**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Active |
| Core Processor | AVR |
| Core Size | 32-Bit Single-Core |
| Speed | 66MHz |
| Connectivity | CANbus, Ethernet, I²C, IrDA, LINbus, SPI, UART/USART, USB |
| Peripherals | Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT |
| Number of I/O | 45 |
| Program Memory Size | 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 11x12b; D/A 2x12b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 125°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 64-TQFP |
| Supplier Device Package | 64-TQFP (10x10) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/at32uc3c2256c-a2zt |

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

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.

## 1.1 Disclaimer

Typical values contained in this data sheet are based on simulations and characterization of other 32-bit AVR Microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

## 1.2 Automotive Quality Grade

The AT32UC3C have been developed and manufactured according to the most stringent requirements of the international standard ISO-TS 16949. This data sheet contains limit values extracted from the results of extensive characterization (Temperature and Voltage). The quality and reliability of the AT32UC3C have been verified during regular product qualification as per AEC-Q100 grade 1.

As indicated in the ordering information paragraph, the product is available in only one temperature grade, Table 1-1.

**Table 1-1.** Temperature Grade Identification for Automotive Products

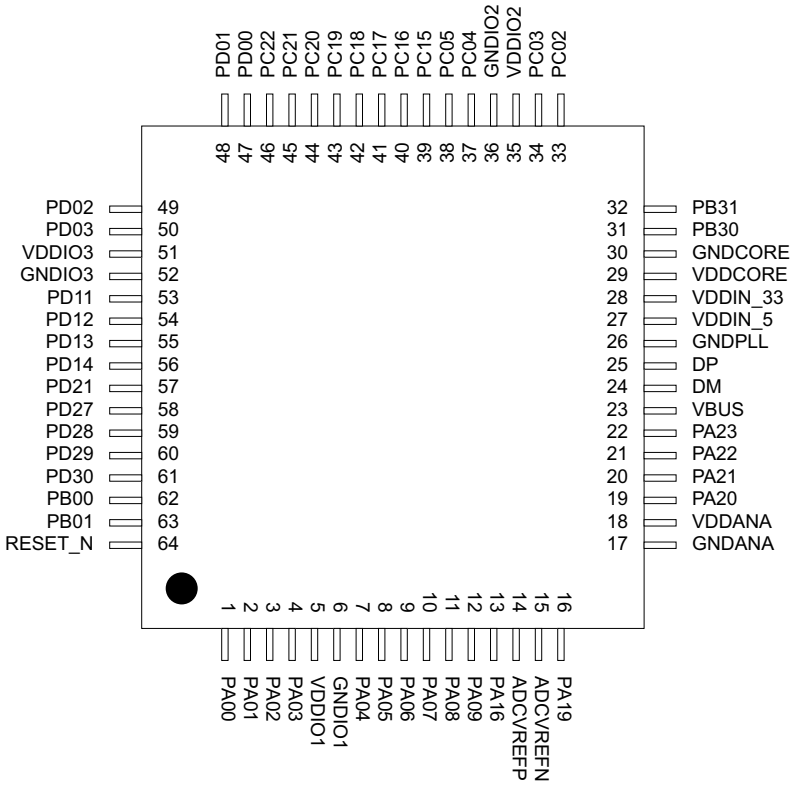
| Temperature(°C) | Temperature Identifier | Comments |
|---|---|---|
| -40;+125 | Z | Full Automotive Temperature Range |

# 3. Package and Pinout

## 3.1 Package

The device pins are multiplexed with peripheral functions as described in Table 3-1 on page 11.

**Figure 3-1.** QFN64/TQFP64 Pinout




Note: on QFN packages, the exposed pad is unconnected.

