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	66MHz
Connectivity	CANbus, Ethernet, I²C, IrDA, LINbus, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	45
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	32K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 11x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at32uc3c2128c-a2ut

- One 4-Channel 20-bit Pulse Width Modulation Controller (PWM)
  - Complementary outputs, with Dead Time Insertion
  - Output Override and Fault Protection
- Two Quadrature Decoders
- One 16-channel 12-bit Pipelined Analog-To-Digital Converter (ADC)
  - Dual Sample and Hold Capability Allowing 2 Synchronous Conversions
  - Single-Ended and Differential Channels, Window Function
- Two 12-bit Digital-To-Analog Converters (DAC), with Dual Output Sample System
- Four Analog Comparators
- Six 16-bit Timer/Counter (TC) Channels
  - External Clock Inputs, PWM, Capture and Various Counting Capabilities
- One Peripheral Event Controller
  - Trigger Actions in Peripherals Depending on Events Generated from Peripherals or from Input Pins
  - Deterministic Trigger
  - 34 Events and 22 Event Actions
- Five Universal Synchronous/Asynchronous Receiver/Transmitters (USART)
  - Independent Baudrate Generator, Support for SPI, LIN, IrDA and ISO7816 interfaces
  - Support for Hardware Handshaking, RS485 Interfaces and Modem Line
- Two Master/Slave Serial Peripheral Interfaces (SPI) with Chip Select Signals
- One Inter-IC Sound (I2S) Controller
  - Compliant with I2S Bus Specification
  - Time Division Multiplexed mode
- Three Master and Three Slave Two-Wire Interfaces (TWI), 400kbit/s I^2C-compatible
- QTouch® Library Support
  - Capacitive Touch Buttons, Sliders, and Wheels
  - QTouch® and QMatrix® Acquisition
- On-Chip Non-intrusive Debug System
  - Nexus Class 2+, Runtime Control, Non-Intrusive Data and Program Trace
  - aWire™ single-pin programming trace and debug interface muxed with reset pin
  - NanoTrace™ provides trace capabilities through JTAG or aWire interface
- 3 package options
  - 64-pin QFN/TQFP (45 GPIO pins)
  - 100-pin TQFP (81 GPIO pins)
  - 144-pin LQFP (123 GPIO pins)
- Two operating voltage ranges:
  - Single 5V Power Supply
  - Single 3.3V Power Supply

1. Description

The AT32UC3C is a complete System-On-Chip microcontroller based on the AVR32UC RISC processor running at frequencies up to 66 MHz. AVR32UC is a high-performance 32-bit RISC microprocessor core, designed for cost-sensitive embedded applications, with particular emphasis on low power consumption, high code density and high performance.

The processor implements a Memory Protection Unit (MPU) and a fast and flexible interrupt controller for supporting modern operating systems and real-time operating systems. Using the Secure Access Unit (SAU) together with the MPU provides the required security and integrity.

Higher computation capabilities are achievable either using a rich set of DSP instructions or using the floating-point instructions.

The AT32UC3C incorporates on-chip Flash and SRAM memories for secure and fast access. For applications requiring additional memory, an external memory interface is provided on AT32UC3C0 derivatives.

The Memory Direct Memory Access controller (MDMA) enables transfers of block of data from memories to memories without processor involvement.

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

The AT32UC3C incorporates on-chip Flash and SRAM memories for secure and fast access. The FlashVault technology allows secure libraries to be programmed into the device. The secure libraries can be executed while the CPU is in Secure State, but not read by non-secure software in the device. The device can thus be shipped to end custumers, who are able to program their own code into the device, accessing the secure libraries, without any risk of compromising the proprietary secure code.

The Power Manager improves design flexibility and security. Power monitoring is supported by on-chip Power-On Reset (POR), Brown-Out Detectors (BOD18, BOD33, BOD50). The CPU runs from the on-chip RC oscillators, the PLLs, or the Multipurpose Oscillators. The Asynchronous Timer (AST) combined with the 32 KHz oscillator keeps track of the time. The AST can operate in counter or calendar mode.

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

The PWM module provides four channels with many configuration options including polarity, edge alignment and waveform non overlap control. The PWM channels can operate independently, with duty cycles set independently from each other, or in interlinked mode, with multiple channels updated at the same time. It also includes safety feature with fault inputs and the ability to lock the PWM configuration registers and the PWM pin assignment.

The AT32UC3C also features many communication interfaces for communication intensive applications. In addition to standard serial interfaces like UART, SPI or TWI, other interfaces like flexible CAN, USB and Ethernet MAC are available. The USART supports different communication modes, like SPI mode and LIN mode.

The Inter-IC Sound Controller (I2SC) provides a 5-bit wide, bidirectional, synchronous, digital audio link with off-chip audio devices. The controller is compliant with the I2S bus specification.

