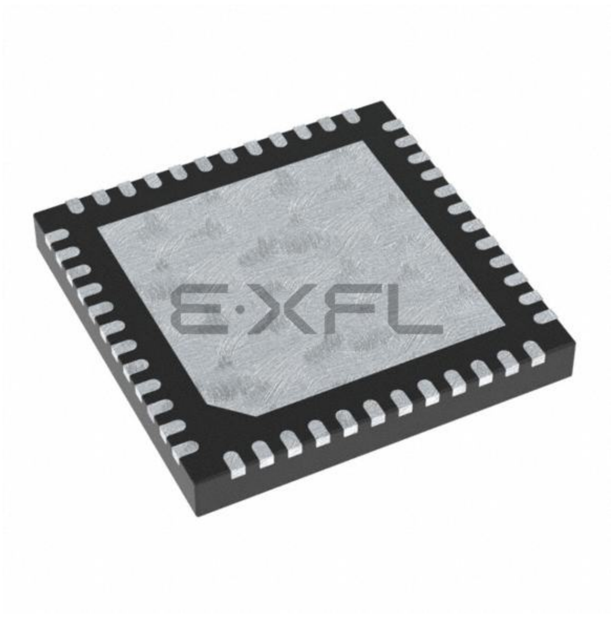
Microchip Technology - AT32UC3L064-ZAUR Datasheet



Welcome to E-XFL.COM

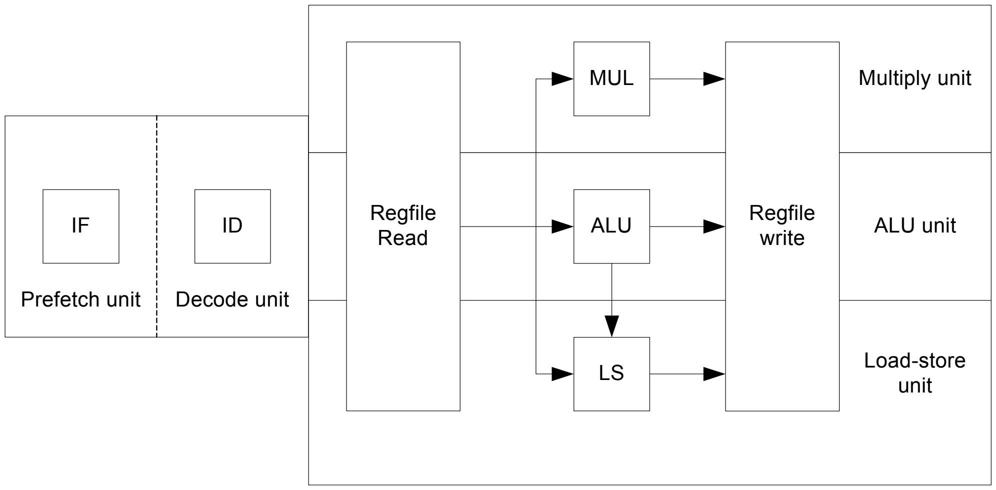
What is "Embedded - Microcontrollers"?

"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	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	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/microchip-technology/at32uc3l064-zaur

Figure 4-2. The AVR32UC Pipeline



4.3.2 AVR32A Microarchitecture Compliance

AVR32UC implements an AVR32A microarchitecture. The AVR32A microarchitecture is targeted at cost-sensitive, lower-end applications like smaller microcontrollers. This microarchitecture does not provide dedicated hardware registers for shadowing of register file registers in interrupt contexts. Additionally, it does not provide hardware registers for the return address registers and return status registers. Instead, all this information is stored on the system stack. This saves chip area at the expense of slower interrupt handling.

4.3.2.1 Interrupt Handling

Upon interrupt initiation, registers R8-R12 are automatically pushed to the system stack. These registers are pushed regardless of the priority level of the pending interrupt. The return address and status register are also automatically pushed to stack. The interrupt handler can therefore use R8-R12 freely. Upon interrupt completion, the old R8-R12 registers and status register are restored, and execution continues at the return address stored popped from stack.

The stack is also used to store the status register and return address for exceptions and *scall*. Executing the *rete* or *rets* instruction at the completion of an exception or system call will pop this status register and continue execution at the popped return address.

4.3.2.2 Java Support

AVR32UC does not provide Java hardware acceleration.

4.3.2.3 Memory Protection

The MPU allows the user to check all memory accesses for privilege violations. If an access is attempted to an illegal memory address, the access is aborted and an exception is taken. The MPU in AVR32UC is specified in the AVR32UC Technical Reference manual.