**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 |
| | 72 | 97 | PC24 | 88 | VDDIO3 | x1/x2 | QDEC1 - QEPA | CANIF - TXLINE[1] | EBI - DATA[5] | PEVC - PAD_EVT [4] | | |
| | | 98 | PC25 | 89 | VDDIO3 | x1/x2 | | TC1 - CLK2 | EBI - DATA[6] | SCIF - GCLK[0] | USART4 - TXD | |
| | | 99 | PC26 | 90 | VDDIO3 | x1/x2 | QDEC1 - QEPI | TC1 - B2 | EBI - DATA[7] | SCIF - GCLK[1] | USART4 - RXD | |
| | | 100 | PC27 | 91 | VDDIO3 | x1/x2 | | TC1 - A2 | EBI - DATA[8] | EIC - EXTINT[0] | USART4 - CTS | |
| | | 101 | PC28 | 92 | VDDIO3 | x1/x2 | SPI1 - NPCS[3] | TC1 - CLK1 | EBI - DATA[9] | | USART4 - RTS | |
| | | 102 | PC29 | 93 | VDDIO3 | x1/x2 | SPI0 - NPCS[1] | TC1 - B1 | EBI - DATA[10] | | | |
| | | 105 | PC30 | 94 | VDDIO3 | x1/x2 | SPI0 - NPCS[2] | TC1 - A1 | EBI - DATA[11] | | | |
| | 73 | 106 | PC31 | 95 | VDDIO3 | x1/x2 | SPI0 - NPCS[3] | TC1 - B0 | EBI - DATA[12] | PEVC - PAD_EVT [5] | USART4 - CLK | |
| 47 | 74 | 107 | PD00 | 96 | VDDIO3 | x1/x2 | SPI0 - MOSI | TC1 - CLK0 | EBI - DATA[13] | QDEC0 - QEPI | USART0 - TXD | |
| 48 | 75 | 108 | PD01 | 97 | VDDIO3 | x1/x2 | SPI0 - MISO | TC1 - A0 | EBI - DATA[14] | TC0 - CLK1 | USART0 - RXD | |
| 49 | 76 | 109 | PD02 | 98 | VDDIO3 | x2/x4 | SPI0 - SCK | TC0 - CLK2 | EBI - DATA[15] | QDEC0 - QEPA | | |
| 50 | 77 | 110 | PD03 | 99 | VDDIO3 | x1/x2 | SPI0 - NPCS[0] | TC0 - B2 | EBI - ADDR[0] | QDEC0 - QEPB | | |
| | | 111 | PD04 | 100 | VDDIO3 | x1/x2 | SPI0 - MOSI | | EBI - ADDR[1] | | | |
| | | 112 | PD05 | 101 | VDDIO3 | x1/x2 | SPI0 - MISO | | EBI - ADDR[2] | | | |
| | | 113 | PD06 | 102 | VDDIO3 | x2/x4 | SPI0 - SCK | | EBI - ADDR[3] | | | |
| | 78 | 114 | PD07 | 103 | VDDIO3 | x1/x2 | USART1 - DTR | EIC - EXTINT[5] | EBI - ADDR[4] | QDEC0 - QEPI | USART4 - TXD | |
| | 79 | 115 | PD08 | 104 | VDDIO3 | x1/x2 | USART1 - DSR | EIC - EXTINT[6] | EBI - ADDR[5] | TC1 - CLK2 | USART4 - RXD | |
| | 80 | 116 | PD09 | 105 | VDDIO3 | x1/x2 | USART1 - DCD | CANIF - RXLINE[0] | EBI - ADDR[6] | QDEC0 - QEPA | USART4 - CTS | |
| | 81 | 117 | PD10 | 106 | VDDIO3 | x1/x2 | USART1 - RI | CANIF - TXLINE[0] | EBI - ADDR[7] | QDEC0 - QEPB | USART4 - RTS | |
| 53 | 84 | 120 | PD11 | 107 | VDDIO3 | x1/x2 | USART1 - TXD | USBC - ID | EBI - ADDR[8] | PEVC - PAD_EVT [6] | MACB - TXD[0] | |
| 54 | 85 | 121 | PD12 | 108 | VDDIO3 | x1/x2 | USART1 - RXD | USBC - VBOF | EBI - ADDR[9] | PEVC - PAD_EVT [7] | MACB - TXD[1] | |
| 55 | 86 | 122 | PD13 | 109 | VDDIO3 | x2/x4 | USART1 - CTS | USART1 - CLK | EBI - SDCK | PEVC - PAD_EVT [8] | MACB - RXD[0] | |
| 56 | 87 | 123 | PD14 | 110 | VDDIO3 | x1/x2 | USART1 - RTS | EIC - EXTINT[7] | EBI - ADDR[10] | PEVC - PAD_EVT [9] | MACB - RXD[1] | |

