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, EBI/EMI, Ethernet, I²C, IrDA, LINbus, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	123
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 16x12b; D/A 4x12b
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/at32uc3c0128c-alur

The Full-Speed USB 2.0 Device interface supports several USB Classes at the same time thanks to the rich End-Point configuration. The On-The-GO (OTG) Host interface allows device like a USB Flash disk or a USB printer to be directly connected to the processor.

The media-independent interface (MII) and reduced MII (RMII) 10/100 Ethernet MAC module provides on-chip solutions for network-connected devices.

The Peripheral Event Controller (PEVC) allows to redirect events from one peripheral or from input pins to another peripheral. It can then trigger, in a deterministic time, an action inside a peripheral without the need of CPU. For instance a PWM waveform can directly trigger an ADC capture, hence avoiding delays due to software interrupt processing.

The AT32UC3C features analog functions like ADC, DAC, Analog comparators. The ADC interface is built around a 12-bit pipelined ADC core and is able to control two independent 8-channel or one 16-channel. The ADC block is able to measure two different voltages sampled at the same time. The analog comparators can be paired to detect when the sensing voltage is within or outside the defined reference window.

Atmel offers the QTouch library for embedding capacitive touch buttons, sliders, and wheels functionality into AVR microcontrollers. The patented charge-transfer signal acquisition offers robust sensing and included fully debounced reporting of touch keys and includes Adjacent Key Suppression® (AKS®) technology for unambiguous detection of key events. The easy-to-use QTouch Suite toolchain allows you to explore, develop, and debug your own touch applications.

AT32UC3C integrates a class 2+ Nexus 2.0 On-Chip Debug (OCD) System, with non-intrusive real-time trace, full-speed read/write memory access in addition to basic runtime control. The Nanotrace interface enables trace feature for aWire- or JTAG-based debuggers. The single-pin aWire interface allows all features available through the JTAG interface to be accessed through the RESET pin, allowing the JTAG pins to be used for GPIO or peripherals.

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 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
		124	PD15	111	VDDIO3	x1/x2	TC0 - A0	USART3 - TXD	EBI - ADDR[11]			
		125	PD16	112	VDDIO3	x1/x2	TC0 - B0	USART3 - RXD	EBI - ADDR[12]			
		126	PD17	113	VDDIO3	x1/x2	TC0 - A1	USART3 - CTS	EBI - ADDR[13]	USART3 - CLK		
		127	PD18	114	VDDIO3	x1/x2	TC0 - B1	USART3 - RTS	EBI - ADDR[14]			
		128	PD19	115	VDDIO3	x1/x2	TC0 - A2		EBI - ADDR[15]			
		129	PD20	116	VDDIO3	x1/x2	TC0 - B2		EBI - ADDR[16]			
57	88	130	PD21	117	VDDIO3	x1/x2	USART3 - TXD	EIC - EXTINT[0]	EBI - ADDR[17]	QDEC1 - QEPI		
	89	131	PD22	118	VDDIO1	x1/x2	USART3 - RXD	TC0 - A2	EBI - ADDR[18]	SCIF - GCLK[0]		
	90	132	PD23	119	VDDIO1	x1/x2	USART3 - CTS	USART3 - CLK	EBI - ADDR[19]	QDEC1 - QEPA		
	91	133	PD24	120	VDDIO1	x1/x2	USART3 - RTS	EIC - EXTINT[8]	EBI - NWE1	QDEC1 - QEPB		
		134	PD25	121	VDDIO1	x1/x2	TC0 - CLK0	USBC - ID	EBI - NWE0		USART4 - CLK	
		135	PD26	122	VDDIO1	x1/x2	TC0 - CLK1	USBC - VBOF	EBI - NRD			
58	92	136	PD27	123	VDDIO1	x1/x2	USART0 - TXD	CANIF - RXLINE[0]	EBI - NCS[1]	TC0 - A0	MACB - RX_ER	
59	93	137	PD28	124	VDDIO1	x1/x2	USART0 - RXD	CANIF - TXLINE[0]	EBI - NCS[2]	TC0 - B0	MACB - RX_DV	
60	94	138	PD29	125	VDDIO1	x1/x2	USART0 - CTS	EIC - EXTINT[6]	USART0 - CLK	TC0 - CLK0	MACB - TX_CLK	
61	95	139	PD30	126	VDDIO1	x1/x2	USART0 - RTS	EIC - EXTINT[3]	EBI - NWAIT	TC0 - A1	MACB - TX_EN	

Note: 1. Refer to "Electrical Characteristics" on page 50 for a description of the electrical properties of the pin types used.

See Section 3.3 for a description of the various peripheral signals.

