



Welcome to [E-XFL.COM](#)

What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Active
Core Processor	AVR
Core Size	32-Bit Single-Core
Speed	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	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	96K 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-VFQFN Exposed Pad
Supplier Device Package	48-QFN (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at32uc3b1512-z1ur

Table 4-4. Oscillator pinout

QFP48 pin	QFP64 pin	Pad	Oscillator pin
30	39	PA18	XIN0
	41	PA28	XIN1
22	30	PA11	XIN32
31	40	PA19	XOUT0
	42	PA29	XOUT1
23	31	PA12	XOUT32

4.3 High Drive Current GPIO

Ones of GPIOs can be used to drive twice current than other GPIO capability (see Electrical Characteristics section).

Table 4-5. High Drive Current GPIO

GPIO Name
PA20
PA21
PA22
PA23

5. Signals Description

The following table gives details on the signal name classified by peripheral.

Table 5-1. Signal Description List

Signal Name	Function	Type	Active Level	Comments
Power				
VDDPLL	PLL Power Supply	Power Input		1.65V to 1.95 V
VDDCORE	Core Power Supply	Power Input		1.65V to 1.95 V
VDDIO	I/O Power Supply	Power Input		3.0V to 3.6V
VDDANA	Analog Power Supply	Power Input		3.0V to 3.6V
VDDIN	Voltage Regulator Input Supply	Power Input		3.0V to 3.6V

Table 9-7. BOD Timing

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
T_{BOD}	Minimum time with VDDCORE < VBOD to detect power failure	Falling VDDCORE from 1.8V to 1.1V		300	800	ns

