Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	AVR
Core Size	32-Bit Single-Core
Speed	66MHz
Connectivity	EBI/EMI, Ethernet, I²C, SPI, SSC, UART/USART, USB OTG
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	109
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	64K x 8
Voltage - Supply (Vcc/Vdd)	1.65V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at32uc3a0512-altr

2. Configuration Summary

The table below lists all AT32UC3A memory and package configurations:

Device	Flash	SRAM	Ext. Bus Interface	Ethernet MAC	Package
AT32UC3A0512	512 Kbytes	64 Kbytes	yes	yes	144 pin LQFP 144 pin BGA
AT32UC3A0256	256 Kbytes	64 Kbytes	yes	yes	144 pin LQFP 144 pin BGA
AT32UC3A0128	128 Kbytes	32 Kbytes	yes	yes	144 pin LQFP 144 pin BGA
AT32UC3A1512	512 Kbytes	64 Kbytes	no	yes	100 pin TQFP
AT32UC3A1256	256 Kbytes	64 Kbytes	no	yes	100 pin TQFP
AT32UC3A1128	128 Kbytes	32 Kbytes	no	yes	100 pin TQFP

3. Abbreviations

- GCLK: Power Manager Generic Clock
- GPIO: General Purpose Input/Output
- HSB: High Speed Bus
- MPU: Memory Protection Unit
- OCD: On Chip Debug
- PB: Peripheral Bus
- PDCA: Peripheral Direct Memory Access Controller (PDC) version A
- USBB: USB On-The-GO Controller version B

4.1 Processor and architecture

4.1.1 AVR32 UC CPU

- 32-bit load/store AVR32A RISC architecture.
  - 15 general-purpose 32-bit registers.
  - 32-bit Stack Pointer, Program Counter and Link Register reside in register file.
  - Fully orthogonal instruction set.
  - Privileged and unprivileged modes enabling efficient and secure Operating Systems.
  - Innovative instruction set together with variable instruction length ensuring industry leading code density.
  - DSP extention with saturating arithmetic, and a wide variety of multiply instructions.
- 3 stage pipeline allows one instruction per clock cycle for most instructions.
  - Byte, half-word, word and double word memory access.
  - Multiple interrupt priority levels.
- MPU allows for operating systems with memory protection.

4.1.2 Debug and Test system

- IEEE1149.1 compliant JTAG and boundary scan
- Direct memory access and programming capabilities through JTAG interface
- Extensive On-Chip Debug features in compliance with IEEE-ISTO 5001-2003 (Nexus 2.0) Class 2+
  - Low-cost NanoTrace supported.
- Auxiliary port for high-speed trace information
- Hardware support for 6 Program and 2 data breakpoints
- Unlimited number of software breakpoints supported
- Advanced Program, Data, Ownership, and Watchpoint trace supported

4.1.3 Peripheral DMA Controller

- Transfers from/to peripheral to/from any memory space without intervention of the processor.
- Next Pointer Support, forbids strong real-time constraints on buffer management.
- Fifteen channels
  - Two for each USART
  - Two for each Serial Synchronous Controller
  - Two for each Serial Peripheral Interface
  - One for each ADC
  - Two for each TWI Interface

4.1.4 Bus system

- High Speed Bus (HSB) matrix with 6 Masters and 6 Slaves handled
  - Handles Requests from the CPU Data Fetch, CPU Instruction Fetch, PDCA, USBB, Ethernet Controller, CPU SAB, and to internal Flash, internal SRAM, Peripheral Bus A, Peripheral Bus B, EBI.
  - Round-Robin Arbitration (three modes supported: no default master, last accessed default master, fixed default master)
  - Burst Breaking with Slot Cycle Limit
  - One Address Decoder Provided per Master

5. Signals Description

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

The signals are multiplexed with GPIO pins as described in ”Peripheral Multiplexing on I/O lines” on page 31.

