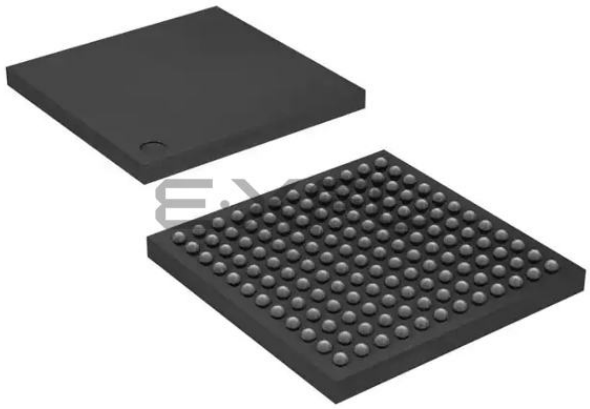


Welcome to E-XFL.COM

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	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	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	32K 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-TFBGA
Supplier Device Package	144-FFBGA (11x11)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at32uc3a0128-ctur

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

Table 9-2. Flash Memory Parameters

Part Number	Flash Size (*FLASH_PW*)	Number of pages (*FLASH_P*)	Page size (*FLASH_W*)	General Purpose Fuse bits (*FLASH_F*)
AT32UC3A0512	512 Kbytes	1024	128 words	32 fuses
AT32UC3A1512	512 Kbytes	1024	128 words	32 fuses
AT32UC3A0256	256 Kbytes	512	128 words	32 fuses
AT32UC3A1256	256 Kbytes	512	128 words	32 fuses
AT32UC3A1128	128 Kbytes	256	128 words	32 fuses
AT32UC3A0128	128 Kbytes	256	128 words	32 fuses

9.3 Bus Matrix Connections

Accesses to unused areas returns an error result to the master requesting such an access.

The bus matrix has the several masters and slaves. Each master has its own bus and its own decoder, thus allowing a different memory mapping per master. The master number in the table below can be used to index the HMATRIX control registers. For example, MCFG0 is associated with the CPU Data master interface.

Table 9-3. High Speed Bus masters

Master 0	CPU Data
Master 1	CPU Instruction
Master 2	CPU SAB
Master 3	PDCA
Master 4	MACB DMA
Master 5	USBB DMA

Each slave has its own arbiter, thus allowing a different arbitration per slave. The slave number in the table below can be used to index the HMATRIX control registers. For example, SCFG3 is associated with the Internal SRAM Slave Interface.

Table 9-4. High Speed Bus slaves

Slave 0	Internal Flash
Slave 1	HSB-PB Bridge 0
Slave 2	HSB-PB Bridge 1
Slave 3	Internal SRAM
Slave 4	USBB DPRAM
Slave 5	EBI

10. Peripherals

10.1 Peripheral address map

Table 10-1. Peripheral Address Mapping

Address		Peripheral Name	Bus
0xE0000000	USBB	USBB Slave Interface - USBB	HSB
0xFFFE0000	USBB	USBB Configuration Interface - USBB	PBB
0xFFFE1000	HMATRIX	HMATRIX Configuration Interface - HMATRIX	PBB
0xFFFE1400	FLASHC	Flash Controller - FLASHC	PBB
0xFFFE1800	MACB	MACB Configuration Interface - MACB	PBB
0xFFFE1C00	SMC	Static Memory Controller Configuration Interface - SMC	PBB
0xFFFE2000	SDRAMC	SDRAM Controller Configuration Interface - SDRAMC	PBB
0xFFFF0000	PDCA	Peripheral DMA Interface - PDCA	PBA
0xFFFF0800	INTC	Interrupt Controller Interface - INTC	PBA
0xFFFF0C00	PM	Power Manager - PM	PBA
0xFFFF0D00	RTC	Real Time Clock - RTC	PBA
0xFFFF0D30	WDT	WatchDog Timer - WDT	PBA
0xFFFF0D80	EIC	External Interrupt Controller - EIC	PBA
0xFFFF1000	GPIO	General Purpose IO Controller - GPIO	PBA
0xFFFF1400	USART0	Universal Synchronous Asynchronous Receiver Transmitter - USART0	PBA
0xFFFF1800	USART1	Universal Synchronous Asynchronous Receiver Transmitter - USART1	PBA

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

  - 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

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.