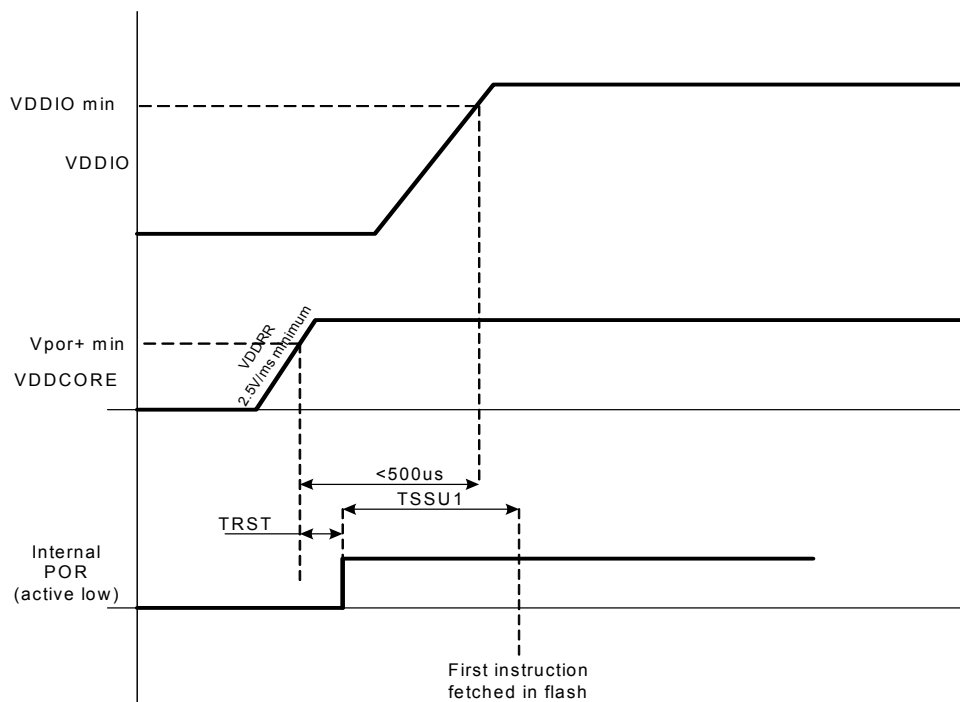
9.4.3 Reset Sequence

Table 9-8. Electrical Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
V_{DDRR}	VDDCORE rise rate to ensure power-on-reset		2.5			V/ms
V_{DDFR}	VDDCORE fall rate to ensure power-on-reset		0.01		400	V/ms
V_{POR+}	Rising threshold voltage: voltage up to which device is kept under reset by POR on rising VDDCORE	Rising VDDCORE: $V_{RESTART} \rightarrow V_{POR+}$	1.4	1.55	1.65	V
V_{POR-}	Falling threshold voltage: voltage when POR resets device on falling VDDCORE	Falling VDDCORE: $1.8V \rightarrow V_{POR+}$	1.2	1.3	1.4	V
$V_{RESTART}$	On falling VDDCORE, voltage must go down to this value before supply can rise again to ensure reset signal is released at V_{POR+}	Falling VDDCORE: $1.8V \rightarrow V_{RESTART}$	-0.1		0.5	V
T_{POR}	Minimum time with VDDCORE < V_{POR-}	Falling VDDCORE: $1.8V \rightarrow 1.1V$		15		μs
T_{RST}	Time for reset signal to be propagated to system			200	400	μs
T_{SSU1}	Time for Cold System Startup: Time for CPU to fetch its first instruction (RCosc not calibrated)		480		960	μs
T_{SSU2}	Time for Hot System Startup: Time for CPU to fetch its first instruction (RCosc calibrated)			420		μs

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

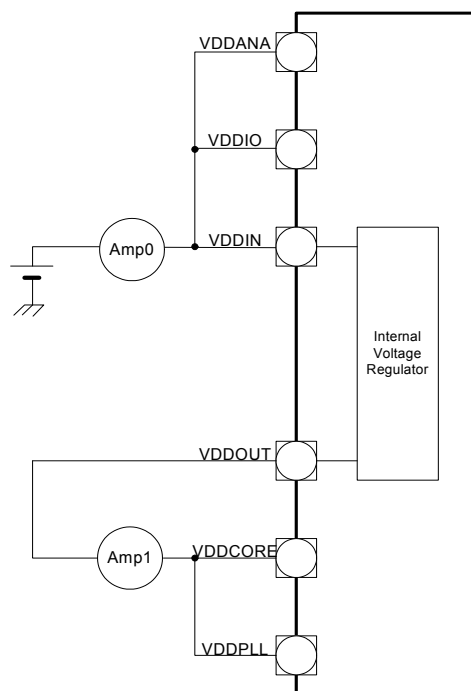
Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
t_{RESET}	RESET_N minimum pulse width		10			ns

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"> - CPU running a recursive Fibonacci Algorithm from flash and clocked from PLL0 at f MHz. - Voltage regulator is on. - XIN0: external clock. Xin1 Stopped. XIN32 stopped. - All peripheral clocks activated with a division by 8. - GPIOs are inactive with internal pull-up, JTAG unconnected with external pull-up and Input pins are connected to GND 	$0.3 \times f(\text{MHz}) + 0.443$	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"> - CPU running in sleep mode - XIN0, Xin1 and XIN32 are stopped. - All peripheral clocks are deactivated. - GPIOs are inactive with internal pull-up, JTAG unconnected with external pull-up and Input pins are connected to GND. 	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"> - CPU running a recursive Fibonacci Algorithm from flash and clocked from PLL0 at f MHz. - Voltage regulator is on. - XIN0: external clock. Xin1 Stopped. XIN32 stopped. - All peripheral clocks activated with a division by 8. - GPIOs are inactive with internal pull-up, JTAG unconnected with external pull-up and Input pins are connected to GND 	$0.359 \times f(\text{MHz}) + 1.53$	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"> - CPU running in sleep mode - XIN0, Xin1 and XIN32 are stopped. - All peripheral clocks are desactivated. - GPIOs are inactive with internal pull-up, JTAG unconnected with external pull-up and Input pins are connected to GND. 	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	

9.6 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

10. Mechanical Characteristics

10.1 Thermal Considerations

10.1.1 Thermal Data

[Table 10-1](#) summarizes the thermal resistance data depending on the package.

Table 10-1. Thermal Resistance Data

Symbol	Parameter	Condition	Package	Typ	Unit
θ_{JA}	Junction-to-ambient thermal resistance	Still Air	TQFP64	49.6	°C/W
θ_{JC}	Junction-to-case thermal resistance		TQFP64	13.5	
θ_{JA}	Junction-to-ambient thermal resistance	Still Air	TQFP48	51.1	°C/W
θ_{JC}	Junction-to-case thermal resistance		TQFP48	13.7	

10.1.2 Junction Temperature

The average chip-junction temperature, T_J , in °C can be obtained from the following:

1. $T_J = T_A + (P_D \times \theta_{JA})$
2. $T_J = T_A + (P_D \times (\theta_{HEATSINK} + \theta_{JC}))$

where:

- θ_{JA} = package thermal resistance, Junction-to-ambient (°C/W), provided in [Table 10-1 on page 55](#).
- θ_{JC} = package thermal resistance, Junction-to-case thermal resistance (°C/W), provided in [Table 10-1 on page 55](#).
- $\theta_{HEAT\ SINK}$ = cooling device thermal resistance (°C/W), provided in the device datasheet.
- P_D = device power consumption (W) estimated from data provided in the section "[Power Consumption](#)" on page 42.
- T_A = ambient temperature (°C).

