



Welcome to [E-XFL.COM](https://www.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	60MHz
Connectivity	I <sup>2</sup> C, IrDA, SPI, SSC, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	44
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	32K x 8
Voltage - Supply (Vcc/Vdd)	1.65V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-VFQFN Exposed Pad
Supplier Device Package	64-QFN (9x9)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/at32uc3b0128-z2ut">https://www.e-xfl.com/product-detail/microchip-technology/at32uc3b0128-z2ut</a>

## 1. Description

The AT32UC3B is a complete System-On-Chip microcontroller based on the AVR32 UC RISC processor running at frequencies up to 60 MHz. AVR32 UC is a high-performance 32-bit RISC microprocessor core, designed for cost-sensitive embedded applications, with particular emphasis on low power consumption, high code density and high performance.

The processor implements a Memory Protection Unit (MPU) and a fast and flexible interrupt controller for supporting modern operating systems and real-time operating systems.

Higher computation capability is achieved using a rich set of DSP instructions.

The AT32UC3B incorporates on-chip Flash and SRAM memories for secure and fast access.

The Peripheral Direct Memory Access controller enables data transfers between peripherals and memories without processor involvement. PDCA drastically reduces processing overhead when transferring continuous and large data streams between modules within the MCU.

The Power Manager improves design flexibility and security: the on-chip Brown-Out Detector monitors the power supply, the CPU runs from the on-chip RC oscillator or from one of external oscillator sources, a Real-Time Clock and its associated timer keeps track of the time.

The Timer/Counter includes three identical 16-bit timer/counter channels. Each channel can be independently programmed to perform frequency measurement, event counting, interval measurement, pulse generation, delay timing and pulse width modulation.

The PWM modules provides seven independent channels with many configuration options including polarity, edge alignment and waveform non overlap control. One PWM channel can trigger ADC conversions for more accurate close loop control implementations.

The AT32UC3B also features many communication interfaces for communication intensive applications. In addition to standard serial interfaces like USART, SPI or TWI, other interfaces like flexible Synchronous Serial Controller and USB are available. The USART supports different communication modes, like SPI mode.

The Synchronous Serial Controller provides easy access to serial communication protocols and audio standards like I<sup>2</sup>S, UART or SPI.

The Full-Speed USB 2.0 Device interface supports several USB Classes at the same time thanks to the rich End-Point configuration. The Embedded Host interface allows device like a USB Flash disk or a USB printer to be directly connected to the processor.

Atmel offers the QTouch library for embedding capacitive touch buttons, sliders, and wheels functionality into AVR microcontrollers. The patented charge-transfer signal acquisition offers robust sensing and included fully debounced reporting of touch keys and includes Adjacent Key Suppression<sup>®</sup> (AKS<sup>®</sup>) technology for unambiguous detection of key events. The easy-to-use QTouch Suite toolchain allows you to explore, develop, and debug your own touch applications.

AT32UC3B integrates a class 2+ Nexus 2.0 On-Chip Debug (OCD) System, with non-intrusive real-time trace, full-speed read/write memory access in addition to basic runtime control. The Nanotrace interface enables trace feature for JTAG-based debuggers.



### 3. Configuration Summary

The table below lists all AT32UC3B memory and package configurations:

**Table 3-1.** Configuration Summary

Feature	AT32UC3B0512	AT32UC3B0256/128/64	AT32UC3B1512	AT32UC3B1256/128/64
Flash	512 KB	256/128/64 KB	512 KB	256/128/64 KB
SRAM	96KB	32/32/16KB	96KB	32/16/16KB
GPIO	44		28	
External Interrupts	8		6	
TWI	1			
USART	3			
Peripheral DMA Channels	7			
SPI	1			
Full Speed USB	Mini-Host + Device		Device	
SSC	1		0	
Audio Bitstream DAC	1	0	1	0
Timer/Counter Channels	3			
PWM Channels	7			
Watchdog Timer	1			
Real-Time Clock Timer	1			
Power Manager	1			
Oscillators	PLL 80-240 MHz (PLL0/PLL1) Crystal Oscillators 0.4-20 MHz (OSC0) Crystal Oscillator 32 KHz (OSC32K) RC Oscillator 115 kHz (RCSYS)			
	Crystal Oscillators 0.4-20 MHz (OSC1)			
10-bit ADC number of channels	8		6	
JTAG	1			
Max Frequency	60 MHz			
Package	TQFP64, QFN64		TQFP48, QFN48	

## 5.2 RESET\_N pin

The RESET\_N pin is a schmitt input and integrates a permanent pull-up resistor to VDDIO. As the product integrates a power-on reset cell, the RESET\_N pin can be left unconnected in case no reset from the system needs to be applied to the product.