12.3 Regulator characteristics

Table 12-2. Electrical characteristics

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
V_{VDDIN}	Supply voltage (input)		3	3.3	3.6	V
V_{VDDOUT}	Supply voltage (output)		1.81	1.85	1.89	V
I_{OUT}	Maximum DC output current with $V_{VDDIN = 3.3V}$				100	mA
	Maximum DC output current with $V_{VDDIN = 2.7V}$				90	mA
I_{SCR}	Static Current of internal regulator	Low Power mode (stop, deep stop or static) at T_A =25°C		10		µA

Table 12-3. Decoupling requirements

Symbol	Parameter	Condition	Typ.	Techno.	Units
C_{IN1}	Input Regulator Capacitor 1		1	NPO	nF
C_{IN2}	Input Regulator Capacitor 2		4.7	X7R	uF
C_{OUT1}	Output Regulator Capacitor 1		470	NPO	pF
C_{OUT2}	Output Regulator Capacitor 2		2.2	X7R	uF

12.4 Analog characteristics

Table 12-4. Electrical characteristics

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
V_{ADVREF}	Analog voltage reference (input)		2.6		3.6	V

Table 12-5. Decoupling requirements

Symbol	Parameter	Condition	Typ.	Techno.	Units
C_{VREF1}	Voltage reference Capacitor 1		10	-	nF
C_{VREF2}	Voltage reference Capacitor 2		1	-	uF

12.4.1 BOD

Table 12-6. BODLEVEL Values

BODLEVEL Value	Typ.	Typ.	Typ.	Units.
00 0000b	1.40	1.47	1.55	V
01 0111b	1.45	1.52	1.6	V
01 1111b	1.55	1.6	1.65	V
10 0111b	1.65	1.69	1.75	V

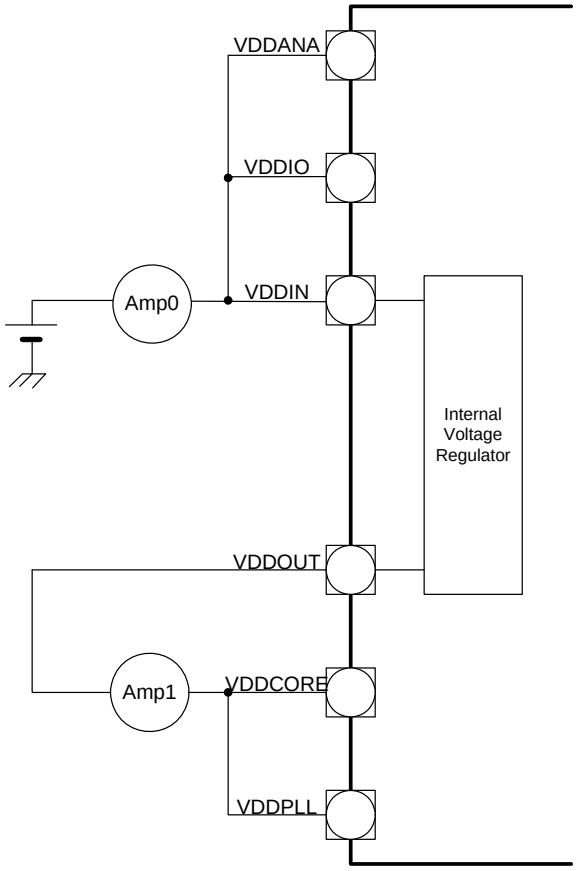
The values in Table 12-6 describes the values of the BODLEVEL in the flash FGPFR register.

12.5 Power Consumption

The values in Table 12-9 and Table 12-10 on page 46 are measured values of power consumption with operating conditions as follows:

- V_{DDIO} = 3.3V
- V_{DDCORE} = V_{DDPLL} = 1.8V
- T_A = 25°C, T_A = 85°C
- I/Os are configured in input, pull-up enabled.

Figure 12-1. Measurement setup



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

12.9 EBI Timings

These timings are given for worst case process, T = 85·C, VDDCORE = 1.65V, VDDIO = 3V and 40 pF load capacitance.

Table 12-22. SMC Clock Signal.