3.2.2 Peripheral Functions

Each GPIO line can be assigned to one of several peripheral functions. The following table describes how the various peripheral functions are selected. The last listed function has priority in case multiple functions are enabled on the same pin.

Table 3-2. Peripheral Functions

Function	Description
GPIO Controller Function multiplexing	GPIO and GPIO peripheral selection A to F
Nexus OCD AUX port connections	OCD trace system

depending on the configuration of the OCD AXS register. For details, see the AVR32UC Technical Reference Manual.

Table 3-5. Nexus OCD AUX port connections

Pin	AXS=0	AXS=1	AXS=2
EVTI_N	PA08	PB19	PA10
MDO[5]	PC05	PC31	PB06
MDO[4]	PC04	PC12	PB15
MDO[3]	PA23	PC11	PB14
MDO[2]	PA22	PB23	PA27
MDO[1]	PA19	PB22	PA26
MDO[0]	PA09	PB20	PA19
EVTO_N	PD29	PD29	PD29
MCKO	PD13	PB21	PB26
MSEO[1]	PD30	PD08	PB25
MSEO[0]	PD14	PD07	PB18

3.2.6 Other Functions

The functions listed in Table 3-6 are not mapped to the normal GPIO functions. The aWire DATA pin will only be active after the aWire is enabled. The aWire DATAOUT pin will only be active after the aWire is enabled and the 2_PIN_MODE command has been sent.

Table 3-6. Other Functions

QFN64/ TQFP64 pin	TQFP100 pin	LQFP144 pin	Pad	Oscillator pin
64	98	142	RESET_N	aWire DATA
3	3	3	PA02	aWire DATAOUT

3.3 Signals Description

The following table give details on the signal name classified by peripherals.

Table 3-7. Signal Description List

Signal Name	Function	Type	Active Level	Comments
Power				
VDDIO1 VDDIO2 VDDIO3	I/O Power Supply	Power Input		4.5V to 5.5V or 3.0V to 3.6 V
VDDANA	Analog Power Supply	Power Input		4.5V to 5.5V or 3.0V to 3.6 V

Table 3-7. Signal Description List

Signal Name	Function	Type	Active Level	Comments
SDCK	SDRAM Clock	Output		
SDCKE	SDRAM Clock Enable	Output		
SDWE	SDRAM Write Enable	Output	Low	
External Interrupt Controller - EIC				
EXTINT[8:1]	External Interrupt Pins	Input		
NMI_N = EXTINT[0]	Non-Maskable Interrupt Pin	Input	Low	
General Purpose Input/Output - GPIOA, GPIOB, GPIOC, GPIOD				
PA[29:19] - PA[16:0]	Parallel I/O Controller GPIOA	I/O		
PB[31:0]	Parallel I/O Controller GPIOB	I/O		
PC[31:0]	Parallel I/O Controller GPIOC	I/O		
PD[30:0]	Parallel I/O Controller GPIOD	I/O		
Inter-IC Sound (I2S) Controller - IISC				
IMCK	I2S Master Clock	Output		
ISCK	I2S Serial Clock	I/O		
ISDI	I2S Serial Data In	Input		
ISDO	I2S Serial Data Out	Output		
IWS	I2S Word Select	I/O		
JTAG				
TCK	Test Clock	Input		
TDI	Test Data In	Input		
TDO	Test Data Out	Output		
TMS	Test Mode Select	Input		
Ethernet MAC - MACB				
COL	Collision Detect	Input		
CRS	Carrier Sense and Data Valid	Input		
MDC	Management Data Clock	Output		
MDIO	Management Data Input/Output	I/O		
RXD[3:0]	Receive Data	Input		

Table 3-7. Signal Description List

Signal Name	Function	Type	Active Level	Comments
RX_CLK	Receive Clock	Input		
RX_DV	Receive Data Valid	Input		
RX_ER	Receive Coding Error	Input		
SPEED	Speed	Output		
TXD[3:0]	Transmit Data	Output		
TX_CLK	Transmit Clock or Reference Clock	Input		
TX_EN	Transmit Enable	Output		
TX_ER	Transmit Coding Error	Output		
WOL	Wake-On-LAN	Output		
Peripheral Event Controller - PEVC				
PAD_EVT[15:0]	Event Input Pins	Input		
Power Manager - PM				
RESET_N	Reset Pin	Input	Low	
Pulse Width Modulator - PWM				
PWMH[3:0] PWML[3:0]	PWM Output Pins	Output		
EXT_FAULT[1:0]	PWM Fault Input Pins	Input		
Quadrature Decoder- QDEC0/QDEC1				
QEPA	QEPA quadrature input	Input		
QEPB	QEPB quadrature input	Input		
QEPI	Index input	Input		
System Controller Interface- SCIF				
XIN0, XIN1, XIN32	Crystal 0, 1, 32K Inputs	Analog		
XOUT0, XOUT1, XOUT32	Crystal 0, 1, 32K Output	Analog		
GCLK0 - GCLK1	Generic Clock Pins	Output		
Serial Peripheral Interface - SPI0, SPI1				
MISO	Master In Slave Out	I/O		
MOSI	Master Out Slave In	I/O		

than the oldest instruction. An instruction B is younger than an instruction A if it was sent down the pipeline later than A.

