allocated to different functions as shown in Table 14-10.

Table 14-10. Generic Clock Allocation

Clock number	Function
0	DFLLIF main reference and GCLK0 pin (CLK_DFLLIF_REF)
1	DFLLIF dithering and SSG reference and GCLK1 pin (CLK_DFLLIF_DITHER)
2	AST and GCLK2 pin
3	PWMA and GCLK3 pin
4	CAT, ACIFB and GCLK4 pin
5	GLOC and GCLK5 pin
6	ABDACB, IISC, and GCLK6 pin
7	USBC and GCLK7 pin
8	PLL0 source clock and GCLK8 pin
9	Master generic clock and GCLK9 pin. Can be used as source for other generic clocks.

Table 14-11. Generic Clock Sources

OSCSEL	Clock/Oscillator	Description
0	RCSYS	System RC oscillator clock
1	OSC32K	Output clock from OSC32K
2	DFLL0	Output clock from DFLL0
3	OSC0	Output clock from Oscillator0
4	RC120M	Output from 120MHz RCOSC
5	CLK_CPU	The clock the CPU runs on
6	CLK_HSB	High Speed Bus clock
7	CLK_PBA	Peripheral Bus A clock
8	CLK_PBB	Peripheral Bus B clock
9	RC32K	Output from 32KHz RCOSC
10	Reserved	
11	CLK_1K	1KHz output clock from OSC32K
12	PLL0	Output clock from PLL0
13-14	Reserved	
15-17	GCLK_IN[0-2]	GCLK_IN[0-2] pins, digital clock input
18	GCLK9	Generic clock 9. Can not be used as an input to itself
19-31	Reserved	

17.7.9 Filter Register

Name: FILTER

Access Type: Read/Write

Offset: 0x020

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
-	INT30	INT29	INT28	INT27	INT26	INT25	INT24
23	22	21	20	19	18	17	16
INT23	INT22	INT21	INT20	INT19	INT18	INT17	INT16
15	14	13	12	11	10	9	8
INT15	INT14	INT13	INT12	INT11	INT10	INT9	INT8
7	6	5	4	3	2	1	0
INT7	INT6	INT5	INT4	INT3	INT2	INT1	NMI

- **INTn: External Interrupt n**
  0: The external interrupt is not filtered.
  1: The external interrupt is filtered.
  Please refer to the Module Configuration section for the number of external interrupts.
- **NMI: Non-Maskable Interrupt**
  0: The Non-Maskable Interrupt is not filtered.
  1: The Non-Maskable Interrupt is filtered.

18.6.4 Value Register

Name: VALUE

Access Type: Read-only

Offset: 0x00C

Reset Value: 0x00000000

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

23	22	21	20	19	18	17	16
VALUE[23:16]							

15	14	13	12	11	10	9	8
VALUE[15:8]							

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

- **VALUE:**
  Result from measurement.

Table 20-5. Maximum Baud Rate Dependent Timeguard Durations

Baud Rate (bit/sec)	Bit time (µs)	Timeguard (ms)
1 200	833	212.50
9 600	104	26.56
14400	69.4	17.71
19200	52.1	13.28
28800	34.7	8.85
33400	29.9	7.63
56000	17.9	4.55
57600	17.4	4.43
115200	8.7	2.21

20.6.3.8 Receiver Time-out

The Time-out Value field in the Receiver Time-out Register (RTOR.TO) enables handling of variable-length frames by detection of selectable idle durations on the RXD line. The value written to TO is loaded to a decremental counter, and unless it is zero, a time-out will occur when the amount of inactive bit periods match the initial counter value. If a time-out has not occurred, the counter will reload and restart every time a new character arrives. A time-out sets the TIMEOUT bit in CSR. Clearing TIMEOUT can be done in two ways:

- Writing a one to the Start Time-out bit (CR.STTTO). This also aborts count down until the next character has been received.
- Writing a one to the Reload and Start Time-out bit (CR.RETTO). This also reloads the counter and restarts count down immediately.