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

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

Table 12-23. SMC Read Signals with Hold Settings

Symbol	Parameter	Min	Units
NRD Controlled (READ_MODE = 1)			
SMC_1	Data Setup before NRD High	12	ns
SMC_2	Data Hold after NRD High	0	
SMC_3	NRD High to NBS0/A0 Change[1]	nrd hold length * t_{CPSMC} - 1.3	
SMC_4	NRD High to NBS1 Change[1]	nrd hold length * t_{CPSMC} - 1.3	
SMC_5	NRD High to NBS2/A1 Change[1]	nrd hold length * t_{CPSMC} - 1.3	
SMC_6	NRD High to NBS3 Change[1]	nrd hold length * t_{CPSMC} - 1.3	
SMC_7	NRD High to A2 - A25 Change[1]	nrd hold length * t_{CPSMC} - 1.3	
SMC_8	NRD High to NCS Inactive[1]	(nrd hold length - ncs rd hold length) * t_{CPSMC} - 2.3	
SMC_9	NRD Pulse Width	nrd pulse length * t_{CPSMC} - 1.4	
NRD Controlled (READ_MODE = 0)			
SMC_{10}	Data Setup before NCS High	11.5	ns
SMC_{11}	Data Hold after NCS High	0	
SMC_{12}	NCS High to NBS0/A0 Change[1]	ncs rd hold length * t_{CPSMC} - 2.3	
SMC_{13}	NCS High to NBS0/A0 Change[1]	ncs rd hold length * t_{CPSMC} - 2.3	
SMC_{14}	NCS High to NBS2/A1 Change[1]	ncs rd hold length * t_{CPSMC} - 2.3	
SMC_{15}	NCS High to NBS3 Change[1]	ncs rd hold length * t_{CPSMC} - 2.3	
SMC_{16}	NCS High to A2 - A25 Change[1]	ncs rd hold length * t_{CPSMC} - 4	
SMC_{17}	NCS High to NRD Inactive[1]	ncs rd hold length - nrd hold length)* t_{CPSMC} - 1.3	
SMC_{18}	NCS Pulse Width	ncs rd pulse length * t_{CPSMC} - 3.6	

Note: 1. hold length = total cycle duration - setup duration - pulse duration. “hold length” is for “ncs rd hold length” or “nrd hold length”.

Table 12-24. SMC Read Signals with no Hold Settings

Symbol	Parameter	Min	Units
NRD Controlled (READ_MODE = 1)			
SMC_{19}	Data Setup before NRD High	13.7	ns
SMC_{20}	Data Hold after NRD High	1	
NRD Controlled (READ_MODE = 0)			
SMC_{21}	Data Setup before NCS High	13.3	ns
SMC_{22}	Data Hold after NCS High	0	

Table 12-25. SMC Write Signals with Hold Settings

Symbol	Parameter	Min	Units
NRD Controlled (READ_MODE = 1)			
SMC_{23}	Data Out Valid before NWE High	(nwe pulse length - 1) * t_{CPSMC} - 0.9	ns
SMC_{24}	Data Out Valid after NWE High(1)	nwe hold length * t_{CPSMC} - 6	
SMC_{25}	NWE High to NBS0/A0 Change(1)	nwe hold length * t_{CPSMC} - 1.9	
SMC_{26}	NWE High to NBS1 Change(1)	nwe hold length * t_{CPSMC} - 1.9	
SMC_{29}	NWE High to NBS2/A1 Change(1)	nwe hold length * t_{CPSMC} - 1.9	
SMC_{30}	NWE High to NBS3 Change(1)	nwe hold length * t_{CPSMC} - 1.9	
SMC_{31}	NWE High to A2 - A25 Change(1)	nwe hold length * t_{CPSMC} - 1.7	
SMC_{32}	NWE High to NCS Inactive(1)	(nwe hold length - ncs wr hold length)* t_{CPSMC} - 2.9	
SMC_{33}	NWE Pulse Width	nwe pulse length * t_{CPSMC} - 0.9	
NRD Controlled (READ_MODE = 0)			
SMC_{34}	Data Out Valid before NCS High	(ncs wr pulse length - 1)* t_{CPSMC} - 4.6	ns
SMC_{35}	Data Out Valid after NCS High(1)	ncs wr hold length * t_{CPSMC} - 5.8	
SMC_{36}	NCS High to NWE Inactive(1)	(ncs wr hold length - nwe hold length)* t_{CPSMC} - 0.6	

Note: 1. hold length = total cycle duration - setup duration - pulse duration. “hold length” is for “ncs wr hold length” or “nwe hold length"