## 5.3 TWI pins

When these pins are used for TWI, the pins are open-drain outputs with slew-rate limitation and inputs with inputs with spike-filtering. When used as GPIO-pins or used for other peripherals, the pins have the same characteristics as GPIO pins.

## 5.4 GPIO pins

All the I/O lines integrate a pull-up resistor. Programming of this pull-up resistor is performed independently for each I/O line through the GPIO Controllers. After reset, I/O lines default as inputs with pull-up resistors disabled, except when indicated otherwise in the column "Reset Value" of the GPIO Controller user interface table.

## 5.5 High drive pins

The four pins PA20, PA21, PA22, PA23 have high drive output capabilities.

## 5.6 Power Considerations

### 5.6.1 Power Supplies

The AT32UC3B has several types of power supply pins:

- **VDDIO:** Powers I/O lines. Voltage is 3.3V nominal.
- **VDDANA:** Powers the ADC Voltage is 3.3V nominal.
- **VDDIN:** Input voltage for the voltage regulator. Voltage is 3.3V nominal.
- **VDDCORE:** Powers the core, memories, and peripherals. Voltage is 1.8V nominal.
- **VDDPLL:** Powers the PLL. Voltage is 1.8V nominal.

The ground pins GND are common to VDDCORE, VDDIO and VDDPLL. The ground pin for VDDANA is GNDANA.

Refer to Electrical Characteristics section for power consumption on the various supply pins.

The main requirement for power supplies connection is to respect a star topology for all electrical connection.

The user must also make sure that the system stack is large enough so that any event is able to push the required registers to stack. If the system stack is full, and an event occurs, the system will enter an UNDEFINED state.

### 6.5.2 Exceptions and Interrupt Requests

When an event other than *scall* or debug request is received by the core, the following actions are performed atomically:

1. The pending event will not be accepted if it is masked. The I3M, I2M, I1M, I0M, EM, and GM bits in the Status Register are used to mask different events. Not all events can be masked. A few critical events (NMI, Unrecoverable Exception, TLB Multiple Hit, and Bus Error) can not be masked. When an event is accepted, hardware automatically sets the mask bits corresponding to all sources with equal or lower priority. This inhibits acceptance of other events of the same or lower priority, except for the critical events listed above. Software may choose to clear some or all of these bits after saving the necessary state if other priority schemes are desired. It is the event source's responsibility to ensure that their events are left pending until accepted by the CPU.
2. When a request is accepted, the Status Register and Program Counter of the current context is stored to the system stack. If the event is an INT0, INT1, INT2, or INT3, registers R8-R12 and LR are also automatically stored to stack. Storing the Status Register ensures that the core is returned to the previous execution mode when the current event handling is completed. When exceptions occur, both the EM and GM bits are set, and the application may manually enable nested exceptions if desired by clearing the appropriate bit. Each exception handler has a dedicated handler address, and this address uniquely identifies the exception source.
3. The Mode bits are set to reflect the priority of the accepted event, and the correct register file bank is selected. The address of the event handler, as shown in Table 6-4, is loaded into the Program Counter.

The execution of the event handler routine then continues from the effective address calculated.

The *rete* instruction signals the end of the event. When encountered, the Return Status Register and Return Address Register are popped from the system stack and restored to the Status Register and Program Counter. If the *rete* instruction returns from INT0, INT1, INT2, or INT3, registers R8-R12 and LR are also popped from the system stack. The restored Status Register contains information allowing the core to resume operation in the previous execution mode. This concludes the event handling.

### 6.5.3 Supervisor Calls

The AVR32 instruction set provides a supervisor mode call instruction. The *scall* instruction is designed so that privileged routines can be called from any context. This facilitates sharing of code between different execution modes. The *scall* mechanism is designed so that a minimal execution cycle overhead is experienced when performing supervisor routine calls from time-critical event handlers.

The *scall* instruction behaves differently depending on which mode it is called from. The behaviour is detailed in the instruction set reference. In order to allow the *scall* routine to return to the correct context, a return from supervisor call instruction, *rets*, is implemented. In the AVR32UC CPU, *scall* and *rets* uses the system stack to store the return address and the status register.