The addresses and priority of simultaneous events are shown in Table 4-4 on page 38. Some of the exceptions are unused in AVR32UC since it has no MMU, coprocessor interface, or floating-point unit.

Table 4-4. Priority and Handler Addresses for Events

Priority	Handler Address	Name	Event source	Stored Return Address
1	0x80000000	Reset	External input	Undefined
2	Provided by OCD system	OCD Stop CPU	OCD system	First non-completed instruction
3	EVBA+0x00	Unrecoverable exception	Internal	PC of offending instruction
4	EVBA+0x04	TLB multiple hit	MPU	PC of offending instruction
5	EVBA+0x08	Bus error data fetch	Data bus	First non-completed instruction
6	EVBA+0x0C	Bus error instruction fetch	Data bus	First non-completed instruction
7	EVBA+0x10	NMI	External input	First non-completed instruction
8	Autovectored	Interrupt 3 request	External input	First non-completed instruction
9	Autovectored	Interrupt 2 request	External input	First non-completed instruction
10	Autovectored	Interrupt 1 request	External input	First non-completed instruction
11	Autovectored	Interrupt 0 request	External input	First non-completed instruction
12	EVBA+0x14	Instruction Address	CPU	PC of offending instruction
13	EVBA+0x50	ITLB Miss	MPU	PC of offending instruction
14	EVBA+0x18	ITLB Protection	MPU	PC of offending instruction
15	EVBA+0x1C	Breakpoint	OCD system	First non-completed instruction
16	EVBA+0x20	Illegal Opcode	Instruction	PC of offending instruction
17	EVBA+0x24	Unimplemented instruction	Instruction	PC of offending instruction
18	EVBA+0x28	Privilege violation	Instruction	PC of offending instruction
19	EVBA+0x2C	Floating-point	UNUSED	
20	EVBA+0x30	Coprocessor absent	Instruction	PC of offending instruction
21	EVBA+0x100	Supervisor call	Instruction	PC(Supervisor Call) +2
22	EVBA+0x34	Data Address (Read)	CPU	PC of offending instruction
23	EVBA+0x38	Data Address (Write)	CPU	PC of offending instruction
24	EVBA+0x60	DTLB Miss (Read)	MPU	PC of offending instruction
25	EVBA+0x70	DTLB Miss (Write)	MPU	PC of offending instruction
26	EVBA+0x3C	DTLB Protection (Read)	MPU	PC of offending instruction
27	EVBA+0x40	DTLB Protection (Write)	MPU	PC of offending instruction
28	EVBA+0x44	DTLB Modified	UNUSED	

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

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

7.9 Timing Characteristics

7.9.1 Startup, Reset, and Wake-up Timing

The startup, reset, and wake-up timings are calculated using the following formula:

$$t = t_{CONST} + N_{CPU} \times t_{CPU}$$

Where t_{CONST} and N_{CPU} are found in Table 7-44. t_{CONST} is the delay relative to RCSYS, t_{CPU} is the period of the CPU clock. If another clock source than RCSYS is selected as CPU clock the startup time of the oscillator, $t_{OSCSTART}$, must be added to the wake-up time in the stop, deepstop, and static sleep modes. Please refer to the source for the CPU clock in the "Oscillator Characteristics" on page 57 for more details about oscillator startup times.

Table 7-44. Maximum Reset and Wake-up Timing

Parameter		Measuring	Max t_{CONST} (in µs)	Max N_{CPU}
Startup time from power-up, using regulator		VDDIN_5 rising (10 mV/ms) Time from V_{VDDIN_5}=0 to the first instruction entering the decode stage of CPU. VDDCORE is supplied by the internal regulator.	2600	0
Startup time from reset release		Time from releasing a reset source (except POR, BOD18, and BOD33) to the first instruction entering the decode stage of CPU.	1240	0
Wake-up	Idle	From wake-up event to the first instruction entering the decode stage of the CPU.	0	19
	Frozen		268	209
	Standby		268	209
	Stop		268+ $t_{OSCSTART}$	212
	Deepstop		268+ $t_{OSCSTART}$	212
	Static		268+ $t_{OSCSTART}$	212

Maximum SPI Frequency, Slave Input Mode

The maximum SPI slave input frequency is given by the following formula:

$$f_{SPCKMAX} = MIN(\frac{f_{CLKSPI} \times 2}{9}, \frac{1}{SPIn})$$

Where $SPIn$ is the MOSI setup and hold time, USPI7 + USPI8 or USPI10 + USPI11 depending on CPOL and NCPHA. f_{CLKSPI} is the maximum frequency of the CLK_SPI. Refer to the SPI chapter for a description of this clock.

