

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 "<u>Embedded -</u> <u>Microcontrollers</u>"

### Details

E·XFI

Product Status	Active
Core Processor	AVR
Core Size	32-Bit Single-Core
Speed	66MHz
Connectivity	EBI/EMI, I <sup>2</sup> C, IrDA, Memory Card, SPI, SSC, UART/USART, USB OTG
Peripherals	Brown-out Detect/Reset, DMA, POR, WDT
Number of I/O	110
Program Memory Size	256КВ (256К х 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128K x 8
Voltage - Supply (Vcc/Vdd)	1.75V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at32uc3a3256-alut

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong



### 3.2.4 JTAG port connections

Table 3-4.	JTAG Pinout
------------	-------------

TFBGA144	QFP144	VFBGA100	Pin name	JTAG pin
K12	107	K9	ТСК	тск
L12	108	K8	TDI	TDI
J11	105	J8	TDO	TDO
J10	104	H7	TMS	TMS

### 3.2.5 Nexus OCD AUX port connections

If the OCD trace system is enabled, the trace system will take control over a number of pins, irrespective of the GPIO configuration. Three differents OCD trace pin mappings are possible, depending on the configuration of the OCD AXS register. For details, see the AVR32 UC Technical Reference Manual.

 Table 3-5.
 Nexus OCD AUX port connections

Pin	AXS=0	AXS=1	AXS=2
EVTI_N	PB05	PA08	PX00
MDO[5]	PA00	PX56	PX06
MDO[4]	PA01	PX57	PX05
MDO[3]	PA03	PX58	PX04
MDO[2]	PA16	PA24	PX03
MDO[1]	PA13	PA23	PX02
MDO[0]	PA12	PA22	PX01
MSEO[1]	PA10	PA07	PX08
MSEO[0]	PA11	PX55	PX07
МСКО	PB07	PX00	PB09
EVTO_N	PB06	PB06	PB06



## 3.5 Power Considerations

### 3.5.1 Power Supplies

The AT32UC3A3 has several types of power supply pins:

- VDDIO: Powers I/O lines. Voltage is 3.3 V nominal
- VDDANA: Powers the ADC. Voltage is 3.3V nominal
- VDDIN: Input voltage for the voltage regulator. Voltage is 3.3V nominal
- VDDCORE: Output voltage from regulator for filtering purpose and provides the supply to the core, memories, and peripherals. Voltage is 1.8V nominal

The ground pin GNDCORE is common to VDDCORE and VDDIN. The ground pin for VDDANA is GNDANA. The ground pins for VDDIO are GNDIO.

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

### 3.5.2 Voltage Regulator

The AT32UC3A3 embeds a voltage regulator that converts from 3.3V to 1.8V with a load of up to 100 mA. The regulator takes its input voltage from VDDIN, and supplies the output voltage on VDDCORE and powers the core, memories and peripherals.

Adequate output supply decoupling is mandatory for VDDCORE to reduce ripple and avoid oscillations.

The best way to achieve this is to use two capacitors in parallel between VDDCORE and GNDCORE:

- One external 470pF (or 1nF) NPO capacitor (COUT1) should be connected as close to the chip as possible.
- One external 2.2µF (or 3.3µF) X7R capacitor (COUT2).

Adequate input supply decoupling is mandatory for VDDIN in order to improve startup stability and reduce source voltage drop.

The input decoupling capacitor should be placed close to the chip, e.g., two capacitors can be used in parallel (1nF NPO and  $4.7\mu$ F X7R).



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



The register file is organized as sixteen 32-bit registers and includes the Program Counter, the Link Register, and the Stack Pointer. In addition, register R12 is designed to hold return values from function calls and is used implicitly by some instructions.

## 4.3 The AVR32UC CPU

The AVR32UC CPU targets low- and medium-performance applications, and provides an advanced OCD system, no caches, and a Memory Protection Unit (MPU). Java acceleration hardware is not implemented.