4.3.2.4 Unaligned Reference Handling

AVR32UC does not support unaligned accesses, except for doubleword accesses. AVR32UC is able to perform word-aligned *st.d* and *ld.d*. Any other unaligned memory access will cause an

Debug state can be entered as described in the *AVR32UC Technical Reference Manual*.

Debug state is exited by the *retd* instruction.

4.4.3.3 *Secure State*

The AVR32 can be set in a secure state, that allows a part of the code to execute in a state with higher security levels. The rest of the code can not access resources reserved for this secure code. Secure State is used to implement FlashVault technology. Refer to the *AVR32UC Technical Reference Manual* for details.

4.4.4 System Registers

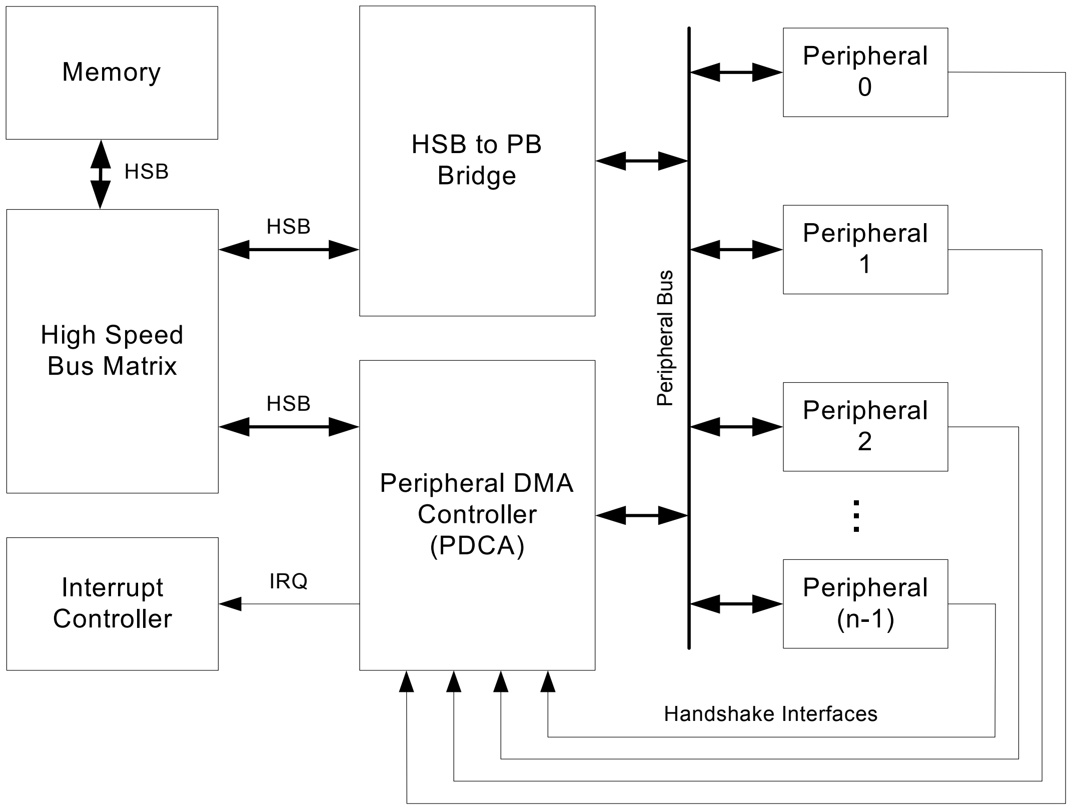
The system registers are placed outside of the virtual memory space, and are only accessible using the privileged *mfsr* and *mtsr* instructions. The table below lists the system registers specified in the AVR32 architecture, some of which are unused in AVR32UC. The programmer is responsible for maintaining correct sequencing of any instructions following a *mtsr* instruction. For detail on the system registers, refer to the *AVR32UC Technical Reference Manual*.

