

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 ² C, IrDA, SPI, SSC, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	28
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	1.65V ~ 3.6V
Data Converters	A/D 6x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-VQFN Exposed Pad
Supplier Device Package	48-QFN (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/atmel/at32uc3b164-z1ut

Table 5-1. Signal Description List (Continued)

Signal Name	Function	Type	Active Level	Comments
VDDOUT	Voltage Regulator Output	Power Output		1.65V to 1.95 V
GNDANA	Analog Ground	Ground		
GND	Ground	Ground		
Clocks, Oscillators, and PLL's				
XIN0, XIN1, XIN32	Crystal 0, 1, 32 Input	Analog		
XOUT0, XOUT1, XOUT32	Crystal 0, 1, 32 Output	Analog		
JTAG				
TCK	Test Clock	Input		
TDI	Test Data In	Input		
TDO	Test Data Out	Output		
TMS	Test Mode Select	Input		
Auxiliary Port - AUX				
MCKO	Trace Data Output Clock	Output		
MDO0 - MDO5	Trace Data Output	Output		
MSEO0 - MSEO1	Trace Frame Control	Output		
EVTI_N	Event In	Output	Low	
EVTO_N	Event Out	Output	Low	
Power Manager - PM				
GCLK0 - GCLK2	Generic Clock Pins	Output		
RESET_N	Reset Pin	Input	Low	
External Interrupt Controller - EIC				
EXTINT0 - EXTINT7	External Interrupt Pins	Input		
KPS0 - KPS7	Keypad Scan Pins	Output		
NMI	Non-Maskable Interrupt Pin	Input	Low	
General Purpose I/O pin- GPIOA, GPIOB				
PA0 - PA31	Parallel I/O Controller GPIOA	I/O		
PB0 - PB11	Parallel I/O Controller GPIOB	I/O		

Table 5-1. Signal Description List (Continued)

Signal Name	Function	Type	Active Level	Comments
Serial Peripheral Interface - SPI0				
MISO	Master In Slave Out	I/O		
MOSI	Master Out Slave In	I/O		
NPCS0 - NPCS3	SPI Peripheral Chip Select	I/O	Low	
SCK	Clock	Output		
Synchronous Serial Controller - SSC				
RX_CLOCK	SSC Receive Clock	I/O		
RX_DATA	SSC Receive Data	Input		
RX_FRAME_SYNC	SSC Receive Frame Sync	I/O		
TX_CLOCK	SSC Transmit Clock	I/O		
TX_DATA	SSC Transmit Data	Output		
TX_FRAME_SYNC	SSC Transmit Frame Sync	I/O		
Timer/Counter - TIMER				
A0	Channel 0 Line A	I/O		
A1	Channel 1 Line A	I/O		
A2	Channel 2 Line A	I/O		
B0	Channel 0 Line B	I/O		
B1	Channel 1 Line B	I/O		
B2	Channel 2 Line B	I/O		
CLK0	Channel 0 External Clock Input	Input		
CLK1	Channel 1 External Clock Input	Input		
CLK2	Channel 2 External Clock Input	Input		
Two-wire Interface - TWI				
SCL	Serial Clock	I/O		
SDA	Serial Data	I/O		
Universal Synchronous Asynchronous Receiver Transmitter - USART0, USART1, USART2				
CLK	Clock	I/O		
CTS	Clear To Send	Input		

Table 5-1. Signal Description List (Continued)

Signal Name	Function	Type	Active Level	Comments
DCD	Data Carrier Detect			Only USART1
DSR	Data Set Ready			Only USART1
DTR	Data Terminal Ready			Only USART1
RI	Ring Indicator			Only USART1
RTS	Request To Send	Output		
RXD	Receive Data	Input		
TXD	Transmit Data	Output		
Analog to Digital Converter - ADC				
AD0 - AD7	Analog input pins	Analog input		
ADVREF	Analog positive reference voltage input	Analog input		2.6 to 3.6V
Audio Bitstream DAC - ABDAC				
DATA0 - DATA1	D/A Data out	Output		
DATAN0 - DATAN1	D/A Data inverted out	Output		
Pulse Width Modulator - PWM				
PWM0 - PWM6	PWM Output Pins	Output		
Universal Serial Bus Device - USBB				
DDM	USB Device Port Data -	Analog		
DDP	USB Device Port Data +	Analog		
VBUS	USB VBUS Monitor and Embedded Host Negotiation	Analog Input		
USBID	ID Pin of the USB Bus	Input		
USB_VBOF	USB VBUS On/off: bus power control port	output		

5.1 JTAG pins

TMS and TDI pins have pull-up resistors. TDO pin is an output, driven at up to VDDIO, and has no pull-up resistor. These 3 pins can be used as GPIO-pins. At reset state, these pins are in GPIO mode.

TCK pin cannot be used as GPIO pin. JTAG interface is enabled when TCK pin is tied low.