Maximum SPI Frequency, Slave Output Mode

The maximum SPI slave output frequency is given by the following formula:

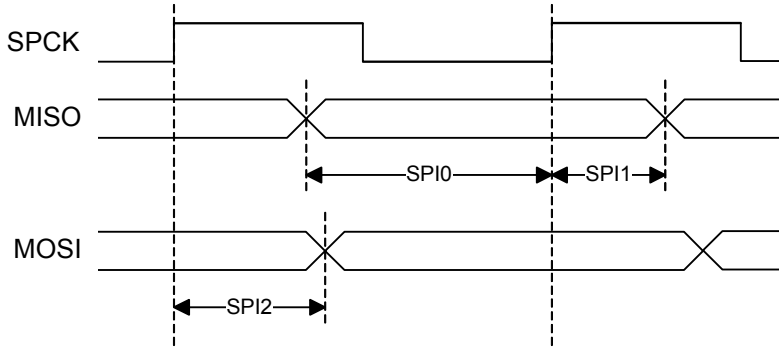
$$f_{SPCKMAX} = MIN(\frac{f_{CLKSPI} \times 2}{9}, f_{PINMAX}, \frac{1}{SPIn + t_{SETUP}})$$

Where $SPIn$ is the MISO delay, USPI6 or USPI9 depending on CPOL and NCPHA. T_{SETUP} is the SPI master setup time. Please refer to the SPI masterdatasheet for T_{SETUP} . f_{CLKSPI} is the maximum frequency of the CLK_SPI. Refer to the SPI chapter for a description of this clock. f_{PINMAX} is the maximum frequency of the SPI pins. Please refer to the I/O Pin Characteristics section for the maximum frequency of the pins.

7.9.4 SPI Timing

7.9.4.1 Master mode

Figure 7-11. SPI Master Mode With (CPOL= NCPHA= 0) or (CPOL= NCPHA= 1)



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.

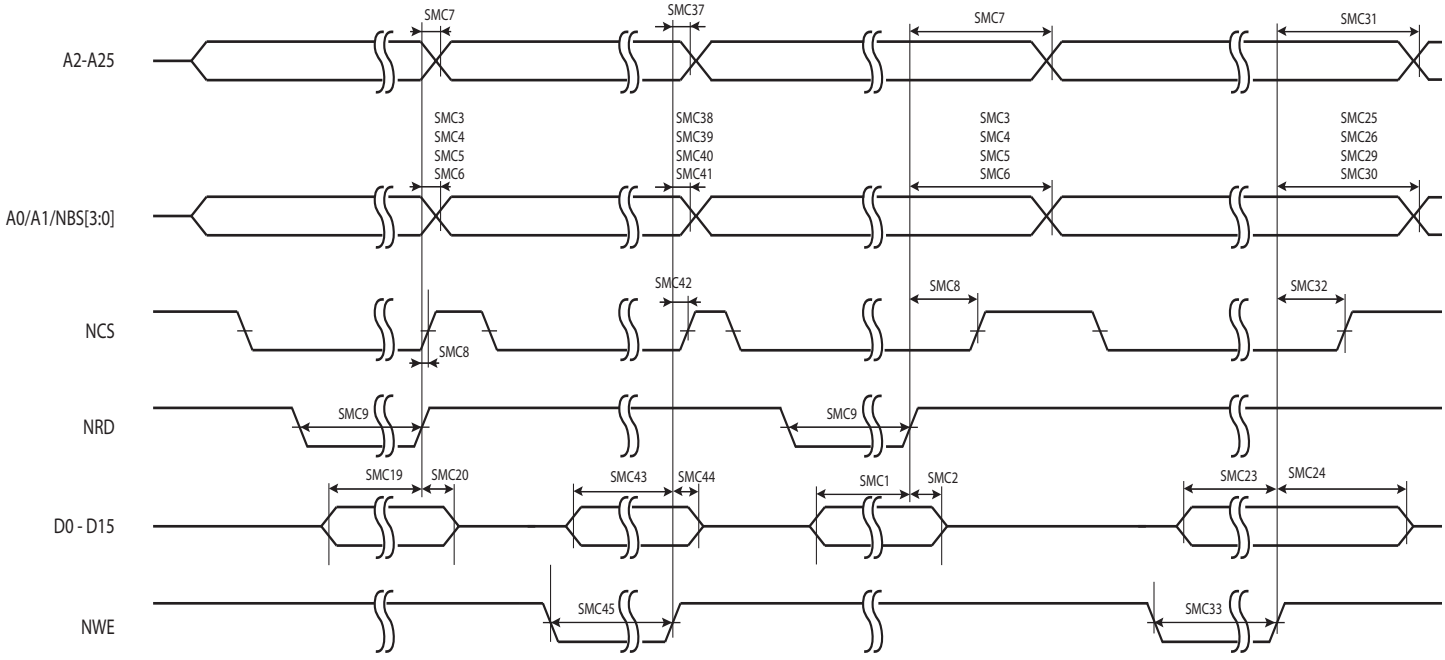
Table 7-53. SMC Read Signals with Hold Settings[1]