### 6.5.4 Debug Requests

The AVR32 architecture defines a dedicated Debug mode. When a debug request is received by the core, Debug mode is entered. Entry into Debug mode can be masked by the DM bit in the



## 7.3 Peripheral Address Map

**Table 7-2.** Peripheral Address Mapping

Address		Peripheral Name
0xFFFE0000	USB	USB 2.0 Interface - USB
0xFFFE1000	HMATRIX	HSB Matrix - HMATRIX
0xFFFE1400	HFLASHC	Flash Controller - HFLASHC
0xFFFF0000	PDCA	Peripheral DMA Controller - PDCA
0xFFFF0800	INTC	Interrupt controller - INTC
0xFFFF0C00	PM	Power Manager - PM
0xFFFF0D00	RTC	Real Time Counter - RTC
0xFFFF0D30	WDT	Watchdog Timer - WDT
0xFFFF0D80	EIM	External Interrupt Controller - EIM
0xFFFF1000	GPIO	General Purpose Input/Output Controller - GPIO
0xFFFF1400	USART0	Universal Synchronous/Asynchronous Receiver/Transmitter - USART0
0xFFFF1800	USART1	Universal Synchronous/Asynchronous Receiver/Transmitter - USART1
0xFFFF1C00	USART2	Universal Synchronous/Asynchronous Receiver/Transmitter - USART2
0xFFFF2400	SPI0	Serial Peripheral Interface - SPI0
0xFFFF2C00	TWI	Two-wire Interface - TWI
0xFFFF3000	PWM	Pulse Width Modulation Controller - PWM
0xFFFF3400	SSC	Synchronous Serial Controller - SSC
0xFFFF3800	TC	Timer/Counter - TC

## 8. Boot Sequence

This chapter summarizes the boot sequence of the AT32UC3B. The behaviour after power-up is controlled by the Power Manager. For specific details, refer to section Power Manager (PM).

### 8.1 Starting of clocks

After power-up, the device will be held in a reset state by the Power-On Reset circuitry, until the power has stabilized throughout the device. Once the power has stabilized, the device will use the internal RC Oscillator as clock source.

On system start-up, the PLLs are disabled. All clocks to all modules are running. No clocks have a divided frequency, all parts of the system receives a clock with the same frequency as the internal RC Oscillator.

### 8.2 Fetching of initial instructions

After reset has been released, the AVR32 UC CPU starts fetching instructions from the reset address, which is 0x8000\_0000. This address points to the first address in the internal Flash.

The code read from the internal Flash is free to configure the system to use for example the PLLs, to divide the frequency of the clock routed to some of the peripherals, and to gate the clocks to unused peripherals.

When powering up the device, there may be a delay before the voltage has stabilized, depending on the rise time of the supply used. The CPU can start executing code as soon as the supply is above the POR threshold, and before the supply is stable. Before switching to a high-speed clock source, the user should use the BOD to make sure the VDDCORE is above the minimum level.



## 9.3 Regulator Characteristics

**Table 9-2.** Electrical Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
$V_{VDDIN}$	Supply voltage (input)		3	3.3	3.6	V
$V_{VDDOUT}$	Supply voltage (output)		1.70	1.8	1.85	V
$I_{OUT}$	Maximum DC output current	$V_{VDDIN} = 3.3V$			100	mA
$I_{SCR}$	Static Current of internal regulator	Low Power mode (stop, deep stop or static) at $T_A = 25^{\circ}C$		10		$\mu A$

**Table 9-3.** Decoupling Requirements

Symbol	Parameter	Conditions	Typ.	Technology	Unit
$C_{IN1}$	Input Regulator Capacitor 1		1	NPO	nF
$C_{IN2}$	Input Regulator Capacitor 2		4.7	X7R	$\mu F$
$C_{OUT1}$	Output Regulator Capacitor 1		470	NPO	pF
$C_{OUT2}$	Output Regulator Capacitor 2		2.2	X7R	$\mu F$

## 9.4 Analog Characteristics

### 9.4.1 ADC Reference

**Table 9-4.** Electrical Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
$V_{ADVREF}$	Analog voltage reference (input)		2.6		3.6	V

**Table 9-5.** Decoupling Requirements

Symbol	Parameter	Conditions	Typ.	Technology	Unit
$C_{VREF1}$	Voltage reference Capacitor 1		10	NPO	nF
$C_{VREF2}$	Voltage reference Capacitor 2		1	NPO	$\mu F$