AVR32UC provides three memory interfaces, one High Speed Bus master for instruction fetch, one High Speed Bus master for data access, and one High Speed Bus slave interface allowing other bus masters to access data RAMs internal to the CPU. Keeping data RAMs internal to the CPU allows fast access to the RAMs, reduces latency, and guarantees deterministic timing. Also, power consumption is reduced by not needing a full High Speed Bus access for memory accesses. A dedicated data RAM interface is provided for communicating with the internal data RAMs.

A local bus interface is provided for connecting the CPU to device-specific high-speed systems, such as floating-point units and fast GPIO ports. This local bus has to be enabled by writing the LOCEN bit in the CPUCR system register. The local bus is able to transfer data between the CPU and the local bus slave in a single clock cycle. The local bus has a dedicated memory range allocated to it, and data transfers are performed using regular load and store instructions. Details on which devices that are mapped into the local bus space is given in the Memories chapter of this data sheet.

Figure 4-1 on page 23 displays the contents of AVR32UC.



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

**Table 4-1.** Instructions with Unaligned Reference Support

Instruction	Supported alignment
ld.d	Word
st.d	Word

### 4.3.6 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

### 4.3.7 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.



	• ) • • • • •		-/
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

 Table 4-3.
 System Registers (Continued)

### 4.5 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 4-4 on page 33. 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 autovectored 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 | 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.

### 4.5.1 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.



Priority	Handler Address	Name	Event source	Stored Return Address
1	0x8000_0000	Reset	External input	Undefined
2	Provided by OCD system	OCD Stop CPU	OCD system	First non-completed instruction
3	EVBA+0x00	Unrecoverable exception	Internal	PC of offending instruction
4	EVBA+0x04	TLB multiple hit	MPU	
5	EVBA+0x08	Bus error data fetch	Data bus	First non-completed instruction
6	EVBA+0x0C	Bus error instruction fetch	Data bus	First non-completed instruction
7	EVBA+0x10	NMI	External input	First non-completed instruction
8	Autovectored	Interrupt 3 request	External input	First non-completed instruction
9	Autovectored	Interrupt 2 request	External input	First non-completed instruction
10	Autovectored	Interrupt 1 request	External input	First non-completed instruction
11	Autovectored	Interrupt 0 request	External input	First non-completed instruction
12	EVBA+0x14	Instruction Address	CPU	PC of offending instruction
13	EVBA+0x50	ITLB Miss	MPU	
14	EVBA+0x18	ITLB Protection	MPU	PC of offending instruction
15	EVBA+0x1C	Breakpoint	OCD system	First non-completed instruction
16	EVBA+0x20	Illegal Opcode	Instruction	PC of offending instruction
17	EVBA+0x24	Unimplemented instruction	Instruction	PC of offending instruction
18	EVBA+0x28	Privilege violation	Instruction	PC of offending instruction
19	EVBA+0x2C	Floating-point	UNUSED	
20	EVBA+0x30	Coprocessor absent	Instruction	PC of offending instruction
21	EVBA+0x100	Supervisor call	Instruction	PC(Supervisor Call) +2
22	EVBA+0x34	Data Address (Read)	CPU	PC of offending instruction
23	EVBA+0x38	Data Address (Write)	CPU	PC of offending instruction
24	EVBA+0x60	DTLB Miss (Read)	MPU	
25	EVBA+0x70	DTLB Miss (Write)	MPU	
26	EVBA+0x3C	DTLB Protection (Read)	MPU	PC of offending instruction
27	EVBA+0x40	DTLB Protection (Write)	MPU	PC of offending instruction
28	EVBA+0x44	DTLB Modified	UNUSED	