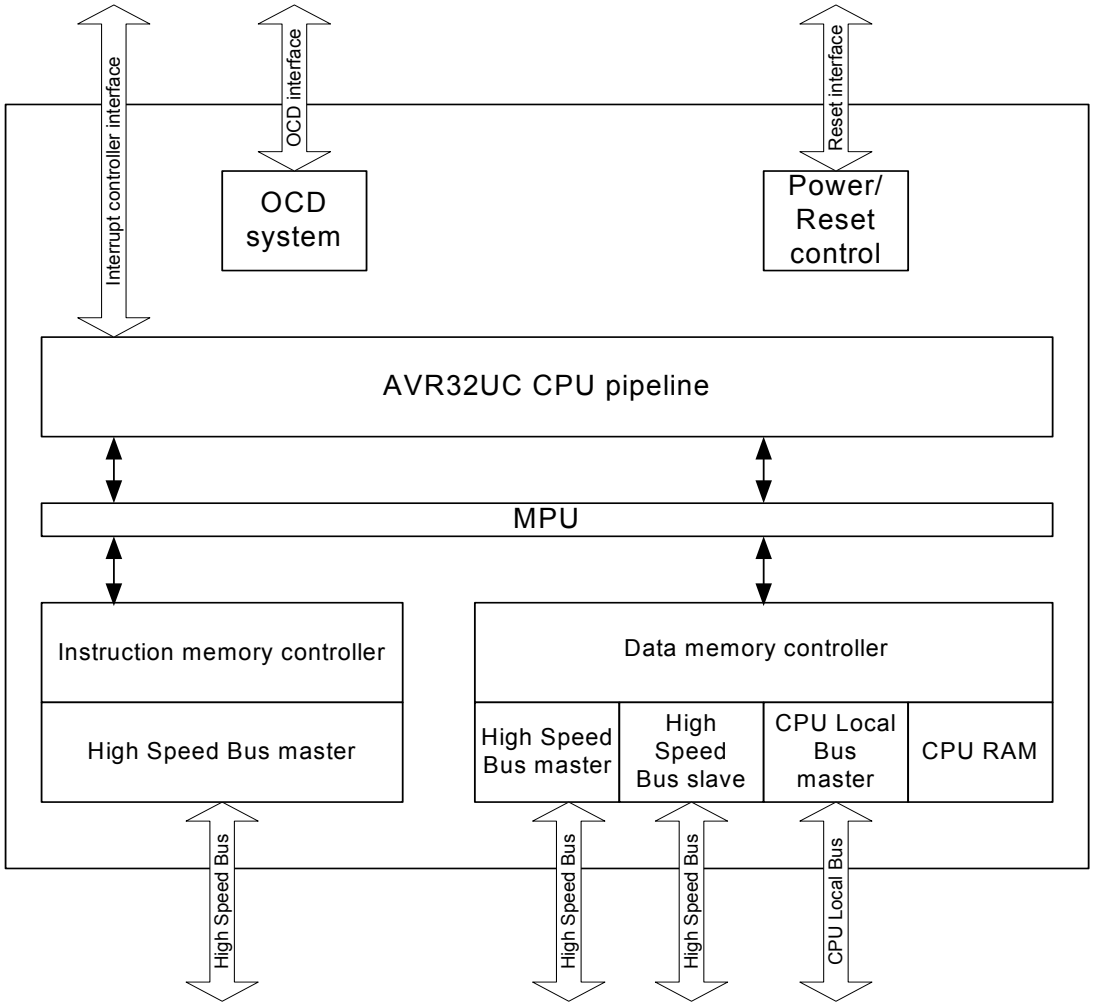
Table 3-1. GPIO Controller Function Multiplexing

TQFP / QFN 64	TQFP 100	LQFP 144	PIN	GPIO	Supply	Pin Type (1)	GPIO function					
							A	B	C	D	E	F
16	25	36	PA19	19	VDDANA	x1/x2	ADCIN8	EIC - EXTINT[1]				
19	28	39	PA20	20	VDDANA	x1/x2	ADCIN9	AC0AP0	AC0AP0 or DAC0A			
20	29	40	PA21	21	VDDANA	x1/x2	ADCIN10	AC0BN0	AC0BN0 or DAC0B			
21	30	41	PA22	22	VDDANA	x1/x2	ADCIN11	AC0AN0	PEVC - PAD_EVT [4]		MACB - SPEED	
22	31	42	PA23	23	VDDANA	x1/x2	ADCIN12	AC0BP0	PEVC - PAD_EVT [5]		MACB - WOL	
	32	43	PA24	24	VDDANA	x1/x2	ADCIN13	SPI1 - NPCS[2]				
	33	44	PA25	25	VDDANA	x1/x2	ADCIN14	SPI1 - NPCS[3]	EIC - EXTINT[0]			
		45	PA26	26	VDDANA	x1/x2	AC0AP1	EIC - EXTINT[1]				
		46	PA27	27	VDDANA	x1/x2	AC0AN1	EIC - EXTINT[2]				
		47	PA28	28	VDDANA	x1/x2	AC0BP1	EIC - EXTINT[3]				
		48	PA29	29	VDDANA	x1/x2	AC0BN1	EIC - EXTINT[0]				
62	96	140	PB00	32	VDDIO1	x1	USART0 - CLK	CANIF - RXLINE[1]	EIC - EXTINT[8]	PEVC - PAD_EVT [10]		
63	97	141	PB01	33	VDDIO1	x1		CANIF - TXLINE[1]		PEVC - PAD_EVT [11]		
	99	143	PB02	34	VDDIO1	x1		USBC - ID	PEVC - PAD_EVT [6]	TC1 - A1		
	100	144	PB03	35	VDDIO1	x1		USBC - VBOF	PEVC - PAD_EVT [7]			
	7	7	PB04	36	VDDIO1	x1/x2	SPI1 - MOSI	CANIF - RXLINE[0]	QDEC1 - QEPI		MACB - TXD[2]	
	8	8	PB05	37	VDDIO1	x1/x2	SPI1 - MISO	CANIF - TXLINE[0]	PEVC - PAD_EVT [12]	USART3 - CLK	MACB - TXD[3]	
	9	9	PB06	38	VDDIO1	x2/x4	SPI1 - SCK		QDEC1 - QEPA	USART1 - CLK	MACB - TX_ER	
		10	PB07	39	VDDIO1	x1/x2	SPI1 - NPCS[0]	EIC - EXTINT[2]	QDEC1 - QEPB		MACB - RX_DV	
		11	PB08	40	VDDIO1	x1/x2	SPI1 - NPCS[1]	PEVC - PAD_EVT [1]	PWM - PWML[0]		MACB - RXD[0]	
		12	PB09	41	VDDIO1	x1/x2	SPI1 - NPCS[2]		PWM - PWMH[0]		MACB - RXD[1]	
		13	PB10	42	VDDIO1	x1/x2	USART1 - DTR	SPI0 - MOSI	PWM - PWML[1]			