### 9.4.2 BOD

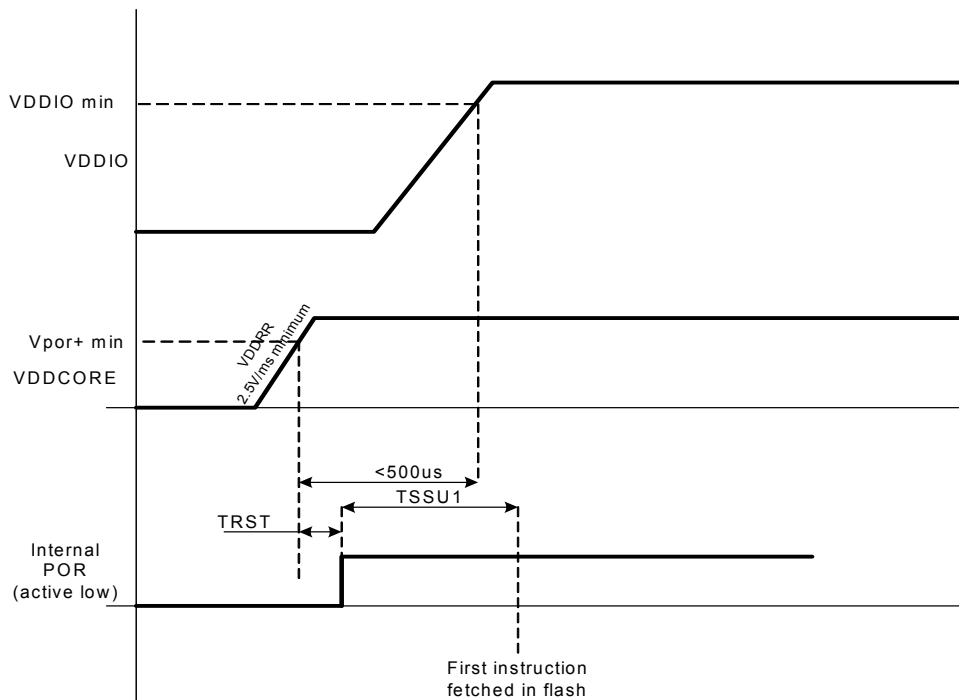
**Table 9-6.** BOD Level Values

Symbol	Parameter Value	Conditions	Min.	Typ.	Max.	Unit
BODLEVEL	00 0000b			1.44		V
	01 0111b			1.52		V
	01 1111b			1.61		V
	10 0111b			1.71		V

Table 9-6 describes the values of the BODLEVEL field in the flash FGPFRR register.

Therefore VDDCORE rise rate (VDDRR) must be equal or superior to 2.5V/ms and VDDIO must reach VDDIO min value before 500 us ( $< TRST + TSSU1$ ) after VDDCORE has reached  $V_{POR+}$  min value.

**Figure 9-4.** Dual Supply Configuration



## 9.4.4 RESET\_N Characteristics

**Table 9-9.** RESET\_N Waveform Parameters

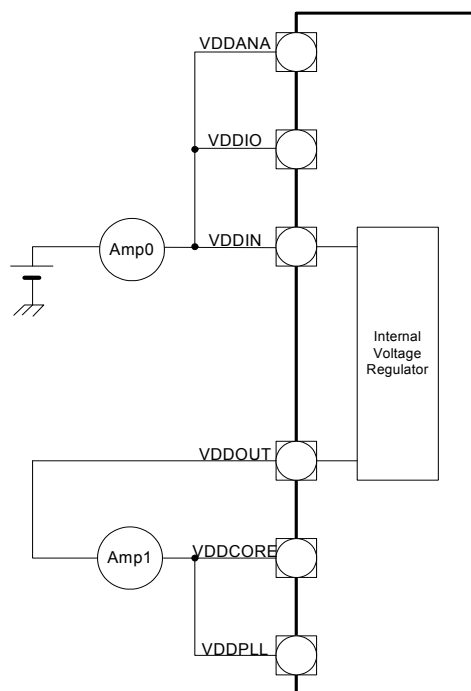
Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
$t_{RESET}$	RESET_N minimum pulse width		10			ns

## 9.5 Power Consumption

The values in [Table 9-10](#), [Table 9-11 on page 43](#) and [Table 9-12 on page 44](#) are measured values of power consumption with operating conditions as follows:

- $V_{DDIO} = V_{DDANA} = 3.3V$
- $V_{DDCORE} = V_{DDPLL} = 1.8V$
- $T_A = 25^{\circ}C$ ,  $T_A = 85^{\circ}C$
- I/Os are configured in input, pull-up enabled.

**Figure 9-5.** Measurement Setup



The following tables represent the power consumption measured on the power supplies.

