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	81
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	64K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 16x12b; D/A 4x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TQFP
Supplier Device Package	100-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at32uc3c1256c-aur

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.

2.2 Configuration Summary

Table 2-1. Configuration Summary

Feature	AT32UC3C0512C/ AT32UC3C0256C/ AT32UC3C0128C/ AT32UC3C064C	AT32UC3C1512C/ AT32UC3C1256C/ AT32UC3C1128C/ AT32UC3C164C	AT32UC3C2512C/ AT32UC3C2256C/ AT32UC3C2128C/ AT32UC3C264C
Flash	512/256/128/64 KB	512/256/128/64 KB	512/256/128/64 KB
SRAM	64/64/32/16 KB	64/64/32/16 KB	64/64/32/16 KB
HSB RAM	4 KB		
EBI	1	0	0
GPIO	123	81	45
External Interrupts	8	8	8
TWI	3	3	2
USART	5	5	4
Peripheral DMA Channels	16	16	16
Peripheral Event System	1	1	1
SPI	2	2	1
CAN channels	2	2	2
USB	1	1	1
Ethernet MAC 10/100	1 RMII/MII	1 RMII/MII	1 RMII only
I2S	1	1	1
Asynchronous Timers	1	1	1
Timer/Counter Channels	6	6	3
PWM channels	4x2		
QDEC	2	2	1
Frequency Meter	1		
Watchdog Timer	1		
Power Manager	1		
Oscillators	PLL 80-240 MHz (PLL0/PLL1) Crystal Oscillator 0.4-20 MHz (OSC0) Crystal Oscillator 32 KHz (OSC32K) RC Oscillator 115 kHz (RCSYS) RC Oscillator 8 MHz (RC8M) RC Oscillator 120 MHz (RC120M)		
	0.4-20 MHz (OSC1)		-
12-bit ADC number of channels	1 16	1 16	1 11
12-bit DAC number of channels	1 4	1 4	1 2

Table 2-1. Configuration Summary

Feature	AT32UC3C0512C/ AT32UC3C0256C/ AT32UC3C0128C/ AT32UC3C064C	AT32UC3C1512C/ AT32UC3C1256C/ AT32UC3C1128C/ AT32UC3C164C	AT32UC3C2512C/ AT32UC3C2256C/ AT32UC3C2128C/ AT32UC3C264C
Analog Comparators	4	4	2
JTAG	1		
aWire	1		
Max Frequency	66 MHz		
Package	LQFP144	TQFP100	TQFP64/QFN64

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
33	51	73	PC02	66	VDDIO2	x1	TWIMS0 - TWD	SPI0 - NPCS[3]	USART2 - RXD	TC1 - CLK1	MACB - MDC	
34	52	74	PC03	67	VDDIO2	x1	TWIMS0 - TWCK	EIC - EXTINT[1]	USART2 - TXD	TC1 - B1	MACB - MDIO	
37	55	77	PC04	68	VDDIO2	x1	TWIMS1 - TWD	EIC - EXTINT[3]	USART2 - TXD	TC0 - B1		
38	56	78	PC05	69	VDDIO2	x1	TWIMS1 - TWCK	EIC - EXTINT[4]	USART2 - RXD	TC0 - A2		
	57	79	PC06	70	VDDIO2	x1	PEVC - PAD_EVT [15]	USART2 - CLK	USART2 - CTS	TC0 - CLK2	TWIMS2 - TWD	TWIMS0 - TWALM
	58	80	PC07	71	VDDIO2	x1	PEVC - PAD_EVT [2]	EBI - NCS[3]	USART2 - RTS	TC0 - B2	TWIMS2 - TWCK	TWIMS1 - TWALM
		81	PC08	72	VDDIO2	x1/x2	PEVC - PAD_EVT [13]	SPI1 - NPCS[1]	EBI - NCS[0]		USART4 - TXD	
		82	PC09	73	VDDIO2	x1/x2	PEVC - PAD_EVT [14]	SPI1 - NPCS[2]	EBI - ADDR[23]		USART4 - RXD	
		83	PC10	74	VDDIO2	x1/x2	PEVC - PAD_EVT [15]	SPI1 - NPCS[3]	EBI - ADDR[22]			
	59	84	PC11	75	VDDIO2	x1/x2	PWM - PWMH[3]	CANIF - RXLINE[1]	EBI - ADDR[21]	TC0 - CLK0		
	60	85	PC12	76	VDDIO2	x1/x2	PWM - PWML[3]	CANIF - TXLINE[1]	EBI - ADDR[20]	USART2 - CLK		
	61	86	PC13	77	VDDIO2	x1/x2	PWM - PWMH[2]	EIC - EXTINT[7]		USART0 - RTS		
	62	87	PC14	78	VDDIO2	x1/x2	PWM - PWML[2]	USART0 - CLK	EBI - SDCKE	USART0 - CTS		
39	63	88	PC15	79	VDDIO2	x1/x2	PWM - PWMH[1]	SPI0 - NPCS[0]	EBI - SDWE	USART0 - RXD	CANIF - RXLINE[1]	
40	64	89	PC16	80	VDDIO2	x1/x2	PWM - PWML[1]	SPI0 - NPCS[1]	EBI - CAS	USART0 - TXD	CANIF - TXLINE[1]	
41	65	90	PC17	81	VDDIO2	x1/x2	PWM - PWMH[0]	SPI0 - NPCS[2]	EBI - RAS	IISC - ISDO		USART3 - TXD
42	66	91	PC18	82	VDDIO2	x1/x2	PWM - PWML[0]	EIC - EXTINT[5]	EBI - SDA10	IISC - ISDI		USART3 - RXD
43	67	92	PC19	83	VDDIO3	x1/x2	PWM - PWML[2]	SCIF - GCLK[0]	EBI - DATA[0]	IISC - IMCK		USART3 - CTS
44	68	93	PC20	84	VDDIO3	x1/x2	PWM - PWMH[2]	SCIF - GCLK[1]	EBI - DATA[1]	IISC - ISCK		USART3 - RTS
45	69	94	PC21	85	VDDIO3	x1/x2	PWM - EXT_ FAULTS[0]	CANIF - RXLINE[0]	EBI - DATA[2]	IISC - IWS		
46	70	95	PC22	86	VDDIO3	x1/x2	PWM - EXT_ FAULTS[1]	CANIF - TXLINE[0]	EBI - DATA[3]		USART3 - CLK	
	71	96	PC23	87	VDDIO3	x1/x2	QDEC1 - QEPB	CANIF - RXLINE[1]	EBI - DATA[4]	PEVC - PAD_EVT [3]		