Table 4-3. System Registers

Reg #	Address	Name	Function
0	0	SR	Status Register
1	4	EVBA	Exception Vector Base Address
2	8	ACBA	Application Call Base Address
3	12	CPUCR	CPU Control Register
4	16	ECR	Exception Cause Register
5	20	RSR_SUP	Unused in AVR32UC
6	24	RSR_INT0	Unused in AVR32UC
7	28	RSR_INT1	Unused in AVR32UC
8	32	RSR_INT2	Unused in AVR32UC
9	36	RSR_INT3	Unused in AVR32UC
10	40	RSR_EX	Unused in AVR32UC
11	44	RSR_NMI	Unused in AVR32UC
12	48	RSR_DBG	Return Status Register for Debug mode
13	52	RAR_SUP	Unused in AVR32UC
14	56	RAR_INT0	Unused in AVR32UC
15	60	RAR_INT1	Unused in AVR32UC
16	64	RAR_INT2	Unused in AVR32UC
17	68	RAR_INT3	Unused in AVR32UC
18	72	RAR_EX	Unused in AVR32UC
19	76	RAR_NMI	Unused in AVR32UC
20	80	RAR_DBG	Return Address Register for Debug mode
21	84	JECR	Unused in AVR32UC
22	88	JOSP	Unused in AVR32UC
23	92	JAVA_LV0	Unused in AVR32UC

7.3 Block Diagram

Figure 7-1. PDCA Block Diagram



7.4 Product Dependencies

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

7.4.1 Power Management

If the CPU enters a sleep mode that disables the PDCA clocks, the PDCA will stop functioning and resume operation after the system wakes up from sleep mode.

7.4.2 Clocks

The PDCA has two bus clocks connected: One High Speed Bus clock (CLK_PDCA_HSB) and one Peripheral Bus clock (CLK_PDCA_PB). These clocks are generated by the Power Manager. Both clocks are enabled at reset, and can be disabled in the Power Manager. It is recommended to disable the PDCA before disabling the clocks, to avoid freezing the PDCA in an undefined state.

7.4.3 Interrupts

The PDCA interrupt request lines are connected to the interrupt controller. Using the PDCA interrupts requires the interrupt controller to be programmed first.

Reset Source	Description
SM33 Reset	Internal regulator supply voltage below the SM33 threshold voltage. This generates a Power-on Reset.
Watchdog Timer	See Watchdog Timer documentation
OCD	See On-Chip Debug documentation

Depending on the reset source, when a reset occurs, some parts of the device are not always reset. Only the Power-on Reset (POR) will force a whole device reset. Refer to the table in the Module Configuration section at the end of this chapter for further details. The latest reset cause can be read in the RCAUSE register, and can be read during the applications boot sequence in order to determine proper action.

12.6.6.1 Power-on Reset Detector

The Power-on Reset 1.8V (POR18) detector monitors the VDDCORE supply pin and generates a Power-on Reset (POR) when the device is powered on. The POR is active until the VDDCORE voltage is above the power-on threshold level (V_{POT}). The POR will be re-generated if the voltage drops below the power-on threshold level. See Electrical Characteristics for parametric details.

The Power-on Reset 3.3V (POR33) detector monitors the internal regulator supply pin and generates a Power-on Reset (POR) when the device is powered on. The POR is active until the internal regulator supply voltage is above the regulator power-on threshold level (V_{POT}). The POR will be re-generated if the voltage drops below the regulator power-on threshold level. See Electrical Characteristics for parametric details.

12.6.6.2 External Reset

The external reset detector monitors the RESET_N pin state. By default, a low level on this pin will generate a reset.

12.6.7 Clock Failure Detector

This mechanism automatically switches the main clock source to the safe RCSYS clock when the main clock source fails. This may happen when an external crystal is selected as a source for the main clock and the crystal is not mounted on the board. The main clock is compared with RCSYS, and if no rising edge of the main clock is detected during one RCSYS period, the clock is considered to have failed.

The detector is enabled by writing a one to the Clock Failure Detection Enable bit in the Clock Failure Detector Control Register (CFDCTRL.CFDEN). As soon as the detector is enabled, the clock failure detector will monitor the divided main clock. Note that the detector does not monitor the main clock if RCSYS is the source of the main clock, or if the main clock is temporarily not available (startup-time after a wake-up, switching timing etc.), or in sleep mode where the main clock is driven by the RCSYS (Stop and DeepStop mode). When a clock failure is detected, the main clock automatically switches to the RCSYS clock and the Clock Failure Detected (CFD) interrupt is generated if enabled. The MCCTRL register is also changed by hardware to indicate that the main clock comes from RCSYS.

12.6.8 Interrupts

The PM has a number of interrupt sources:

- AE - Access Error,

12.7.17 Wake Cause Register

Name: WCAUSE

Access Type: Read-only

Offset: 0x184

Reset Value: Latest Wake Source

31	30	29	28	27	26	25	24
WCAUSE[31:24]							

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

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

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

A bit in this register is set on wake up caused by the peripheral referred to in Table 12-12 on page 181.