Table 3-1. GPIO Controller Function Multiplexing

TQFP / QFN 64	TQFP 100	LQFP 144	PIN	GPIO	Supply	Pin Type (1)	GPIO function					
							A	B	C	D	E	F
		14	PB11	43	VDDIO1	x1/x2	USART1 - DSR	SPI0 - MISO	PWM - PWMH[1]			
		15	PB12	44	VDDIO1	x1/x2	USART1 - DCD	SPI0 - SCK	PWM - PWML[2]			
		16	PB13	45	VDDIO1	x1/x2	USART1 - RI	SPI0 - NPCS[0]	PWM - PWMH[2]		MACB - RX_ER	
		17	PB14	46	VDDIO1	x1/x2	USART1 - RTS	SPI0 - NPCS[1]	PWM - PWML[3]		MACB - MDC	
		18	PB15	47	VDDIO1	x1/x2	USART1 - CTS	USART1 - CLK	PWM - PWMH[3]		MACB - MDIO	
		19	PB16	48	VDDIO1	x1/x2	USART1 - RXD	SPI0 - NPCS[2]	PWM - EXT_ FAULTS[0]		CANIF - RXLINE[0]	
		20	PB17	49	VDDIO1	x1/x2	USART1 - TXD	SPI0 - NPCS[3]	PWM - EXT_ FAULTS[1]		CANIF - TXLINE[0]	
		57	PB18	50	VDDIO2	x1/x2	TC0 - CLK2		EIC - EXTINT[4]			
	42	58	PB19	51	VDDIO2	x1/x2	TC0 - A0	SPI1 - MOSI	IISC - ISDO		MACB - CRS	
	43	59	PB20	52	VDDIO2	x1/x2	TC0 - B0	SPI1 - MISO	IISC - ISDI	ACIFA1 - ACAOUT	MACB - COL	
	44	60	PB21	53	VDDIO2	x2/x4	TC0 - CLK1	SPI1 - SCK	IISC - IMCK	ACIFA1 - ACBOUT	MACB - RXD[2]	
	45	61	PB22	54	VDDIO2	x1/x2	TC0 - A1	SPI1 - NPCS[3]	IISC - ISCK	SCIF - GCLK[0]	MACB - RXD[3]	
	46	62	PB23	55	VDDIO2	x1/x2	TC0 - B1	SPI1 - NPCS[2]	IISC - IWS	SCIF - GCLK[1]	MACB - RX_CLK	
		63	PB24	56	VDDIO2	x1/x2	TC0 - CLK0	SPI1 - NPCS[1]				
		64	PB25	57	VDDIO2	x1/x2	TC0 - A2	SPI1 - NPCS[0]	PEVC - PAD_EVT [8]			
		65	PB26	58	VDDIO2	x2/x4	TC0 - B2	SPI1 - SCK	PEVC - PAD_EVT [9]		MACB - TX_EN	
		66	PB27	59	VDDIO2	x1/x2	QDEC0 - QEPA	SPI1 - MISO	PEVC - PAD_EVT [10]	TC1 - CLK0	MACB - TXD[0]	
		67	PB28	60	VDDIO2	x1/x2	QDEC0 - QEPB	SPI1 - MOSI	PEVC - PAD_EVT [11]	TC1 - B0	MACB - TXD[1]	
		68	PB29	61	VDDIO2	x1/x2	QDEC0 - QEPI	SPI0 - NPCS[0]	PEVC - PAD_EVT [12]	TC1 - A0		
31	47	69	PB30	62	VDDIO2	x1						
32	48	70	PB31	63	VDDIO2	x1						
	49	71	PC00	64	VDDIO2	x1/x2	USBC - ID	SPI0 - NPCS[1]	USART2 - CTS	TC1 - B2	CANIF - TXLINE[1]	
	50	72	PC01	65	VDDIO2	x1/x2	USBC - VBOF	SPI0 - NPCS[2]	USART2 - RTS	TC1 - A2	CANIF - RXLINE[1]	

Figure 4-1. Overview of the AVR32UC CPU



4.3.1 Pipeline Overview

AVR32UC has three pipeline stages, Instruction Fetch (IF), Instruction Decode (ID), and Instruction Execute (EX). The EX stage is split into three parallel subsections, one arithmetic/logic (ALU) section, one multiply (MUL) section, and one load/store (LS) section.

Instructions are issued and complete in order. Certain operations require several clock cycles to complete, and in this case, the instruction resides in the ID and EX stages for the required number of clock cycles. Since there is only three pipeline stages, no internal data forwarding is required, and no data dependencies can arise in the pipeline.

Figure 4-2 on page 28 shows an overview of the AVR32UC pipeline stages.

4.4 Programming Model

4.4.1 Register File Configuration

The AVR32UC register file is shown below.