Table 3-7. Signal Description List

Signal Name	Function	Type	Active Level	Comments
ADCVREFN	Analog negative reference connected to external capacitor	Analog		
Auxiliary Port - AUX				
MCKO	Trace Data Output Clock	Output		
MDO[5:0]	Trace Data Output	Output		
MSEO[1:0]	Trace Frame Control	Output		
EVTI_N	Event In	Output	Low	
EVTO_N	Event Out	Output	Low	
aWire - AW				
DATA	aWire data	I/O		
DATAOUT	aWire data output for 2-pin mode	I/O		
Controller Area Network Interface - CANIF				
RXLINE[1:0]	CAN channel rxline	I/O		
TXLINE[1:0]	CAN channel txline	I/O		
DAC Interface - DACIFB0/1				
DAC0A, DAC0B	DAC0 output pins of S/H A	Analog		
DAC1A, DAC1B	DAC output pins of S/H B	Analog		
DACREF	Analog reference voltage input	Analog		
External Bus Interface - EBI				
ADDR[23:0]	Address Bus	Output		
CAS	Column Signal	Output	Low	
DATA[15:0]	Data Bus	I/O		
NCS[3:0]	Chip Select	Output	Low	
NRD	Read Signal	Output	Low	
NWAIT	External Wait Signal	Input	Low	
NWE0	Write Enable 0	Output	Low	
NWE1	Write Enable 1	Output	Low	
RAS	Row Signal	Output	Low	
SDA10	SDRAM Address 10 Line	Output		

Debug state can be entered as described in the *AVR32UC Technical Reference Manual*.

Debug state is exited by the *retd* instruction.

4.4.3.3 Secure State

The AVR32 can be set in a secure state, that allows a part of the code to execute in a state with higher security levels. The rest of the code can not access resources reserved for this secure code. Secure State is used to implement FlashVault technology. Refer to the *AVR32UC Technical Reference Manual* for details.

4.4.4 System Registers

The system registers are placed outside of the virtual memory space, and are only accessible using the privileged *mfsr* and *mtsr* instructions. The table below lists the system registers specified in the AVR32 architecture, some of which are unused in AVR32UC. The programmer is responsible for maintaining correct sequencing of any instructions following a *mtsr* instruction. For detail on the system registers, refer to the *AVR32UC Technical Reference Manual*.