Refer to [Section 9.3 on page 38](#) for decoupling capacitors values and regulator characteristics.

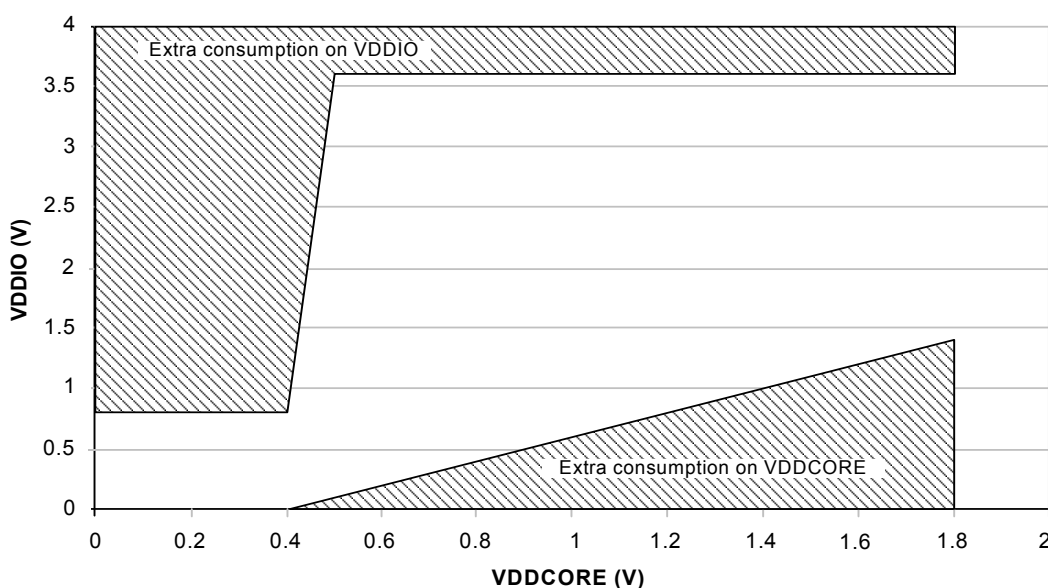
For decoupling recommendations for VDDIO, VDDANA, VDDCORE and VDDPLL, please refer to the Schematic checklist.

5.6.2.2 Dual Power Supply

In case of dual power supply, VDDIN and VDDOUT should be connected to ground to prevent from leakage current.

To avoid over consumption during the power up sequence, VDDIO and VDDCORE voltage difference needs to stay in the range given [Figure 5-3](#).

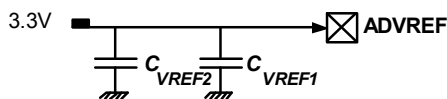
Figure 5-3. VDDIO versus VDDCORE during power up sequence



5.6.3 Analog-to-Digital Converter (ADC) reference.

The ADC reference (ADVREF) must be provided from an external source. Two decoupling capacitors must be used to insure proper decoupling.

Figure 5-4. ADVREF Decoupling



Refer to [Section 9.4 on page 38](#) for decoupling capacitors values and electrical characteristics.

In case ADC is not used, the ADVREF pin should be connected to GND to avoid extra consumption.

The following table shows the instructions with support for unaligned addresses. All other instructions require aligned addresses.

Table 6-1. Instructions with Unaligned Reference Support

Instruction	Supported alignment
ld.d	Word
st.d	Word

6.3.6 Unimplemented Instructions

The following instructions are unimplemented in AVR32UC, and will cause an Unimplemented Instruction Exception if executed:

- All SIMD instructions
- All coprocessor instructions if no coprocessors are present
- retj, incjosp, popjc, pushjc
- tlbr, tlbs, tlbw
- cache

6.3.7 CPU and Architecture Revision

Three major revisions of the AVR32UC CPU currently exist.

The Architecture Revision field in the CONFIG0 system register identifies which architecture revision is implemented in a specific device.

AVR32UC CPU revision 3 is fully backward-compatible with revisions 1 and 2, ie. code compiled for revision 1 or 2 is binary-compatible with revision 3 CPUs.

Figure 6-5. The Status Register Low Halfword



6.4.3 Processor States

6.4.3.1 Normal RISC State

The AVR32 processor supports several different execution contexts as shown in [Table 6-2 on page 23](#).

Table 6-2. Overview of Execution Modes, their Priorities and Privilege Levels.

Priority	Mode	Security	Description
1	Non Maskable Interrupt	Privileged	Non Maskable high priority interrupt mode
2	Exception	Privileged	Execute exceptions
3	Interrupt 3	Privileged	General purpose interrupt mode
4	Interrupt 2	Privileged	General purpose interrupt mode
5	Interrupt 1	Privileged	General purpose interrupt mode
6	Interrupt 0	Privileged	General purpose interrupt mode
N/A	Supervisor	Privileged	Runs supervisor calls
N/A	Application	Unprivileged	Normal program execution mode

Mode changes can be made under software control, or can be caused by external interrupts or exception processing. A mode can be interrupted by a higher priority mode, but never by one with lower priority. Nested exceptions can be supported with a minimal software overhead.