Figure 20-12. Receiver Time-out Block Diagram

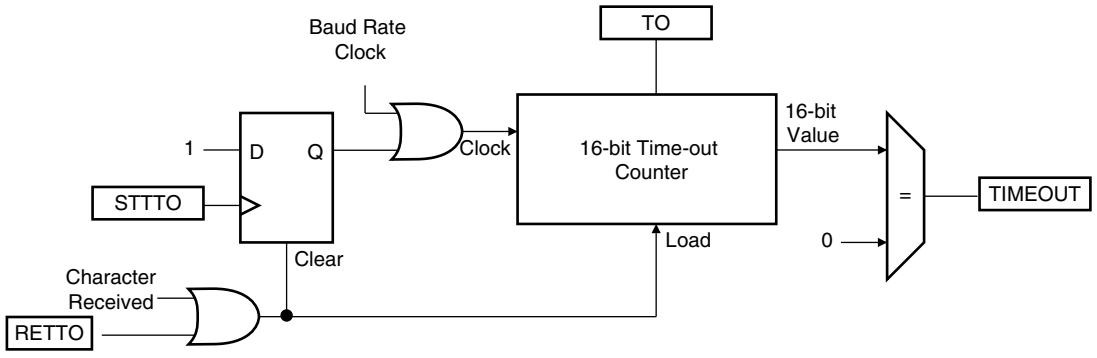


Table 20-6. Maximum Time-out Period

Baud Rate (bit/sec)	Bit Time (µs)	Time-out (ms)
600	1 667	109 225
1 200	833	54 613
2 400	417	27 306
4 800	208	13 653

- DLM=1: the response data length is defined by the Identifier bits according to the table below.

Table 20-8. Response Data Length if DLM = 1

IDCHR[5]	IDCHR[4]	Response Data Length [bytes]
0	0	2
0	1	2
1	0	4
1	1	8

20.6.5.11 Checksum

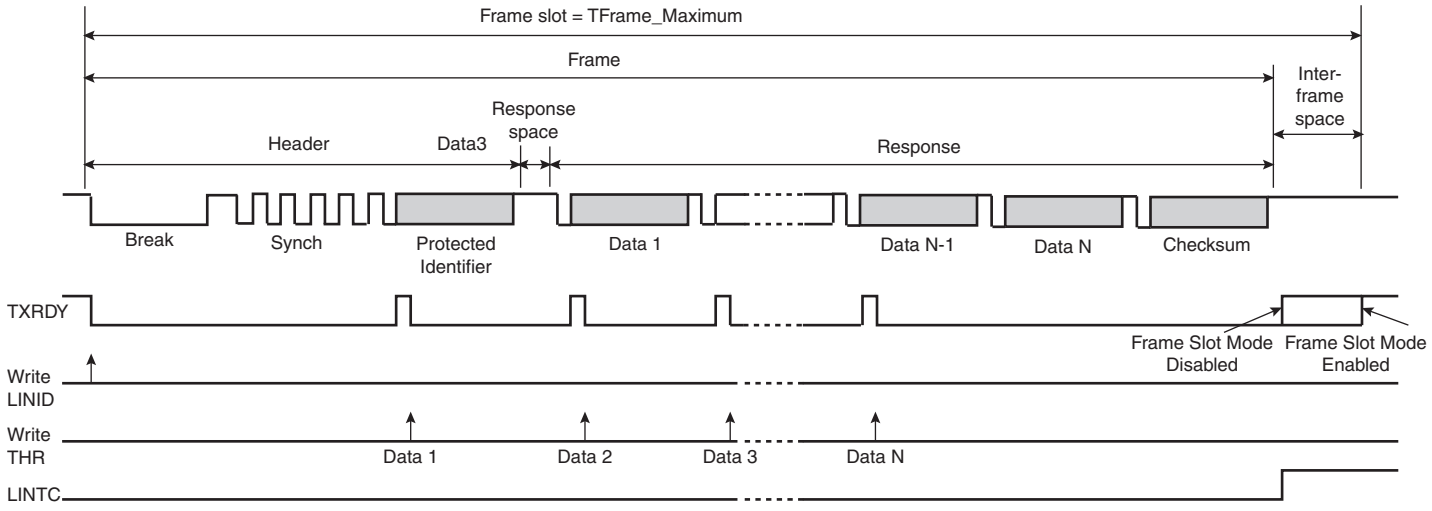
The last frame field is the checksum. It is configured by the Checksum Type (LINMR.CHKTYP), and the Checksum Disable (LINMR.CHKDIS) bits. TXRDY will not be set after the last THR data write if enabled. Writing a one to CHKDIS will disable the automatic checksum generation/checking, and the user may send/check this last byte manually, disguised as a normal data. The checksum is an inverted 8-bit sum with carry, either:

- over all data bytes, called a classic checksum. This is used for LIN 1.3 compliant slaves, and automatically managed when CHKDIS=0, and CHKTYP=1.
- over all data bytes and the protected identifier, called an enhanced checksum. This is used for LIN 2.0 compliant slaves, and automatically managed when CHKDIS=0, and CHKTYP=0.

20.6.5.12 Frame Slot Mode

A LIN master can be configured to use frame slots with a pre-defined minimum length. Writing a one to the Frame Slot Mode Disable bit (LINMR.FSDIS) disables this mode. This mode will not allow TXRDY to be set after a frame transfer until the entire frame slot duration has elapsed, in effect preventing the master from sending a new header. The LIN Transfer Complete bit (CSR.LINTC) will still be set after the checksum has been sent. Writing a one to CR.RSTST clears LINTC.

Figure 20-26. Frame Slot Mode with Automatic Checksum



The minimum frame slot size is determined by TFrame_Maximum, and calculated below (all values in bit periods):

- THeader_Nominal = 34

21.8.8 Interrupt Mask Register

Name: IMR

Access Type: Read-only

Offset: 0x1C

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
-	-	-	-	-	UNDES	TXEMPTY	NSSR
7	6	5	4	3	2	1	0
-	-	-	-	OVRES	MODF	TDRE	RDRF

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.

21.8.12 Chip Select Register 3

Name: CSR3

Access Type: Read/Write

Offset: 0x3C

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
DLYBCT							

23	22	21	20	19	18	17	16
DLYBS							

15	14	13	12	11	10	9	8
SCBR							

7	6	5	4	3	2	1	0
BITS				CSAAT	CSNAAT	NCPHA	CPOL

- **DLYBCT: Delay Between Consecutive Transfers**

  This field defines the delay between two consecutive transfers with the same peripheral without removing the chip select. The delay is always inserted after each transfer and before removing the chip select if needed.

  When DLYBCT equals zero, no delay between consecutive transfers is inserted and the clock keeps its duty cycle over the character transfers.

  Otherwise, the following equation determines the delay:

$$\text{Delay Between Consecutive Transfers} = \frac{32 \times DLYBCT}{CLKSPI}$$

- **DLYBS: Delay Before SPCK**

  This field defines the delay from NPCS valid to the first valid SPCK transition.

  When DLYBS equals zero, the NPCS valid to SPCK transition is 1/2 the SPCK clock period.

  Otherwise, the following equations determine the delay:

$$\text{Delay Before SPCK} = \frac{DLYBS}{CLKSPI}$$

- **SCBR: Serial Clock Baud Rate**

  In Master Mode, the SPI Interface uses a modulus counter to derive the SPCK baud rate from the CLK_SPI. The Baud rate is selected by writing a value from 1 to 255 in the SCBR field. The following equations determine the SPCK baud rate:

$$\text{SPCK Baudrate} = \frac{CLKSPI}{SCBR}$$

  Writing the SCBR field to zero is forbidden. Triggering a transfer while SCBR is zero can lead to unpredictable results.

  At reset, SCBR is zero and the user has to write it to a valid value before performing the first transfer.

  If a clock divider (SCBRn) field is set to one and the other SCBR fields differ from one, access on CSn is correct but no correct access will be possible on other CS.

A trigger such as an external event or a software trigger can modify CVn at any time. If a trigger occurs while CVn is incrementing, CVn then decrements. If a trigger is received while CVn is decrementing, CVn then increments. See Figure 26-11 on page 655.

RC Compare cannot be programmed to generate a trigger in this configuration.

At the same time, RC Compare can stop the counter clock (CMRn.CPCSTOP = 1) and/or disable the counter clock (CMRn.CPCDIS = 1).