Table 4-3. System Registers

Reg #	Address	Name	Function
0	0	SR	Status Register
1	4	EVBA	Exception Vector Base Address
2	8	ACBA	Application Call Base Address
3	12	CPUCR	CPU Control Register
4	16	ECR	Exception Cause Register
5	20	RSR_SUP	Unused in AVR32UC
6	24	RSR_INT0	Unused in AVR32UC
7	28	RSR_INT1	Unused in AVR32UC
8	32	RSR_INT2	Unused in AVR32UC
9	36	RSR_INT3	Unused in AVR32UC
10	40	RSR_EX	Unused in AVR32UC
11	44	RSR_NMI	Unused in AVR32UC
12	48	RSR_DBG	Return Status Register for Debug mode
13	52	RAR_SUP	Unused in AVR32UC
14	56	RAR_INT0	Unused in AVR32UC
15	60	RAR_INT1	Unused in AVR32UC
16	64	RAR_INT2	Unused in AVR32UC
17	68	RAR_INT3	Unused in AVR32UC
18	72	RAR_EX	Unused in AVR32UC
19	76	RAR_NMI	Unused in AVR32UC
20	80	RAR_DBG	Return Address Register for Debug mode
21	84	JECR	Unused in AVR32UC
22	88	JOSP	Unused in AVR32UC
23	92	JAVA_LV0	Unused in AVR32UC

Table 4-3. System Registers (Continued)

Reg #	Address	Name	Function
24	96	JAVA_LV1	Unused in AVR32UC
25	100	JAVA_LV2	Unused in AVR32UC
26	104	JAVA_LV3	Unused in AVR32UC
27	108	JAVA_LV4	Unused in AVR32UC
28	112	JAVA_LV5	Unused in AVR32UC
29	116	JAVA_LV6	Unused in AVR32UC
30	120	JAVA_LV7	Unused in AVR32UC
31	124	JTBA	Unused in AVR32UC
32	128	JBCR	Unused in AVR32UC
33-63	132-252	Reserved	Reserved for future use
64	256	CONFIG0	Configuration register 0
65	260	CONFIG1	Configuration register 1
66	264	COUNT	Cycle Counter register
67	268	COMPARE	Compare register
68	272	TLBEHI	Unused in AVR32UC
69	276	TLBELO	Unused in AVR32UC
70	280	PTBR	Unused in AVR32UC
71	284	TLBEAR	Unused in AVR32UC
72	288	MMUCR	Unused in AVR32UC
73	292	TLBARLO	Unused in AVR32UC
74	296	TLBARHI	Unused in AVR32UC
75	300	PCCNT	Unused in AVR32UC
76	304	PCNT0	Unused in AVR32UC
77	308	PCNT1	Unused in AVR32UC
78	312	PCCR	Unused in AVR32UC
79	316	BEAR	Bus Error Address Register
80	320	MPUAR0	MPU Address Register region 0
81	324	MPUAR1	MPU Address Register region 1
82	328	MPUAR2	MPU Address Register region 2
83	332	MPUAR3	MPU Address Register region 3
84	336	MPUAR4	MPU Address Register region 4
85	340	MPUAR5	MPU Address Register region 5
86	344	MPUAR6	MPU Address Register region 6
87	348	MPUAR7	MPU Address Register region 7
88	352	MPUPSR0	MPU Privilege Select Register region 0
89	356	MPUPSR1	MPU Privilege Select Register region 1

Table 4-3. System Registers (Continued)

Reg #	Address	Name	Function
90	360	MPUPSR2	MPU Privilege Select Register region 2
91	364	MPUPSR3	MPU Privilege Select Register region 3
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	412	SS_STATUS	Secure State Status Register
104	416	SS_ADRF	Secure State Address Flash Register
105	420	SS_ADRR	Secure State Address RAM Register
106	424	SS_ADR0	Secure State Address 0 Register
107	428	SS_ADR1	Secure State Address 1 Register
108	432	SS_SP_SYS	Secure State Stack Pointer System Register
109	436	SS_SP_APP	Secure State Stack Pointer Application Register
110	440	SS_RAR	Secure State Return Address Register
111	444	SS_RSR	Secure State Return Status Register
112-191	448-764	Reserved	Reserved for future use
192-255	768-1020	IMPL	IMPLEMENTATION DEFINED

4.5 Exceptions and Interrupts

In the AVR32 architecture, events are used as a common term for exceptions and interrupts. AVR32UC incorporates a powerful event handling scheme. The different event sources, like Illegal Op-code and interrupt requests, have different priority levels, ensuring a well-defined behavior when multiple events are received simultaneously. Additionally, pending events of a higher priority class may preempt handling of ongoing events of a lower priority class.

When an event occurs, the execution of the instruction stream is halted, and execution is passed to an event handler at an address specified in Table 4-4 on page 38. 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 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 interrupt requests, yielding a uniform event handling scheme.

An interrupt controller does the priority handling of the 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.

The user must also make sure that the system stack is large enough so that any event is able to push the required registers to stack. If the system stack is full, and an event occurs, the system will enter an UNDEFINED state.