Table 5-1. Signal Description List

Signal Name	Function	Type	Active Level	Comments
Power				
VDDPLL	Power supply for PLL	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
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		

Table 6-3. BGA144 Package Pinout A1..M8

	1	2	3	4	5	6	7	8
A	VDDIO	PB07	PB05	PB02	PB03	PB01	PC00	PA28
B	PB08	GND	PB06	PB04	VDDIO	PB00	PC01	VDDPLL
C	PB09	PX33	PA29	PC02	PX28	PX26	PX22	PX21
D	PB11	PB13	PB12	PX30	PX29	PX25	PX24	PX20
E	PB10	VDDIO	PX32	PX31	VDDIO	PX27	PX23	VDDANA
F	PA30	PB14	PX34	PB16	TCK	GND	GND	PX16
G	TMS	PC03	PX36	PX35	PX37	GND	GND	PA16
H	TDO	VDDCORE	PX38	PX39	VDDIO	PA01	PA10	VDDCORE
J	TDI	PB17	PB15	PX00	PX01	PA00	PA03	PA04
K	PC05	PC04	PB19	PB20	PX02	PB29	PB30	PA02
L	PB21	GND	PB18	PB24	VDDOUT	PX04	PB31	VDDIN
M	PB22	PB23	PB25	PB26	PX03	PB27	PB28	RESET_N

Table 6-4. BGA144 Package Pinout A9..M12

	9	10	11	12
A	PA26	PA25	PA24	PA23
B	PA27	PA21	GND	PA22
C	ADVREF	GNDANA	PX19	PA19
D	PA18	PA20	DP	DM
E	PX18	PX17	VDDIO	VBUS
F	PA17	PX15	PA15	PA14
G	PA13	PA12	PA11	NC
H	PX11	PA08	VDDCORE	VDDCORE
J	PX14	PA07	PX13	PA09
K	PX08	GND	PA05	PX12
L	PX06	PX10	GND	PA06
M	PX05	PX07	PX09	VDDIO

Note: NC is not connected.

The following GPIO registers are mapped on the local bus:

Table 10-2. Local bus mapped GPIO registers

Port	Register	Mode	Local Bus Address	Access
0	Output Driver Enable Register (ODER)	WRITE	0x4000_0040	Write-only
		SET	0x4000_0044	Write-only
		CLEAR	0x4000_0048	Write-only
		TOGGLE	0x4000_004C	Write-only
	Output Value Register (OVR)	WRITE	0x4000_0050	Write-only
		SET	0x4000_0054	Write-only
		CLEAR	0x4000_0058	Write-only
		TOGGLE	0x4000_005C	Write-only
	Pin Value Register (PVR)	-	0x4000_0060	Read-only
1	Output Driver Enable Register (ODER)	WRITE	0x4000_0140	Write-only
		SET	0x4000_0144	Write-only
		CLEAR	0x4000_0148	Write-only
		TOGGLE	0x4000_014C	Write-only
	Output Value Register (OVR)	WRITE	0x4000_0150	Write-only
		SET	0x4000_0154	Write-only
		CLEAR	0x4000_0158	Write-only
		TOGGLE	0x4000_015C	Write-only
	Pin Value Register (PVR)	-	0x4000_0160	Read-only
2	Output Driver Enable Register (ODER)	WRITE	0x4000_0240	Write-only
		SET	0x4000_0244	Write-only
		CLEAR	0x4000_0248	Write-only
		TOGGLE	0x4000_024C	Write-only
	Output Value Register (OVR)	WRITE	0x4000_0250	Write-only
		SET	0x4000_0254	Write-only
		CLEAR	0x4000_0258	Write-only
		TOGGLE	0x4000_025C	Write-only
	Pin Value Register (PVR)	-	0x4000_0260	Read-only

10.4.3 SPIs

Each SPI can be connected to an internally divided clock:

Table 10-6. SPI clock connections

SPI	Source	Name	Connection
0	Internal	CLK_DIV	PBA clock or PBA clock / 32
1			

10.5 Nexus OCD AUX port connections

If the OCD trace system is enabled, the trace system will take control over a number of pins, irrespectively of the PIO configuration. Two different 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 10-7. Nexus OCD AUX port connections

Pin	AXS=0	AXS=1
EVTI_N	PB19	PA08
MDO[5]	PB16	PA27
MDO[4]	PB14	PA26
MDO[3]	PB13	PA25
MDO[2]	PB12	PA24
MDO[1]	PB11	PA23
MDO[0]	PB10	PA22
EVTO_N	PB20	PB20
MCKO	PB21	PA21
MSEO[1]	PB04	PA07
MSEO[0]	PB17	PA28

10.6 PDC handshake signals

The PDC and the peripheral modules communicate through a set of handshake signals. The following table defines the valid settings for the Peripheral Identifier (PID) in the PDC Peripheral Select Register (PSR).