Figure 4-3. The AVR32UC Register File

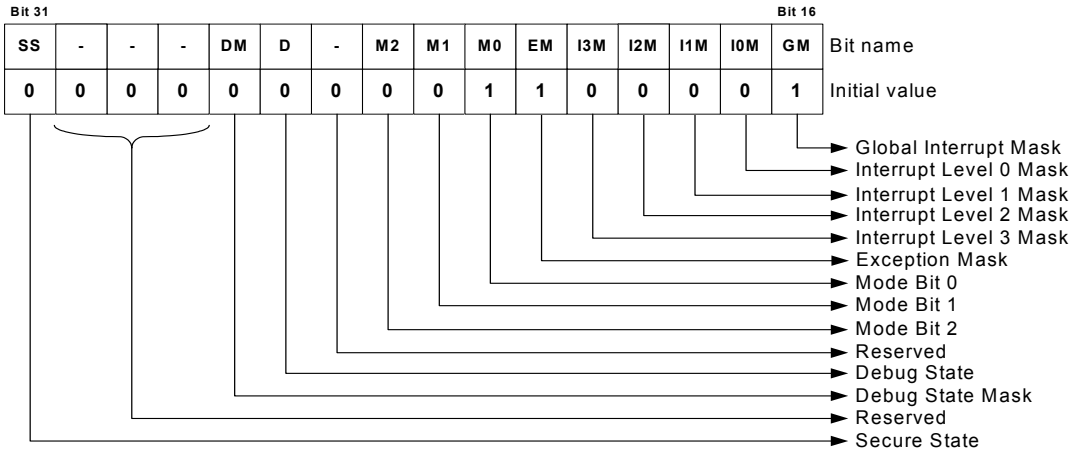
Application	Supervisor	INT0	INT1	INT2	INT3	Exception	NMI	Secure
Bit 31 Bit 0	Bit 31 Bit 0	Bit 31 Bit 0	Bit 31 Bit 0	Bit 31 Bit 0	Bit 31 Bit 0	Bit 31 Bit 0	Bit 31 Bit 0	Bit 31 Bit 0
PC	PC	PC	PC	PC	PC	PC	PC	PC
LR	LR	LR	LR	LR	LR	LR	LR	LR
SP_APP	SP_SYS	SP_SYS	SP_SYS	SP_SYS	SP_SYS	SP_SYS	SP_SYS	SP_SEC
R12	R12	R12	R12	R12	R12	R12	R12	R12
R11	R11	R11	R11	R11	R11	R11	R11	R11
R10	R10	R10	R10	R10	R10	R10	R10	R10
R9	R9	R9	R9	R9	R9	R9	R9	R9
R8	R8	R8	R8	R8	R8	R8	R8	R8
R7	R7	R7	R7	R7	R7	R7	R7	R7
R6	R6	R6	R6	R6	R6	R6	R6	R6
R5	R5	R5	R5	R5	R5	R5	R5	R5
R4	R4	R4	R4	R4	R4	R4	R4	R4
R3	R3	R3	R3	R3	R3	R3	R3	R3
R2	R2	R2	R2	R2	R2	R2	R2	R2
R1	R1	R1	R1	R1	R1	R1	R1	R1
R0	R0	R0	R0	R0	R0	R0	R0	R0
SR	SR	SR	SR	SR	SR	SR	SR	SR

SS_STATUS
SS_ADRF
SS_ADRR
SS_ADR0
SS_ADR1
SS_SP_SYS
SS_SP_APP
SS_RAR
SS_RSR

4.4.2 Status Register Configuration

The Status Register (SR) is split into two halfwords, one upper and one lower, see Figure 4-4 and Figure 4-5. The lower word contains the C, Z, N, V, and Q condition code flags and the R, T, and L bits, while the upper halfword contains information about the mode and state the processor executes in. Refer to the *AVR32 Architecture Manual* for details.

Figure 4-4. The Status Register High Halfword



4.5.3 Supervisor Calls

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

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

4.5.4 Debug Requests

The AVR32 architecture defines a dedicated Debug mode. When a debug request is received by the core, Debug mode is entered. Entry into Debug mode can be masked by the DM bit in the status register. Upon entry into Debug mode, hardware sets the SR.D bit and jumps to the Debug Exception handler. By default, Debug mode executes in the exception context, but with dedicated Return Address Register and Return Status Register. These dedicated registers remove the need for storing this data to the system stack, thereby improving debuggability. The Mode bits in the Status Register can freely be manipulated in Debug mode, to observe registers in all contexts, while retaining full privileges.

Debug mode is exited by executing the *retd* instruction. This returns to the previous context.

4.5.5 Entry Points for Events

Several different event handler entry points exist. In AVR32UC, the reset address is 0x80000000. This places the reset address in the boot flash memory area.

TLB miss exceptions and *scall* have a dedicated space relative to EVBA where their event handler can be placed. This speeds up execution by removing the need for a jump instruction placed at the program address jumped to by the event hardware. All other exceptions have a dedicated event routine entry point located relative to EVBA. The handler routine address identifies the exception source directly.

AVR32UC uses the ITLB and DTLB protection exceptions to signal a MPU protection violation. ITLB and DTLB miss exceptions are used to signal that an access address did not map to any of the entries in the MPU. TLB multiple hit exception indicates that an access address did map to multiple TLB entries, signalling an error.

All interrupt requests have entry points located at an offset relative to EVBA. This autovector offset is specified by an interrupt controller. The programmer must make sure that none of the autovector offsets interfere with the placement of other code. The autovector offset has 14 address bits, giving an offset of maximum 16384 bytes.

Special considerations should be made when loading EVBA with a pointer. Due to security considerations, the event handlers should be located in non-writeable flash memory, or optionally in a privileged memory protection region if an MPU is present.

If several events occur on the same instruction, they are handled in a prioritized way. The priority ordering is presented in Table 4-4 on page 38. If events occur on several instructions at different locations in the pipeline, the events on the oldest instruction are always handled before any events on any younger instruction, even if the younger instruction has events of higher priority

5. Memories

5.1 Embedded Memories

- Internal High-Speed Flash (See Table 5-1 on page 40)
  - 512 Kbytes
  - 256 Kbytes
  - 128 Kbytes
  - 64 Kbytes
    - 0 Wait State Access at up to 33 MHz in Worst Case Conditions
    - 1 Wait State Access at up to 66 MHz in Worst Case Conditions
    - Pipelined Flash Architecture, allowing burst reads from sequential Flash locations, hiding penalty of 1 wait state access
    - Pipelined Flash Architecture typically reduces the cycle penalty of 1 wait state operation to only 15% compared to 0 wait state operation
    - 100 000 Write Cycles, 15-year Data Retention Capability
    - Sector Lock Capabilities, Bootloader Protection, Security Bit
    - 32 Fuses, Erased During Chip Erase
    - User Page For Data To Be Preserved During Chip Erase
- Internal High-Speed SRAM, Single-cycle access at full speed (See Table 5-1 on page 40)
  - 64 Kbytes
  - 32 Kbytes
  - 16 Kbytes
- Supplementary Internal High-Speed System SRAM (HSB RAM), Single-cycle access at full speed
  - Memory space available on System Bus for peripherals data.
  - 4 Kbytes

6.2 Startup Considerations

This chapter summarizes the boot sequence of the AT32UC3C. The behavior after power-up is controlled by the Power Manager. For specific details, refer to the Power Manager chapter.