## Table 4-4. Priority and Handler Addresses for Events



### **Table 5-2.**Peripheral Address Mapping

0xFFFF0C00	РМ	Power Manager - PM
0xFFFF0D00	RTC	Real Time Counter - RTC
0xFFFF0D30	WDT	Watchdog Timer - WDT
0xFFFF0D80	EIC	External Interrupt Controller - EIC
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
0xFFFF2000	USART3	Universal Synchronous/Asynchronous Receiver/Transmitter - USART3
0xFFFF2400	SPI0	Serial Peripheral Interface - SPI0
0xFFFF2800	SPI1	Serial Peripheral Interface - SPI1
0xFFFF2C00	TWIMO	Two-wire Master Interface - TWIM0
0xFFFF3000	TWIM1	Two-wire Master Interface - TWIM1
0xFFFF3400	SSC	Synchronous Serial Controller - SSC
0xFFFF3800	TC0	Timer/Counter - TC0
0xFFFF3C00	ADC	Analog to Digital Converter - ADC
0xFFFF4000	ABDAC	Audio Bitstream DAC - ABDAC
0xFFFF4400	TC1	Timer/Counter - TC1



 Table 5-2.
 Peripheral Address Mapping

0xFFFF5000	TWIS0	Two-wire Slave Interface - TWIS0
0xFFFF5400	TWIS1	Two-wire Slave Interface - TWIS1

## 5.4 CPU Local Bus Mapping

Some of the registers in the GPIO module are mapped onto the CPU local bus, in addition to being mapped on the Peripheral Bus. These registers can therefore be reached both by accesses on the Peripheral Bus, and by accesses on the local bus.

Mapping these registers on the local bus allows cycle-deterministic toggling of GPIO pins since the CPU and GPIO are the only modules connected to this bus. Also, since the local bus runs at CPU speed, one write or read operation can be performed per clock cycle to the local busmapped GPIO registers.

The following GPIO registers are mapped on the local bus:

 Table 5-3.
 Local Bus Mapped GPIO Registers

Port	Register	Mode	Local Bus Address	Access
0	Output Driver Enable Register (ODER)	WRITE	0x40000040	Write-only
		SET	0x40000044	Write-only
		CLEAR	0x40000048	Write-only
		TOGGLE	0x4000004C	Write-only
	Output Value Register (OVR)	WRITE	0x40000050	Write-only
		SET	0x40000054	Write-only
		CLEAR	0x40000058	Write-only
		TOGGLE	0x4000005C	Write-only
	Pin Value Register (PVR)	-	0x40000060	Read-only
1	Output Driver Enable Register (ODER)	WRITE	0x40000140	Write-only
		SET	0x40000144	Write-only
		CLEAR	0x40000148	Write-only
		TOGGLE	0x4000014C	Write-only
	Output Value Register (OVR)	WRITE	0x40000150	Write-only
		SET	0x40000154	Write-only
		CLEAR	0x40000158	Write-only
		TOGGLE	0x4000015C	Write-only
	Pin Value Register (PVR)	-	0x40000160	Read-only



## 7.2 DC Characteristics

The following characteristics are applicable to the operating temperature range:  $T_A = -40^{\circ}C$  to 85°C, unless otherwise specified and are certified for a junction temperature up to $T_J = 100^{\circ}C$ .