**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 | | |

#### 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

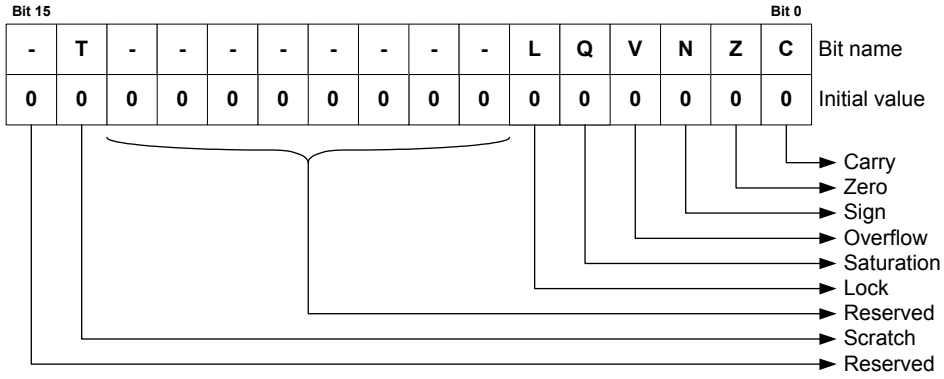
#### 4.3.2.7 *CPU and Architecture Revision*

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.

**Figure 4-5.** The Status Register Low Halfword




### 4.4.3 Processor States

#### *4.4.3.1 Normal RISC State*

The AVR32 processor supports several different execution contexts as shown in Table 4-2.

**Table 4-2.** Overview of Execution Modes, their Priorities and Privilege Levels.

| Priority | Mode | Security | Description |
|---|---|---|---|
| 1 | Non Maskable Interrupt | Privileged | Non Maskable high priority interrupt mode |
| 2 | Exception | Privileged | Execute exceptions |
| 3 | Interrupt 3 | Privileged | General purpose interrupt mode |
| 4 | Interrupt 2 | Privileged | General purpose interrupt mode |
| 5 | Interrupt 1 | Privileged | General purpose interrupt mode |
| 6 | Interrupt 0 | Privileged | General purpose interrupt mode |
| N/A | Supervisor | Privileged | Runs supervisor calls |
| N/A | Application | Unprivileged | Normal program execution mode |

Mode changes can be made under software control, or can be caused by external interrupts or exception processing. A mode can be interrupted by a higher priority mode, but never by one with lower priority. Nested exceptions can be supported with a minimal software overhead.

When running an operating system on the AVR32, user processes will typically execute in the application mode. The programs executed in this mode are restricted from executing certain instructions. Furthermore, most system registers together with the upper halfword of the status register cannot be accessed. Protected memory areas are also not available. All other operating modes are privileged and are collectively called System Modes. They have full access to all privileged and unprivileged resources. After a reset, the processor will be in supervisor mode.

#### *4.4.3.2 Debug State*

The AVR32 can be set in a debug state, which allows implementation of software monitor routines that can read out and alter system information for use during application development. This implies that all system and application registers, including the status registers and program counters, are accessible in debug state. The privileged instructions are also available.

All interrupt levels are by default disabled when debug state is entered, but they can individually be switched on by the monitor routine by clearing the respective mask bit in the status register.

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 |

## 5.2 Physical Memory Map

The system bus is implemented as a bus matrix. All system bus addresses are fixed, and they are never remapped in any way, not even in boot. Note that AVR32UC CPU uses unsegmented translation, as described in the AVR32 Architecture Manual. The 32-bit physical address space is mapped as follows:

