# Atmel - AT32UC3C2512C-Z2ZT Datasheet

**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 | 50MHz |
| 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 | 512KB (512K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 64K x 8 |
| Voltage - Supply (Vcc/Vdd) | 3V ~ 5.5V |
| Data Converters | A/D 11x12b; D/A 2x12b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 125°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 64-VFQFN Exposed Pad |
| Supplier Device Package | 64-QFN (9x9) |
| Purchase URL | https://www.e-xfl.com/product-detail/atmel/at32uc3c2512c-z2zt |

## 2.2 Configuration Summary

**Table 2-1.** Configuration Summary

| Feature | AT32UC3C0512C | AT32UC3C1512C | AT32UC3C2512C |
|---|---|---|---|
| Flash | 512 KB | 512 KB | 512 KB |
| SRAM | 64KB | 64KB | 64KB |
| 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<br>RMII/MII | 1<br>RMII/MII | 1<br>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)<br>Crystal Oscillator 0.4-20 MHz (OSC0)<br>Crystal Oscillator 32 KHz (OSC32K)<br>RC Oscillator 115 kHz (RCSYS)<br>RC Oscillator 8 MHz (RC8M)<br>RC Oscillator 120 MHz (RC120M) | | |
| | 0.4-20 MHz (OSC1) | | - |
| 12-bit ADC<br>number of channels | 1<br>16 | 1<br>16 | 1<br>11 |
| 12-bit DAC<br>number of channels | 1<br>4 | 1<br>4 | 1<br>2 |
| Analog Comparators | 4 | 4 | 2 |
| JTAG | 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 |
| | | 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. Pin type x1 is pin with drive strength of x1. Pin type x1/x2 is pin with programmable drive strength of x1 or x2. Pin type x2/x4 is pin with programmable drive strength of x2 or x4. The drive strength is programmable through ODCR0, ODCR0S, ODCR0C, ODCR0T registers of GPIO. Refer to "Electrical Characteristics" on page 49 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 |
| aWire DATAOUT | aWire output in two-pin mode |
| JTAG port connections | JTAG debug port |
| Oscillators | OSC0, OSC32 |

### 3.2.3 Oscillator Pinout

The oscillators are not mapped to the normal GPIO functions and their muxings are controlled by registers in the System Control Interface (SCIF). Please refer to the SCIF chapter for more information about this.

**Table 3-3.** Oscillator pinout

| QFN64/ TQFP64 pin | TQFP100 pin | LQFP144 pin | Pad | Oscillator pin |
|---|---|---|---|---|
| 31 | 47 | 69 | PB30 | xin0 |
| | 99 | 143 | PB02 | xin1 |
| 62 | 96 | 140 | PB00 | xin32 |
| 32 | 48 | 70 | PB31 | xout0 |
| | 100 | 144 | PB03 | xout1 |
| 63 | 97 | 141 | PB01 | xout32 |

### 3.2.4 JTAG port connections

If the JTAG is enabled, the JTAG will take control over a number of pins, irrespectively of the I/O Controller configuration.

**Table 3-4.** JTAG pinout

| QFN64/ TQFP64 pin | TQFP100 pin | LQFP144 pin | Pin name | JTAG pin |
|---|---|---|---|---|
| 2 | 2 | 2 | PA01 | TDI |
| 3 | 3 | 3 | PA02 | TDO |
| 4 | 4 | 4 | PA03 | TMS |
| 1 | 1 | 1 | PA00 | TCK |

### 3.2.5 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 GPIO configuration. Three different OCD trace pin mappings are possible,

**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]<br>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 | | |

#### 4.3.2.5 *Unaligned Reference Handling*

AVR32UC does not support unaligned accesses, except for doubleword accesses. AVR32UC is able to perform word-aligned *st.d* and *ld.d*. Any other unaligned memory access will cause an address exception. Doubleword-sized accesses with word-aligned pointers will automatically be performed as two word-sized accesses.

The following table shows the instructions with support for unaligned addresses. All other instructions require aligned addresses.

**Table 4-1.** Instructions with Unaligned Reference Support

| Instruction | Supported Alignment |
|---|---|
| ld.d | Word |
| st.d | Word |

#### 4.3.2.6 *Unimplemented Instructions*

The following instructions are unimplemented in AVR32UC, and will cause an Unimplemented Instruction Exception if executed:

- All SIMD instructions
- All coprocessor instructions if no coprocessors are present
- retj, incjosp, popjc, pushjc
- tlbr, tlbs, tlbw
- cache

#### 4.3.2.7 *CPU and Architecture Revision*

Three major revisions of the AVR32UC CPU currently exist. The device described in this datasheet uses CPU revision 3.