When running an operating system on the AVR32, user processes will typically execute in the application mode. The programs executed in this mode are restricted from executing certain instructions. Furthermore, most system registers together with the upper halfword of the status register cannot be accessed. Protected memory areas are also not available. All other operating modes are privileged and are collectively called System Modes. They have full access to all privileged and unprivileged resources. After a reset, the processor will be in supervisor mode.

6.4.3.2 Debug State

The AVR32 can be set in a debug state, which allows implementation of software monitor routines that can read out and alter system information for use during application development. This implies that all system and application registers, including the status registers and program counters, are accessible in debug state. The privileged instructions are also available.

Table 6-3. System Registers (Continued)

Reg #	Address	Name	Function
92	368	MPUPSR4	MPU Privilege Select Register region 4
93	372	MPUPSR5	MPU Privilege Select Register region 5
94	376	MPUPSR6	MPU Privilege Select Register region 6
95	380	MPUPSR7	MPU Privilege Select Register region 7
96	384	MPUCRA	Unused in this version of AVR32UC
97	388	MPUCRB	Unused in this version of AVR32UC
98	392	MPUBRA	Unused in this version of AVR32UC
99	396	MPUBRB	Unused in this version of AVR32UC
100	400	MPUAPRA	MPU Access Permission Register A
101	404	MPUAPRB	MPU Access Permission Register B
102	408	MPUCR	MPU Control Register
103-191	448-764	Reserved	Reserved for future use
192-255	768-1020	IMPL	IMPLEMENTATION DEFINED

6.5 Exceptions and Interrupts

AVR32UC incorporates a powerful exception handling scheme. The different exception sources, like Illegal Op-code and external interrupt requests, have different priority levels, ensuring a well-defined behavior when multiple exceptions are received simultaneously. Additionally, pending exceptions of a higher priority class may preempt handling of ongoing exceptions of a lower priority class.

When an event occurs, the execution of the instruction stream is halted, and execution control is passed to an event handler at an address specified in [Table 6-4 on page 29](#). Most of the handlers are placed sequentially in the code space starting at the address specified by EVBA, with four bytes between each handler. This gives ample space for a jump instruction to be placed there, jumping to the event routine itself. A few critical handlers have larger spacing between them, allowing the entire event routine to be placed directly at the address specified by the EVBA-relative offset generated by hardware. All external interrupt sources have autovector interrupt service routine (ISR) addresses. This allows the interrupt controller to directly specify the ISR address as an address relative to EVBA. The autovector offset has 14 address bits, giving an offset of maximum 16384 bytes. The target address of the event handler is calculated as $(EVBA \mid event_handler_offset)$, not $(EVBA + event_handler_offset)$, so EVBA and exception code segments must be set up appropriately. The same mechanisms are used to service all different types of events, including external interrupt requests, yielding a uniform event handling scheme.

An interrupt controller does the priority handling of the external interrupts and provides the autovector offset to the CPU.

6.5.1 System Stack Issues

Event handling in AVR32UC uses the system stack pointed to by the system stack pointer, SP_SYS, for pushing and popping R8-R12, LR, status register, and return address. Since event code may be timing-critical, SP_SYS should point to memory addresses in the IRAM section, since the timing of accesses to this memory section is both fast and deterministic.

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. Electrical Characteristics

9.1 Absolute Maximum Ratings*

Operating Temperature.....	-40°C to +85°C
Storage Temperature	-60°C to +150°C
Voltage on GPIO Pins with respect to Ground for TCK, RESET_N, PA03, PA04, PA05, PA06, PA07, PA08, PA11, PA12, PA18, PA19, PA28, PA29, PA30, PA31	-0.3 to 3.6V
Voltage on GPIO Pins with respect to Ground except for TCK, RESET_N, PA03, PA04, PA05, PA06, PA07, PA08, PA11, PA12, PA18, PA19, PA28, PA29, PA30, PA31.....	-0.3 to 5.5V
Maximum Operating Voltage (VDDCORE, VDDPLL)	1.95V
Maximum Operating Voltage (VDDIO,VDDIN,VDDANA) .	3.6V
Total DC Output Current on all I/O Pin	
for 48-pin package	200 mA
for 64-pin package	265 mA

*NOTICE: Stresses beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Figure 9-1. MCU Cold Start-Up RESET_N tied to VDDIN