From the first equation, the user can derive the estimated lifetime of the chip and decide if a cooling device is necessary or not. If a cooling device is to be fitted on the chip, the second equation should be used to compute the resulting average chip-junction temperature T_J in °C.

10.3 Soldering Profile

Table 10-14 gives the recommended soldering profile from J-STD-20.

Table 10-14. Soldering Profile

Profile Feature	Green Package
Average Ramp-up Rate (217°C to Peak)	3°C/s
Preheat Temperature 175°C ±25°C	Min. 150°C, Max. 200°C
Temperature Maintained Above 217°C	60-150s
Time within 5°C of Actual Peak Temperature	30s
Peak Temperature Range	260°C
Ramp-down Rate	6°C/s
Time 25°C to Peak Temperature	Max. 8mn

Note: It is recommended to apply a soldering temperature higher than 250°C.
A maximum of three reflow passes is allowed per component.

12. Errata

12.1 AT32UC3B0512, AT32UC3B1512

12.1.1 Rev D

- PWM

1. PWM channel interrupt enabling triggers an interrupt

When enabling a PWM channel that is configured with center aligned period (CALG=1), an interrupt is signalled.

Fix/Workaround

When using center aligned mode, enable the channel and read the status before channel interrupt is enabled.

2. PWN counter restarts at 0x0001

The PWM counter restarts at 0x0001 and not 0x0000 as specified. Because of this the first PWM period has one more clock cycle.

Fix/Workaround

- The first period is 0x0000, 0x0001, ..., period.
- Consecutive periods are 0x0001, 0x0002, ..., period.

3. PWM update period to a 0 value does not work

It is impossible to update a period equal to 0 by the using the PWM update register (PWM_CUPD).

Fix/Workaround

Do not update the PWM_CUPD register with a value equal to 0.

4. SPI

5. SPI Slave / PDCA transfer: no TX UNDERRUN flag

There is no TX UNDERRUN flag available, therefore in SPI slave mode, there is no way to be informed of a character lost in transmission.

Fix/Workaround

For PDCA transfer: none.

6. SPI bad serial clock generation on 2nd chip_select when SCBR=1, CPOL=1, and NCPHA=0

When multiple chip selects (CS) are in use, if one of the baudrates equal 1 while one (CSRn.SCBR=1) of the others do not equal 1, and CSRn.CPOL=1 and CSRn.NCPHA=0, then an additional pulse will be generated on SCK.

Fix/Workaround

When multiple CS are in use, if one of the baudrates equals 1, the others must also equal 1 if CSRn.CPOL=1 and CSRn.NCPHA=0.

7. TC

8. **Channel chaining skips first pulse for upper channel**

When chaining two channels using the Block Mode Register, the first pulse of the clock between the channels is skipped.

Fix/Workaround

Configure the lower channel with RA = 0x1 and RC = 0x2 to produce a dummy clock cycle for the upper channel. After the dummy cycle has been generated, indicated by the SR.CPCS bit, reconfigure the RA and RC registers for the lower channel with the real values.

*- 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. **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.

4. **USART**5. **ISO7816 info register US_NER cannot be read**

The NER register always returns zero.

Fix/Workaround

None.

6. **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.

7. **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

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.

- *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. **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.

4. **Flash**

5. **Reset vector is 80000020h rather than 80000000h**

Reset vector is 80000020h rather than 80000000h.

Fix/Workaround

The flash program code must start at the address 80000020h. The flash memory range 80000000h-80000020h must be programmed with 00000000h.

- *USART*

1. **ISO7816 info register US_NER cannot be read**

The NER register always returns zero.

Fix/Workaround

None.

2. **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.

3. **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 trans-

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.

- 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 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.

- 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.