Table 12-12. Wake Cause

Bit	Wake Cause
0	CAT
1	ACIFB
2	ADCIFB
3	TWI Slave 0
4	TWI Slave 1
5	WAKE_N
6	ADCIFB Pen Detect
15:7	-
16	EIC
17	AST
31:18	-

13.6.20 Temperature Sensor Configuration Register

Name:	TSENS
Access Type:	Read/Write
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
-	-	-			-	-	EN

- **EN: Temperature Sensor Enable**

  0: The Temperature Sensor is disabled.

  1: The Temperature Sensor 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.

13.6.23 32kHz RC Oscillator Configuration Register

Name:	RC32KCR
Access Type:	Read/Write
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
-	-	-	-	-	-	-	EN

- **EN: RC32K Enable**
  0: The 32 kHz RC oscillator is disabled.
  1: The 32 kHz RC oscillator 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.

15. Watchdog Timer (WDT)

Rev: 4.0.2.0

15.1 Features

- **Watchdog Timer counter with 32-bit counter**
- **Timing window watchdog**
- **Clocked from system RC oscillator or the 32 KHz crystal oscillator**
- **Configuration lock**
- **WDT may be enabled at reset by a fuse**

15.2 Overview

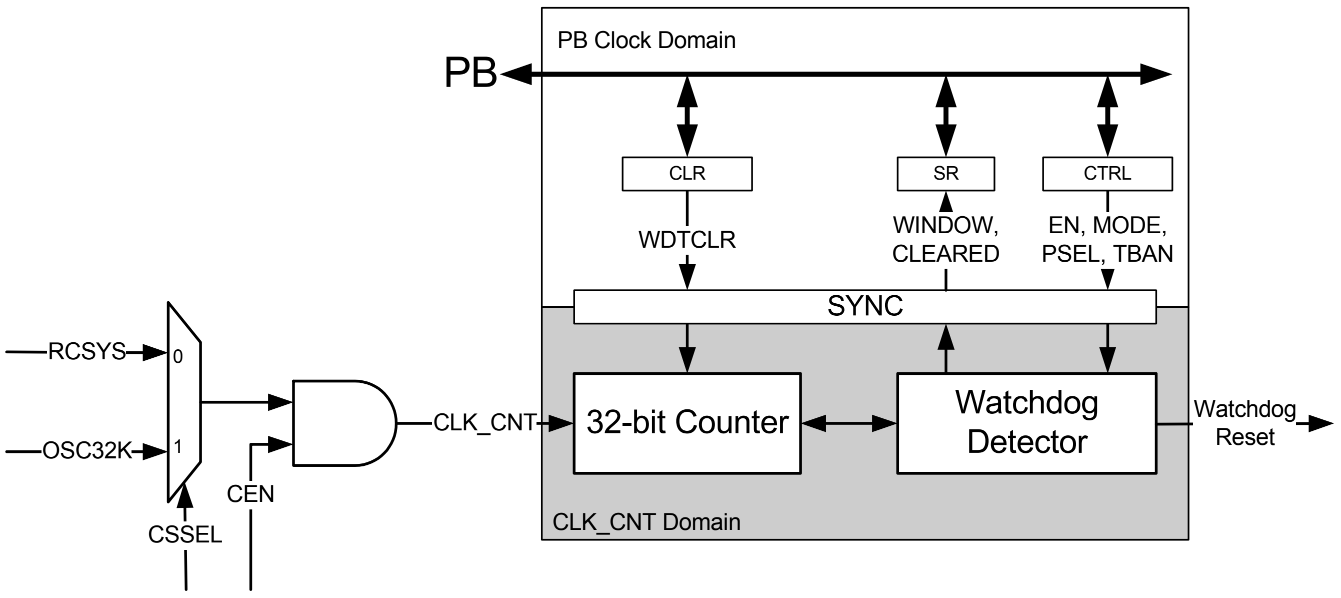
The Watchdog Timer (WDT) will reset the device unless it is periodically serviced by the software. This allows the device to recover from a condition that has caused the system to be unstable.

The WDT has an internal counter clocked from the system RC oscillator or the 32 KHz crystal oscillator.

The WDT counter must be periodically cleared by software to avoid a watchdog reset. If the WDT timer is not cleared correctly, the device will reset and start executing from the boot vector.

15.3 Block Diagram

Figure 15-1. WDT Block Diagram



15.4 Product Dependencies

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

18.7.18 Lock Register

Name:	LOCK
Access:	Read/Write, Set, Clear, Toggle
Offset:	0x1A0, 0x1A4, 0x1A8, 0x1AC
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: Lock State**

  0: Pin is unlocked. The corresponding bit can be changed in any GPIO register for this port.