Table 10-8. PDC Handshake Signals

PID Value	Peripheral module & direction
0	ADC
1	SSC - RX
2	USART0 - RX
3	USART1 - RX

Table 10-9. GPIO Controller Function Multiplexing

41	58	PA16	GPIO 16	SSC - TX_DATA	SPI1 - MOSI	EBI - ADDR[21]
42	60	PA17	GPIO 17	SSC - RX_DATA	SPI1 - MISO	EBI - ADDR[22]
43	62	PA18	GPIO 18	SSC - RX_CLOCK	SPI1 - NPCS[1]	MACB - WOL
44	64	PA19	GPIO 19	SSC - RX_FRAME_SYNC	SPI1 - NPCS[2]	
45	66	PA20	GPIO 20	EIM - EXTINT[8]	SPI1 - NPCS[3]	
51	73	PA21	GPIO 21	ADC - AD[0]	EIM - EXTINT[0]	USB - USB_ID
52	74	PA22	GPIO 22	ADC - AD[1]	EIM - EXTINT[1]	USB - USB_VBOF
53	75	PA23	GPIO 23	ADC - AD[2]	EIM - EXTINT[2]	DAC - DATA[1]
54	76	PA24	GPIO 24	ADC - AD[3]	EIM - EXTINT[3]	DAC - DATAN[1]
55	77	PA25	GPIO 25	ADC - AD[4]	EIM - SCAN[0]	EBI - NCS[0]
56	78	PA26	GPIO 26	ADC - AD[5]	EIM - SCAN[1]	EBI - ADDR[20]
57	79	PA27	GPIO 27	ADC - AD[6]	EIM - SCAN[2]	EBI - ADDR[21]
58	80	PA28	GPIO 28	ADC - AD[7]	EIM - SCAN[3]	EBI - ADDR[22]
83	122	PA29	GPIO 29	TWI - SDA	USART2 - RTS	
84	123	PA30	GPIO 30	TWI - SCL	USART2 - CTS	
65	88	PB00	GPIO 32	MACB - TX_CLK	USART2 - RTS	USART3 - RTS
66	90	PB01	GPIO 33	MACB - TX_EN	USART2 - CTS	USART3 - CTS
70	96	PB02	GPIO 34	MACB - TXD[0]	DAC - DATA[0]	
71	98	PB03	GPIO 35	MACB - TXD[1]	DAC - DATAN[0]	
72	100	PB04	GPIO 36	MACB - CRS	USART3 - CLK	EBI - NCS[3]
73	102	PB05	GPIO 37	MACB - RXD[0]	DAC - DATA[1]	
74	104	PB06	GPIO 38	MACB - RXD[1]	DAC - DATAN[1]	
75	106	PB07	GPIO 39	MACB - RX_ER		
76	111	PB08	GPIO 40	MACB - MDC		
77	113	PB09	GPIO 41	MACB - MDIO		
78	115	PB10	GPIO 42	MACB - TXD[2]	USART3 - RXD	EBI - SDCK
81	119	PB11	GPIO 43	MACB - TXD[3]	USART3 - TXD	EBI - SDCKE
82	121	PB12	GPIO 44	MACB - TX_ER	TC - CLK0	EBI - RAS
87	126	PB13	GPIO 45	MACB - RXD[2]	TC - CLK1	EBI - CAS
88	127	PB14	GPIO 46	MACB - RXD[3]	TC - CLK2	EBI - SDWE
95	134	PB15	GPIO 47	MACB - RX_DV		
96	136	PB16	GPIO 48	MACB - COL	USB - USB_ID	EBI - SDA10
98	139	PB17	GPIO 49	MACB - RX_CLK	USB - USB_VBOF	EBI - ADDR[23]
99	141	PB18	GPIO 50	MACB - SPEED	ADC - TRIGGER	PWM - PWM[6]
100	143	PB19	GPIO 51	PWM - PWM[0]	PM - GCLK[0]	EIM - SCAN[4]
1	3	PB20	GPIO 52	PWM - PWM[1]	PM - GCLK[1]	EIM - SCAN[5]
2	5	PB21	GPIO 53	PWM - PWM[2]	PM - GCLK[2]	EIM - SCAN[6]
3	6	PB22	GPIO 54	PWM - PWM[3]	PM - GCLK[3]	EIM - SCAN[7]
6	9	PB23	GPIO 55	TC - A0	USART1 - DCD	