12.10 JTAG Timings

12.10.1 JTAG Interface Signals

Table 12-29. JTAG Interface Timing specification

Symbol	Parameter	Conditions	Min	Max	Units
$JTAG_0$	TCK Low Half-period	(1)	6		ns
$JTAG_1$	TCK High Half-period	(1)	3		ns
$JTAG_2$	TCK Period	(1)	9		ns
$JTAG_3$	TDI, TMS Setup before TCK High	(1)	1		ns
$JTAG_4$	TDI, TMS Hold after TCK High	(1)	0		ns
$JTAG_5$	TDO Hold Time	(1)	4		ns
$JTAG_6$	TCK Low to TDO Valid	(1)		6	ns
$JTAG_7$	Device Inputs Setup Time	(1)			ns
$JTAG_8$	Device Inputs Hold Time	(1)			ns
$JTAG_9$	Device Outputs Hold Time	(1)			ns
$JTAG_{10}$	TCK to Device Outputs Valid	(1)			ns

Note: 1. V_{VDDIO} from 3.0V to 3.6V, maximum external capacitor = 40pF

Table 12-30. SPI Timings

Symbol	Parameter	Conditions	Min	Max	Units
SPI_0	MISO Setup time before SPCK rises (master)	3.3V domain(1)	22 + (t_{CPMCK})/2(2)		ns
SPI_1	MISO Hold time after SPCK rises (master)	3.3V domain(1)	0		ns
SPI_2	SPCK rising to MOSI Delay (master)	3.3V domain(1)		7	ns
SPI_3	MISO Setup time before SPCK falls (master)	3.3V domain(1)	22 + (t_{CPMCK})/2(2)		ns
SPI_4	MISO Hold time after SPCK falls (master)	3.3V domain (1)	0		ns
SPI_5	SPCK falling to MOSI Delay (master)	3.3V domain (1)		7	ns
SPI_6	SPCK falling to MISO Delay (slave)	3.3V domain (1)		26.5	ns
SPI_7	MOSI Setup time before SPCK rises (slave)	3.3V domain (1)	0		ns
SPI_8	MOSI Hold time after SPCK rises (slave)	3.3V domain (1)	1.5		ns
SPI_9	SPCK rising to MISO Delay (slave)	3.3V domain (1)		27	ns
SPI_{10}	MOSI Setup time before SPCK falls (slave)	3.3V domain (1)	0		ns
SPI_{11}	MOSI Hold time after SPCK falls (slave)	3.3V domain (1)	1		ns

Notes: 1. 3.3V domain: V_{VDDIO} from 3.0V to 3.6V, maximum external capacitor = 40 pF.
2. t_{CPMCK}: Master Clock period in ns.

12.12 MACB Characteristics

Table 12-31. Ethernet MAC Signals

Symbol	Parameter	Conditions	Min (ns)	Max (ns)
$EMAC_1$	Setup for EMDIO from EMDC rising	Load: 20pF(2)		
$EMAC_2$	Hold for EMDIO from EMDC rising	Load: 20pF(2)		
$EMAC_3$	EMDIO toggling from EMDC falling	Load: 20pF(2)		

Notes: 1. f: MCK frequency (MHz)
2. V_{VDDIO} from 3.0V to 3.6V, maximum external capacitor = 20 pF