## 9.5.1 Power Consumption for Different Sleep Modes

**Table 9-10.** Power Consumption for Different Sleep Modes for AT32UC3B064, AT32UC3B0128, AT32UC3B0256, AT32UC3B164, AT32UC3B1128, AT32UC3B1256

Mode	Conditions	Typ.	Unit
Active	<ul style="list-style-type: none"> <li>- CPU running a recursive Fibonacci Algorithm from flash and clocked from PLL0 at f MHz.</li> <li>- Voltage regulator is on.</li> <li>- XIN0: external clock. Xin1 Stopped. XIN32 stopped.</li> <li>- All peripheral clocks activated with a division by 8.</li> <li>- GPIOs are inactive with internal pull-up, JTAG unconnected with external pull-up and Input pins are connected to GND</li> </ul>	$0.3 \times f(\text{MHz}) + 0.443$	mA/MHz
	Same conditions at 60 MHz	18.5	mA
Idle	See Active mode conditions	$0.117 \times f(\text{MHz}) + 0.28$	mA/MHz
	Same conditions at 60 MHz	7.3	mA
Frozen	See Active mode conditions	$0.058 \times f(\text{MHz}) + 0.115$	mA/MHz
	Same conditions at 60 MHz	3.6	mA
Standby	See Active mode conditions	$0.042 \times f(\text{MHz}) + 0.115$	mA/MHz
	Same conditions at 60 MHz	2.7	mA
Stop	<ul style="list-style-type: none"> <li>- CPU running in sleep mode</li> <li>- XIN0, Xin1 and XIN32 are stopped.</li> <li>- All peripheral clocks are deactivated.</li> <li>- GPIOs are inactive with internal pull-up, JTAG unconnected with external pull-up and Input pins are connected to GND.</li> </ul>	37.8	μA
Deepstop	See Stop mode conditions	24.9	μA
Static	See Stop mode conditions	Voltage Regulator On	13.9
		Voltage Regulator Off	8.9

Notes: 1. Core frequency is generated from XIN0 using the PLL so that  $140 \text{ MHz} < f_{\text{PLL0}} < 160 \text{ MHz}$  and  $10 \text{ MHz} < f_{\text{XIN0}} < 12 \text{ MHz}$ .

**Table 9-11.** Power Consumption for Different Sleep Modes for AT32UC3B0512, AT32UC3B1512

Mode	Conditions	Typ.	Unit
Active	<ul style="list-style-type: none"> <li>- CPU running a recursive Fibonacci Algorithm from flash and clocked from PLL0 at f MHz.</li> <li>- Voltage regulator is on.</li> <li>- XIN0: external clock. Xin1 Stopped. XIN32 stopped.</li> <li>- All peripheral clocks activated with a division by 8.</li> <li>- GPIOs are inactive with internal pull-up, JTAG unconnected with external pull-up and Input pins are connected to GND</li> </ul>	$0.359 \times f(\text{MHz}) + 1.53$	mA/MHz
	Same conditions at 60 MHz	24	mA
Idle	See Active mode conditions	$0.146 \times f(\text{MHz}) + 0.291$	mA/MHz
	Same conditions at 60 MHz	9	mA

**Table 9-11.** Power Consumption for Different Sleep Modes for AT32UC3B0512, AT32UC3B1512

Mode	Conditions	Typ.	Unit
Frozen	See Active mode conditions	$0.0723 \times f(\text{MHz}) + 0.156$	mA/MHz
	Same conditions at 60 MHz	4.5	mA
Standby	See Active mode conditions	$0.0537 \times f(\text{MHz}) + 0.166$	mA/MHz
	Same conditions at 60 MHz	3.4	mA
Stop	<ul style="list-style-type: none"> <li>- CPU running in sleep mode</li> <li>- XIN0, Xin1 and XIN32 are stopped.</li> <li>- All peripheral clocks are desactivated.</li> <li>- GPIOs are inactive with internal pull-up, JTAG unconnected with external pull-up and Input pins are connected to GND.</li> </ul>	62	μA
Deepstop	See Stop mode conditions	30	μA
Static	See Stop mode conditions	Voltage Regulator On	15.5
		Voltage Regulator Off	7.5

Notes: 1. Core frequency is generated from XIN0 using the PLL so that  $140 \text{ MHz} < f_{\text{PLL0}} < 160 \text{ MHz}$  and  $10 \text{ MHz} < f_{\text{XIN0}} < 12 \text{ MHz}$ .

**Table 9-12.** Peripheral Interface Power Consumption in Active Mode

Peripheral	Conditions	Consumption	Unit
INTC	AT32UC3B064 AT32UC3B0128 AT32UC3B0256 AT32UC3B164 AT32UC3B1128 AT32UC3B1256 AT32UC3B0512 AT32UC3B1512	20	μA/MHz
GPIO		16	
PDCA		12	
USART		14	
USB		23	
ADC		8	
TWI		7	
PWM		18	
SPI		8	
SSC		11	
TC		11	
ABDAC	AT32UC3B0512 AT32UC3B1512	6	