  - Optional Manchester Encoding
- RS485 with driver control signal
- ISO7816, T = 0 or T = 1 Protocols for interfacing with smart cards
  - NACK handling, error counter with repetition and iteration limit
- IrDA modulation and demodulation
  - Communication at up to 115.2 Kbps
- Test Modes
  - Remote Loopback, Local Loopback, Automatic Echo
- SPI Mode
  - Master or Slave
  - Serial Clock Programmable Phase and Polarity
  - SPI Serial Clock (SCK) Frequency up to Internal Clock Frequency PBA/4
- Supports Connection of Two Peripheral DMA Controller Channels (PDC)
  - Offers Buffer Transfer without Processor Intervention

10.11.8 Serial Synchronous Controller

- Provides serial synchronous communication links used in audio and telecom applications (with CODECs in Master or Slave Modes, I2S, TDM Buses, Magnetic Card Reader, etc.)
- Contains an independent receiver and transmitter and a common clock divider
- Offers a configurable frame sync and data length
- Receiver and transmitter can be programmed to start automatically or on detection of different event on the frame sync signal
- Receiver and transmitter include a data signal, a clock signal and a frame synchronization signal

10.11.9 Timer Counter

- Three 16-bit Timer Counter Channels
- Wide range of functions including:
  - Frequency Measurement
  - Event Counting
  - Interval Measurement
  - Pulse Generation
  - Delay Timing
  - Pulse Width Modulation
  - Up/down Capabilities
- Each channel is user-configurable and contains:
  - Three external clock inputs
  - Five internal clock inputs
  - Two multi-purpose input/output signals
- Two global registers that act on all three TC Channels

10.11.10 Pulse Width Modulation Controller

- 7 channels, one 20-bit counter per channel
- Common clock generator, providing Thirteen Different Clocks
  - A Modulo n counter providing eleven clocks
  - Two independent Linear Dividers working on modulo n counter outputs
- Independent channel programming
  - Independent Enable Disable Commands
  - Independent Clock
  - Independent Period and Duty Cycle, with Double Bufferization
  - Programmable selection of the output waveform polarity
  - Programmable center or left aligned output waveform

10.11.11 Ethernet 10/100 MAC

- Compatibility with IEEE Standard 802.3
- 10 and 100 Mbits per second data throughput capability
- Full- and half-duplex operations
- MII or RMII interface to the physical layer
- Register Interface to address, data, status and control registers
- DMA Interface, operating as a master on the Memory Controller
- Interrupt generation to signal receive and transmit completion
- 28-byte transmit and 28-byte receive FIFOs
- Automatic pad and CRC generation on transmitted frames
- Address checking logic to recognize four 48-bit addresses
- Support promiscuous mode where all valid frames are copied to memory
- Support physical layer management through MDIO interface control of alarm and update time/calendar data

10.11.12 Audio Bitstream DAC

- Digital Stereo DAC
- Oversampled D/A conversion architecture
  - Oversampling ratio fixed 128x
  - FIR equalization filter
  - Digital interpolation filter: Comb4
  - 3rd Order Sigma-Delta D/A converters
- Digital bitstream outputs
- Parallel interface
- Connected to Peripheral DMA Controller for background transfer without CPU intervention

11. Boot Sequence

This chapter summarizes the boot sequence of the AT32UC3A. The behaviour after power-up is controlled by the Power Manager. For specific details, refer to Section 13. ”Power Manager (PM)” on page 53.