The Architecture Revision field in the CONFIG0 system register identifies which architecture revision is implemented in a specific device.

AVR32UC CPU revision 3 is fully backward-compatible with revisions 1 and 2, ie. code compiled for revision 1 or 2 is binary-compatible with revision 3 CPUs.

### 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

# 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 49 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.

  - 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.

## 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 | 25 MHz |
| 1 | 2 cycles | 50 MHz |

**Table 7-16.** Flash Characteristics

| Symbol | Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $t_{FPP}$ | Page programming time | $f_{CLK\_HSB}$ = 50 MHz | | 17 | | ms |
| $t_{FPE}$ | Page erase time | | | 17 | | |
| $t_{FFP}$ | Fuse programming time | | | 1.3 | | |
| $t_{FEA}$ | Full chip erase time (EA) | | | 18.3 | | |
| $t_{FCE}$ | 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) | | 10 k | | | cycles |
| $N_{FFUSE}$ | General Purpose fuses endurance (write/bit) | | 500 | | | cycles |
| $t_{RET}$ | Data retention | | 15 | | | years |

**Table 7-29.** ADC Decoupling requirements

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|---|
| $C_{ADCREFPN}$ | ADCREFP/ADCREFN capacitance | No voltage reference appplied on ADCREFP/ADCREFN | | 100 | | nF |

**Table 7-30.** ADC Inputs

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|---|
| $V_{ADCINn}$ | ADC input voltage range | | 0 | | $V_{VDDANA}$ | V |
| $C_{ONCHIP}$ | Internal Capacitance | ADC used without S/H | | | 5 | pF |
| | | ADC used with S/H | | | 4 | |
| $R_{ONCHIP}$ | Switch resistance | ADC used without S/H | | | 5.1 | kΩ |
| | | ADC used with S/H | | | 4.6 | |

**Figure 7-3.** ADC input

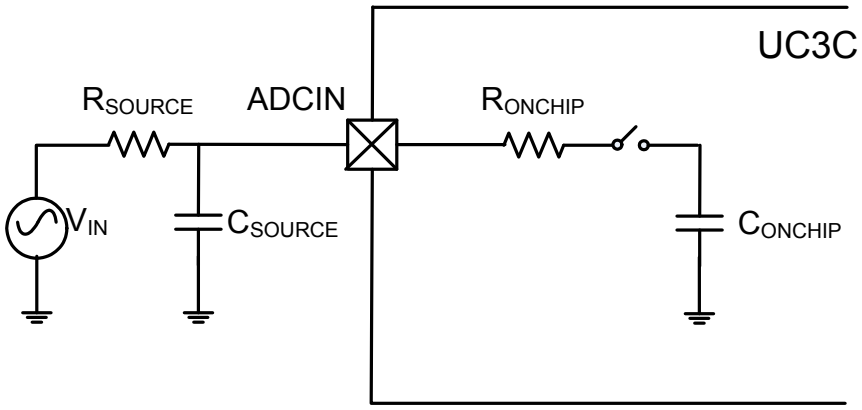



**Table 7-31.** ADC Transfer Characteristics 12-bit Resolution Mode[1]

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|---|
| RES | Resolution | Differential mode, $V_{VDDANA}$ = 3V, $V_{ADCREF0}$ = 1V, ADCFIA.SEQCFGn.SRES = 0 ($F_{adc}$ = 1.2MHz) | | | 12 | Bit |
| INL | Integral Non-Linearity | | | | 6 | LSB |
| DNL | Differential Non-Linearity | | | | 5 | LSB |
| | Offset error | | -10 | | 10 | mV |
| | Gain error | | -30 | | 30 | mV |

**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 | | | | 2 | LSB |
| DNL | Differential Non-Linearity | | | | 2 | LSB |
| | Offset error | | -30 | | 30 | mV |
| | Gain error | | -30 | | 30 | 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 |