4.5.2 Exceptions and Interrupt Requests

When an event other than *scall* or debug request is received by the core, the following actions are performed atomically:

1. The pending event will not be accepted if it is masked. The I3M, I2M, I1M, I0M, EM, and GM bits in the Status Register are used to mask different events. Not all events can be masked. A few critical events (NMI, Unrecoverable Exception, TLB Multiple Hit, and Bus Error) can not be masked. When an event is accepted, hardware automatically sets the mask bits corresponding to all sources with equal or lower priority. This inhibits acceptance of other events of the same or lower priority, except for the critical events listed above. Software may choose to clear some or all of these bits after saving the necessary state if other priority schemes are desired. It is the event source's responsability to ensure that their events are left pending until accepted by the CPU.
2. When a request is accepted, the Status Register and Program Counter of the current context is stored to the system stack. If the event is an INT0, INT1, INT2, or INT3, registers R8-R12 and LR are also automatically stored to stack. Storing the Status Register ensures that the core is returned to the previous execution mode when the current event handling is completed. When exceptions occur, both the EM and GM bits are set, and the application may manually enable nested exceptions if desired by clearing the appropriate bit. Each exception handler has a dedicated handler address, and this address uniquely identifies the exception source.
3. The Mode bits are set to reflect the priority of the accepted event, and the correct register file bank is selected. The address of the event handler, as shown in Table 4-4 on page 38, is loaded into the Program Counter.

The execution of the event handler routine then continues from the effective address calculated.

The *rete* instruction signals the end of the event. When encountered, the Return Status Register and Return Address Register are popped from the system stack and restored to the Status Register and Program Counter. If the *rete* instruction returns from INT0, INT1, INT2, or INT3, registers R8-R12 and LR are also popped from the system stack. The restored Status Register contains information allowing the core to resume operation in the previous execution mode. This concludes the event handling.

Table 5-3. Peripheral Address Mapping

0xFFFF3800	TWIM0	Two-wire Master Interface - TWIM0
0xFFFF3C00	TWIM1	Two-wire Master Interface - TWIM1
0xFFFF4000	TWIS0	Two-wire Slave Interface - TWIS0
0xFFFF4400	TWIS1	Two-wire Slave Interface - TWIS1
0xFFFF4800	IISC	Inter-IC Sound (I2S) Controller - IISC
0xFFFF4C00	PWM	Pulse Width Modulation Controller - PWM
0xFFFF5000	QDEC0	Quadrature Decoder - QDEC0
0xFFFF5400	QDEC1	Quadrature Decoder - QDEC1
0xFFFF5800	TC1	Timer/Counter - TC1
0xFFFF5C00	PEVC	Peripheral Event Controller - PEVC
0xFFFF6000	ACIFA0	Analog Comparators Interface - ACIFA0
0xFFFF6400	ACIFA1	Analog Comparators Interface - ACIFA1
0xFFFF6800	DACIFB0	DAC interface - DACIFB0
0xFFFF6C00	DACIFB1	DAC interface - DACIFB1
0xFFFF7000	AW	aWire - AW

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 bus-mapped GPIO registers.

6. Supply and Startup Considerations

6.1 Supply Considerations

6.1.1 Power Supplies

The AT32UC3C has several types of power supply pins:

- **VDDIO pins (VDDIO1, VDDIO2, VDDIO3): Power I/O lines. Two voltage ranges are available: 5V or 3.3V nominal. The VDDIO pins should be connected together.**
- **VDDANA: Powers the Analog part of the device (Analog I/Os, ADC, ACs, DACs). 2 voltage ranges available: 5V or 3.3V nominal.**
- **VDDIN_5: Input voltage for the 1.8V and 3.3V regulators. Two Voltage ranges are available: 5V or 3.3V nominal.**
- **VDDIN_33:**
  - **USB I/O power supply**
  - **if the device is 3.3V powered: Input voltage, voltage is 3.3V nominal.**
  - **if the device is 5V powered: stabilization for the 3.3V voltage regulator, requires external capacitors**
- **VDDCORE: Stabilization for the 1.8V voltage regulator, requires external capacitors.**
- **GNDCORE: Ground pins for the voltage regulators and the core.**
- **GNDANA: Ground pin for Analog part of the design**
- **GNDPLL: Ground pin for the PLLs**
- **GNDIO pins (GNDIO1, GNDIO2, GNDIO3): Ground pins for the I/O lines. The GNDIO pins should be connected together.**

See ”Electrical Characteristics” on page 50 for power consumption on the various supply pins.

For decoupling recommendations for the different power supplies, please refer to the schematic checklist.