Figure 26-10. WAVSEL = 1 Without Trigger

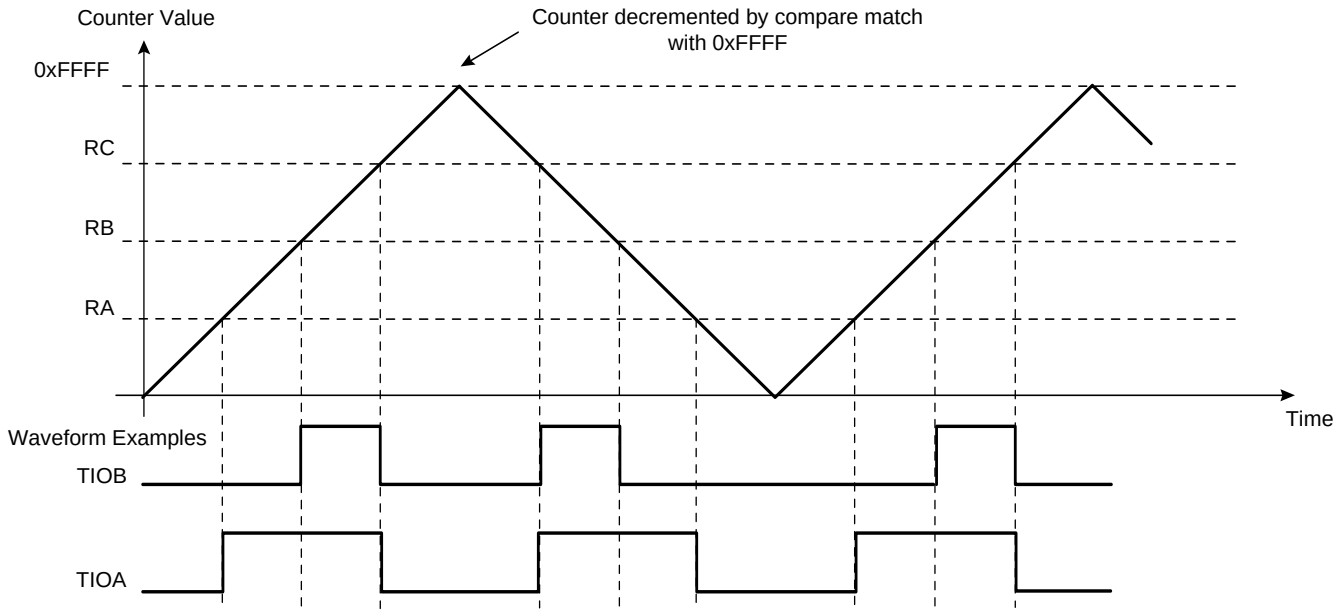
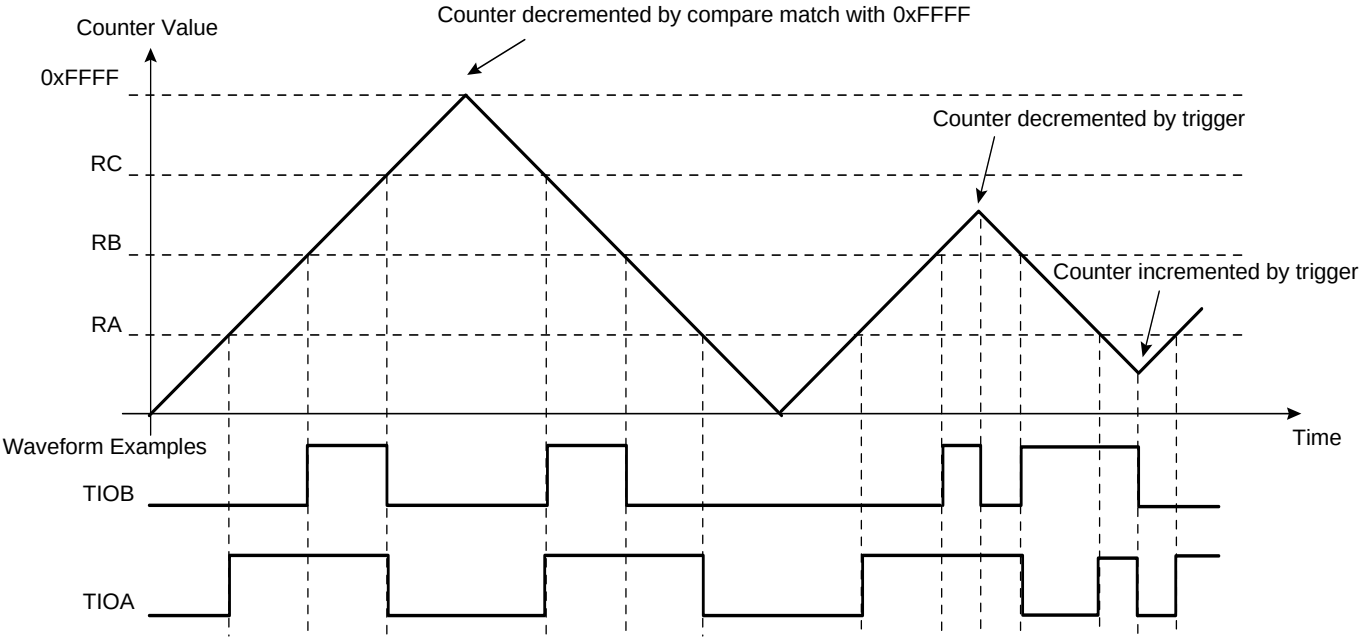