  1: Pin is locked. The corresponding bit can not be changed in any GPIO register for this port.

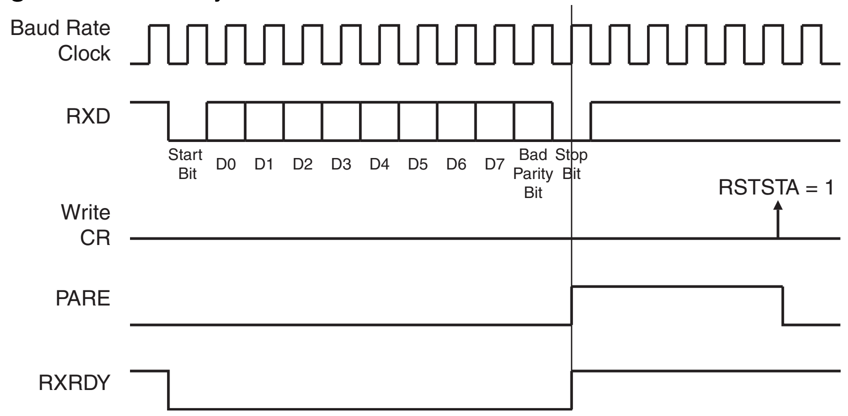
  The value of LOCK determines which bits are locked in the lockable registers.

  The LOCK, LOCKC, and LOCKT registers are protected, which means they can only be written immediately after a write to the UNLOCK register with the proper KEY and OFFSET.

  LOCKS is not protected, and can be written at any time.

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

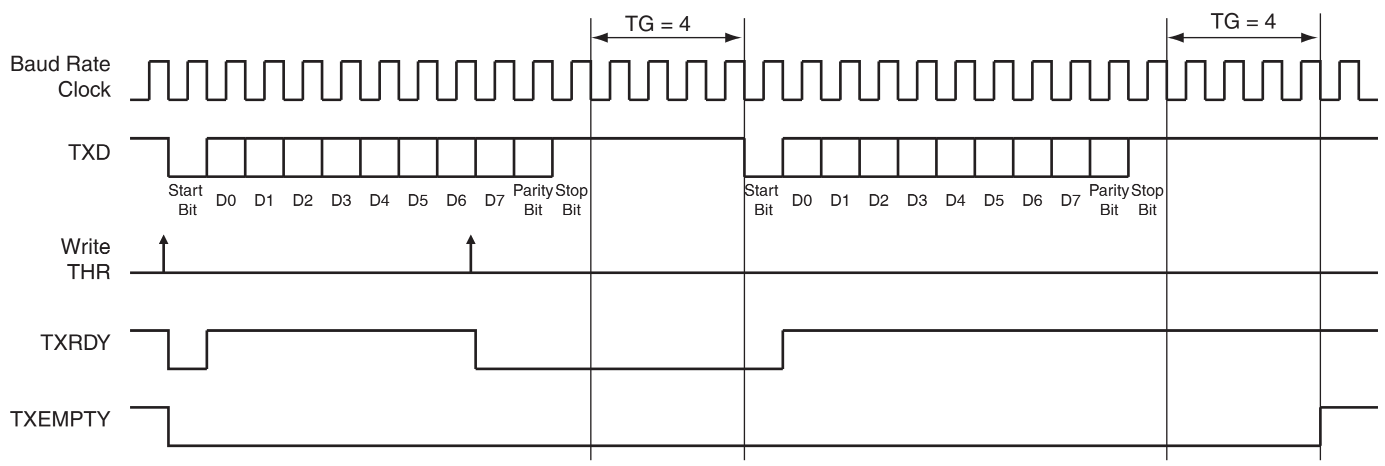
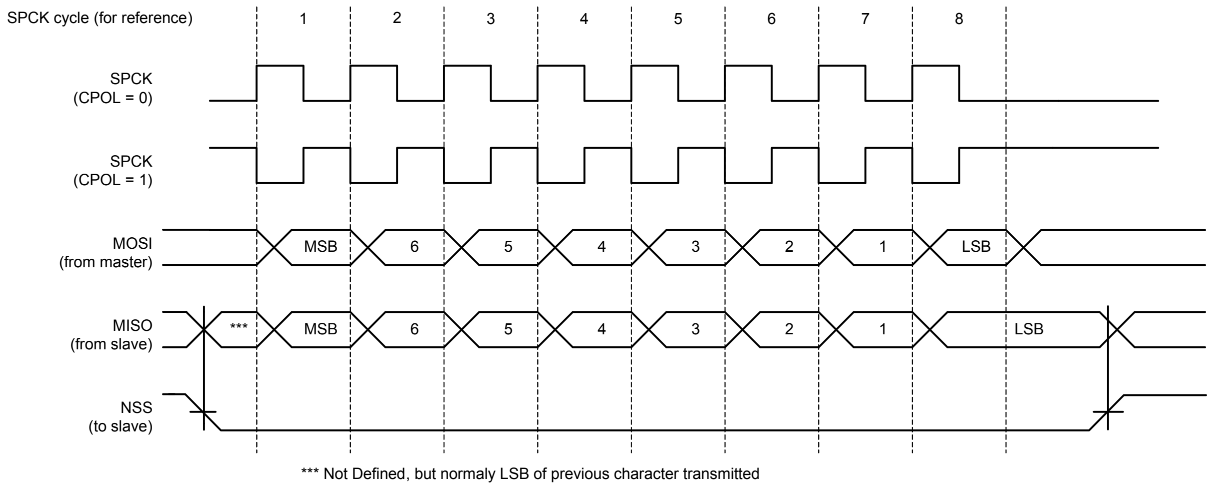