**Table 5-1.** AT32UC3C Physical Memory Map

| Device | Start Address | AT32UC3 Derivatives | | | |
|---|---|---|---|---|---|
| | | C0512C | C1512C<br>C2512C | C1256C<br>C2256C | C2128C |
| Embedded SRAM | 0x0000_0000 | 64 KB | 64 KB | 64 KB | 32 KB |
| Embedded Flash | 0x8000_0000 | 512 KB | 512 KB | 256 KB | 128 KB |
| SAU | 0x9000_0000 | 1 KB | 1 KB | 1 KB | 1 KB |
| HSB SRAM | 0xA000_0000 | 4 KB | 4 KB | 4 KB | 4 KB |
| EBI SRAM CS0 | 0xC000_0000 | 16 MB | - | - | - |
| EBI SRAM CS2 | 0xC800_0000 | 16 MB | - | - | - |
| EBI SRAM CS3 | 0xCC00_0000 | 16 MB | - | - | - |
| EBI SRAM CS1<br>/SDRAM CS0 | 0xD000_0000 | 128 MB | - | - | - |
| HSB-PB Bridge C | 0xFFFD_0000 | 64 KB | 64 KB | 64 KB | 64 KB |
| HSB-PB Bridge B | 0xFFFE_0000 | 64 KB | 64 KB | 64 KB | 64 KB |
| HSB-PB Bridge A | 0xFFFF_0000 | 64 KB | 64 KB | 64 KB | 64 KB |

**Table 5-2.** Flash Memory Parameters

| Part Number | Flash Size<br>(FLASH_PW) | Number of pages<br>(FLASH_P) | Page size<br>(FLASH_W) |
|---|---|---|---|
| AT32UC3C0512C<br>AT32UC3C1512C<br>AT32UC3C2512C | 512 Kbytes | 1024 | 128 words |
| AT32UC3C1256C<br>AT32UC3C2256C | 256 Kbytes | 512 | 128 words |
| AT32UC3C2128C | 128 Kbytes | 256 | 128 words |

## 5.3 Peripheral Address Map

**Table 5-3.** Peripheral Address Mapping

| Address | Peripheral Name | |
|---|---|---|
| **0xFFFD0000** | PDCA | Peripheral DMA Controller - PDCA |

**Table 5-4.** Local bus mapped GPIO registers

| Port | Register | Mode | Local Bus Address | Access |
|---|---|---|---|---|
| B | Output Driver Enable Register (ODER) | WRITE | 0x40000140 | Write-only |
| | | SET | 0x40000144 | Write-only |
| | | CLEAR | 0x40000148 | Write-only |
| | | TOGGLE | 0x4000014C | Write-only |
| | Output Value Register (OVR) | WRITE | 0x40000150 | Write-only |
| | | SET | 0x40000154 | Write-only |
| | | CLEAR | 0x40000158 | Write-only |
| | | TOGGLE | 0x4000015C | Write-only |
| | Pin Value Register (PVR) | - | 0x40000160 | Read-only |
| C | Output Driver Enable Register (ODER) | WRITE | 0x40000240 | Write-only |
| | | SET | 0x40000244 | Write-only |
| | | CLEAR | 0x40000248 | Write-only |
| | | TOGGLE | 0x4000024C | Write-only |
| | Output Value Register (OVR) | WRITE | 0x40000250 | Write-only |
| | | SET | 0x40000254 | Write-only |
| | | CLEAR | 0x40000258 | Write-only |
| | | TOGGLE | 0x4000025C | Write-only |
| | Pin Value Register (PVR) | - | 0x40000260 | Read-only |
| D | Output Driver Enable Register (ODER) | WRITE | 0x40000340 | Write-only |
| | | SET | 0x40000344 | Write-only |
| | | CLEAR | 0x40000348 | Write-only |
| | | TOGGLE | 0x4000034C | Write-only |
| | Output Value Register (OVR) | WRITE | 0x40000350 | Write-only |
| | | SET | 0x40000354 | Write-only |
| | | CLEAR | 0x40000358 | Write-only |
| | | TOGGLE | 0x4000035C | Write-only |
| | Pin Value Register (PVR) | - | 0x40000360 | Read-only |