6.1.2 Voltage Regulators

The AT32UC3C embeds two voltage regulators:

- One 1.8V internal regulator that converts from VDDIN_5 to 1.8V. The regulator supplies the output voltage on VDDCORE.
- One 3.3V internal regulator that converts from VDDIN_5 to 3.3V. The regulator supplies the USB pads on VDDIN_33. If the USB is not used or if VDDIN_5 is within the 3V range, the 3.3V regulator can be disabled through the VREG33CTL field of the VREGCTRL SCIF register.

6.1.3 Regulators Connection

The AT32UC3C supports two power supply configurations.

- 5V single supply mode
- 3.3V single supply mode

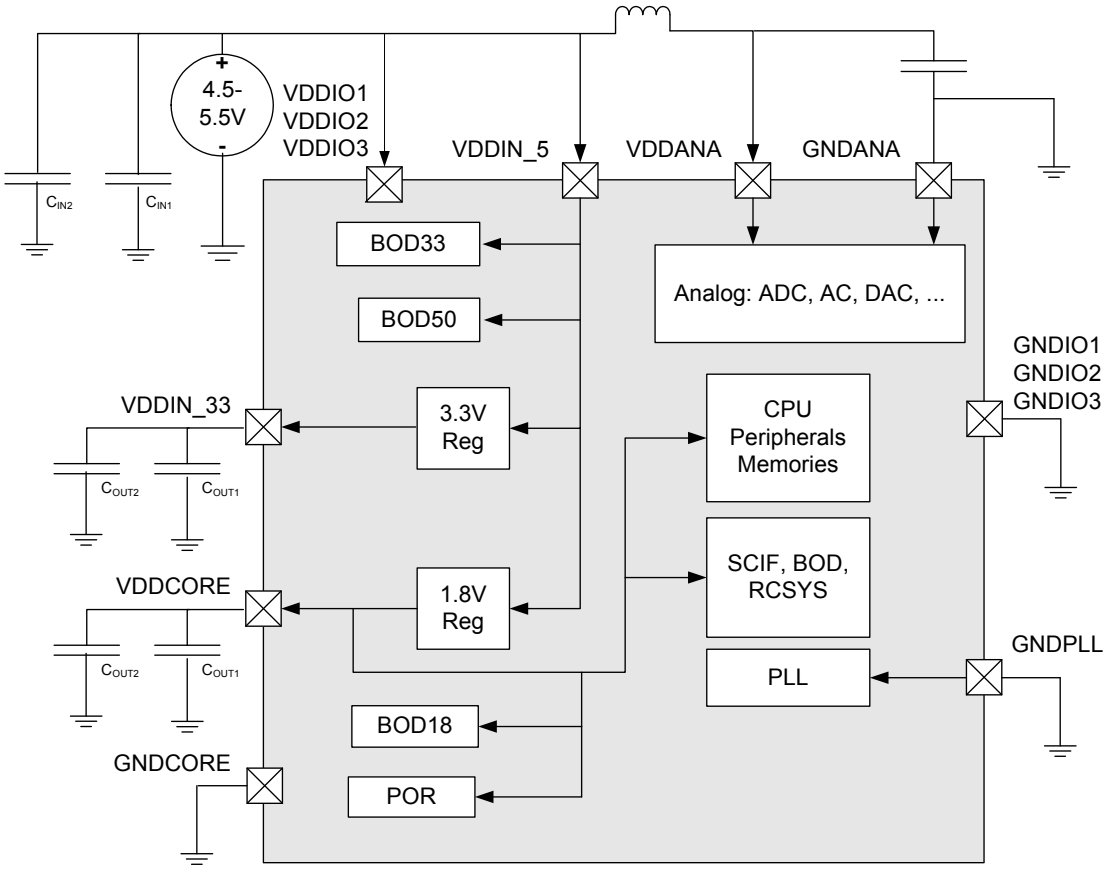
6.1.3.1 5V Single Supply Mode

In 5V single supply mode, the 1.8V internal regulator is connected to the 5V source (VDDIN_5 pin) and its output feeds VDDCORE.

The 3.3V regulator is connected to the 5V source (VDDIN_5 pin) and its output feeds the USB pads. If the USB is not used, the 3.3V regulator can be disabled through the VREG33CTL field of the VREGCTRL SCIF register.

Figure 6-1 on page 47 shows the power schematics to be used for 5V single supply mode. All I/O lines and analog blocks will be powered by the same power (VDDIN_5 = VDDIO1 = VDDIO2 = VDDIO3 = VDDANA).

Figure 6-1. 5V Single Power Supply mode



6.1.3.2 *3.3V Single Supply Mode*

In 3.3V single supply mode, the VDDIN_5 and VDDIN_33 pins should be connected together externally. The 1.8V internal regulator is connected to the 3.3V source (VDDIN_5 pin) and its output feeds VDDCORE.