12.2.2 Rev. G

- PWM

1. **PWM channel interrupt enabling triggers an interrupt**
When enabling a PWM channel that is configured with center aligned period (CALG=1), an interrupt is signalled.
Fix/Workaround
When using center aligned mode, enable the channel and read the status before channel interrupt is enabled.
2. **PWN counter restarts at 0x0001**
The PWM counter restarts at 0x0001 and not 0x0000 as specified. Because of this the first PWM period has one more clock cycle.
Fix/Workaround
- The first period is 0x0000, 0x0001, ..., period.
- Consecutive periods are 0x0001, 0x0002, ..., period.
3. **PWM update period to a 0 value does not work**
It is impossible to update a period equal to 0 by the using the PWM update register (PWM_CUPD).
Fix/Workaround
Do not update the PWM_CUPD register with a value equal to 0.
4. **SPI**
5. **SPI Slave / PDCA transfer: no TX UNDERRUN flag**
There is no TX UNDERRUN flag available, therefore in SPI slave mode, there is no way to be informed of a character lost in transmission.
Fix/Workaround
For PDCA transfer: none.
6. **SPI bad serial clock generation on 2nd chip_select when SCBR=1, CPOL=1, and NCPHA=0**
When multiple chip selects (CS) are in use, if one of the baudrates equal 1 while one (CSRn.SCBR=1) of the others do not equal 1, and CSRn.CPOL=1 and CSRn.NCPHA=0, then an additional pulse will be generated on SCK.
Fix/Workaround
When multiple CS are in use, if one of the baudrates equals 1, the others must also equal 1 if CSRn.CPOL=1 and CSRn.NCPHA=0.
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.

12.2.3 Rev. F

- PWM

1. **PWM channel interrupt enabling triggers an interrupt**
 When enabling a PWM channel that is configured with center aligned period (CALG=1), an interrupt is signalled.
Fix/Workaround
 When using center aligned mode, enable the channel and read the status before channel interrupt is enabled.
2. **PWN counter restarts at 0x0001**
 The PWM counter restarts at 0x0001 and not 0x0000 as specified. Because of this the first PWM period has one more clock cycle.
Fix/Workaround
 - The first period is 0x0000, 0x0001, ..., period.
 - Consecutive periods are 0x0001, 0x0002, ..., period.
3. **PWM update period to a 0 value does not work**
 It is impossible to update a period equal to 0 by the using the PWM update register (PWM_CUPD).
Fix/Workaround
 Do not update the PWM_CUPD register with a value equal to 0.
4. **SPI**
5. **SPI Slave / PDCA transfer: no TX UNDERRUN flag**
 There is no TX UNDERRUN flag available, therefore in SPI slave mode, there is no way to be informed of a character lost in transmission.
Fix/Workaround
 For PDCA transfer: none.
6. **SPI bad serial clock generation on 2nd chip_select when SCBR=1, CPOL=1, and NCPHA=0**
 When multiple chip selects (CS) are in use, if one of the baudrates equal 1 while one (CSRn.SCBR=1) of the others do not equal 1, and CSRn.CPOL=1 and CSRn.NCPHA=0, then an additional pulse will be generated on SCK.
Fix/Workaround
 When multiple CS are in use, if one of the baudrates equals 1, the others must also equal 1 if CSRn.CPOL=1 and CSRn.NCPHA=0.
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.

3. 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.

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

13. Datasheet Revision History

Please note that the referring page numbers in this section are referred to this document. The referring revision in this section are referring to the document revision.

13.1 Rev. L– 01/2012

1. Updated Mechanical Characteristics section.

13.2 Rev. K– 02/2011

1. Updated USB section.
2. Updated Configuration Summary section.
3. Updated Electrical Characteristics section.
4. Updated Errata section.

13.3 Rev. J– 12/2010

1. Updated USB section.
2. Updated USART section.
3. Updated TWI section.
4. Updated PWM section.
5. Updated Electrical Characteristics section.

13.4 Rev. I – 06/2010

1. Updated SPI section.
2. Updated Electrical Characteristics section.

13.5 Rev. H – 10/2009

1. Update datasheet architecture.
2. Add AT32UC3B0512 and AT32UC3B1512 devices description.

13.6 Rev. G – 06/2009

1. Open Drain Mode removed from GPIO section.
2. Updated Errata section.

13.7 Rev. F – 04/2008

1. Updated Errata section.

13.8 Rev. E – 12/2007

1. Updated Memory Protection section.

13.9 Rev. D – 11/2007

1. Updated Processor Architecture section.
2. Updated Electrical Characteristics section.

13.10 Rev. C – 10/2007

1. Updated Features sections.
2. Updated block diagram with local bus figure
3. Add schematic for HMatrix master/slave connection.
4. Updated Features sections with local bus.
5. Added SPI feature to USART section.
6. Updated USBB section.
7. Updated ADC trigger selection in ADC section.
8. Updated JTAG and Boundary Scan section with programming procedure.
9. Add description for silicon revision D

13.11 Rev. B – 07/2007

1. Updated registered trademarks
2. Updated address page.