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

| Symbol | Parameter | Condition | | Min | Typ | Max | Units |
|---|---|---|---|---|---|---|---|
| $t_{RISE}$ | Rise time[3] | $V_{VDD}$ = 3.0V | load = 10pF, pin drive x1[2] | | | 8.4 | ns |
| | | | load = 10pF, pin drive x2[2] | | | 3.8 | |
| | | | load = 10pF, pin drive x4[2] | | | 2.1 | |
| | | | load = 30pF, pin drive x1[2] | | | 17.5 | |
| | | | load = 30pF, pin drive x2[2] | | | 8.2 | |
| | | | load = 30pF, pin drive x4[2] | | | 4.2 | |
| | | $V_{VDD}$ = 4.5V | load = 10pF, pin drive x1[2] | | | 5.9 | |
| | | | load = 10pF, pin drive x2[2] | | | 2.6 | |
| | | | load = 10pF, pin drive x4[2] | | | 1.5 | |
| | | | load = 30pF, pin drive x1[2] | | | 12.2 | |
| | | | load = 30pF, pin drive x2[2] | | | 5.7 | |
| | | | load = 30pF, pin drive x4[2] | | | 3.0 | |
| $t_{FALL}$ | Fall time[3] | $V_{VDD}$ = 3.0V | load = 10pF, pin drive x1[2] | | | 8.5 | ns |
| | | | load = 10pF, pin drive x2[2] | | | 3.9 | |
| | | | load = 10pF, pin drive x4[2] | | | 2.1 | |
| | | | load = 30pF, pin drive x1[2] | | | 17.6 | |
| | | | load = 30pF, pin drive x2[2] | | | 8.1 | |
| | | | load = 30pF, pin drive x4[2] | | | 4.3 | |
| | | $V_{VDD}$ = 4.5V | load = 10pF, pin drive x1[2] | | | 5.9 | |
| | | | load = 10pF, pin drive x2[2] | | | 2.7 | |
| | | | load = 10pF, pin drive x4[2] | | | 1.5 | |
| | | | load = 30pF, pin drive x1[2] | | | 12.2 | |
| | | | load = 30pF, pin drive x2[2] | | | 5.7 | |
| | | | load = 30pF, pin drive x4[2] | | | 3.0 | |
| $I_{LEAK}$ | Input leakage current | Pull-up resistors disabled | | | | 2.0 | µA |
| $C_{IN}$ | Input capacitance | PA00-PA29, PB00-PB31, PC00-PC01, PC08-PC31, PD00-PD30 | | | 7.5 | | pF |
| | | PC02, PC03, PC04, PC05, PC06, PC07 | | | 2 | | |

Note:
1. $V_{VDD}$ corresponds to either $V_{VDDIO1}$, $V_{VDDIO2}$, $V_{VDDIO3}$, or $V_{VDDANA}$, depending on the supply for the pin. Refer to Section 3-1 on page 11 for details.
2. drive x1 capability pins are: PB00, PB01, PB02, PB03, PB30, PB31, PC02, PC03, PC04, PC05, PC06, PC07 - drive x2 /x4 capability pins are: PB06, PB21, PB26, PD02, PD06, PD13 - drive x1/x2 capability pins are the remaining PA, PB, PC, PD pins. The drive strength is programmable through ODCR0, ODCR0S, ODCR0C, ODCR0T registers of GPIO.
3. 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.

### 7.8.4 3.3V Brown Out Detector (BOD33) Characteristics

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

**Table 7-23.** BOD33.LEVEL Values

| BOD33.LEVEL Value | Parameter | Min | Max | Units |
|---|---|---|---|---|
| 17 | | 2.27 | 2.52 | V |
| 22 | | 2.36 | 2.61 | |
| 27 | | 2.45 | 2.71 | |
| 31 | threshold at power-up sequence | 2.52 | 2.79 | |
| 33 | | 2.56 | 2.83 | |
| 39 | | 2.67 | 2.95 | |
| 44 | | 2.76 | 3.05 | |
| 49 | | 2.85 | 3.15 | |
| 53 | | 2.91 | 3.23 | |
| 60 | | 3.05 | 3.37 | |

### 7.8.5 5V Brown Out Detector (BOD50) Characteristics

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

**Table 7-25.** BOD50.LEVEL Values

| BOD50.LEVEL Value | Parameter | Min | Max | Units |
|---|---|---|---|---|
| 16 | | 3.28 | 3.61 | V |
| 25 | | 3.52 | 3.87 | |
| 35 | | 3.78 | 4.17 | |
| 44 | | 4.02 | 4.43 | |
| 53 | | 4.25 | 4.69 | |
| 61 | | 4.47 | 4.92 | |