**Figure 7-9.** USART in SPI Slave Mode With (CPOL= CPHA= 0) or (CPOL= CPHA= 1)

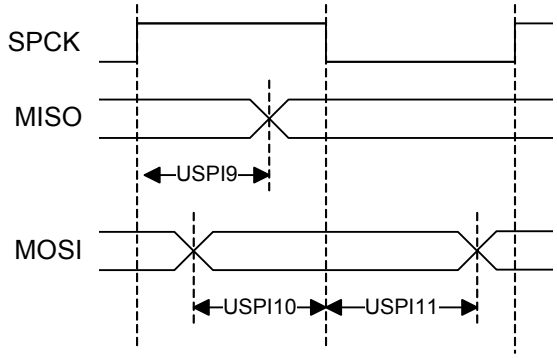



**Figure 7-10.** USART in SPI Slave Mode NPCS Timing

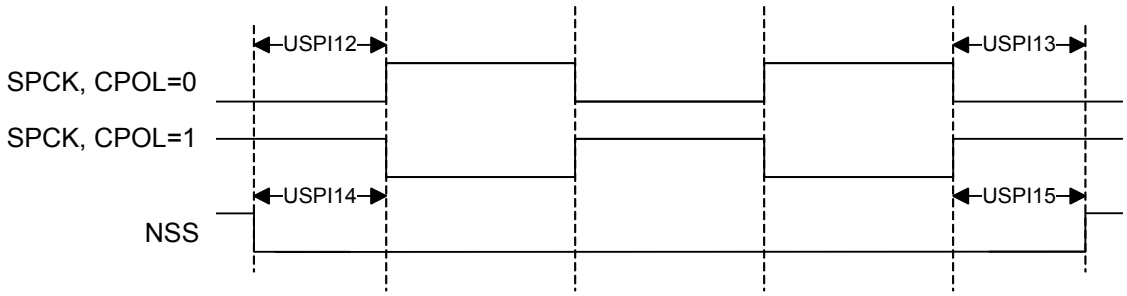



**Table 7-47.** USART in SPI mode Timing, Slave Mode[1]

| Symbol | Parameter | Conditions | Min | Max | Units |
|---|---|---|---|---|---|
| USPI6 | SPCK falling to MISO delay | external capacitor = 40pF | | 28.5 | ns |
| USPI7 | MOSI setup time before SPCK rises | | $t_{SAMPLE}$[2] + $t_{CLK\_USART}$ | | ns |
| USPI8 | MOSI hold time after SPCK rises | | 0 | | ns |
| USPI9 | SPCK rising to MISO delay | | | 30 | ns |
| USPI10 | MOSI setup time before SPCK falls | | $t_{SAMPLE}$[2] + $t_{CLK\_USART}$ | | ns |
| USPI11 | MOSI hold time after SPCK falls | | 0 | | ns |
| USPI12 | NSS setup time before SPCK rises | | 35 | | ns |
| USPI13 | NSS hold time after SPCK falls | | 0 | | ns |
| USPI14 | NSS setup time before SPCK falls | | 35 | | ns |
| USPI15 | NSS hold time after SPCK rises | | 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.

2. Where: $t_{SAMPLE} = t_{SPCK} - \left( \left\lfloor \frac{t_{SPCK}}{2 \times t_{CLKUSART}} \right\rfloor + \frac{1}{2} \right) \times t_{CLKUSART}$