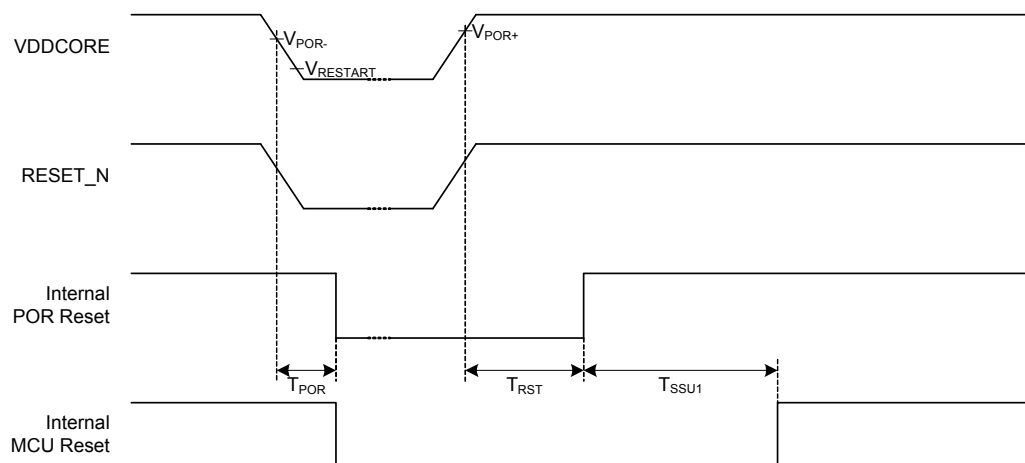


Figure 9-2. MCU Cold Start-Up RESET_N Externally Driven

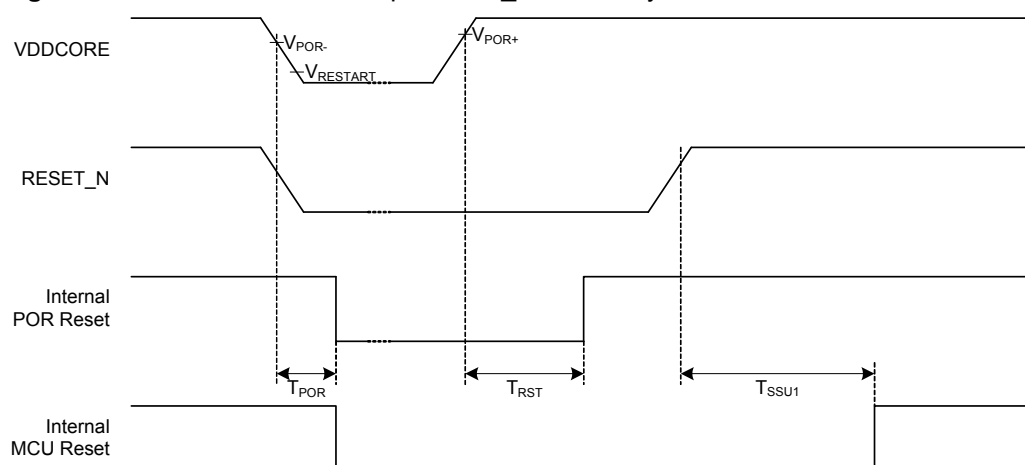
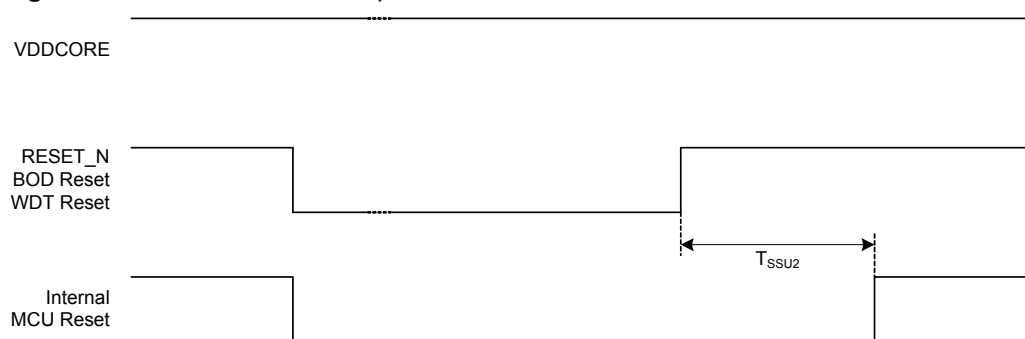


Figure 9-3. MCU Hot Start-Up



In dual supply configuration, the power up sequence must be carefully managed to ensure a safe startup of the device in all conditions.

The power up sequence must ensure that the internal logic is safely powered when the internal reset (Power On Reset) is released and that the internal Flash logic is safely powered when the CPU fetch the first instructions.