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”.

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 7-19. SDRAMC Signals relative to SDCK.

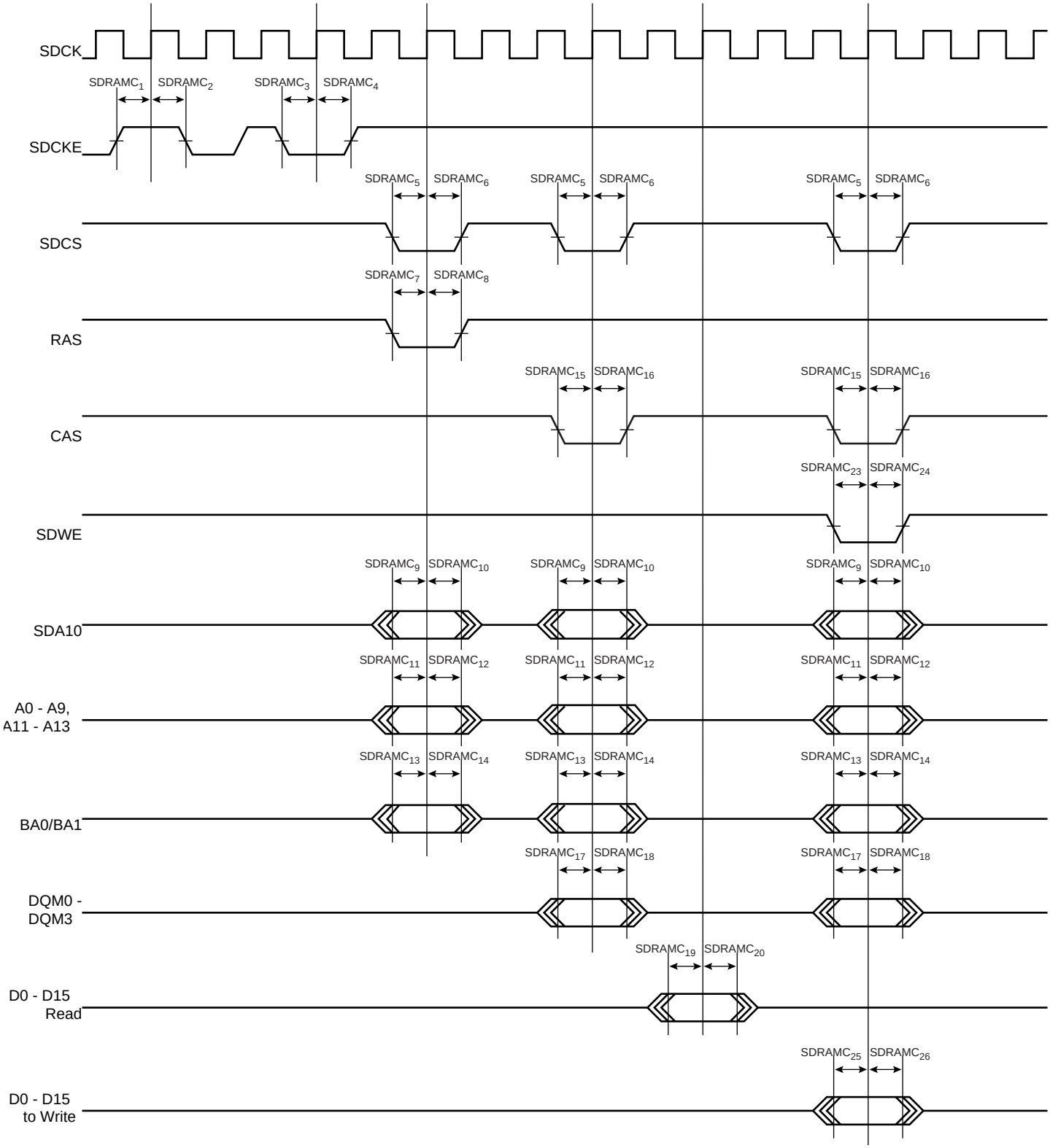
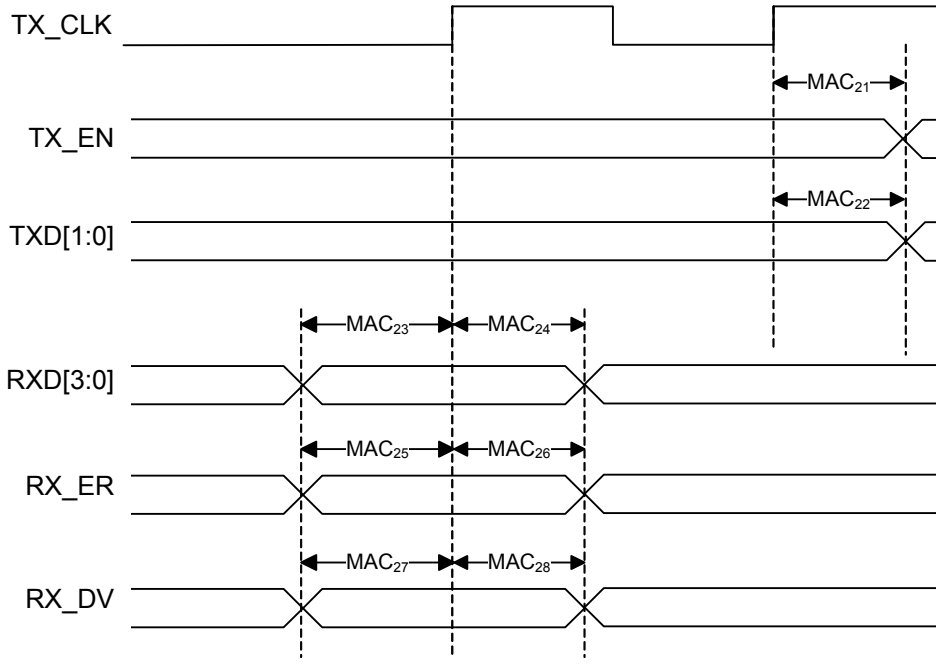