**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)

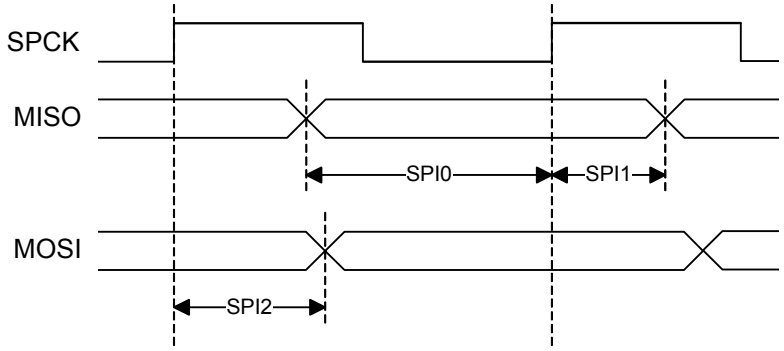
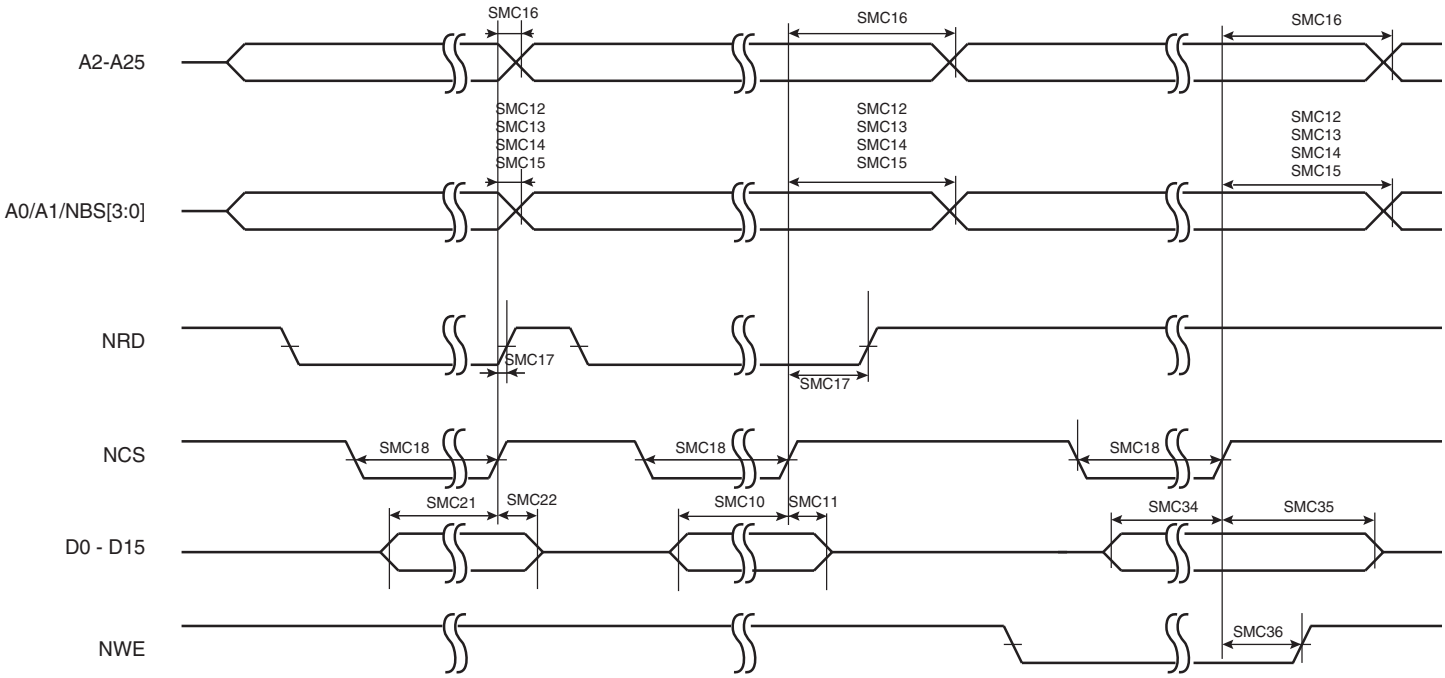


**Table 7-56.** SMC Write Signals with No Hold Settings (NWE Controlled only)[1]

| Symbol | Parameter | Conditions | Min | Units |
|---|---|---|---|---|
| $SMC_{37}$ | NWE rising to A2-A25 valid | $V_{VDD}$ = 3.0V, drive strength of the pads set to the lowest, external capacitor = 40pF | 9.1 | ns |
| $SMC_{38}$ | NWE rising to NBS0/A0 valid | | 7.9 | |
| $SMC_{40}$ | NWE rising to A1/NBS2 change | | 9.1 | |
| $SMC_{42}$ | NWE rising to NCS rising | | 8.7 | |
| $SMC_{43}$ | Data Out valid before NWE rising | | (nwe pulse length - 1) * $t_{CPSMC}$ - 1.5 | |
| $SMC_{44}$ | Data Out valid after NWE rising | | 8.7 | |
| $SMC_{45}$ | NWE pulse width | | nwe pulse length * $t_{CPSMC}$ - 0.1 | |

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.

**Figure 7-17.** SMC Signals for NCS Controlled Accesses



### 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 | 2.6 | 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 | | 0.6 | | ns |
| $MAC_8$ | TX_ER toggling from TX_CLK rising | | 17.3 | 19.6 | ns |
| $MAC_9$ | TX_EN toggling from TX_CLK rising | | 15.5 | 16.2 | ns |
| $MAC_{10}$ | TXD toggling from TX_CLK rising | | 14.9 | 19.2 | ns |
| $MAC_{11}$ | Setup for RXD from RX_CLK | | 1.3 | | ns |
| $MAC_{12}$ | Hold for RXD from RX_CLK | | 2 | | ns |
| $MAC_{13}$ | Setup for RX_ER from RX_CLK | | 3.6 | | 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 | | 1.4 | | ns |