11.1 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 recieves a clock with the same frequency as the internal RC Oscillator.

11.2 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.

Table 12-7. BOD Timing

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

12.4.2 POR

Table 12-8. Electrical Characteristic

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Units.
V_{DDRR}	VDDCORE rise rate to ensure power-on-reset		0.01			V/ms
V_{SSFR}	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: $V_{RESTART}$ -> V_{POR+}	1.35	1.5	1.6	V
V_{POR-}	Falling threshold voltage: voltage when POR resets device on falling VDDCORE	Falling VDDCORE: 1.8V -> V_{POR+}	1.25	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 -> $V_{RESTART}$	-0.1		0.5	V
T_{POR}	Minimum time with VDDCORE < V_{POR-}	Falling VDDCORE: 1.8V -> 1.1V		15		us
T_{RST}	Time for reset signal to be propagated to system			200	400	us

These figures represent the power consumption measured on the power supplies.

Table 12-9. Power Consumption for Different Modes

Mode	Conditions		Typ.	Unit
Active	Typ : Ta =25 °C CPU running from flash (1). VDDIN=3.3 V. VDDCORE =1.8V. CPU clocked from PLL0 at f MHz Voltage regulator is on. XIN0 : external clock. (1) XIN1 stopped. XIN32 stopped PLL0 running All peripheral clocks activated. GPIOs on internal pull-up. JTAG unconnected with ext pull-up.	f = 12 MHz	9	mA
		f = 24 MHz	15	mA
		f = 36MHz	20	mA
		f = 50 MHz	28	mA
		f = 66 MHz	36.3	mA
Idle	Typ : Ta = 25 °C CPU running from flash (1). VDDIN=3.3 V. VDDCORE =1.8V. CPU clocked from PLL0 at f MHz Voltage regulator is on. XIN0 : external clock. XIN1 stopped. XIN32 stopped PLL0 running All peripheral clocks activated. GPIOs on internal pull-up. JTAG unconnected with ext pull-up.	f = 12 MHz	5	mA
		f = 24 MHz	10	mA
		f = 36MHz	14	mA
		f = 50 MHz	19	mA
		f = 66 MHz	25.5	mA
Frozen	Typ : Ta = 25 °C CPU running from flash (1). CPU clocked from PLL0 at f MHz Voltage regulator is on. XIN0 : external clock. XIN1 stopped. XIN32 stopped PLL0 running All peripheral clocks activated. GPIOs on internal pull-up. JTAG unconnected with ext pull-up.	f = 12 MHz	3	mA
		f = 24 MHz	6	mA
		f = 36MHz	9	mA
		f = 50 MHz	13	mA
		f = 66 MHz	16.8	mA
Standby	Typ : Ta = 25 °C CPU running from flash (1). CPU clocked from PLL0 at f MHz Voltage regulator is on. XIN0 : external clock. XIN1 stopped. XIN32 stopped PLL0 running All peripheral clocks activated. GPIOs on internal pull-up. JTAG unconnected with ext pull-up.	f = 12 MHz	1	mA
		f = 24 MHz	2	mA
		f = 36MHz	3	mA
		f = 50 MHz	4	mA
		f = 66 MHz	4.8	mA