Figure 20-4. SPI Transfer Format (NCPHA = 0, 8 bits per transfer)



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.

22.9.8 Interrupt Enable Register

Name:	IER
Access Type:	Write-only
Offset:	0x1C
Reset Value:	0x00000000

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	-	-	TCOMP	-	TXRDY	RXRDY

Writing a zero to a bit in this register has no effect.

Writing a one to a bit in this register will write a one to the corresponding bit in IMR.

26.7.3 Resistive Touch Screen Pin Connections

Table 26-2. Resistive Touch Screen Pin Connections

ADCIFB Pin	TS Signal, APOE == 0	TS Signal, APOE == 1
ADP0	Xp through a resistor	No Connect
ADP1	Yp through a resistor	No Connect
ADtspo+0	Xp	Xp
ADtspo+1	Xm	Xm
ADtspo+2	Yp	Yp
ADtspo+3	Ym	Ym

The resistive touch screen film signals connects to the ADCIFB using the AD and ADP pins. The X_P (top) and X_M (bottom) film signals are connected to ADtspo+0 and ADtspo+1 pins, and the Y_P (right) and Y_M (left) signals are connected to ADtspo+2 and ADtspo+3 pins. The tspo index is configurable through the Touch Screen Pin Offset (TSPO) field in the Mode Register (MR) and allows the user to configure which AD pins to use for resistive touch screen applications. Writing a zero to the TSPO field instructs the ADCIFB to use AD0 through AD3, where AD0 is connected to X_P, AD1 is connected to X_M and so on. Writing a one to the TSPO field instructs the ADCIFB to use AD1 through AD4 for resistive touch screen sequencing, where AD1 is connected to X_P and AD0 is free to be used as an ordinary ADC channel.

When the Analog Pin Output Enable (APOE) bit in the Mode Register (MR) is zero, the AD pins are used to measure input voltage and drive the GND sequences, while the ADP pins are used to drive the VDD sequences. This arrangement allows the user to reduce the voltage seen at the AD input pins by inserting external resistors between ADP0 and X_P and ADP1 and Y_P signals which are again directly connected to the AD pins. It is important that the voltages observed at the AD pins are not higher than the maximum allowed ADC input voltage. See Figure 26-3 on page 597 for details regarding how to connect the resistive touch screen films to the AD and ADP pins.

By adding a resistor between ADP0 and X_P, and ADP1 and Y_P, the maximum voltage observed at the AD pins can be controlled by the following voltage divider expressions:

$$V(AD_{tspo+0}) = \frac{R_{filmx}}{R_{filmx} + R_{resistorx}} \cdot V(DP_0)$$

The Rfilmx parameter is the film resistance observed when measuring between X_P and X_M. The Rresistorx parameter is the resistor size inserted between ADP0 and X_P. The definition of Rfilmy and Rresistory is the same but for ADP1, Y_P, and Y_M instead.

Table 27-2. I/O Line Description

Pin Name	Pin Description	Type
ACBPn	Positive reference pin for Analog Comparator B n	Analog
ACBNn	Negative reference pin for Analog Comparator B n	Analog
ACREFN	Reference Voltage for all comparators selectable for INN	Analog

The signal names corresponds to the groups A and B of analog comparators. For normal mode, the mapping from input signal names in the block diagram to the signal names is given in Table 27-3.