Symbol	Parameter	Conditions		Min.	Тур.	Max.	Unit
VVDDIO	DC Supply Peripheral I/Os			3.0		3.6	V
Vvddana	DC Analog Supply			3.0		3.6	V
		All I/O pins except TWC RESET_N, TCK, TDI	K, TWD,	-0.3		+0.8	V
V <sub>IL</sub>	Input Low-level Voltage	TWCK, TWD		V <sub>VDDIO</sub> x0.7		V <sub>VDDIO</sub> +0.5	V
		RESET_N, TCK, TDI		+0.8V			V
.,		All I/O pins except TWC	K, TWD	2.0		3.6	V
V <sub>IH</sub>	Input High-level Voltage	TWCK, TWD					V
V <sub>OL</sub>	Output Low-level Voltage	$I_{OL} = -2mA$ for Pin drive x1 $I_{OL} = -4mA$ for Pin drive x2 $I_{OL} = -8mA$ for Pin drive x3				0.4	V
V <sub>OH</sub>	Output High-level Voltage	$I_{OH} = 2mA$ for Pin drive x1 $I_{OH} = 4mA$ for Pin drive x2 $I_{OH} = 8mA$ for Pin drive x3		V <sub>VDDIO</sub> -0.4			V
I <sub>LEAK</sub>	Input Leakage Current	Pullup resistors disabled			0.05	1	μA
C <sub>IN</sub>	Input Capacitance				7		pF
R <sub>PULLUP</sub>	Pull-up Resistance	All I/O pins except RES TDI, TMS	ET_N, TCK,	9	15	25	KΩ
		RESET_N, TCK, TDI, TMS		5		25	KΩ
I <sub>O</sub>	Output Current Pin drive 1x Pin drive 2x Pin drive 3x					2.0 4.0 8.0	mA
1	Static Current	On V <sub>VDDIN</sub> = 3.3V,	$T_A = 25^{\circ}C$		30		μΑ
I <sub>SC</sub> Sta	Static Guilent	CPU in static mode	$T_A = 85^{\circ}C$		175		μA

Table 7-1.	DC Characteristics
------------	--------------------



 Table 7-2.
 Normal I/O Pin Characteristics

Symbol	Parameter	Conditions	drive x2	drive x2	drive x3	Unit
		10pf	40	66	100	MHz
f <sub>MAX</sub>	Output frequency	30pf	18.2	35.7	61.6	MHz
		60pf	7.5	18.5	36.3	MHz
		10pf	2.7	1.4	0.9	ns
t <sub>RISE</sub>	Rise time	30pf	6.9	3.5	1.9	ns
		60pf	13.4	6.7	3.5	ns
		10pf	3.2	1.7	0.9	ns
t <sub>FALL</sub>	Fall time	30pf	8.6	4.3	2.26	ns
		60pf	16.5	8.3	4.3	ns

## 7.4 Regulator characteristics

### Table 7-3. Electrical Characteristics

Symbol	Parameter	Conditions	Min.	Тур.	Max.	Unit
V <sub>VDDIN</sub>	Supply voltage (input)		3.0	3.3	3.6	V
V <sub>VDDCORE</sub>	Supply voltage (output)		1.75	1.85	1.95	V
Ιουτ	Maximum DC output current	VVDDIN = 3.3V			100	mA

## Table 7-4. Decoupling Requirements

Symbol	Parameter	Conditions	Тур.	Technology	Unit
C <sub>IN1</sub>	Input Regulator Capacitor 1		1	NPO	nF
C <sub>IN2</sub>	Input Regulator Capacitor 2		4.7	X7R	μF
C <sub>OUT1</sub>	Output Regulator Capacitor 1		470	NPO	pF
C <sub>OUT2</sub>	Output Regulator Capacitor 2		2.2	X7R	μF













VDDIN VDDIO





## 7.9 ADC Characteristics

Parameter Conditions		Min.	Тур.	Max.	Unit
ADC Clock Frequency	10-bit resolution mode			5	MHz
ADC Clock Frequency	8-bit resolution mode			8	MHz
Startup Time	Return from Idle Mode			20	μs
Track and Hold Acquisition Time		600			ns
Conversion Time	ADC Clock = 5 MHz			2	μs
Conversion Time	ADC Clock = 8 MHz			1.25	μs
Throughout Data	ADC Clock = 5 MHz			384 <sup>(1)</sup>	kSPS
	ADC Clock = 8 MHz			533 <sup>(2)</sup>	kSPS

 Table 7-22.
 Channel Conversion Time and ADC Clock

1. Corresponds to 13 clock cycles: 3 clock cycles for track and hold acquisition time and 10 clock cycles for conversion.