Figure 26-11. WAVSEL = 1 With Trigger



26.7.2 Channel Mode Register: Capture Mode

Name: CMR

Access Type: Read/Write

Offset: 0x04 + n * 0x40

Reset Value: 0x00000000

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

23	22	21	20	19	18	17	16
-	-	-	-	LDRB		LDRA	

15	14	13	12	11	10	9	8
WAVE	CPCTRG	-	-	-	ABETRG	ETRGEDG	

7	6	5	4	3	2	1	0
LDBDIS	LDBSTOP	BURST		CLKI	TCCLKS		

- **LDRB: RB Loading Selection**

LDRB	Edge
0	none
1	rising edge of TIOA
2	falling edge of TIOA
3	each edge of TIOA

- **LDRA: RA Loading Selection**

LDRA	Edge
0	none
1	rising edge of TIOA
2	falling edge of TIOA
3	each edge of TIOA

- **WAVE**
  1: Capture mode is disabled (Waveform mode is enabled).
  0: Capture mode is enabled.
- **CPCTRG: RC Compare Trigger Enable**
  1: RC Compare resets the counter and starts the counter clock.
  0: RC Compare has no effect on the counter and its clock.
- **ABETRG: TIOA or TIOB External Trigger Selection**
  1: TIOA is used as an external trigger.

28.7.8 Interrupt Mask Register

Name: IMR

Access Type: Read-only

Offset: 0x1C

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
-	-	-	-	-	TXUR	TXRDY	-

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.

clear an active interrupt request, write a one to the corresponding bit in the Interrupt Clear Register (ICR).

The source for the interrupt requests are the status bits in the Status Register (SR). The SR shows the ADCIFB status at the time the register is read. The Interrupt Status Register (ISR) shows the status since the last write to the Interrupt Clear Register. The combination of ISR and SR allows the user to react to status change conditions but also allows the user to read the current status at any time.

29.6.12 Peripheral Events

The Peripheral Event System can be used together with the ADCIFB to allow any peripheral event generator to be used as a trigger source. To enable peripheral events to trigger a conversion sequence the user must write the Peripheral Event Trigger value (0x7) to the Trigger Mode (TRGMOD) field in the Trigger Register (TRGR). Refer to Table 29-4 on page 730. The user must also configure a peripheral event generator to emit peripheral events for the ADCIFB to trigger on. Refer to the Peripheral Event System chapter for details.

29.6.13 Sleep Mode

Before entering sleep modes the user must make sure the ADCIFB is idle and that the Analog-to-Digital Converter cell is inactive. To deactivate the Analog-to-Digital Converter cell the SLEEP bit in the ADC Configuration Register (ACR) must be written to one and the ADCIFB must be idle. To make sure the ADCIFB is idle, write a zero the Trigger Mode (TRGMOD) field in the Trigger Register (TRGR) and wait for the READY bit in the Status Register (SR) to be set.