## 9.9 USB Transceiver Characteristics

### 9.9.1 Electrical Characteristics

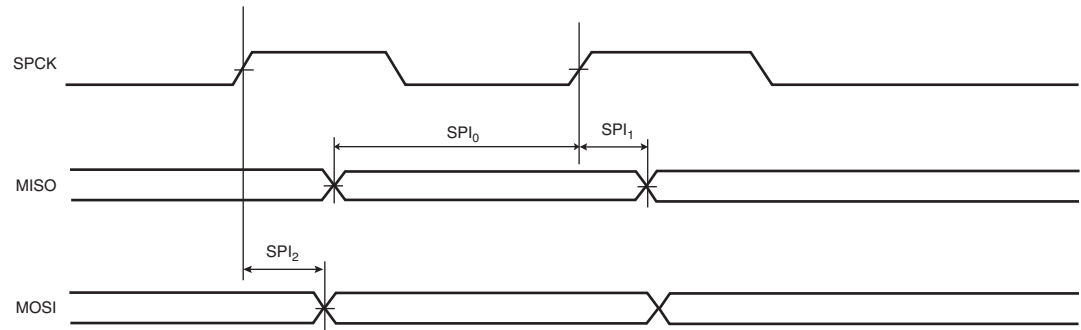
**Table 9-25.** Electrical Parameters

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
$R_{EXT}$	Recommended external USB series resistor	In series with each USB pin with $\pm 5\%$		39		$\Omega$

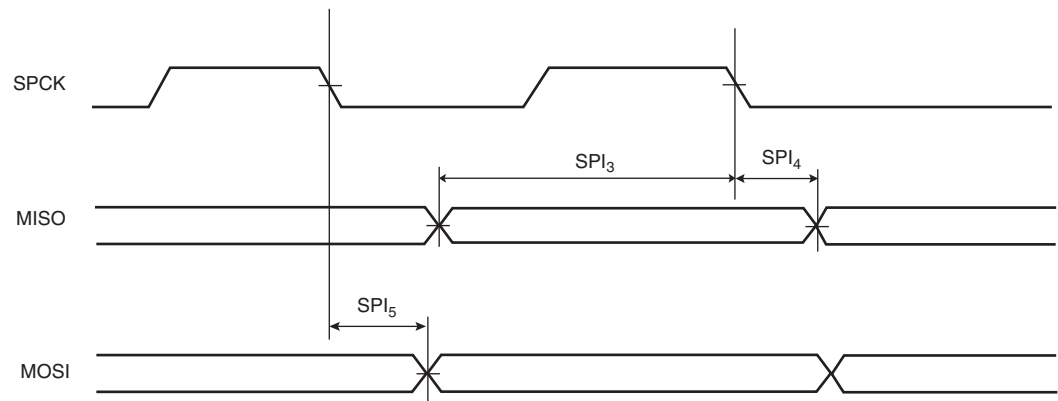
The USB on-chip buffers comply with the Universal Serial Bus (USB) v2.0 standard. All AC parameters related to these buffers can be found within the USB 2.0 electrical specifications.

## 9.11 SPI Characteristics

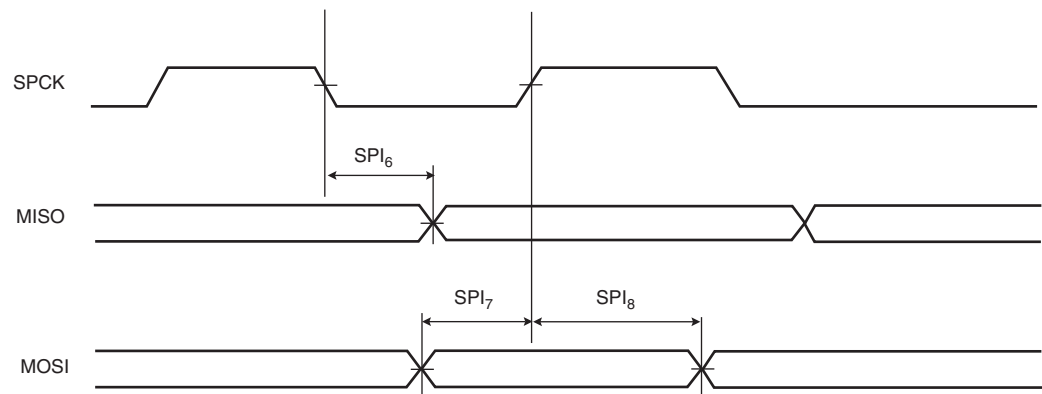
**Figure 9-7.** SPI Master mode with (CPOL = NCPHA = 0) or (CPOL= NCPHA= 1)