9.6 System Clock Characteristics

These parameters are given in the following conditions:

- $V_{DDCORE} = 1.8V$
- Ambient Temperature = 25°C

9.6.1 CPU/HSB Clock Characteristics

Table 9-13. Core Clock Waveform Parameters

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
$1/(t_{CPCPU})$	CPU Clock Frequency				60	MHz
t_{CPCPU}	CPU Clock Period		16.6			ns

9.6.2 PBA Clock Characteristics

Table 9-14. PBA Clock Waveform Parameters

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
$1/(t_{CPPBA})$	PBA Clock Frequency				60	MHz
t_{CPPBA}	PBA Clock Period		16.6			ns

9.6.3 PBB Clock Characteristics

Table 9-15. PBB Clock Waveform Parameters

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
$1/(t_{CPPBB})$	PBB Clock Frequency				60	MHz
t_{CPPBB}	PBB Clock Period		16.6			ns

Figure 10-4. QFN-48 package drawing

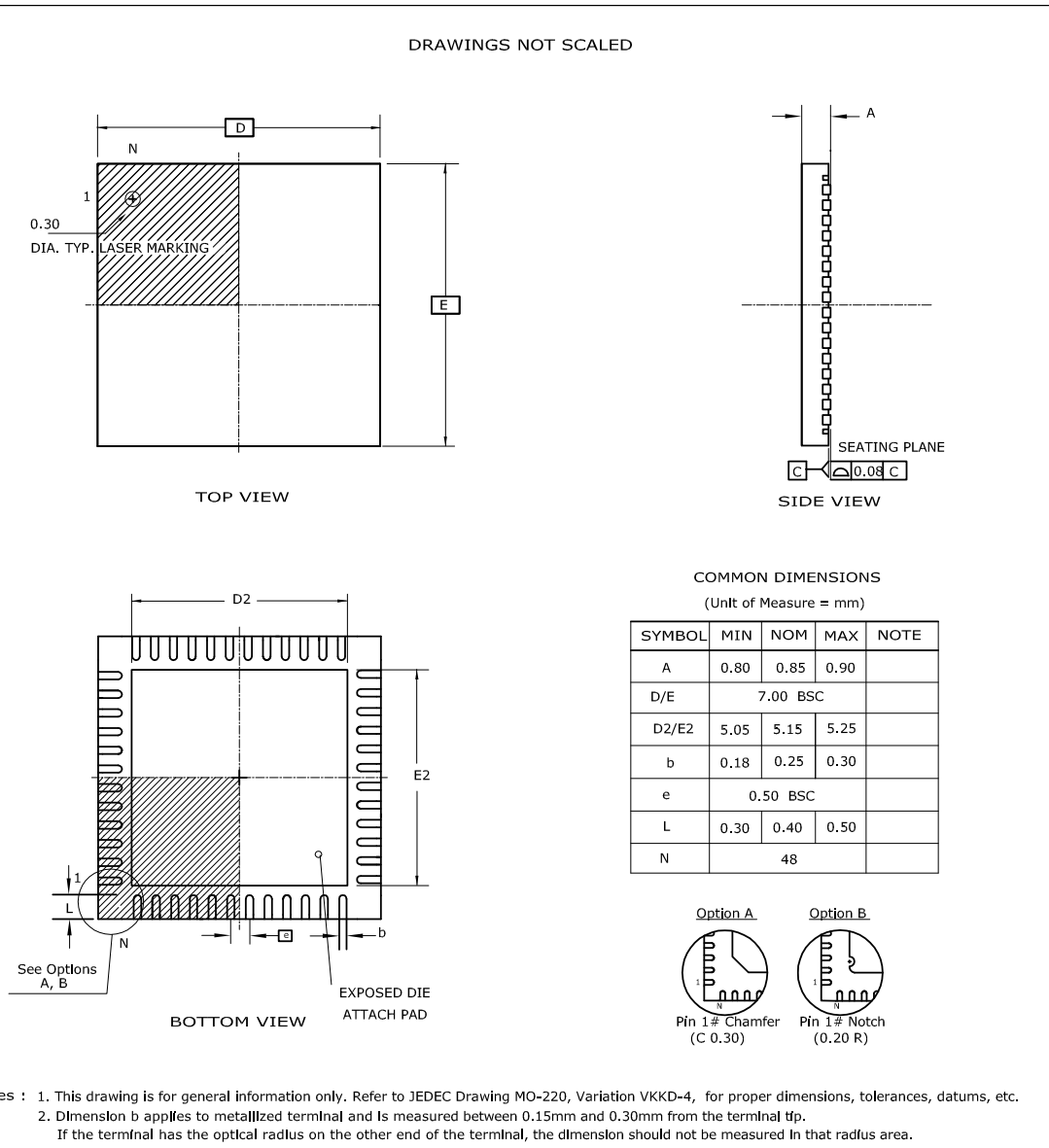


Table 10-11. Device and Package Maximum Weight

Weight	100 mg
--------	--------

Table 10-12. Package Characteristics

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

Table 10-13. Package Reference

JEDEC Drawing Reference	M0-220
JESD97 Classification	e3

11. Ordering Information

Device	Ordering Code	Package	Conditioning	Temperature Operating Range
AT32UC3B0512	AT32UC3B0512-A2UES	TQFP 64	-	Industrial (-40°C to 85°C)
	AT32UC3B0512-A2UR	TQFP 64	Reel	Industrial (-40°C to 85°C)
	AT32UC3B0512-A2UT	TQFP 64	Tray	Industrial (-40°C to 85°C)
	AT32UC3B0512-Z2UES	QFN 64	-	Industrial (-40°C to 85°C)
	AT32UC3B0512-Z2UR	QFN 64	Reel	Industrial (-40°C to 85°C)
	AT32UC3B0512-Z2UT	QFN 64	Tray	Industrial (-40°C to 85°C)
AT32UC3B0256	AT32UC3B0256-A2UT	TQFP 64	Tray	Industrial (-40°C to 85°C)
	AT32UC3B0256-A2UR	TQFP 64	Reel	Industrial (-40°C to 85°C)
	AT32UC3B0256-Z2UT	QFN 64	Tray	Industrial (-40°C to 85°C)
	AT32UC3B0256-Z2UR	QFN 64	Reel	Industrial (-40°C to 85°C)
AT32UC3B0128	AT32UC3B0128-A2UT	TQFP 64	Tray	Industrial (-40°C to 85°C)
	AT32UC3B0128-A2UR	TQFP 64	Reel	Industrial (-40°C to 85°C)
	AT32UC3B0128-Z2UT	QFN 64	Tray	Industrial (-40°C to 85°C)
	AT32UC3B0128-Z2UR	QFN 64	Reel	Industrial (-40°C to 85°C)
AT32UC3B064	AT32UC3B064-A2UT	TQFP 64	Tray	Industrial (-40°C to 85°C)