12.7.2 Main Oscillators Characteristics

Table 12-15. Main Oscillator Characteristics

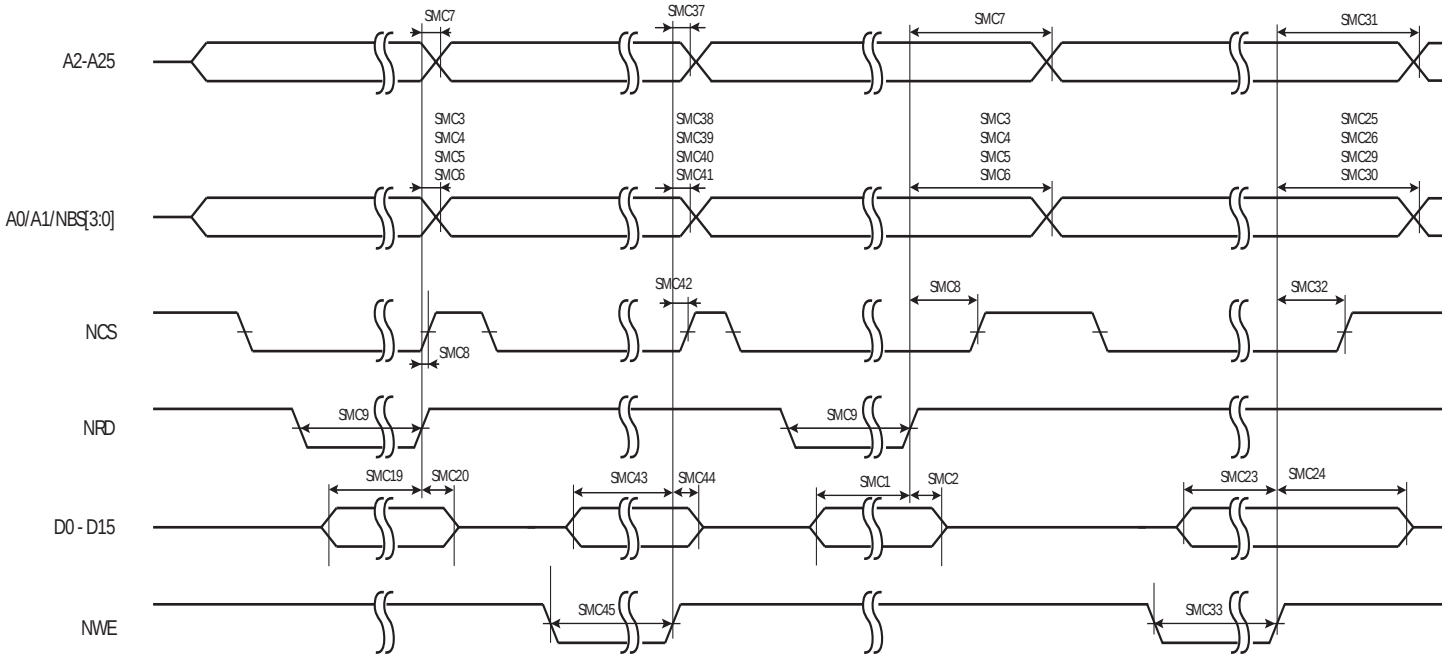
Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$1/(t_{CPMAIN})$	Crystal Oscillator Frequency		0.45		16	MHz
C_{L1}, C_{L2}	Internal Load Capacitance ($C_{L1} = C_{L2}$)			12		pF
	Duty Cycle		40	50	60	%
t_{ST}	Startup Time				TBD	ms
$1/(t_{CPXIN})$	XIN Clock Frequency	External clock			50	MHz
		Crystal	0.45		16	MHz
t_{CHXIN}	XIN Clock High Half-period		0.4 x t_{CPXIN}		0.6 x t_{CPXIN}	
t_{CLXIN}	XIN Clock Low Half-period		0.4 x t_{CPXIN}		0.6 x t_{CPXIN}	
C_{IN}	XIN Input Capacitance			7		pF

12.7.3 PLL Characteristics

Table 12-16. Phase Lock Loop Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
F_{OUT}	Output Frequency		80		240	MHz
F_{IN}	Input Frequency		4		16	MHz
I_{PLL}	Current Consumption	active mode (Fout=80Mhz)		250		µA
		active mode (Fout=240Mhz)		600		µA

Figure 12-3. SMC Signals for NRD and NRW Controlled Accesses.



12.9.1 SDRAM Signals

These timings are given for 10 pF load on SDCK and 40 pF on other signals.

Table 12-27. SDRAM Clock Signal.

Symbol	Parameter	Max[1]	Units
$1/(t_{CPSDCK})$	SDRAM Controller Clock Frequency	$1/(t_{cpcpu})$	MHz

Note: 1. The maximum frequency of the SDRAMC interface is the same as the max frequency for the HSB.