Table 7-61. Ethernet MAC RMII Specific Signals[1]

Symbol	Parameter	Conditions	Min.	Max.	Unit
MAC_{21}	TX_EN toggling from TX_CLK rising	V_{VDD} = 3.0V, drive strength of the pads set to the highest, external capacitor = 10pF on MACB pins	11.7	12.5	ns
MAC_{22}	TXD toggling from TX_CLK rising		11.7	12.5	ns
MAC_{23}	Setup for RXD from TX_CLK		4.5		ns
MAC_{24}	Hold for RXD from TX_CLK		0		ns
MAC_{25}	Setup for RX_ER from TX_CLK		3.4		ns
MAC_{26}	Hold for RX_ER from TX_CLK		0		ns
MAC_{27}	Setup for RX_DV from TX_CLK		4.4		ns
MAC_{28}	Hold for RX_DV from TX_CLK		0		ns

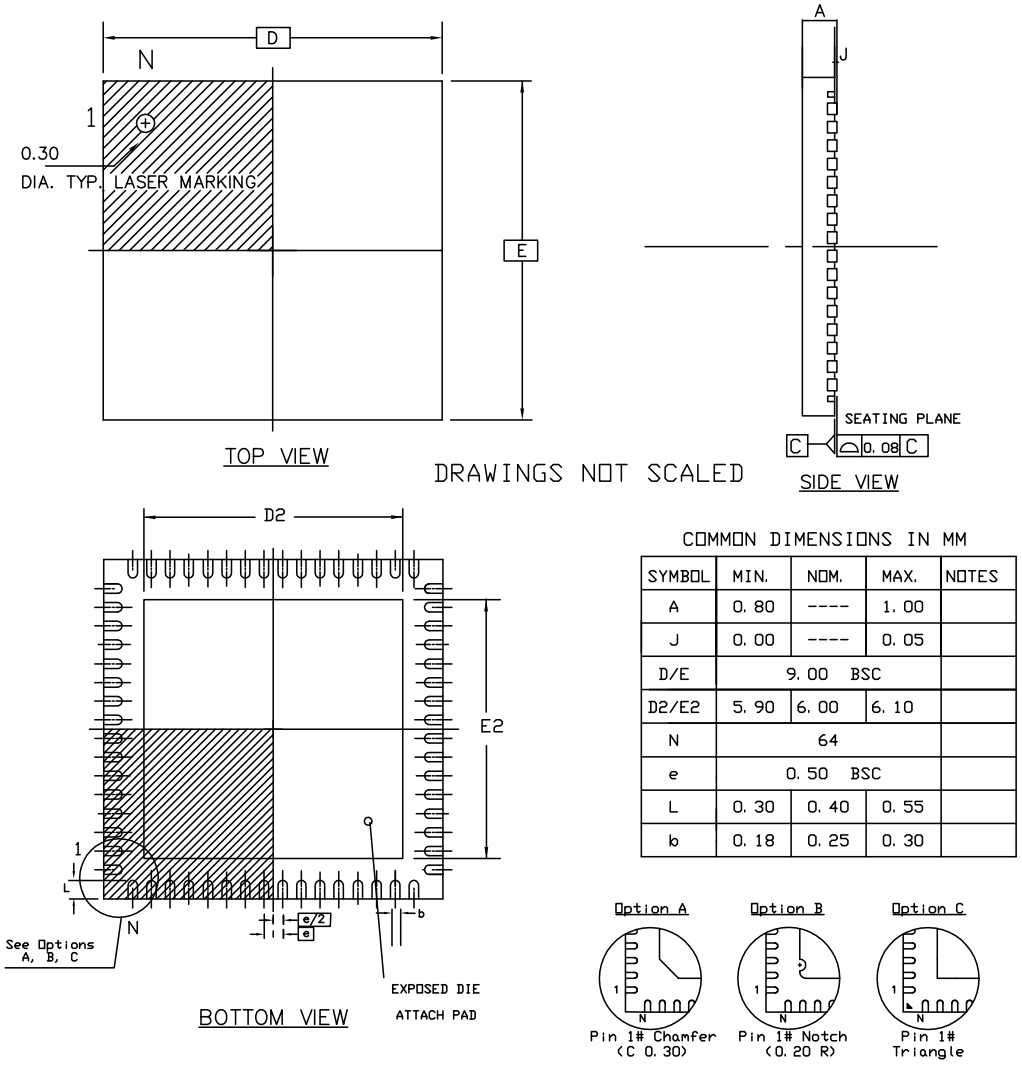
Note: 1. These values are based on simulation and characterization of other AVR microcontrollers manufactured in the same process technology. These values are not covered by test limits in production.

Figure 7-21. Ethernet MAC RMII Mode



8.2 Package Drawings

Figure 8-1. QFN-64 package drawing



SYMBOL	MIN.	NOM.	MAX.	NOTES
A	0.80	----	1.00	
J	0.00	----	0.05	
D/E		9.00 BSC		
D2/E2	5.90	6.00	6.10	
N		64		
e		0.50 BSC		
L	0.30	0.40	0.55	
b	0.18	0.25	0.30	

Note: The exposed pad is not connected to anything internally, but should be soldered to ground to increase board level reliability.