Note that by deactivating the Analog-to-Digital Converter cell, a startup time penalty as defined in the STARTUP field in the ADC Configuration Register (ACR) will apply on the next conversion.

29.6.14 Conversion Performances

For performance and electrical characteristics of the ADCIFB, refer to the Electrical Characteristics chapter.

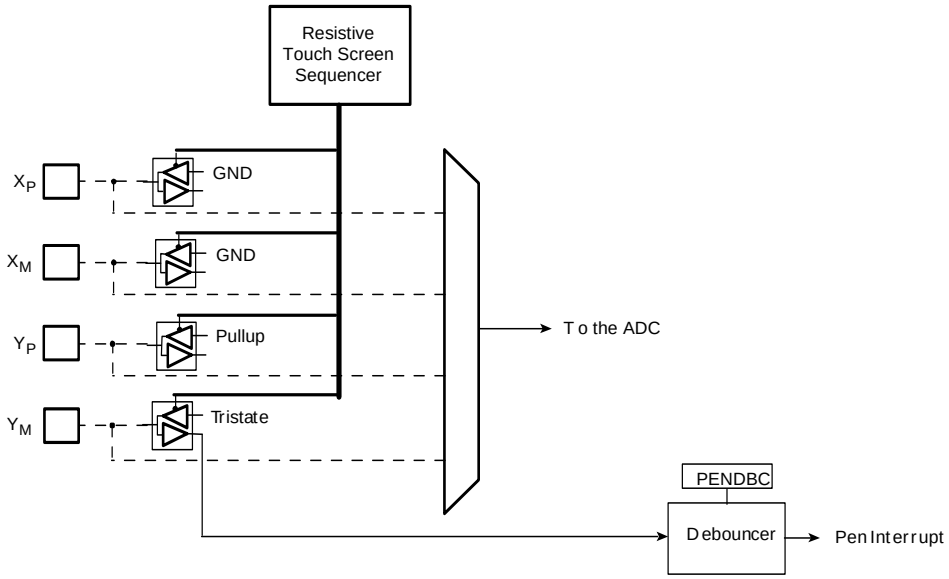
29.7 Resistive Touch Screen

The ADCIFB embeds an integrated Resistive Touch Screen Sequencer that can be used to calculate contact coordinates on a resistive touch screen film. When instructed to start, the integrated Resistive Touch Screen Sequencer automatically applies a sequence of voltage patterns to the resistive touch screen films and the Analog-to-Digital Conversion cell is used to measure the effects. The resulting measurements can be used to calculate the horizontal and vertical contact coordinates. It is recommended to use a high resistance touch screen for optimal resolution.

The resistive touch screen film is connected to the ADCIFB using the AD and ADP pins. See Section 29.7.3 for details.

Resistive Touch Screen Mode is enabled by writing a one to the Touch Screen ADC Mode field in the Mode Register (MR.TSAMOD). In this mode, channels TSPO+0 though TSPO+3 are automatically enabled where TSPO refers to the Touch Screen Pin Offset field in the Mode Register (MR.TSPO). For each conversion sequence, all enabled channels before TSPO+0 and after TSPO+3 are converted as ordinary ADC channels, producing 1 conversion result each. When the sequencer enters the TSPO+0 channel the Resistive Touch Screen Sequencer will take over control and convert the next 4 channels as described in Section 29.7.4.

Figure 29-4. Resistive Touch Screen Pen Detect



The Resistive Touch Screen Pen Detect can be used to generate an ADCIFB interrupt request or it can be used to trig a conversion, so that a position can be measured as soon as a contact is detected.

The Pen Detect Mode generates two types of status signals, reported in the Status Register (SR):

- The bit PENCNT is set when current flows and remains set until current stops.
- The bit NOCNT is set when no current flows and remains set until current flows.

Before a current change is reflected in the SR, the new status must be stable for the duration of the debouncing time.

Both status conditions can generate an interrupt request if the corresponding bit in the Interrupt Mask Register (IMR) is one. Refer to Section 29.6.11 on page 717.