**Figure 9-8.** SPI Master mode with (CPOL=0 and NCPHA=1) or (CPOL=1 and NCPHA=0)

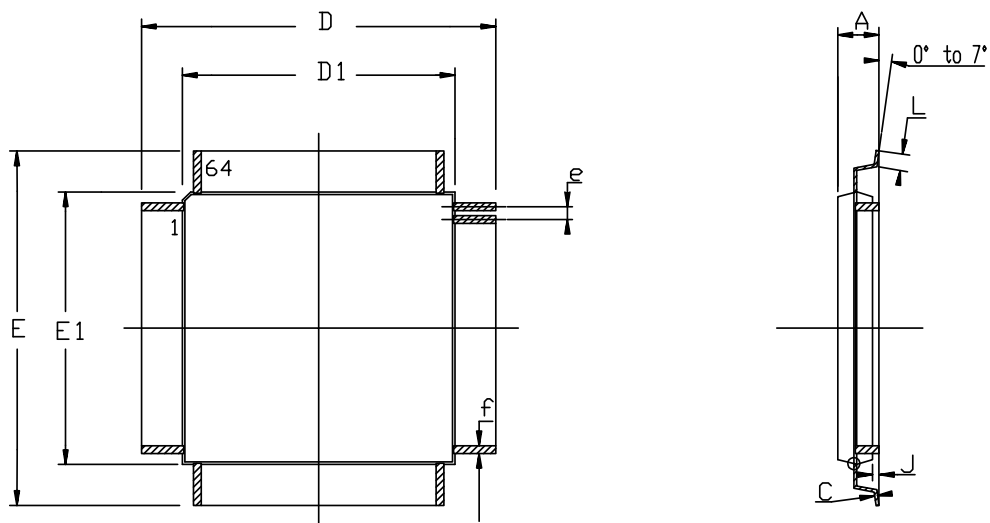


**Figure 9-9.** SPI Slave mode with (CPOL=0 and NCPHA=1) or (CPOL=1 and NCPHA=0)



## 10.2 Package Drawings

Figure 10-1. TQFP-64 package drawing



COMMON DIMENSIONS IN MM

SYMBOL	Min	Max	NOTES
A	----	1.20	
A1	0.95	1.05	
C	0.09	0.20	
D	12.00 BSC		
D1	10.00 BSC		
E	12.00 BSC		
E1	10.00 BSC		
J	0.05	0.15	
L	0.45	0.75	
e	0.50 BSC		
f	0.17	0.27	

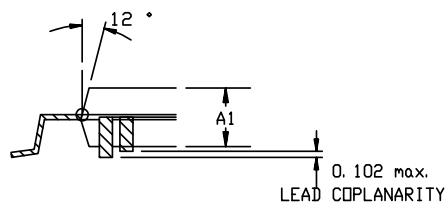


Table 10-2. Device and Package Maximum Weight

Weight	300 mg
--------	--------

Table 10-3. Package Characteristics

Moisture Sensitivity Level	Jedec J-STD-20D-MSL3
----------------------------	----------------------

Table 10-4. Package Reference

JEDEC Drawing Reference	MS-026
JESD97 Classification	e3



the receive buffer is full. In the interrupt handler code, write a one to the RTSDIS bit in the USART Control Register (CR). This will drive the RTS output high. After the next DMA transfer is started and a receive buffer is available, write a one to the RTSEN bit in the USART CR so that RTS will be driven low.

**8. Corruption after receiving too many bits in SPI slave mode**

If the USART is in SPI slave mode and receives too much data bits (ex: 9bits instead of 8 bits) by the SPI master, an error occurs. After that, the next reception may be corrupted even if the frame is correct and the USART has been disabled, reset by a soft reset and re-enabled.

**Fix/Workaround**

None.

**9. USART slave synchronous mode external clock must be at least 9 times lower in frequency than CLK\_USART**

When the USART is operating in slave synchronous mode with an external clock, the frequency of the signal provided on CLK must be at least 9 times lower than CLK\_USART.

**Fix/Workaround**

When the USART is operating in slave synchronous mode with an external clock, provide a signal on CLK that has a frequency at least 9 times lower than CLK\_USART.

**10. HMATRIX**

**11. In the PRAS and PRBS registers, the MxPR fields are only two bits**

In the PRAS and PRBS registers, the MxPR fields are only two bits wide, instead of four bits. The unused bits are undefined when reading the registers.

**Fix/Workaround**

Mask undefined bits when reading PRAS and PRBS.

*- DSP Operations*

**1. Hardware breakpoints may corrupt MAC results**

Hardware breakpoints on MAC instructions may corrupt the destination register of the MAC instruction.

**Fix/Workaround**

Place breakpoints on earlier or later instructions.

**8. SPI disable does not work in SLAVE mode**

SPI disable does not work in SLAVE mode.