Table 27-3. Signal Name Mapping

Pin Name	Channel Number	Normal Mode
ACAP0/ACAN0	0	ACP0/ACN0
ACBP0/ACBN0	1	ACP1/ACN1
ACAP1/ACAN1	2	ACP2/ACN2
ACBP1/ACBN1	3	ACP3/ACN3
ACAP2/ACAN2	4	ACP4/ACN4
ACBP2/ACBN2	5	ACP5/ACN5
ACAP3/ACAN3	6	ACP6/ACN6
ACBP3/ACBN3	7	ACP7/ACN7

27.5 Product Dependencies

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

27.5.1 I/O Lines

The ACIFB pins are multiplexed with other peripherals. The user must first program the I/O Controller to give control of the pins to the ACIFB.

27.5.2 Power Management

If the CPU enters a sleep mode that disables clocks used by the ACIFB, the ACIFB will stop functioning and resume operation after the system wakes up from sleep mode.

27.5.3 Clocks

The clock for the ACIFB bus interface (CLK_ACIFB) is generated by the Power Manager. This clock is enabled at reset, and can be disabled in the Power Manager. It is recommended to disable the ACIFB before disabling the clock, to avoid freezing the ACIFB in an undefined state.

The ACIFB uses a GCLK as clock source for the Analog Comparators. The user must set up this GCLK at the right frequency. The CLK_ACIFB clock of the interface must be at least 4x the GCLK frequency used in the comparators. The GCLK is used both for measuring the startup time of a comparator, and to give a frequency for the comparisons done in Continuous Measurement Mode, see Section 27.6.

Refer to the Electrical Characteristics chapter for GCLK frequency limitations.

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.

28.7.26 Autonomous Touch Current Count Register

Name:	ATCURR
Access Type:	Read-only
Offset:	0x70
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
COUNT[15:8]							

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

- **COUNT: Count value**

  The current count acquired by the autonomous touch sensor. This is useful for autonomous touch debugging purposes.

each of which are controlled by a pair of OCD registers which determine the range of addresses (or single address) which should produce data trace messages.

31.3.8.4 *Ownership Trace*

Program and data trace operate on virtual addresses. In cases where an operating system runs several processes in overlapping virtual memory segments, the Ownership Trace feature can be used to identify the process switch. When the O/S activates a process, it will write the process ID number to an OCD register, which produces an Ownership Trace Message, allowing the debugger to switch context for the subsequent program and data trace messages. As the use of this feature depends on the software running on the CPU, it can also be used to extract other types of information from the system.

31.3.8.5 *Watchpoint Messages*

The breakpoint modules normally used to generate program and data breakpoints can also be used to generate Watchpoint messages, allowing a debugger to monitor program and data events without halting the CPU. Watchpoints can be enabled independently of breakpoints, so a breakpoint module can optionally halt the CPU when the trigger condition occurs. Data trace modules can also be configured to produce watchpoint messages instead of regular data trace messages.

31.3.8.6 *Event In and Event Out Pins*

The AUX port also contains an Event In pin (EVTI_N) and an Event Out pin (EVTO_N). EVTI_N can be used to trigger a breakpoint when an external event occurs. It can also be used to trigger specific program and data trace synchronization messages, allowing an external event to be correlated to the program flow.

When the CPU enters debug mode, a Debug Status message is transmitted on the trace port. All trace messages can be timestamped when they are received by the debug tool. However, due to the latency of the transmit queue buffering, the timestamp will not be 100% accurate. To improve this, EVTO_N can toggle every time a message is inserted into the transmit queue, allowing trace messages to be timestamped precisely. EVTO_N can also toggle when a breakpoint module triggers, or when the CPU enters debug mode, for any reason. This can be used to measure precisely when the respective internal event occurs.

31.3.8.7 *Software Quality Analysis (SQA)*

Software Quality Analysis (SQA) deals with two important issues regarding embedded software development. *Code coverage* involves identifying untested parts of the embedded code, to improve test procedures and thus the quality of the released software. *Performance analysis* allows the developer to precisely quantify the time spent in various parts of the code, allowing bottlenecks to be identified and optimized.

Program trace must be used to accomplish these tasks without instrumenting (altering) the code to be examined. However, traditional program trace cannot reconstruct the current PC value without correlating the trace information with the source code, which cannot be done on-the-fly. This limits program trace to a relatively short time segment, determined by the size of the trace buffer in the debug tool.