	AT32UC3B064-A2UR	TQFP 64	Reel	Industrial (-40°C to 85°C)
	AT32UC3B064-Z2UT	QFN 64	Tray	Industrial (-40°C to 85°C)
	AT32UC3B064-Z2UR	QFN 64	Reel	Industrial (-40°C to 85°C)
AT32UC3B1512	AT32UC3B1512-Z1UT	QFN 48	-	Industrial (-40°C to 85°C)
	AT32UC3B1512-Z1UR	QFN 48	-	Industrial (-40°C to 85°C)
AT32UC3B1256	AT32UC3B1256-AUT	TQFP 48	Tray	Industrial (-40°C to 85°C)
	AT32UC3B1256-AUR	TQFP 48	Reel	Industrial (-40°C to 85°C)
	AT32UC3B1256-Z1UT	QFN 48	Tray	Industrial (-40°C to 85°C)
	AT32UC3B1256-Z1UR	QFN 48	Reel	Industrial (-40°C to 85°C)
AT32UC3B1128	AT32UC3B1128-AUT	TQFP 48	Tray	Industrial (-40°C to 85°C)
	AT32UC3B1128-AUR	TQFP 48	Reel	Industrial (-40°C to 85°C)
	AT32UC3B1128-Z1UT	QFN 48	Tray	Industrial (-40°C to 85°C)
	AT32UC3B1128-Z1UR	QFN 48	Reel	Industrial (-40°C to 85°C)
AT32UC3B164	AT32UC3B164-AUT	TQFP 48	Tray	Industrial (-40°C to 85°C)
	AT32UC3B164-AUR	TQFP 48	Reel	Industrial (-40°C to 85°C)
	AT32UC3B164-Z1UT	QFN 48	Tray	Industrial (-40°C to 85°C)
	AT32UC3B164-Z1UR	QFN 48	Reel	Industrial (-40°C to 85°C)

7. SPI Glitch on RXREADY flag in slave mode when enabling the SPI or during the first transfer

In slave mode, the SPI can generate a false RXREADY signal during enabling of the SPI or during the first transfer.

Fix/Workaround

1. Set slave mode, set required CPOL/CPHA.
2. Enable SPI.
3. Set the polarity CPOL of the line in the opposite value of the required one.
4. Set the polarity CPOL to the required one.
5. Read the RXHOLDING register.

Transfers can now begin and RXREADY will now behave as expected.

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.

13. 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.

14. 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

16. 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 Power Manager (PM) before going to any sleep mode where the OSC0 is disabled, or pull down or up XIN0 and XOUT0 with 1Mohm resistor.

17. SSC

18. Additional delay on TD output

A delay from 2 to 3 system clock cycles is added to TD output when:

TCMR.START = Receive Start,

TCMR.STTDLY = more than ZERO,

RCMR.START = Start on falling edge / Start on Rising edge / Start on any edge,

RFMR.FSOS = None (input).

Fix/Workaround

None.