2. Corresponds to 15 clock cycles: 5 clock cycles for track and hold acquisition time and 10 clock cycles for conversion.

### Table 7-23. ADC Power Consumption

Parameter	Conditions	Min.	Тур.	Max.	Unit
Current Consumption on VDDANA <sup>(1)</sup>	On 13 samples with ADC clock = 5 MHz			1.25	mA

1. Including internal reference input current

### Table 7-24.Analog Inputs

Parameter	Conditions	Min.	Тур.	Max.	Unit
Input Voltage Range		0		VDDANA	V
Input Leakage Current				1	μA
Input Capacitance			7		рF
Input Resistance			350	850	Ohm

## Table 7-25. Transfer Characteristics in 8-bit mode

Parameter	Conditions	Min.	Тур.	Max.	Unit
Resolution			8		Bit
Absoluto Accuracy	ADC Clock = 5 MHz			0.8	LSB
Absolute Accuracy	ADC Clock = 8 MHz			1.5	LSB
Integral Non linearity	ADC Clock = 5 MHz		0.35	0.5	LSB
	ADC Clock = 8 MHz		0.5	1.5	LSB
Differential Nen linearity	ADC Clock = 5 MHz		0.3	0.5	LSB
Differential Non-intearity	ADC Clock = 8 MHz		0.5	1.5	LSB
Offset Error	ADC Clock = 5 MHz	-1.5		1.5	LSB
Gain Error	ADC Clock = 5 MHz	-0.5		0.5	LSB



Parameter	Conditions	Min.	Тур.	Max.	Unit
Resolution			10		Bit
Absolute Accuracy	ADC Clock = 5 MHz			3	LSB
Integral Non-linearity	ADC Clock = 5 MHz		1.5	2	LSB
Differential Neg linearity	ADC Clock = 5 MHz		1	2	LSB
Diferential Non-intearity	ADC Clock = 2.5 MHz		0.6	1	LSB
Offset Error	ADC Clock = 5 MHz	-2		2	LSB
Gain Error	ADC Clock = 5 MHz	-2		2	LSB

## **Table 7-26.**Transfer Characteristics in 10-bit mode

## 7.10 USB Transceiver Characteristics

### 7.10.1 Electrical Characteristics

### Table 7-27. Electrical Parameters

Symbol	Parameter	Conditions	Min.	Тур.	Max.	Unit
R <sub>EXT</sub>	Recommended External USB Series Resistor	In series with each USB pin with ±5%		39		Ω
R <sub>BIAS</sub>	VBIAS External Resistor <sup>(1)</sup>	±1%		6810		Ω
C <sub>BIAS</sub>	VBIAS External Capcitor			10		pF

1. 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.

### 7.10.2 Static Power Consumption

### Table 7-28. Static Power Consumption

Symbol	Parameter	Conditions	Min.	Тур.	Max.	Unit
I <sub>BIAS</sub>	Bias current consumption on VBG				1	μA
	HS Transceiver and I/O current consumption				8	μA
I <sub>VDDUTMI</sub>	FS/HS Transceiver and I/O current consumption	If cable is connected, add 200µA (typical) due to Pull-up/Pull-down current consumption			3	μΑ

### 7.10.3 Dynamic Power Consumption

### **Table 7-29.**Dynamic Power Consumption

Symbol	Parameter	Conditions	Min.	Тур.	Max.	Unit
I <sub>BIAS</sub>	Bias current consumption on VBG			0.7	0.8	mA



## 8.3 Soldering Profile

Table 8-5 gives the recommended soldering profile from J-STD-20.