The 3.3V regulator should be disabled once the circuit is running through the VREG33CTL field of the VREGCTRL SCIF register.

Figure 6-2 on page 48 shows the power schematics to be used for 3.3V single supply mode. All I/O lines and analog blocks will be powered by the same power (VDDIN_5 = VDDIN_33 = VDDIO1 = VDDIO2 = VDDIO3 = VDDANA).

  - PLL1 stopped
- Clocks
  - External clock on XIN0 as main clock source.
  - CPU, HSB, and PB clocks undivided

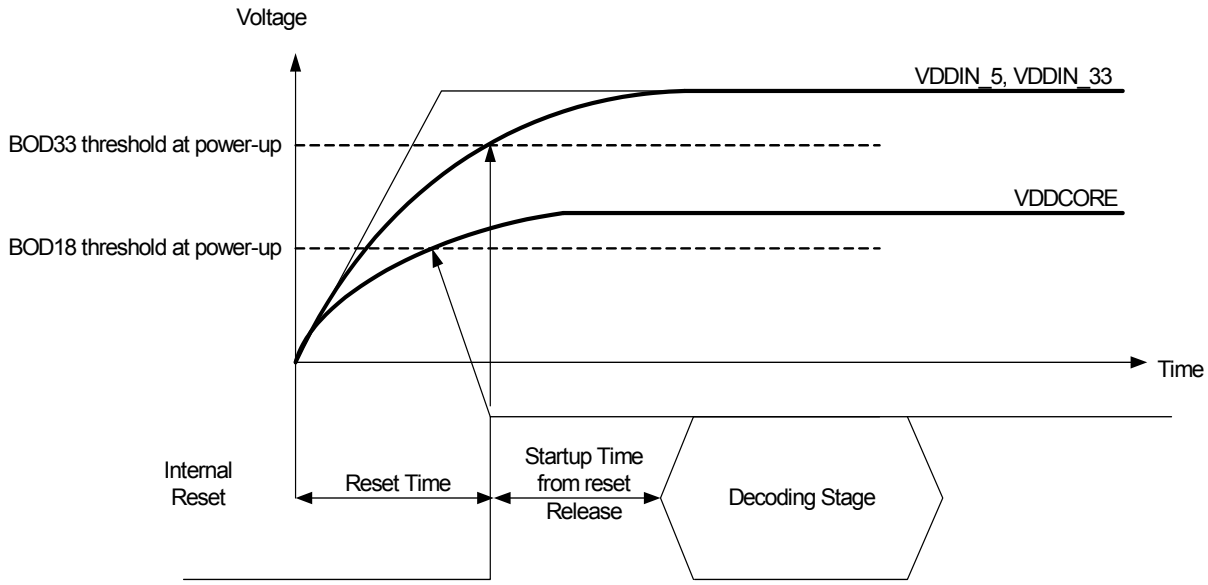
Consumption active is the added current consumption when the module clock is turned on and when the module is doing a typical set of operations.

Table 7-5. Typical Current Consumption by Peripheral[2]

Peripheral	Typ Consumption Active	Unit
ACIFA[1]	3	µA/MHz
ADCIFA[1]	7	
AST	3	
CANIF	25	
DACIFB[1]	3	
EBI	23	
EIC	0.5	
FREQM	0.5	
GPIO	37	
INTC	3	
MDMA	4	
PDCA	24	
PEVC	15	
PWM	40	
QDEC	3	
SAU	3	
SDRAMC	2	
SMC	9	
SPI	5	
TC	8	
TWIM	2	
TWIS	2	
USART	10	
USBC	5	
WDT	2	

Notes: 1. Includes the current consumption on VDDANA.
2. These numbers are valid for the measured condition only and must not be extrapolated to other frequencies.

Figure 7-5. Startup and Reset Time



7.9.2 RESET_N characteristics

Table 7-45. RESET_N Clock Waveform Parameters

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
t_{RESET}	RESET_N minimum pulse length		2 * T_{RCSYS}			clock cycles