6.2.1 Starting of clocks

At power-up, the BOD33 and the BOD18 are enabled. The device will be held in a reset state by the power-up circuitry, until the VDDIN_33 (resp. VDDCORE) has reached the reset threshold of the BOD33 (resp BOD18). Refer to the Electrical Characteristics for the BOD thresholds. Once the power has stabilized, the device will use the System RC Oscillator (RCSYS, 115KHz typical frequency) as clock source. The BOD18 and BOD33 are kept enabled or are disabled according to the fuse settings (See the Fuse Setting section in the Flash Controller chapter).

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 receive a clock with the same frequency as the internal RC Oscillator.

6.2.2 Fetching of initial instructions

After reset has been released, the AVR32UC CPU starts fetching instructions from the reset address, which is 0x8000_0000. This address points to the first address in the internal Flash.

The internal Flash uses VDDIO voltage during read and write operations. It is recommended to use the BOD33 to monitor this voltage and make sure the VDDIO is above the minimum level (3.0V).

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 7-2. Supply Rise Rates and Order

Symbol	Parameter	Rise Rate		
		Min	Max	Comment
V_{VDDIN_5}	DC supply internal 3.3V regulator	0.01 V/ms	1.25 V/us	
V_{VDDIN_33}	DC supply internal 1.8V regulator	0.01 V/ms	1.25 V/us	
V_{VDDIO1} V_{VDDIO2} V_{VDDIO3}	DC supply peripheral I/O	0.01 V/ms	1.25 V/us	Rise after or at the same time as VDDIN_5, VDDIN_33
V_{VDDANA}	DC supply peripheral I/O and analog part	0.01 V/ms	1.25 V/us	Rise after or at the same time as VDDIN_5, VDDIN_33

7.3 Maximum Clock Frequencies

These parameters are given in the following conditions:

- $V_{VDDCORE}$ > 1.85V
- Temperature = -40°C to 85°C

Table 7-3. Clock Frequencies

Symbol	Parameter	Conditions	Min	Max	Units
f_{CPU}	CPU clock frequency			66	MHz
f_{PBA}	PBA clock frequency			66	MHz
f_{PBB}	PBB clock frequency			66	MHz
f_{PBC}	PBC clock frequency			66	MHz
f_{GCLK0}	GCLK0 clock frequency	Generic clock for USBC		50(1)	MHz
f_{GCLK1}	GCLK1 clock frequency	Generic clock for CANIF		66(1)	MHz
f_{GCLK2}	GCLK2 clock frequency	Generic clock for AST		80(1)	MHz
f_{GCLK4}	GCLK4 clock frequency	Generic clock for PWM		133(1)	MHz
f_{GCLK11}	GCLK11 clock frequency	Generic clock for IISC		50(1)	MHz

Note: 1. These values are based on simulation and characterization of other AVR microcontrollers manufactured in the same process technology. These values are not covered by test limits in production.

7.4 Power Consumption

The values in Table 7-4 are measured values of power consumption under the following conditions, except where noted:

- Operating conditions core supply (Figure 7-1)
  - V_{VDDIN_5} = V_{VDDIN_33} = 3.3V
  - $V_{VDDCORE}$ = 1.85V, supplied by the internal regulator
  - V_{VDDIO1} = V_{VDDIO2} = V_{VDDIO3} = 3.3V
  - V_{VDDANA} = 3.3V

7.5 I/O Pin Characteristics

Table 7-6. Normal I/O Pin Characteristics[1]

Symbol	Parameter	Condition		Min	Typ	Max	Units
R_{PULLUP}	Pull-up resistance	V_{VDD} = 3V		5		26	kOhm
		V_{VDD} = 5V		5		16	kOhm
$R_{PULLDOWN}$	Pull-down resistance			2		16	kOhm
V_{IL}	Input low-level voltage	V_{VDD} = 3V				$0.3*V_{VDDIO}$	V
		V_{VDD} = 4.5V				$0.3*V_{VDDIO}$	
V_{IH}	Input high-level voltage	V_{VDD} = 3.6V		$0.7*V_{VDDIO}$			V
		V_{VDD} = 5.5V		$0.7*V_{VDDIO}$			
V_{OL}	Output low-level voltage	I_{OL} = -3.5mA, pin drive x1[2]				0.45	V
		I_{OL} = -7mA, pin drive x2[2]					
		I_{OL} = -14mA, pin drive x4[2]					
V_{OH}	Output high-level voltage	I_{OH} = 3.5mA, pin drive x1[2]		V_{VDD} - 0.8			V
		I_{OH} = 7mA, pin drive x2[2]					
		I_{OH} = 14mA, pin drive x4[2]					
f_{MAX}	Output frequency[3]	V_{VDD} = 3.0V	load = 10pF, pin drive x1[2]			35	MHz
			load = 10pF, pin drive x2[2]			55	
			load = 10pF, pin drive x4[2]			70	
			load = 30pF, pin drive x1[2]			15	
			load = 30pF, pin drive x2[2]			30	
			load = 30pF, pin drive x4[2]			45	
		V_{VDD} =4.5V	load = 10pF, pin drive x1[2]			50	
			load = 10pF, pin drive x2[2]			80	
			load = 10pF, pin drive x4[2]			95	
			load = 30pF, pin drive x1[2]			25	
			load = 30pF, pin drive x2[2]			40	
			load = 30pF, pin drive x4[2]			65	

Table 7-8. Crystal Oscillator Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{OUT}	Crystal oscillator frequency		0.4		20	MHz
C_i	Internal equivalent load capacitance			1.7		pF
$t_{STARTUP}$	Startup time	f_{OUT} = 8MHz SCIF.OSCCTRL.GAIN = 1(1)		975		us
		f_{OUT} = 16MHz SCIF.OSCCTRL.GAIN = 2(1)		1100		us

Notes: 1. Please refer to the SCIF chapter for details.