### 7.8.6 Analog to Digital Converter (ADC) and sample and hold (S/H) Characteristics

**Table 7-27.** ADC and S/H characteristics

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|---|
| $f_{ADC}$ | ADC clock frequency | 12-bit resolution mode, $V_{VDDANA}$ = 3V | | | 1.2 | MHz |
| | | 10-bit resolution mode, $V_{VDDANA}$ = 3V | | | 1.6 | |
| | | 8-bit resolution mode, $V_{VDDANA}$ = 3V | | | 2.2 | |
| | | 12-bit resolution mode, $V_{VDDANA}$ = 4.5V | | | 1.5 | |
| | | 10-bit resolution mode, $V_{VDDANA}$ = 4.5V | | | 2 | |
| | | 8-bit resolution mode, $V_{VDDANA}$ = 4.5V | | | 2.4 | |
| $t_{STARTUP}$ | Startup time | ADC cold start-up | | | 1 | ms |
| | | ADC hot start-up | | | 24 | ADC clock cycles |
| $t_{CONV}$ | Conversion time (latency) | (ADCIFA.SEQCFGn.SRES)/2 + 2, ADCIFA.CFG.SHD = 1 | 6 | | 8 | ADC clock cycles |
| | | (ADCIFA.SEQCFGn.SRES)/2 + 3, ADCIFA.CFG.SHD = 0 | 7 | | 9 | |
| | Throughput rate | 12-bit resolution, ADC clock = 1.2 MHz, $V_{VDDANA}$ = 3V | | | 1.2 | MSPS |
| | | 10-bit resolution, ADC clock = 1.6 MHz, $V_{VDDANA}$ = 3V | | | 1.6 | |
| | | 12-bit resolution, ADC clock = 1.5 MHz, $V_{VDDANA}$ = 4.5V | | | 1.5 | |
| | | 10-bit resolution, ADC clock = 2 MHz, $V_{VDDANA}$ = 4.5V | | | 2 | |

**Table 7-28.** ADC Reference Voltage

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|---|
| $V_{ADCREF0}$ | ADCREF0 input voltage range | 5V Range | 1 | | 3.5 | V |
| | | 3V Range | 1 | | $V_{VDDANA}$-0.7 | |
| $V_{ADCREF1}$ | ADCREF1 input voltage range | 5V Range | 1 | | 3.5 | V |
| | | 3V Range | 1 | | $V_{VDDANA}$-0.7 | |
| $V_{ADCREFP}$ | ADCREFP input voltage | 5V Range - Voltage reference applied on ADCREFP | 1 | | 3.5 | V |
| | | 3V Range - Voltage reference applied on ADCREFP | 1 | | $V_{VDDANA}$-0.7 | |
| $V_{ADCREFN}$ | ADCREFN input voltage | Voltage reference applied on ADCREFN | | GNDANA | | V |
| | Internal 1V reference | | | 1.0 | | V |
| | Internal 0.6*VDDANA reference | | | 0.6*$V_{VDDANA}$ | | V |