7.9.6 JTAG Timing

Figure 7-16. JTAG Interface Signals

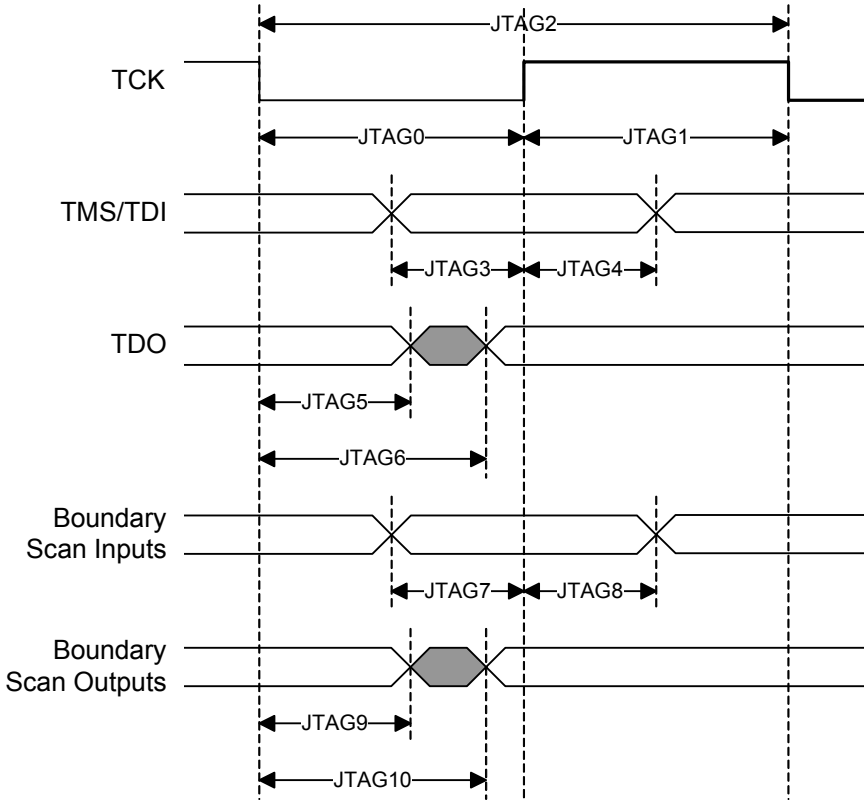


Table 7-51. JTAG Timings[1]

Symbol	Parameter	Conditions	Min	Max	Units
JTAG0	TCK Low Half-period	external capacitor = 40pF	21.5		ns
JTAG1	TCK High Half-period		8.5		ns
JTAG2	TCK Period		29		ns
JTAG3	TDI, TMS Setup before TCK High		6.5		ns
JTAG4	TDI, TMS Hold after TCK High		0		ns
JTAG5	TDO Hold Time		12.5		ns
JTAG6	TCK Low to TDO Valid			21.5	ns
JTAG7	Boundary Scan Inputs Setup Time		0		ns
JTAG8	Boundary Scan Inputs Hold Time		4.5		ns
JTAG9	Boundary Scan Outputs Hold Time		11		ns
JTAG10	TCK to Boundary Scan Outputs Valid			18	ns

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.

7.9.9 MACB Characteristics

Table 7-59. Ethernet MAC Signals[1]

Symbol	Parameter	Conditions	Min.	Max.	Unit
MAC_1	Setup for MDIO from MDC rising	V_{VDD} = 3.0V, drive strength of the pads set to the highest, external capacitor = 10pF on MACB pins	0	2.5	ns
MAC_2	Hold for MDIO from MDC rising		0	0.7	ns
MAC_3	MDIO toggling from MDC falling		0	1.1	ns

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 7-60. Ethernet MAC MII Specific Signals[1]

Symbol	Parameter	Conditions	Min.	Max.	Unit
MAC_4	Setup for COL from TX_CLK rising	V_{VDD} = 3.0V, drive strength of the pads set to the highest, external capacitor = 10pF on MACB pins	0		ns
MAC_5	Hold for COL from TX_CLK rising		0		ns
MAC_6	Setup for CRS from TX_CLK rising		0.5		ns
MAC_7	Hold for CRS from TX_CLK rising		0.5		ns
MAC_8	TX_ER toggling from TX_CLK rising		16.4	18.6	ns
MAC_9	TX_EN toggling from TX_CLK rising		14.5	15.3	ns
MAC_{10}	TXD toggling from TX_CLK rising		13.9	18.2	ns
MAC_{11}	Setup for RXD from RX_CLK		1.3		ns
MAC_{12}	Hold for RXD from RX_CLK		1.8		ns
MAC_{13}	Setup for RX_ER from RX_CLK		3.4		ns
MAC_{14}	Hold for RX_ER from RX_CLK		0		ns
MAC_{15}	Setup for RX_DV from RX_CLK		0.7		ns
MAC_{16}	Hold for RX_DV from RX_CLK		1.3n		ns

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.