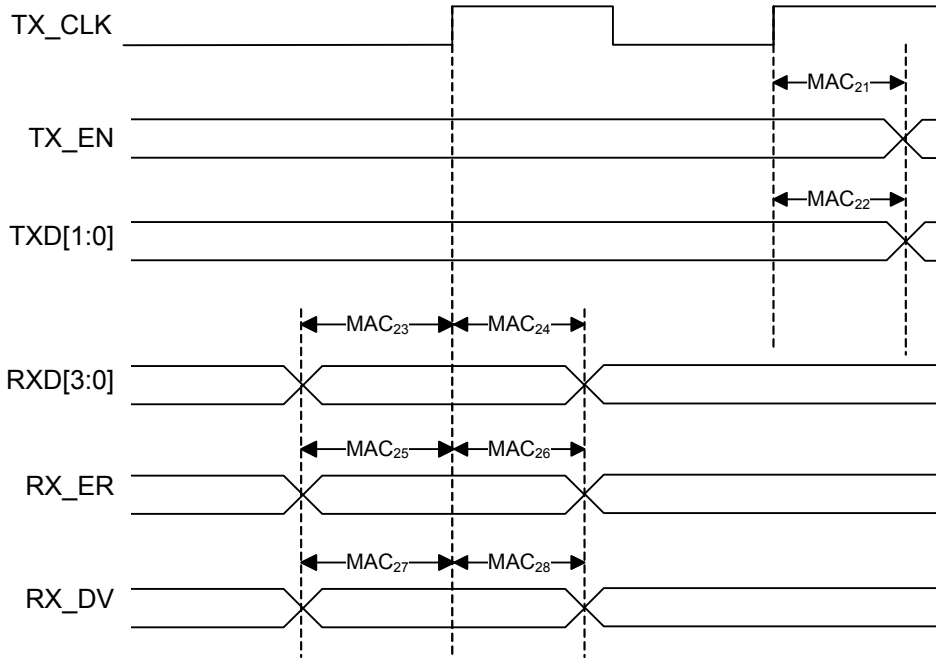
Note: 1. These values are based on simulation and characterization of other AVR microcontrollers manufactured in the same process technology. These values are not covered by test limits in production.

**Table 7-61.** 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 | 12.5 | 13.4 | ns |
| $MAC_{22}$ | TXD toggling from TX_CLK rising | | 12.5 | 13.4 | ns |
| $MAC_{23}$ | Setup for RXD from TX_CLK | | 4.7 | | ns |
| $MAC_{24}$ | Hold for RXD from TX_CLK | | 0 | | ns |
| $MAC_{25}$ | Setup for RX_ER from TX_CLK | | 3.6 | | ns |
| $MAC_{26}$ | Hold for RX_ER from TX_CLK | | 0 | | ns |
| $MAC_{27}$ | Setup for RX_DV from TX_CLK | | 4.6 | | 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. Mechanical Characteristics

## 8.1 Thermal Considerations

### 8.1.1 Thermal Data

Table 8-1 summarizes the thermal resistance data depending on the package.

**Table 8-1.** Thermal Resistance Data

| Symbol | Parameter | Condition | Package | Typ | Unit |
|---|---|---|---|---|---|
| $\theta_{JA}$ | Junction-to-ambient thermal resistance | No air flow | QFN64 | 20.0 | °C/W |
| $\theta_{JC}$ | Junction-to-case thermal resistance | | QFN64 | 0.8 | |
| $\theta_{JA}$ | Junction-to-ambient thermal resistance | No air flow | TQFP64 | 40.5 | °C/W |
| $\theta_{JC}$ | Junction-to-case thermal resistance | | TQFP64 | 8.7 | |
| $\theta_{JA}$ | Junction-to-ambient thermal resistance | No air flow | TQFP100 | 39.3 | °C/W |
| $\theta_{JC}$ | Junction-to-case thermal resistance | | TQFP100 | 8.5 | |
| $\theta_{JA}$ | Junction-to-ambient thermal resistance | No air flow | LQFP144 | 38.1 | °C/W |
| $\theta_{JC}$ | Junction-to-case thermal resistance | | LQFP144 | 8.4 | |

### 8.1.2 Junction Temperature

The average chip-junction temperature, $T_J$, in °C can be obtained from the following:

1. $T_J = T_A + (P_D \times \theta_{JA})$
2. $T_J = T_A + (P_D \times (\theta_{HEATSINK} + \theta_{JC}))$

where:

- $\theta_{JA}$ = package thermal resistance, Junction-to-ambient (°C/W), provided in Table 8-1 on page 89.
- $\theta_{JC}$ = package thermal resistance, Junction-to-case thermal resistance (°C/W), provided in Table 8-1 on page 89.
- $\theta_{HEAT\ SINK}$ = cooling device thermal resistance (°C/W), provided in the device datasheet.
- $P_D$ = device power consumption (W) estimated from data provided in the section "Power Consumption" on page 50.
- $T_A$ = ambient temperature (°C).