Table 8-2. Device and Package Maximum Weight

200	mg

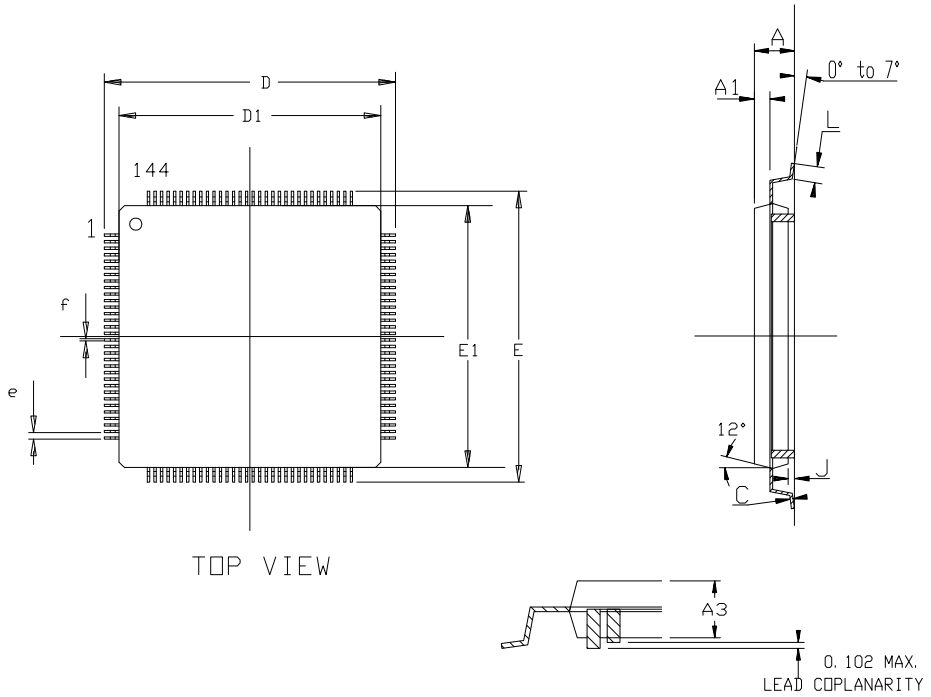
Table 8-3. Package Characteristics

Moisture Sensitivity Level	Jdec J-STD0-20D - MSL 3

Table 8-4. Package Reference

JEDEC Drawing Reference	MS-026
JESD97 Classification	E3

Figure 8-4. LQFP-144 package drawing



	MM			INCH		
	Min	Nom	Max	Min	Nom	Max
A	–	–	1.60	–	–	.063
C	0.09	–	0.20	.004	–	.008
A3	1.35	1.40	1.45	.053	.055	.057
D	21.90	22.00	22.10	.862	.866	.870
D1	19.90	20.00	20.10	.783	.787	.791
E	21.90	22.00	22.10	.862	.866	.870
E1	19.90	20.00	20.10	.783	.787	.791
J	0.05	–	0.15	.002	–	.006
L	0.45	0.60	0.75	.018	.024	.030
e	0.50 BSC			.0197 BSC		
f	0.22 BSC			.009 BSC		

Table 8-11. Device and Package Maximum Weight

1300	mg

Table 8-12. Package Characteristics

Moisture Sensitivity Level	Jdec J-STD0-20D - MSL 3

Table 8-13. Package Reference

JEDEC Drawing Reference	MS-026
JESD97 Classification	E3

Fix/Workaround
None.

3 **In host mode, the disconnection during OUT transition is not supported**
In USB host mode, a pipe can not work if the previous USB device disconnection has occurred during a USB transfer.
Fix/Workaround
Reset the USBC (USBCON.USB=0 and =1) after a device disconnection (UHINT.DDISCI).

4 **In USB host mode, entering suspend mode can fail**
In USB host mode, entering suspend mode can fail when UHCON.SOFE=0 is done just after a SOF reception (UHINT.HSOFI).
Fix/Workaround
Check that UHNUM.FLENHIGH is below 185 in Full speed and below 21 in Low speed before clearing UHCON.SOFE.

5 **In USB host mode, entering suspend mode for low speed device can fail when the USB freeze (USBCON.FRZCLK=1) is done just after UHCON.SOFE=0.**
Fix/Workaround
When entering suspend mode (UHCON.SOFE is cleared), check that USBFSM.DRDSTATE is not equal to three before freezing the clock (USBCON.FRZCLK=1).

10.1.11 WDT

1 **WDT Control Register does not have synchronization feedback**