Table 8-5.	Soldering Profile
------------	-------------------

Profile Feature	Green Package
Average Ramp-up Rate (217°C to Peak)	3°C/Second max
Preheat Temperature 175°C ±25°C	150-200°C
Time Maintained Above 217°C	60-150 seconds
Time within 5°C of Actual Peak Temperature	30 seconds
Peak Temperature Range	260 (+0/-5°C)
Ramp-down Rate	6°C/Second max.
Time 25°C to Peak Temperature	8 minutes max

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



### 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.

### 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.

### 10.1.8 Power Manager

## OSC32 not functionnal in Crystal Modes (OSC32CTRL.MODE=1 or OSC32CTRL.MODE=2)

OSC32 clock output is not active even if the oscillation signal is present on XIN32/XOUT32 pins.

OSC32RDY bit may still set even if the CLK32 is not active.

External clock mode (OSC32CTRL.MODE=0) is not affected.

### Fix/Workaround

None.

# 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.

### 10.1.9 PDCA

### PCONTROL.CHxRES is non-functional

PCONTROL.CHxRES is non-functional. Counters are reset at power-on, and cannot be reset by software.

#### Fix/Workaround

Software needs to keep history of performance counters.

### Transfer error will stall a transmit peripheral handshake interface

If a transfer error is encountered on a channel transmitting to a peripheral, the peripheral handshake of the active channel will stall and the PDCA will not do any more transfers on the affected peripheral handshake interface.

### Fix/Workaround

Disable and then enable the peripheral after the transfer error.



### Fix/Workaround

Set to 1b bit CORRS4 of the ECCHRS mode register (MD). In C-code: \*((volatile int\*) (0xFFFE2404))= 0x400.

DMACA data transfer fails when CTLx.SRC\_TR\_WIDTH is not equal to CTLx.DST\_TR\_WIDTH Fix/Workaround For any DMACA transfer make sure CTLx.SRC\_TR\_WIDTH = CTLx.DST\_TR\_WIDTH.

### 3.3V supply monitor is not available

FGPFRLO[30:29] are reserved and should not be used by the application. **Fix/Workaround** None.

Service access bus (SAB) can not access DMACA registers Fix/Workaround None.

### 10.2.2 Processor and Architecture

### 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.

### 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.

### 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.

10.2.3 MPU

# 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.

### 10.2.4 USB

### 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).



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

Fix/Workaround None.

### 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.

### **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.

### 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.

### Multiply instructions do not work on RevD

All the multiply instructions do not work. **Fix/Workaround** Do not use the multiply instructions.

10.3.3 MPU

## 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.

### 10.3.4 USB

#### 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.



## 11.7 Rev. B - 08/09

1. Updated the datasheet with new device AT32UC3A4.

## 11.8 Rev. A - 03/09

1. Initial revision.



1	Descr	iption	3
2	Overv	<i>r</i> iew	4
	2.1	Block Diagram	4
	2.2	Configuration Summary	5
3	Packa	nge and Pinout	6
	3.1	Package	6
	3.2	Peripheral Multiplexing on I/O lines	9
	3.3	Signal Descriptions	14
	3.4	I/O Line Considerations	19
	3.5	Power Considerations	20
4	Proce	essor and Architecture	21
	4.1	Features	21
	4.2	AVR32 Architecture	21
	4.3	The AVR32UC CPU	22
	4.4	Programming Model	26
	4.5	Exceptions and Interrupts	
5	Мето	ories	34
	5.1	Embedded Memories	34
	5.2	Physical Memory Map	34
	5.3	Peripheral Address Map	35
	5.4	CPU Local Bus Mapping	37
6	Boot S	Sequence	39
	6.1	Starting of Clocks	
	6.2	Fetching of Initial Instructions	
7	Electr	ical Characteristics	40
	7.1	Absolute Maximum Ratings*	40
	7.2	DC Characteristics	41
	7.3	I/O pin Characteristics	42
	7.4	Regulator characteristics	43
	7.5	Analog characteristics	44
	7.6	Power Consumption	48
	7.7	System Clock Characteristics	51
	7.8	Oscillator Characteristics	52
	7.9	ADC Characteristics	54