9. Ordering Information

Table 9-1. Ordering Information

Device	Ordering Code	Carrier Type	Package	Temperature Operating Range
AT32UC3C0512C	AT32UC3C0512C-ALUT	Tray	LQFP 144	Industrial (-40°C to 85°C)
	AT32UC3C0512C-ALUR	Tape & Reel		
AT32UC3C0256C	AT32UC3C0256C-ALUT	Tray		
	AT32UC3C0256C-ALUR	Tape & Reel		
AT32UC3C0128C	AT32UC3C0128C-ALUT	Tray		
	AT32UC3C0128C-ALUR	Tape & Reel		
AT32UC3C064C	AT32UC3C064C-ALUT	Tray		
	AT32UC3C064C-ALUR	Tape & Reel		
AT32UC3C1512C	AT32UC3C1512C-AUT	Tray	TQFP 100	
	AT32UC3C1512C-AUR	Tape & Reel		
AT32UC3C1256C	AT32UC3C1256C-AUT	Tray		
	AT32UC3C1256C-AUR	Tape & Reel		
AT32UC3C1128C	AT32UC3C1128C-AUT	Tray		
	AT32UC3C1128C-AUR	Tape & Reel		
AT32UC3C164C	AT32UC3C164C-AUT	Tray		
	AT32UC3C164C-AUR	Tape & Reel		
AT32UC3C2512C	AT32UC3C2512C-A2UT	Tray	TQFP 64	
	AT32UC3C2512C-A2UR	Tape & Reel		
	AT32UC3C2512C-Z2UT	Tray	QFN 64	
	AT32UC3C2512C-Z2UR	Tape & Reel		
AT32UC3C2256C	AT32UC3C2256C-A2UT	Tray	TQFP 64	
	AT32UC3C2256C-A2UR	Tape & Reel		
	AT32UC3C2256C-Z2UT	Tray	QFN 64	
	AT32UC3C2256C-Z2UR	Tape & Reel		
AT32UC3C2128C	AT32UC3C2128C-A2UT	Tray	TQFP 64	
	AT32UC3C2128C-A2UR	Tape & Reel		
	AT32UC3C2128C-Z2UT	Tray	QFN 64	
	AT32UC3C2128C-Z2UR	Tape & Reel		
AT32UC3C264C	AT32UC3C264C-A2UT	Tray	TQFP 64	
	AT32UC3C264C-A2UR	Tape & Reel		
	AT32UC3C264C-Z2UT	Tray	QFN 64	
	AT32UC3C264C-Z2UR	Tape & Reel		

2 **Requesting clocks in idle sleep modes will mask all other PB clocks than the requested**
In idle or frozen sleep mode, all the PB clocks will be frozen if the TWIS or the AST need to wake the cpu up.
Fix/Workaround
Disable the TWIS or the AST before entering idle or frozen sleep mode.

3 **TWIS may not wake the device from sleep mode**
If the CPU is put to a sleep mode (except Idle and Frozen) directly after a TWI Start condition, the CPU may not wake upon a TWIS address match. The request is NACKed.
Fix/Workaround
When using the TWI address match to wake the device from sleep, do not switch to sleep modes deeper than Frozen. Another solution is to enable asynchronous EIC wake on the TWIS clock (TWCK) or TWIS data (TWD) pins, in order to wake the system up on bus events.

10.2.6 SCIF

1 **PLLCOUNT value larger than zero can cause PLLEN glitch**
Initializing the PLLCOUNT with a value greater than zero creates a glitch on the PLLEN signal during asynchronous wake up.
Fix/Workaround
The lock-masking mechanism for the PLL should not be used.
The PLLCOUNT field of the PLL Control Register should always be written to zero.

2 **PLL lock might not clear after disable**
Under certain circumstances, the lock signal from the Phase Locked Loop (PLL) oscillator may not go back to zero after the PLL oscillator has been disabled. This can cause the propagation of clock signals with the wrong frequency to parts of the system that use the PLL clock.
Fix/Workaround
PLL must be turned off before entering STOP, DEEPSTOP or STATIC sleep modes. If PLL has been turned off, a delay of 30us must be observed after the PLL has been enabled again before the SCIF.PLL0LOCK bit can be used as a valid indication that the PLL is locked.

3 **BOD33 reset locks the device**
If BOD33 is enabled as a reset source (SCIF.BOD33.CTRL=0x1) and when VDDIN_33 power supply voltage falls below the BOD33 voltage (SCIF.BOD33.LEVEL), the device is locked permanently under reset even if the power supply goes back above BOD33 reset level. In order to unlock the device, an external reset event should be applied on RESET_N.
Fix/Workaround
Use an external BOD on VDDIN_33 or an external reset source.

10.2.7 SPI

1 **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.

Table of Contents

Headquarters

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

International

Atmel Asia
Unit 1-5 & 16, 19/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
Hong Kong
Tel: (852) 2245-6100
Fax: (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-
Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

Atmel Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Product Contact

Web Site
www.atmel.com

Technical Support
avr32@atmel.com

Sales Contact
www.atmel.com/contacts

Literature Requests
www.atmel.com/literature