The OCD system in AT32UC3L016/32/64 extends program trace with SQA capabilities, allowing the debug tool to reconstruct the PC value on-the-fly. Code coverage and performance analysis can thus be reported for an unlimited execution sequence.

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 register. For a write operation, scan in the new contents of the register.
9. Return to Run-Test/Idle.

For any operation, the full 7 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 31-19. MEMORY_SERVICE Details

Instructions	Details
IR input value	**10100** (0x14)
IR output value	peb01
DR Size	34 bits
DR input value (Address phase)	**aaaaaaar** xxxxxxxx xxxxxxxx xxxxxxxx xx
DR input value (Data read phase)	**xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx** xx
DR input value (Data write phase)	**dddddddd dddddddd dddddddd dddddddd** xx
DR output value (Address phase)	xx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxeb
DR output value (Data read phase)	eb **dddddddd dddddddd dddddddd dddddddd**
DR output value (Data write phase)	xx **xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxeb**

31.5.3.3 MEMORY_SIZED_ACCESS

This instruction allows access to the entire Service Access Bus data area. Data is accessed through a 36-bit byte index, a 2-bit size, a direction bit, and 8, 16, or 32 bits of data. Not all units mapped on the SAB bus may support all sizes of accesses, e.g., some may only support word accesses.

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_SIZED_ACCESS instruction is selected, and toggles between address and data mode each time a data scan completes with the busy bit cleared.

The size field is encoded as i Table 31-20.

Table 31-20. Size Field Semantics

Size field value	Access size	Data alignment
00	Byte (8 bits)	Address modulo 4 : data alignment 0: **dddddddd** xxxxxxxx xxxxxxxx xxxxxxxx 1: xxxxxxxx **dddddddd** xxxxxxxx xxxxxxxx 2: xxxxxxxx xxxxxxxx **dddddddd** xxxxxxxx 3: xxxxxxxx xxxxxxxx xxxxxxxx **dddddddd**
01	Halfword (16 bits)	Address modulo 4 : data alignment 0: **dddddddd dddddddd** xxxxxxxx xxxxxxxx 1: Not allowed 2: xxxxxxxx xxxxxxxx **dddddddd dddddddd** 3: Not allowed
10	Word (32 bits)	Address modulo 4 : data alignment 0: **dddddddd dddddddd dddddddd dddddddd** 1: Not allowed 2: Not allowed 3: Not allowed
11	Reserved	N/A

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), 2-bit access size, and the 36-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 36 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 31-21. MEMORY_SIZED_ACCESS Details

Instructions	Details
IR input value	**10101** (0x15)
IR output value	peb01
DR Size	39 bits
DR input value (Address phase)	aaaaaaaa aaaaaaaa aaaaaaaa aaaaaaaa aaaassr
DR input value (Data read phase)	**xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx** xxxxxxx
DR input value (Data write phase)	**dddddddd dddddddd dddddddd dddddddd** xxxxxxx

31.6.8.5 MEMORY_READWRITE_STATUS

After a MEMORY_WRITE command this response is sent by AW. The response can also be sent after a MEMORY_READ command if AW encountered an error when receiving the address. The response contains 3 bytes, where the first is the status of the command and the 2 next contains the byte count when the first error occurred. The first byte is encoded this way:

Table 31-55. MEMORY_READWRITE_STATUS Status Byte

status byte	Description
0x00	Write successful
0x01	SAB busy
0x02	Bus error (wrong address)
Other	Reserved

Table 31-56. MEMORY_READWRITE_STATUS Details

Response	Details
Response value	0xC2
Additional data	Status byte and byte count (2 bytes)

31.6.8.6 BAUD_RATE

The current baud rate in the AW. See Section 31.6.6.7 for more details.

Table 31-57. BAUD_RATE Details

Response	Details
Response value	0xC3
Additional data	Baud rate

31.6.8.7 STATUS_INFO

A status message from AW.

Table 31-58. STATUS_INFO Contents

Bit number	Name	Description
15-9	Reserved	
8	Protected	The protection bit in the internal flash is set. SAB access is restricted. This bit will read as one during reset.
7	SAB busy	The SAB bus is busy with a previous transfer. This could indicate that the CPU is running on a very slow clock, the CPU clock has stopped for some reason or that the part is in constant reset.
6	Chip erase ongoing	The Chip erase operation has not finished.
5	CPU halted	This bit will be set if the CPU is halted. This bit will read as zero during reset.
4-1	Reserved	
0	Reset status	This bit will be set if AW has reset the CPU using the RESET command.