19. TF output is not correct

TF output is not correct (at least emitted one serial clock cycle later than expected) when:

TFMR.FSOS = Driven Low during data transfer/ Driven High during data transfer

TCMR.START = Receive start

RFMR.FSOS = None (Input)

RCMR.START = any on RF (edge/level)

Fix/Workaround

None.

20. Frame Synchro and Frame Synchro Data are delayed by one clock cycle

The frame synchro and the frame synchro data are delayed from 1 SSC_CLOCK when:

- Clock is CKDIV

- The START is selected on either a frame synchro edge or a level

- Frame synchro data is enabled

- Transmit clock is gated on output (through CKO field)

Fix/Workaround

Transmit or receive CLOCK must not be gated (by the mean of CKO field) when START condition is performed on a generated frame synchro.

21. USB

22. UPCFGn.INTFRQ is irrelevant for isochronous pipe

As a consequence, isochronous IN and OUT tokens are sent every 1ms (Full Speed), or every 125uS (High Speed).

Fix/Workaround

For higher polling time, the software must freeze the pipe for the desired period in order to prevent any "extra" token.

- ADC

1. Sleep Mode activation needs additional A to D conversion

If the ADC sleep mode is activated when the ADC is idle the ADC will not enter sleep mode before after the next AD conversion.

Fix/Workaround

Activate the sleep mode in the mode register and then perform an AD conversion.

SR.CPCS bit, reconfigure the RA and RC registers for the lower channel with the real values.

- *OCD*

1. The auxiliary trace does not work for CPU/HSB speed higher than 50MHz

The auxiliary trace does not work for CPU/HSB speed higher than 50MHz.

Fix/Workaround

Do not use the auxiliary trace for CPU/HSB speed higher than 50MHz.

- *Processor and Architecture*

1. LDM instruction with PC in the register list and without ++ increments Rp

For LDM with PC in the register list: the instruction behaves as if the ++ field is always set, ie the pointer is always updated. This happens even if the ++ field is cleared. Specifically, the increment of the pointer is done in parallel with the testing of R12.

Fix/Workaround

None.

2. RETE instruction does not clear SREG[L] from interrupts

The RETE instruction clears SREG[L] as expected from exceptions.

Fix/Workaround

When using the STCOND instruction, clear SREG[L] in the stacked value of SR before returning from interrupts with RETE.

3. RETS behaves incorrectly when MPU is enabled

RETS behaves incorrectly when MPU is enabled and MPU is configured so that system stack is not readable in unprivileged mode.

Fix/Workaround

Make system stack readable in unprivileged mode, or return from supervisor mode using rete instead of rets. This requires:

1. Changing the mode bits from 001 to 110 before issuing the instruction. Updating the mode bits to the desired value must be done using a single mtsr instruction so it is done atomically. Even if this step is generally described as not safe in the UC technical reference manual, it is safe in this very specific case.
2. Execute the RETE instruction.

4. Privilege violation when using interrupts in application mode with protected system stack

If the system stack is protected by the MPU and an interrupt occurs in application mode, an MPU DTLB exception will occur.

Fix/Workaround

Make a DTLB Protection (Write) exception handler which permits the interrupt request to be handled in privileged mode.

5. USART

6. ISO7816 info register US_NER cannot be read

The NER register always returns zero.

Fix/Workaround

None.

7. ISO7816 Mode T1: RX impossible after any TX

RX impossible after any TX.

Fix/Workaround

SOFT_RESET on RX+ Config US_MR + Config_US_CR.

8. The RTS output does not function correctly in hardware handshaking mode

The RTS signal is not generated properly when the USART receives data in hardware handshaking mode. When the Peripheral DMA receive buffer becomes full, the RTS output should go high, but it will stay low.

Fix/Workaround

Do not use the hardware handshaking mode of the USART. If it is necessary to drive the RTS output high when the Peripheral DMA receive buffer becomes full, use the normal mode of the USART. Configure the Peripheral DMA Controller to signal an interrupt when 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.

9. 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.

10. 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.

11. HMATRIX**12. 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.

- FLASHC**1. Reading from on-chip flash may fail after a flash fuse write operation (FLASHC LP, UP, WGPB, EGPB, SSB, PGPFB, EAGPF commands).**

After a flash fuse write operation (FLASHC LP, UP, WGPB, EGPB, SSB, PGPFB, EAGPF commands), the following flash read access may return corrupted data. This erratum does not affect write operations to regular flash memory.

Fix/Workaround

The flash fuse write operation (FLASHC LP, UP, WGPB, EGPB, SSB, PGPFB, EAGPF commands) must be issued from internal RAM. After the write operation, perform a dummy flash page write operation (FLASHC WP). Content and location of this page is not important

- SSC

1. SSC does not trigger RF when data is low

The SSC cannot transmit or receive data when CKS = CKDIV and CKO = none, in TCMR or RCMR respectively.

Fix/Workaround

Set CKO to a value that is not "none" and bypass the output of the TK/RK pin with the GPIO.