Table 12-32. Ethernet MAC MII Specific Signals

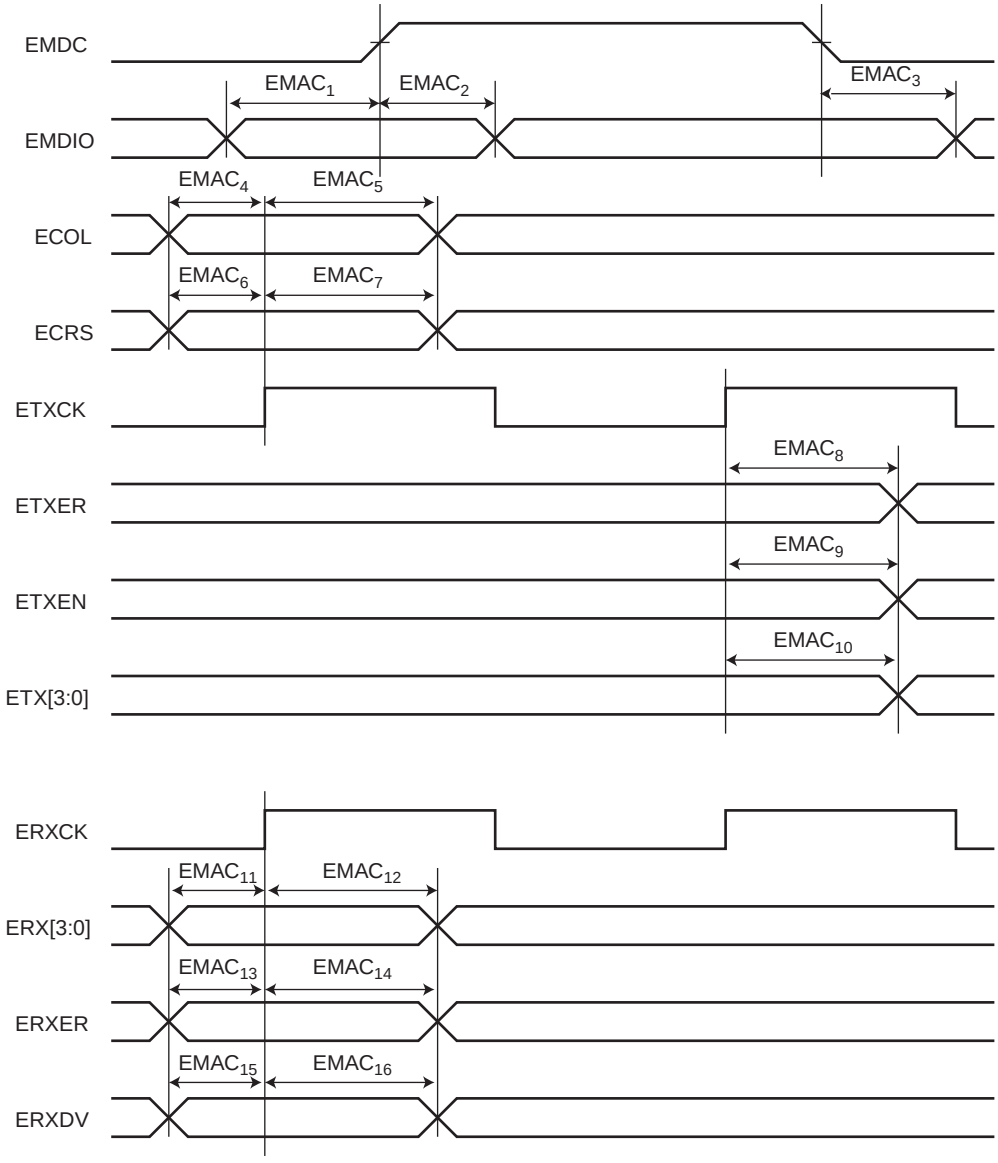
Symbol	Parameter	Conditions	Min (ns)	Max (ns)
$EMAC_4$	Setup for ECOL from ETXCK rising	Load: 20pF (1)	3	
$EMAC_5$	Hold for ECOL from ETXCK rising	Load: 20pF (1)	0	
$EMAC_6$	Setup for ECRS from ETXCK rising	Load: 20pF (1)	3	
$EMAC_7$	Hold for ECRS from ETXCK rising	Load: 20pF (1)	0	
$EMAC_8$	ETXER toggling from ETXCK rising	Load: 20pF (1)		15
$EMAC_9$	ETXEN toggling from ETXCK rising	Load: 20pF (1)		15
$EMAC_{10}$	ETX toggling from ETXCK rising	Load: 20pF (1)		15
$EMAC_{11}$	Setup for ERX from ERXCK	Load: 20pF (1)	1	

Table 12-32. Ethernet MAC MII Specific Signals

Symbol	Parameter	Conditions	Min (ns)	Max (ns)
$EMAC_{12}$	Hold for ERX from ERXCK	Load: 20pF (1)	1.5	
$EMAC_{13}$	Setup for ERXER from ERXCK	Load: 20pF (1)	1	
$EMAC_{14}$	Hold for ERXER from ERXCK	Load: 20pF (1)	0.5	
$EMAC_{15}$	Setup for ERXDV from ERXCK	Load: 20pF (1)	1.5	
$EMAC_{16}$	Hold for ERXDV from ERXCK	Load: 20pF (1)	1	