From the first equation, the user can derive the estimated lifetime of the chip and decide if a cooling device is necessary or not. If a cooling device is to be fitted on the chip, the second equation should be used to compute the resulting average chip-junction temperature $T_J$ in °C.

# 10. Errata

## 10.1 rev E

### 10.1.1 ADCIFA

1 **ADCREFP/ADCREFN can not be selected as an external ADC reference by setting the ADCIFA.CFG.EXREF bit to one**
**Fix/Workaround**
A voltage reference can be applied on ADCREFP/ADCREFN pins if the ADCIFA.CFG.EXREF bit is set to zero, the ADCIFA.CFG.RS bit is set to zero and the voltage reference applied on ADCREFP/ADCREFN pins is higher than the internal 1V reference.

### 10.1.2 AST

1 **AST wake signal is released one AST clock cycle after the BUSY bit is cleared**
After writing to the Status Clear Register (SCR) the wake signal is released one AST clock cycle after the BUSY bit in the Status Register (SR.BUSY) is cleared. If entering sleep mode directly after the BUSY bit is cleared the part will wake up immediately.
**Fix/Workaround**
Read the Wake Enable Register (WER) and write this value back to the same register. Wait for BUSY to clear before entering sleep mode.

### 10.1.3 aWire

1 **aWire MEMORY_SPEED_REQUEST command does not return correct CV**
The aWire MEMORY_SPEED_REQUEST command does not return a CV corresponding to the formula in the aWire Debug Interface chapter.
**Fix/Workaround**
Issue a dummy read to address 0x100000000 before issuing the MEMORY_SPEED_REQUEST command and use this formula instead:

$$f_{sab} = \frac{7 f_{aw}}{CV - 3}$$

### 10.1.4 Power Manager

1 **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 rev D

### 10.2.1 ADCIFA

1. **ADCREFP/ADCREFN can not be selected as an external ADC reference by setting the ADCIFA.CFG.EXREF bit to one**
   **Fix/Workaround**
   A voltage reference can be applied on ADCREFP/ADCREFN pins if the ADCIFA.CFG.EXREF bit is set to zero, the ADCIFA.CFG.RS bit is set to zero and the voltage reference applied on ADCREFP/ADCREFN pins is higher than the internal 1V reference.

### 10.2.2 AST

1. **AST wake signal is released one AST clock cycle after the BUSY bit is cleared**
   After writing to the Status Clear Register (SCR) the wake signal is released one AST clock cycle after the BUSY bit in the Status Register (SR.BUSY) is cleared. If entering sleep mode directly after the BUSY bit is cleared the part will wake up immediately.
   **Fix/Workaround**
   Read the Wake Enable Register (WER) and write this value back to the same register. Wait for BUSY to clear before entering sleep mode.

### 10.2.3 aWire

1. **aWire MEMORY_SPEED_REQUEST command does not return correct CV**
   The aWire MEMORY_SPEED_REQUEST command does not return a CV corresponding to the formula in the aWire Debug Interface chapter.
   **Fix/Workaround**
   Issue a dummy read to address 0x100000000 before issuing the MEMORY_SPEED_REQUEST command and use this formula instead:

$$f_{sab} = \frac{7 f_{aw}}{CV - 3}$$

### 10.2.4 GPIO

1. **Clearing Interrupt flags can mask other interrupts**
   When clearing interrupt flags in a GPIO port, interrupts on other pins of that port, happening in the same clock cycle will not be registered.
   **Fix/Workaround**
   Read the PVR register of the port before and after clearing the interrupt to see if any pin change has happened while clearing the interrupt. If any change occurred in the PVR between the reads, they must be treated as an interrupt.

### 10.2.5 Power Manager

1. **Clock Failure Detector (CFD) can be issued while turning off the CFD**
   While turning off the CFD, the CFD bit in the Status Register (SR) can be set. This will change the main clock source to RCSYS.
   **Fix/Workaround**
   Solution 1: Enable CFD interrupt. If CFD interrupt is issues after turning off the CFD, switch back to original main clock source.
   Solution 2: Only turn off the CFD while running the main clock on RCSYS.