- USB

1. USB No end of host reset signaled upon disconnection

In host mode, in case of an unexpected device disconnection whereas a usb reset is being sent by the usb controller, the UHCON.RESET bit may not be cleared by the hardware at the end of the reset.

Fix/Workaround

A software workaround consists in testing (by polling or interrupt) the disconnection (UHINT.DDISCI == 1) while waiting for the end of reset (UHCON.RESET == 0) to avoid being stuck.

2. USBFSM and UHADDR1/2/3 registers are not available

Do not use USBFSM register.

Fix/Workaround

Do not use USBFSM register and use HCON[6:0] field instead for all the pipes.

- Cycle counter

1. CPU Cycle Counter does not reset the COUNT system register on COMPARE match.

The device revision B does not reset the COUNT system register on COMPARE match. In this revision, the COUNT register is clocked by the CPU clock, so when the CPU clock stops, so does incrementing of COUNT.

Fix/Workaround

None.

- ADC

1. ADC possible miss on DRDY when disabling a channel

The ADC does not work properly when more than one channel is enabled.

Fix/Workaround

Do not use the ADC with more than one channel enabled at a time.

2. ADC OVRE flag sometimes not reset on Status Register read

The OVRE flag does not clear properly if read simultaneously to an end of conversion.

Fix/Workaround

None.

3. Sleep Mode activation needs additional A to D conversion

If the ADC sleep mode is activated when the ADC is idle the ADC will not enter sleep mode before after the next AD conversion.

Fix/Workaround

Activate the sleep mode in the mode register and then perform an AD conversion.

2. **The command Quick Page Read User Page(QPRUP) is not functional**
The command Quick Page Read User Page(QPRUP) is not functional.
Fix/Workaround
None.
3. **PAGEN Semantic Field for Program GP Fuse Byte is WriteData[7:0], ByteAddress[1:0] on revision B instead of WriteData[7:0], ByteAddress[2:0]**
PAGEN Semantic Field for Program GP Fuse Byte is WriteData[7:0], ByteAddress[1:0] on revision B instead of WriteData[7:0], ByteAddress[2:0].
Fix/Workaround
None.
4. **Reading from on-chip flash may fail after a flash fuse write operation (FLASHC LP, UP, WGPB, EGPB, SSB, PGPFB, EAGPF commands).**
After a flash fuse write operation (FLASHC LP, UP, WGPB, EGPB, SSB, PGPFB, EAGPF commands), the following flash read access may return corrupted data. This erratum does not affect write operations to regular flash memory.
Fix/Workaround
The flash fuse write operation (FLASHC LP, UP, WGPB, EGPB, SSB, PGPFB, EAGPF commands) must be issued from internal RAM. After the write operation, perform a dummy flash page write operation (FLASHC WP). Content and location of this page is not important and filling the write buffer with all one (FFh) will leave the current flash content unchanged. It is then safe to read and fetch code from the flash.

5.

- RTC

1. **Writes to control (CTRL), top (TOP) and value (VAL) in the RTC are discarded if the RTC peripheral bus clock (PBA) is divided by a factor of four or more relative to the HSB clock**
Writes to control (CTRL), top (TOP) and value (VAL) in the RTC are discarded if the RTC peripheral bus clock (PBA) is divided by a factor of four or more relative to the HSB clock.
Fix/Workaround
Do not write to the RTC registers using the peripheral bus clock (PBA) divided by a factor of four or more relative to the HSB clock.
2. **The RTC CLKEN bit (bit number 16) of CTRL register is not available**
The RTC CLKEN bit (bit number 16) of CTRL register is not available.
Fix/Workaround
Do not use the CLKEN bit of the RTC on Rev B.

Table of Contents

1 Description 3

2 Overview 4

 2.1 Blockdiagram 4

3 Configuration Summary 5

4 Package and Pinout 6

 4.1 Package 6

 4.2 Peripheral Multiplexing on I/O lines 7

 4.3 High Drive Current GPIO 10

5 Signals Description 10

 5.1 JTAG pins 13

 5.2 RESET_N pin 14

 5.3 TWI pins 14

 5.4 GPIO pins 14

 5.5 High drive pins 14

 5.6 Power Considerations 14

6 Processor and Architecture 17

 6.1 Features 17

 6.2 AVR32 Architecture 17

 6.3 The AVR32UC CPU 18

 6.4 Programming Model 22

 6.5 Exceptions and Interrupts 26

 6.6 Module Configuration 30

7 Memories 31

 7.1 Embedded Memories 31

 7.2 Physical Memory Map 31

 7.3 Peripheral Address Map 32

 7.4 CPU Local Bus Mapping 33

8 Boot Sequence 34

 8.1 Starting of clocks 34

 8.2 Fetching of initial instructions 34

9 Electrical Characteristics 35

 9.1 Absolute Maximum Ratings* 35