When writing to the Timeout Prescale Select (PSEL), Time Ban Prescale Select (TBAN), Enable (EN), or WDT Mode (MODE) fieldss of the WDT Control Register (CTRL), a synchronizer is started to propagate the values to the WDT clcok domain. This synchronization takes a finite amount of time, but only the status of the synchronization of the EN bit is reflected back to the user. Writing to the synchronized fields during synchronization can lead to undefined behavior.
Fix/Workaround
-When writing to the affected fields, the user must ensure a wait corresponding to 2 clock cycles of both the WDT peripheral bus clock and the selected WDT clock source.
-When doing writes that changes the EN bit, the EN bit can be read back until it reflects the written value.

10.2.12 WDT

1 **Clearing the Watchdog Timer (WDT) counter in second half of timeout period will issue a Watchdog reset**

If the WDT counter is cleared in the second half of the timeout period, the WDT will immediately issue a Watchdog reset.

Fix/Workaround

Use twice as long timeout period as needed and clear the WDT counter within the first half of the timeout period. If the WDT counter is cleared after the first half of the timeout period, you will get a Watchdog reset immediately. If the WDT counter is not cleared at all, the time before the reset will be twice as long as needed.

2 **WDT Control Register does not have synchronization feedback**

When writing to the Timeout Prescale Select (PSEL), Time Ban Prescale Select (TBAN), Enable (EN), or WDT Mode (MODE) fieldss of the WDT Control Register (CTRL), a synchronizer is started to propagate the values to the WDT clcok domain. This synchronization takes a finite amount of time, but only the status of the synchronization of the EN bit is reflected back to the user. Writing to the synchronized fields during synchronization can lead to undefined behavior.

Fix/Workaround

-When writing to the affected fields, the user must ensure a wait corresponding to 2 clock cycles of both the WDT peripheral bus clock and the selected WDT clock source.

-When doing writes that changes the EN bit, the EN bit can be read back until it reflects the written value.