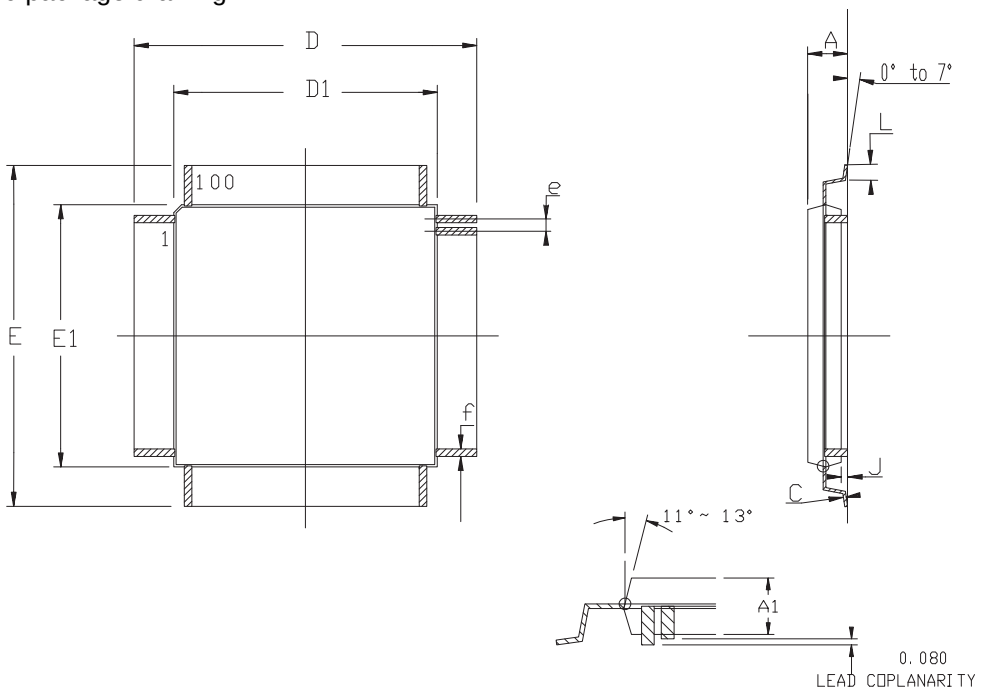
Note: 1. V_{VDDIO} from 3.0V to 3.6V, maximum external capacitor = 20 pF

Figure 12-10. Ethernet MAC MII Mode



13.2 Package Drawings

Figure 13-1. TQFP-100 package drawing



	MM		INCH	
	Min	Max	Min	Max
A	----	1.20	----	.047
A1	0.95	1.05	.037	.041
C	0.09	0.20	.004	.008
D	16.00 BSC		.630 BSC	
D1	14.00 BSC		.551 BSC	
E	16.00 BSC		.630 BSC	
E1	14.00 BSC		.551 BSC	
J	0.05	0.15	.002	.006
L	0.45	0.75	.018	.030
e	0.50 BSC		.020 BSC	
f	0.17	0.27	.007	.011

Table 13-2. Device and Package Maximum Weight

500	mg

Table 13-3. Package Characteristics

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

Table 13-4. Package Reference

JEDEC Drawing Reference	MS-026
JESD97 Classification	E3

None.

15.2.9 USART

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

15.2.10 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 specific case.
   2. Execute the RETE instruction.

15.4 Rev. H

15.4.1 PWM

1. **PWM channel interrupt enabling triggers an interrupt**
   When enabling a PWM channel that is configured with center aligned period (CALG=1), an interrupt is signalled.

   Fix/Workaround
   When using center aligned mode, enable the channel and read the status before channel interrupt is enabled.

2. **PWM counter restarts at 0x0001**
   The PWM counter restarts at 0x0001 and not 0x0000 as specified. Because of this the first PWM period has one more clock cycle.
   Fix/Workaround
   - The first period is 0x0000, 0x0001, ..., period
   - Consecutive periods are 0x0001, 0x0002, ..., period

3. **PWM update period to a 0 value does not work**
   It is impossible to update a period equal to 0 by the using the PWM update register (PWM_CUPD).

   Fix/Workaround
   Do not update the PWM_CUPD register with a value equal to 0.

15.4.2 ADC

1. **Sleep Mode activation needs additional A to D conversion**
   If the ADC sleep mode is activated when the ADC is idle the ADC will not enter sleep mode before after the next AD conversion.

   Fix/Workaround
   Activate the sleep mode in the mode register and then perform an AD conversion.

15.4.3 SPI

1. **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.

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

   Fix/Workaround
   Do not set FDIV = 1

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

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.

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.

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

Atmel Asia Limited
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 Munich GmbH
Business Campus
Parkring 4
D-85748 Garching b. Munich
GERMANY
Tel: (+49) 89-31970-0
Fax: (+49) 89-3194621

Atmel Japan
16F, Shin Osaki Kangyo Bldg.
1-6-4 Osaka Shinagawa-ku
Tokyo 104-0032
JAPAN
Tel: (+81) 3-6417-0300
Fax: (+81) 3-6417-0370