Table 12-28. SDRAM Clock Signal.

Symbol	Parameter	Min	Units
$SDRAMC_1$	SDCKE High before SDCK Rising Edge	7.4	ns
$SDRAMC_2$	SDCKE Low after SDCK Rising Edge	3.2	
$SDRAMC_3$	SDCKE Low before SDCK Rising Edge	7	
$SDRAMC_4$	SDCKE High after SDCK Rising Edge	2.9	
$SDRAMC_5$	SDCS Low before SDCK Rising Edge	7.5	
$SDRAMC_6$	SDCS High after SDCK Rising Edge	1.6	
$SDRAMC_7$	RAS Low before SDCK Rising Edge	7.2	
$SDRAMC_8$	RAS High after SDCK Rising Edge	2.3	
$SDRAMC_9$	SDA10 Change before SDCK Rising Edge	7.6	
$SDRAMC_{10}$	SDA10 Change after SDCK Rising Edge	1.9	

Table 12-28. SDRAM Clock Signal.

Symbol	Parameter	Min	Units
$SDRAMC_{11}$	Address Change before SDCK Rising Edge	6.2	ns
$SDRAMC_{12}$	Address Change after SDCK Rising Edge	2.2	
$SDRAMC_{13}$	Bank Change before SDCK Rising Edge	6.3	
$SDRAMC_{14}$	Bank Change after SDCK Rising Edge	2.4	
$SDRAMC_{15}$	CAS Low before SDCK Rising Edge	7.4	
$SDRAMC_{16}$	CAS High after SDCK Rising Edge	1.9	
$SDRAMC_{17}$	DQM Change before SDCK Rising Edge	6.4	
$SDRAMC_{18}$	DQM Change after SDCK Rising Edge	2.2	
$SDRAMC_{19}$	D0-D15 in Setup before SDCK Rising Edge	9	
$SDRAMC_{20}$	D0-D15 in Hold after SDCK Rising Edge	0	
$SDRAMC_{23}$	SDWE Low before SDCK Rising Edge	7.6	
$SDRAMC_{24}$	SDWE High after SDCK Rising Edge	1.8	
$SDRAMC_{25}$	D0-D15 Out Valid before SDCK Rising Edge	7.1	
$SDRAMC_{26}$	D0-D15 Out Valid after SDCK Rising Edge	1.5	

Workaround/fix
The same PID should not be assigned to more than one channel.

15.3.7 GPIO

1. **Some GPIO VIH (input high voltage) are 3.6V max instead of 5V tolerant**
Only 11 GPIOs remain 5V tolerant (VIHmax=5V):PB01, PB02, PB03, PB10, PB19, PB20, PB21, PB22, PB23, PB27, PB28**.**
Workaround/fix
None.

15.3.8 USART

1. **ISO7816 info register US_NER cannot be read**
The NER register always returns zero.
Fix/Workaround
None.

15.3.9 TWI

1. **The TWI RXRDY flag in SR register is not reset when a software reset is performed.**
Fix/Workaround
After a Software Reset, the register TWI RHR must be read.

15.3.10 SDRAMC

1. **Code execution from external SDRAM does not work**
Code execution from SDRAM does not work.

Fix/Workaround
Do not run code from SDRAM.

15.3.11 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. **Exceptions when system stack is protected by MPU**
RETS behaves incorrectly when MPU is enabled and MPU is configured so that system stack is not readable in unprivileged mode.
Fix/Woraround
Workaround 1: Make system stack readable in unprivileged mode,
or
Workaround 2: Return from supervisor mode using rete instead of rets. This requires :
1. Changing the mode bits from 001b to 110b 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 described in general as not safe in the UC technical reference guide, it is safe in this very

15.5 Rev. E

15.5.1 SPI

1. **SPI FDIV option does not work**
   Selecting clock signal using FDIV = 1 does not work as specified.

   Fix/Workaround
   Do not set FDIV = 1.

2. **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.