30.9.6 Interrupt Status Register

Name: ISR

Access Type: Read-only

Offset: 0x1C

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
-	-	-	-	WFINT3	WFINT2	WFINT1	WFINT0
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
SUTINT7	ACINT7	SUTINT6	ACINT6	SUTINT5	ACINT5	SUTINT4	ACINT4
7	6	5	4	3	2	1	0
SUTINT3	ACINT3	SUTINT2	ACINT2	SUTINT1	ACINT1	SUTINT0	ACINT0

- **WFINTn: Window Mode Interrupt Status**
  - 0: No Window Mode Interrupt is pending.
  - 1: Window Mode Interrupt is pending.
  - This bit is cleared when the corresponding bit in ICR is written to one.
  - This bit is set when the corresponding channel pair operating in window mode generated an interrupt.
- **SUTINTn: ACn Startup Time Interrupt Status**
  - 0: No Startup Time Interrupt is pending.
  - 1: Startup Time Interrupt is pending.
  - This bit is cleared when the corresponding bit in ICR is written to one.
  - This bit is set when the startup time of the corresponding AC has passed.
- **ACINTn: ACn Interrupt Status**
  - 0: No Normal Mode Interrupt is pending.
  - 1: Normal Mode Interrupt is pending.
  - This bit is cleared when the corresponding bit in ICR is written to one.
  - This bit is set when the corresponding channel generated an interrupt.

32.8 Module Configuration

The specific configuration for each GLOC instance is listed in the following tables.The GLOC bus clocks listed here are connected to the system bus clocks. Please refer to the Power Manager chapter for details.

Table 32-4. GLOC Configuration

Feature	GLOC
Number of LUT units	2

Table 32-5. GLOC Clocks

Clock Name	Description
CLK_GLOC	Clock for the GLOC bus interface
GCLK	The generic clock used for the GLOC is GCLK5

Table 32-6. Register Reset Values

Register	Reset Value
VERSION	0x00000100
PARAMETER	0x00000002

34.3.7.1 *Cyclic Redundancy Check (CRC)*

The MSU can be used to automatically calculate the CRC of a block of data in memory. The MSU will then read out each word in the specified memory block and report the CRC32-value in an MSU register.

34.3.7.2 *NanoTrace*

The MSU additionally supports NanoTrace. This is a 32-bit AVR-specific feature, in which trace data is output to memory instead of the AUX port. This allows the trace data to be extracted by the debugger through the SAB, enabling trace features for aWire- or JTAG-based debuggers. The user must write MSU registers to configure the address and size of the memory block to be used for NanoTrace. The NanoTrace buffer can be anywhere in the physical address range, including internal and external RAM, through an EBI, if present. This area may not be used by the application running on the CPU.

34.3.8 AUX-based Debug Features

Utilizing the Auxiliary (AUX) port gives access to a wide range of advanced debug features. Of prime importance are the trace features, which allow an external debugger to receive continuous information on the program execution in the CPU. Additionally, Event In and Event Out pins allow external events to be correlated with the program flow.

Debug tools utilizing the AUX port should connect to the device through a Nexus-compliant Mictor-38 connector, as described in the AVR32UC Technical Reference manual. This connector includes the JTAG signals and the RESET_N pin, giving full access to the programming and debug features in the device.

15.7 Module Configuration 351

16 Watchdog Timer (WDT) 352

16.1 Features 352
16.2 Overview 352
16.3 Block Diagram 352
16.4 Product Dependencies 352
16.5 Functional Description 353
16.6 User Interface 358
16.7 Module Configuration 364

17 External Interrupt Controller (EIC) 365

17.1 Features 365
17.2 Overview 365
17.3 Block Diagram 365
17.4 I/O Lines Description 366
17.5 Product Dependencies 366
17.6 Functional Description 366
17.7 User Interface 370
17.8 Module Configuration 386

18 Frequency Meter (FREQM) 387

18.1 Features 387
18.2 Overview 387
18.3 Block Diagram 387
18.4 Product Dependencies 387
18.5 Functional Description 388
18.6 User Interface 390
18.7 Module Configuration 401

19 General-Purpose Input/Output Controller (GPIO) 403

19.1 Features 403
19.2 Overview 403
19.3 Block Diagram 403
19.4 I/O Lines Description 404
19.5 Product Dependencies 404
19.6 Functional Description 405
19.7 User Interface 410
19.8 Module Configuration 433