7.6.2 32KHz Crystal Oscillator (OSC32K) Characteristics

7.6.2.1 Digital Clock Characteristics

The following table describes the characteristics for the oscillator when a digital clock is applied on XIN32.

Table 7-9. Digital 32KHz Clock Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Units
f_{CPXIN}	XIN32 clock frequency			32.768	5000	KHz
t_{CPXIN}	XIN32 clock period		200			ns
t_{CHXIN}	XIN32 clock high half-priod		0.4 x t_{CPXIN}		0.6 x t_{CPXIN}	ns
t_{CLXIN}	XIN32 clock low half-priod		0.4 x t_{CPXIN}		0.6 x t_{CPXIN}	ns
C_{IN}	XIN32 input capacitance			2		pF

7.6.2.2 Crystal Oscillator Characteristics

Figure 7-2 and the equation above also applies to the 32KHz oscillator connection. The user must choose a crystal oscillator where the crystal load capacitance C_L is within the range given in the table. The exact value of C_L can then be found in the crystal datasheet..

Table 7-10. 32 KHz Crystal Oscillator Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{OUT}	Crystal oscillator frequency			32 768		Hz
$t_{STARTUP}$	Startup time	R_S = 50 kOhm, C_L = 12.5pF		2		s
C_L	Crystal load capacitance		6		15	pF
C_i	Internal equivalent load capacitance			1.4		pF

7.7 Flash Characteristics

Table 7-15 gives the device maximum operating frequency depending on the number of flash wait states. The FSW bit in the FLASHC FSR register controls the number of wait states used when accessing the flash memory.

Table 7-15. Maximum Operating Frequency

Flash Wait States	Read Mode	Maximum Operating Frequency
0	1 cycle	33 MHz
1	2 cycles	66 MHz

Table 7-16. Flash Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
t_{FPP}	Page programming time	f_{CLK_HSB} = 66 MHz		4.3		ms
t_{FPE}	Page erase time			4.3		
t_{FFP}	Fuse programming time			0.6		
t_{FEA}	Full chip erase time (EA)			4.9		
t_{FCE}	JTAG chip erase time (CHIP_ERASE)	f_{CLK_HSB} = 115 kHz		640		

Table 7-17. Flash Endurance and Data Retention

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
N_{FARRAY}	Array endurance (write/page)		100k			cycles
N_{FFUSE}	General Purpose fuses endurance (write/bit)		1k			cycles
t_{RET}	Data retention		15			years

7.8.4 3.3V Brown Out Detector (BOD33) Characteristics

The values in Table 7-23 describe the values of the BOD33.LEVEL field in the SCIF module.

Table 7-23. BOD33.LEVEL Values

BOD33.LEVEL Value	Parameter	Min	Max	Units
17		2.21	2.55	V
22		2.30	2.64	
27		2.39	2.74	
31	threshold at power-up sequence	2.46	2.82	
33		2.50	2.86	
39		2.60	2.98	
44		2.69	3.08	
49		2.78	3.18	
53		2.85	3.27	
60		2.98	3.41	

7.8.5 5V Brown Out Detector (BOD50) Characteristics

The values in Table 7-25 describe the values of the BOD50.LEVEL field in the SCIF module.

Table 7-25. BOD50.LEVEL Values

BOD50.LEVEL Value	Parameter	Min	Max	Units
16		3.20	3.65	V
25		3.42	3.92	
35		3.68	4.22	
44		3.91	4.48	
53		4.15	4.74	
61		4.36	4.97	

Table 7-36. ADC and S/H Transfer Characteristics (Continued)10-bit Resolution Mode and S/H gain from 1 to 16[1]

Symbol	Parameter	Conditions	Min	Typ	Max	Units
RES	Resolution	Differential mode, V_{VDDANA} = 5V, $V_{ADCREF0}$ = 3V, ADCFIA.SEQCFGn.SRES = 1, S/H gain from 1 to 16 (F_{adc} = 1.5MHz)			10	Bit
INL	Integral Non-Linearity				1.5	LSB
DNL	Differential Non-Linearity				1.5	LSB
	Offset error		-25		25	mV
	Gain error		-15		15	mV

Note: 1. The measures are done without any I/O activity on VDDANA/GNDANA power domain.

7.8.7 Digital to Analog Converter (DAC) Characteristics

Table 7-37. Channel Conversion Time and DAC Clock

Symbol	Parameter	Conditions	Min	Typ	Max	Units
f_{DAC}	DAC clock frequency				1	MHz
$t_{STARTUP}$	Startup time				3	µs
t_{CONV}	Conversion time (latency)	No S/H enabled, internal DAC			1	µs
		One S/H			1.5	µs
		Two S/H			2	µs
	Throughput rate				$1/t_{CONV}$	MSPS

Table 7-38. External Voltage Reference Input

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{DACREF}	DACREF input voltage range		1.2		V_{VDDANA}-0.7	V

Table 7-39. DAC Outputs

Symbol	Parameter	Conditions	Min	Typ	Max	Units
	Output range	with external DAC reference	0.2		V_{DACREF}	V
		with internal DAC reference	0.2		V_{VDDANA}-0.7	
C_{LOAD}	Output capacitance		0		100	pF
R_{LOAD}	Output resitance		2			kΩ

TWIM and TWIS user interface registers. Please refer to the TWIM and TWIS sections for more information.