3. **SPI Bad serial clock generation on 2nd chip select when SCBR=1, CPOL=1 and CNCPHA=0**
   When multiple CS are in use, if one of the baudrate equals to 1 and one of the others doesn't equal to 1, and CPOL=1 and CPHA=0, then an additional pulse will be generated on SCK.

   Fix/Workaround
   When multiple CS are in use, if one of the baudrate equals to 1, the other must also equal 1 if CPOL=1 and CPHA=0.

4. **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 befin and RXREADY will now behave as expected.

5. **SPI CSNAAT bit 2 in register CSR0...CSR3 is not available.**
   Fix/Workaround

   Do not use this bit.

6. **SPI disable does not work in SLAVE mode.**
   Fix/Workaround
   Read the last received data, then perform a Software Reset.

7. **SPI Bad Serial Clock Generation on 2nd chip_select when SCBR = 1, CPOL=1 and NCPHA=0**
   When multiple CS are in use, if one of the baudrate equals to 1 and one of the others doesn't equal to 1, and CPOL=1 and CPHA=0, then an aditional pulse will be generated on SCK.

6. **CPU Cycle Counter does not reset the COUNT system register on COMPARE match.**
The device revision E 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.

7. **Memory Protection Unit (MPU) is non functional.**
Fix/Workaround
Do not use the MPU.

8. **The following alternate GPIO function C are not available in revE**
MACB-WOL on GPIO9 (PA09), MACB-WOL on GPIO18 (PA18), USB-USB_ID on GPIO21 (PA21), USB-USB_VBOF on GPIO22 (PA22), and all function B and C on GPIO70 to GPIO101 (PX00 to PX39).
Fix/Workaround
Do not use these alternate B and C functions on the listed GPIO pins.

9. **Clock connection table on Rev E**

Here is the table of Rev E

Figure 15-1. Timer/Counter clock connections on RevE

Source	Name	Connection
Internal	TIMER_CLOCK1	32 KHz Oscillator
	TIMER_CLOCK2	PBA Clock / 4
	TIMER_CLOCK3	PBA Clock / 8
	TIMER_CLOCK4	PBA Clock / 16
	TIMER_CLOCK5	PBA Clock / 32
External	XC0	
	XC1	
	XC2	

10. **Local Bus fast GPIO not available in RevE.**
Fix/Workaround
Do not use on this silicon revision.

11. **Spurious interrupt may corrupt core SR mode to exception**
If the rules listed in the chapter `Masking interrupt requests in peripheral modules' of the AVR32UC Technical Reference Manual are not followed, a spurious interrupt may occur. An interrupt context will be pushed onto the stack while the core SR mode will indicate an exception. A RETE instruction would then corrupt the stack..

Fix/Workaround
Follow the rules of the AVR32UC Technical Reference Manual. To increase software robustness, if an exception mode is detected at the beginning of an interrupt handler, change the stack interrupt context to an exception context and issue a RETE instruction.

12. **CPU cannot operate on a divided slow clock (internal RC oscillator)**
Fix/Workaround
Do not run the CPU on a divided slow clock.

13. **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.

14. **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.

15. **Exceptions when system stack is protected by MPU**
RETS behaves incorrectly when MPU is enabled and MPU is configured so that system stack is not readable in unprivileged mode.
Fix/Woraround
Workaround 1: Make system stack readable in unprivileged mode,
or
Workaround 2: Return from supervisor mode using rete instead of rets. This requires :
1. Changing the mode bits from 001b to 110b 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 described in general as not safe in the UC technical reference guide, it is safe in this very specific case.
2. Execute the RETE instruction.

15.5.6 SDRAMC

1. **Code execution from external SDRAM does not work**
Code execution from SDRAM does not work.

Fix/Workaround
Do not run code from SDRAM.

2. **SDRAM SDCKE rise at the same time as SDCK while exiting self-refresh mode**
SDCKE rise at the same time as SDCK while exiting self-refresh mode.

Fix/Workaround
None.

15.5.7 USART

1. **USART Manchester Encoder Not Working**
Manchester encoding/decoding is not working.

Fix/Workaround
Do not use manchester encoding.