**Fix/Workaround**

Read the last received data, then perform a software reset by writing a one to the Software Reset bit in the Control Register (CR.SWRST).

**9. SPI data transfer hangs with CSR0.CSAAT==1 and MR.MODFDIS==0**

When CSR0.CSAAT==1 and mode fault detection is enabled (MR.MODFDIS==0), the SPI module will not start a data transfer.

**Fix/Workaround**

Disable mode fault detection by writing a one to MR.MODFDIS.

**10. Disabling SPI has no effect on the SR.TDRE bit**

Disabling SPI has no effect on the SR.TDRE bit whereas the write data command is filtered when SPI is disabled. Writing to TDR when SPI is disabled will not clear SR.TDRE. If SPI is disabled during a PDCA transfer, the PDCA will continue to write data to TDR until its buffer is empty, and this data will be lost.

**Fix/Workaround**

Disable the PDCA, add two NOPs, and disable the SPI. To continue the transfer, enable the SPI and PDCA.

**11. Power Manager**

**12. If the BOD level is higher than VDDCORE, the part is constantly reset**

If the BOD level is set to a value higher than VDDCORE and enabled by fuses, the part will be in constant reset.

**Fix/Workaround**

Apply an external voltage on VDDCORE that is higher than the BOD level and is lower than VDDCORE max and disable the BOD.

**2. When the main clock is RCSYS, TIMER\_CLOCK5 is equal to PBA clock**

When the main clock is generated from RCSYS, TIMER\_CLOCK5 is equal to PBA Clock and not PBA Clock / 128.

**Fix/Workaround**

None.

**13. Clock sources will not be stopped in STATIC sleep mode if the difference between CPU and PBx division factor is too high**

If the division factor between the CPU/HSB and PBx frequencies is more than 4 when going to a sleep mode where the system RC oscillator is turned off, then high speed clock sources will not be turned off. This will result in a significantly higher power consumption during the sleep mode.

**Fix/Workaround**

Before going to sleep modes where the system RC oscillator is stopped, make sure that the factor between the CPU/HSB and PBx frequencies is less than or equal to 4.

**14. Increased Power Consumption in VDDIO in sleep modes**

If the OSC0 is enabled in crystal mode when entering a sleep mode where the OSC0 is disabled, this will lead to an increased power consumption in VDDIO.

**Fix/Workaround**

Disable the OSC0 through the System Control Interface (SCIF) before going to any sleep mode where the OSC0 is disabled, or pull down or up XIN0 and XOUT0 with 1 Mohm resistor.

- *DSP Operations*

**1. Hardware breakpoints may corrupt MAC results**

Hardware breakpoints on MAC instructions may corrupt the destination register of the MAC instruction.

**Fix/Workaround**

Place breakpoints on earlier or later instructions.

- *USART*

1. **USART Manchester Encoder Not Working**  
Manchester encoding/decoding is not working.  
**Fix/Workaround**  
Do not use manchester encoding.
2. **USART RXBREAK problem when no timeguard**  
In asynchronous mode the RXBREAK flag is not correctly handled when the timeguard is 0 and the break character is located just after the stop bit.  
**Fix/Workaround**  
If the NBSTOP is 1, timeguard should be different from 0.
3. **USART Handshaking: 2 characters sent / CTS rises when TX**  
If CTS switches from 0 to 1 during the TX of a character, if the Holding register is not empty, the TXHOLDING is also transmitted.  
**Fix/Workaround**  
None.
4. **USART PDC and TIMEGUARD not supported in MANCHESTER**  
Manchester encoding/decoding is not working.  
**Fix/Workaround**  
Do not use manchester encoding.
5. **USART SPI mode is non functional on this revision**  
USART SPI mode is non functional on this revision.  
**Fix/Workaround**  
Do not use the USART SPI mode.

- *HMATRIX*

1. **HMatrix fixed priority arbitration does not work**  
Fixed priority arbitration does not work.  
**Fix/Workaround**  
Use Round-Robin arbitration instead.

- *Clock characteristic*

1. **PBA max frequency**  
The Peripheral bus A (PBA) max frequency is 30MHz instead of 60MHz.  
**Fix/Workaround**  
Do not set the PBA maximum frequency higher than 30MHz.

- *FLASHC*

1. **The address of Flash General Purpose Fuse Register Low (FGPFRLO) is 0xFFFE140C on revB instead of 0xFFFE1410**  
The address of Flash General Purpose Fuse Register Low (FGPFRLO) is 0xFFFE140C on revB instead of 0xFFFE1410.  
**Fix/Workaround**  
None.

it is done atomically. Even if this step is described in general as not safe in the UC technical reference guide, it is safe in this very specific case.

2. Execute the RETE instruction.