Table 7-50. TWI-Bus Timing Requirements

Symbol	Parameter	Mode	Minimum		Maximum		Unit
			Requirement	Device	Requirement	Device	
t_r	TWCK and TWD rise time	Standard(1)	-		1000		ns
		Fast(1)	20 + 0.1 C_b		300		
t_f	TWCK and TWD fall time	Standard(1)	-		300		ns
		Fast(1)	20 + 0.1 C_b		300		
t_{HD-STA}	(Repeated) START hold time	Standard(1)	4.0	t_{clkpb}	-		µs
		Fast(1)	0.6				
t_{SU-STA}	(Repeated) START set-up time	Standard(1)	4.7	t_{clkpb}	-		µs
		Fast(1)	0.6				
t_{SU-STO}	STOP set-up time	Standard(1)	4.0	$4t_{clkpb}$	-		µs
		Fast(1)	0.6				
t_{HD-DAT}	Data hold time	Standard(1)	0.3(2)	$2t_{clkpb}$	3.45	??	µs
		Fast(1)			0.9		
$t_{SU-DAT-I2C}$	Data set-up time	Standard(1)	250	$2t_{clkpb}$	-		ns
		Fast(1)	100				
t_{SU-DAT}		-	-	t_{clkpb}	-		-
$t_{LOW-I2C}$	TWCK LOW period	Standard(1)	4.7	$4t_{clkpb}$	-		µs
		Fast(1)	1.3				
t_{LOW}		-	-	t_{clkpb}	-		-
t_{HIGH}	TWCK HIGH period	Standard(1)	4.0	$8t_{clkpb}$	-		µs
		Fast(1)	0.6				
f_{TWCK}	TWCK frequency	Standard(1)	-		100	$\frac{1}{12t_{clkpb}}$	kHz
		Fast(1)			400		

Notes: 1. Standard mode: $f_{TWCK} \leq 100$ kHz ; fast mode: $f_{TWCK} > 100$ kHz .
2. A device must internally provide a hold time of at least 300 ns for TWD with reference to the falling edge of TWCK.

Notations:

C_b = total capacitance of one bus line in pF
t_{clkpb} = period of TWI peripheral bus clock
$t_{prescaled}$ = period of TWI internal prescaled clock (see chapters on TWIM and TWIS)
The maximum $t_{HD;DAT}$ has only to be met if the device does not stretch the LOW period ($t_{LOW-I2C}$) of TWCK.

7.9.7 EBI Timings

See EBI I/O lines description for more details.

Table 7-52. SMC Clock Signal.

Symbol	Parameter	Max[1]	Units
$1/(t_{CPSMC})$	SMC Controller clock frequency	f_{cpu}	MHz

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

Table 7-53. SMC Read Signals with Hold Settings[1]

Symbol	Parameter	Conditions	Min	Units
NRD Controlled (READ_MODE = 1)				
SMC_1	Data setup before NRD high	V_{VDD} = 3.0V, drive strength of the pads set to the lowest, external capacitor = 40pF	32.5	ns
SMC_2	Data hold after NRD high		0	
SMC_3	NRD high to NBS0/A0 change[2]		nrd hold length * t_{CPSMC} - 1.5	
SMC_4	NRD high to NBS1 change[2]		nrd hold length * t_{CPSMC} - 0	
SMC_5	NRD high to NBS2/A1 change[2]		nrd hold length * t_{CPSMC} - 0	
SMC_7	NRD high to A2 - A25 change[2]		nrd hold length * t_{CPSMC} - 5.6	
SMC_8	NRD high to NCS inactive[2]		(nrd hold length - ncs rd hold length) * t_{CPSMC} - 1.3	
SMC_9	NRD pulse width		nrd pulse length * t_{CPSMC} - 0.6	
NRD Controlled (READ_MODE = 0)				
SMC_{10}	Data setup before NCS high	V_{VDD} = 3.0V, drive strength of the pads set to the lowest, external capacitor = 40pF	34.1	ns
SMC_{11}	Data hold after NCS high		0	
SMC_{12}	NCS high to NBS0/A0 change[2]		ncs rd hold length * t_{CPSMC} - 3	
SMC_{13}	NCS high to NBS0/A0 change[2]		ncs rd hold length * t_{CPSMC} - 2	
SMC_{14}	NCS high to NBS2/A1 change[2]		ncs rd hold length * t_{CPSMC} - 1.1	
SMC_{16}	NCS high to A2 - A25 change[2]		ncs rd hold length * t_{CPSMC} - 7.2	
SMC_{17}	NCS high to NRD inactive[2]		(ncs rd hold length - nrd hold length) * t_{CPSMC} - 2.2	
SMC_{18}	NCS pulse width		ncs rd pulse length * t_{CPSMC} - 3	

Note: 1. These values are based on simulation and characterization of other AVR microcontrollers manufactured in the same process technology. These values are not covered by test limits in production.

2. hold length = total cycle duration - setup duration - pulse duration. “hold length” is for “ncs rd hold length” or “nrd hold length”.

Table 7-54. SMC Read Signals with no Hold Settings[1]

Symbol	Parameter	Conditions	Min	Units
NRD Controlled (READ_MODE = 1)				
SMC_{19}	Data setup before NRD high	V_{VDD} = 3.0V, drive strength of the pads set to the lowest, external capacitor = 40pF	32.5	ns
SMC_{20}	Data hold after NRD high		0	
NRD Controlled (READ_MODE = 0)				
SMC_{21}	Data setup before NCS high	V_{VDD} = 3.0V, drive strength of the pads set to the lowest, external capacitor = 40pF	28.5	ns
SMC_{22}	Data hold after NCS high		0	

Note: 1. These values are based on simulation and characterization of other AVR microcontrollers manufactured in the same process technology. These values are not covered by test limits in production.

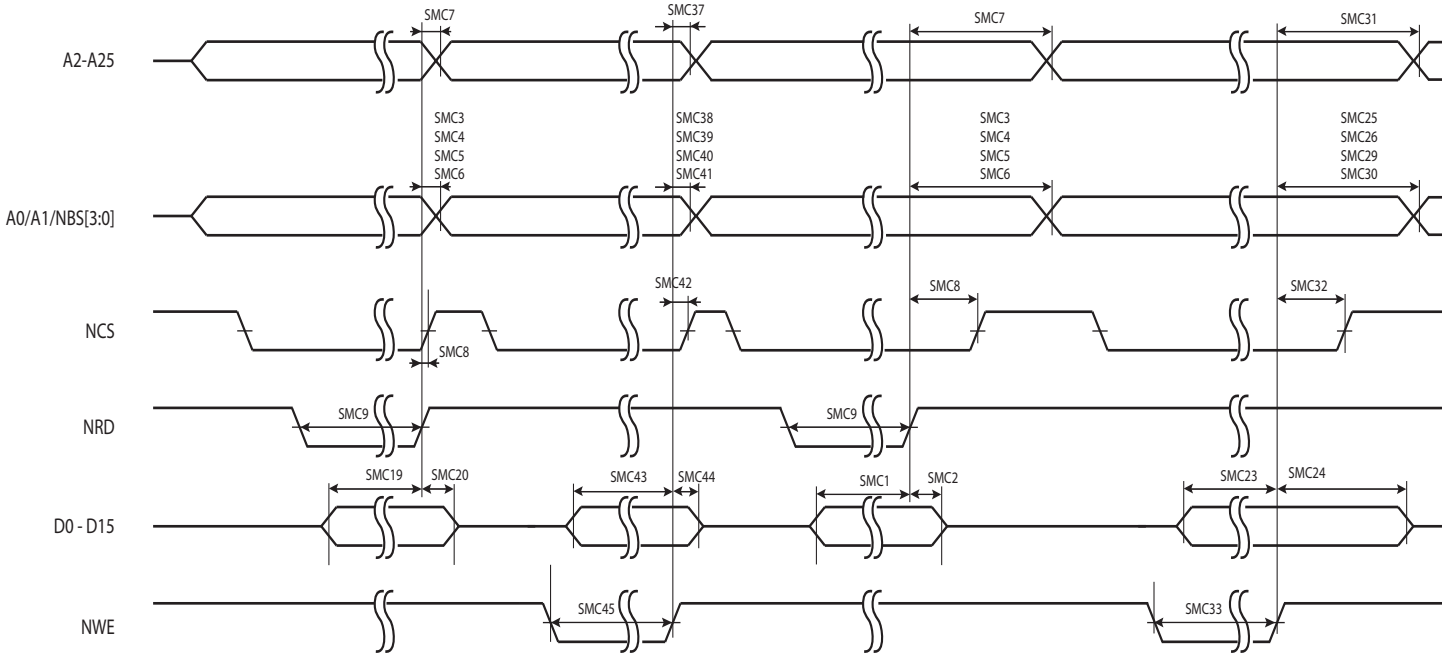
Table 7-55. SMC Write Signals with Hold Settings[1]

Symbol	Parameter	Conditions	Min	Units
NRD Controlled (READ_MODE = 1)				
SMC_{23}	Data Out valid before NWE high	V_{VDD} = 3.0V, drive strength of the pads set to the lowest, external capacitor = 40pF	(nwe pulse length - 1) * t_{CPSMC} - 1.4	ns
SMC_{24}	Data Out valid after NWE high[2]		nwe pulse length * t_{CPSMC} - 4.7	
SMC_{25}	NWE high to NBS0/A0 change[2]		nwe pulse length * t_{CPSMC} - 2.7	
SMC_{29}	NWE high to NBS2/A1 change[2]		nwe pulse length * t_{CPSMC} - 0.7	
SMC_{31}	NWE high to A2 - A25 change[2]		nwe pulse length * t_{CPSMC} - 6.8	
SMC_{32}	NWE high to NCS inactive[2]		(nwe hold pulse - ncs wr hold length) * t_{CPSMC} - 2.5	
SMC_{33}	NWE pulse width		nwe pulse length * t_{CPSMC} - 0.2	
NRD Controlled (READ_MODE = 0)				
SMC_{34}	Data Out valid before NCS high	V_{VDD} = 3.0V, drive strength of the pads set to the lowest, external capacitor = 40pF	(ncs wr pulse length - 1) * t_{CPSMC} - 2.2	ns
SMC_{35}	Data Out valid after NCS high[2]		ncs wr hold length * t_{CPSMC} - 5.1	
SMC_{36}	NCS high to NWE inactive[2]		(ncs wr hold length - nwe hold length) * t_{CPSMC} - 2	

Note: 1. These values are based on simulation and characterization of other AVR microcontrollers manufactured in the same process technology. These values are not covered by test limits in production.

2. hold length = total cycle duration - setup duration - pulse duration. “hold length” is for “ncs wr hold length” or “nwe hold length”

Figure 7-18. SMC Signals for NRD and NRW Controlled Accesses[1]



Note: 1. These values are based on simulation and characterization of other AVR microcontrollers manufactured in the same process technology. These values are not covered by test limits in production.

7.9.8 SDRAM Signals

Table 7-57. SDRAM Clock Signal

Symbol	Parameter	Max[1]	Units
$1/(t_{CPSDCK})$	SDRAM Controller clock frequency	f_{cpu}	MHz

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

Figure 7-20. Ethernet MAC MII Mode