**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

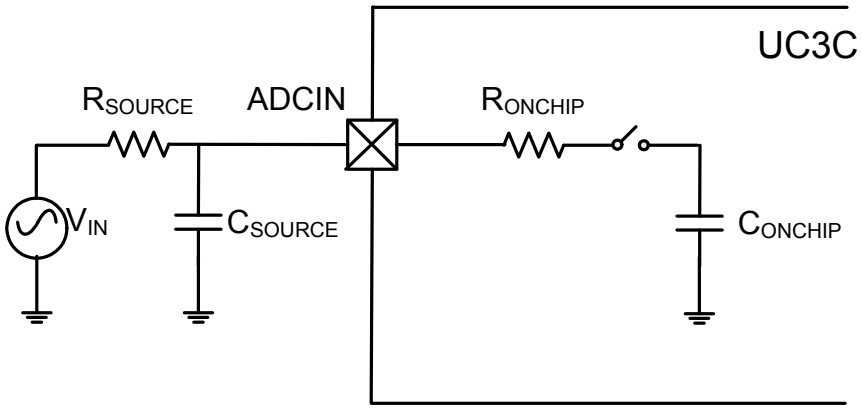



**Table 7-31.** ADC Transfer Characteristics 12-bit Resolution Mode[1]

| 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 |

### 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.

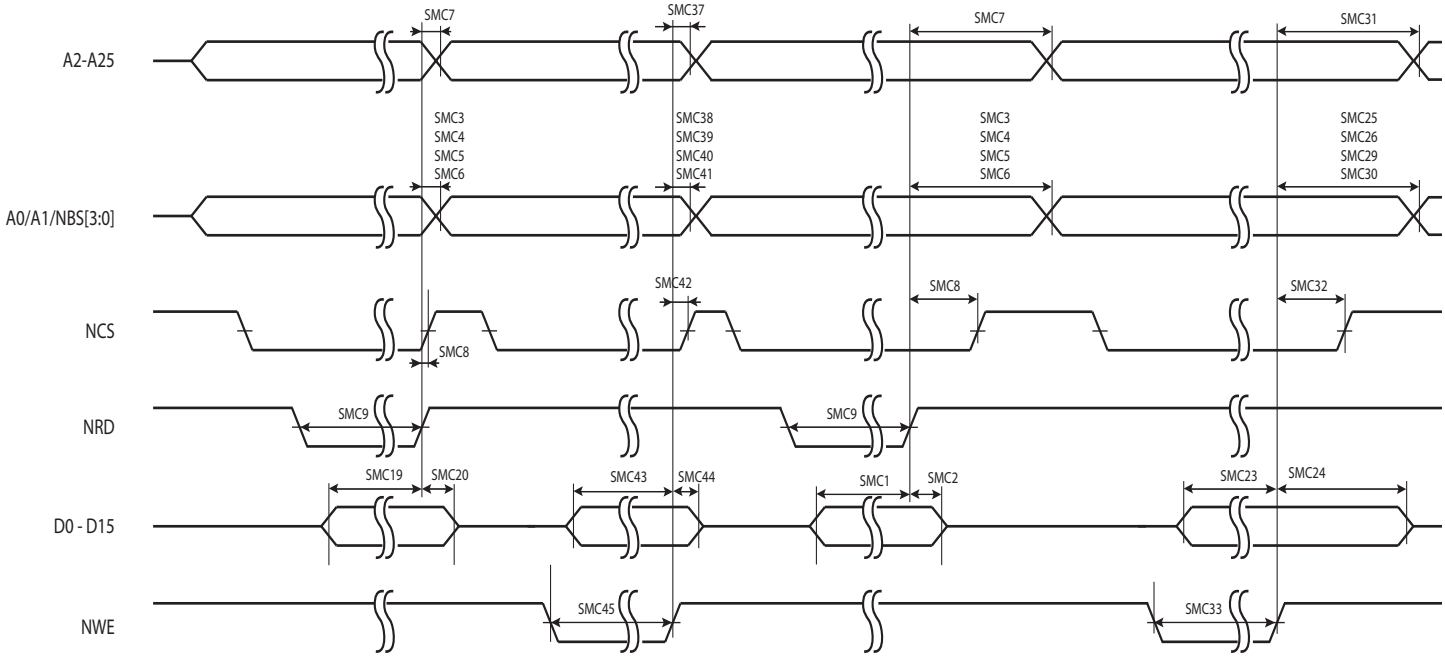
**Table 7-53.** SMC Read Signals with Hold Settings[1]

| 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 | 34.4 | 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 * $t_{CPSMC}$ - 5.9 | |
| $SMC_8$ | 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 * $t_{CPSMC}$ - 0.9 | |
| **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 | 36.1 | ns |
| $SMC_{11}$ | Data hold after NCS high | | 0 | |
| $SMC_{12}$ | NCS high to NBS0/A0 change[2] | | ncs rd hold length * $t_{CPSMC}$ - 3.2 | |
| $SMC_{13}$ | NCS high to NBS0/A0 change[2] | | ncs rd hold length * $t_{CPSMC}$ - 2.2 | |
| $SMC_{14}$ | NCS high to NBS2/A1 change[2] | | ncs rd hold length * $t_{CPSMC}$ - 1.2 | |
| $SMC_{16}$ | NCS high to A2 - A25 change[2] | | ncs rd hold length * $t_{CPSMC}$ - 7.6 | |
| $SMC_{17}$ | NCS high to NRD inactive[2] | | (ncs rd hold length - nrd hold length) * $t_{CPSMC}$ - 2.4 | |
| $SMC_{18}$ | NCS pulse width | | ncs rd pulse length * $t_{CPSMC}$ - 3.3 | |

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.

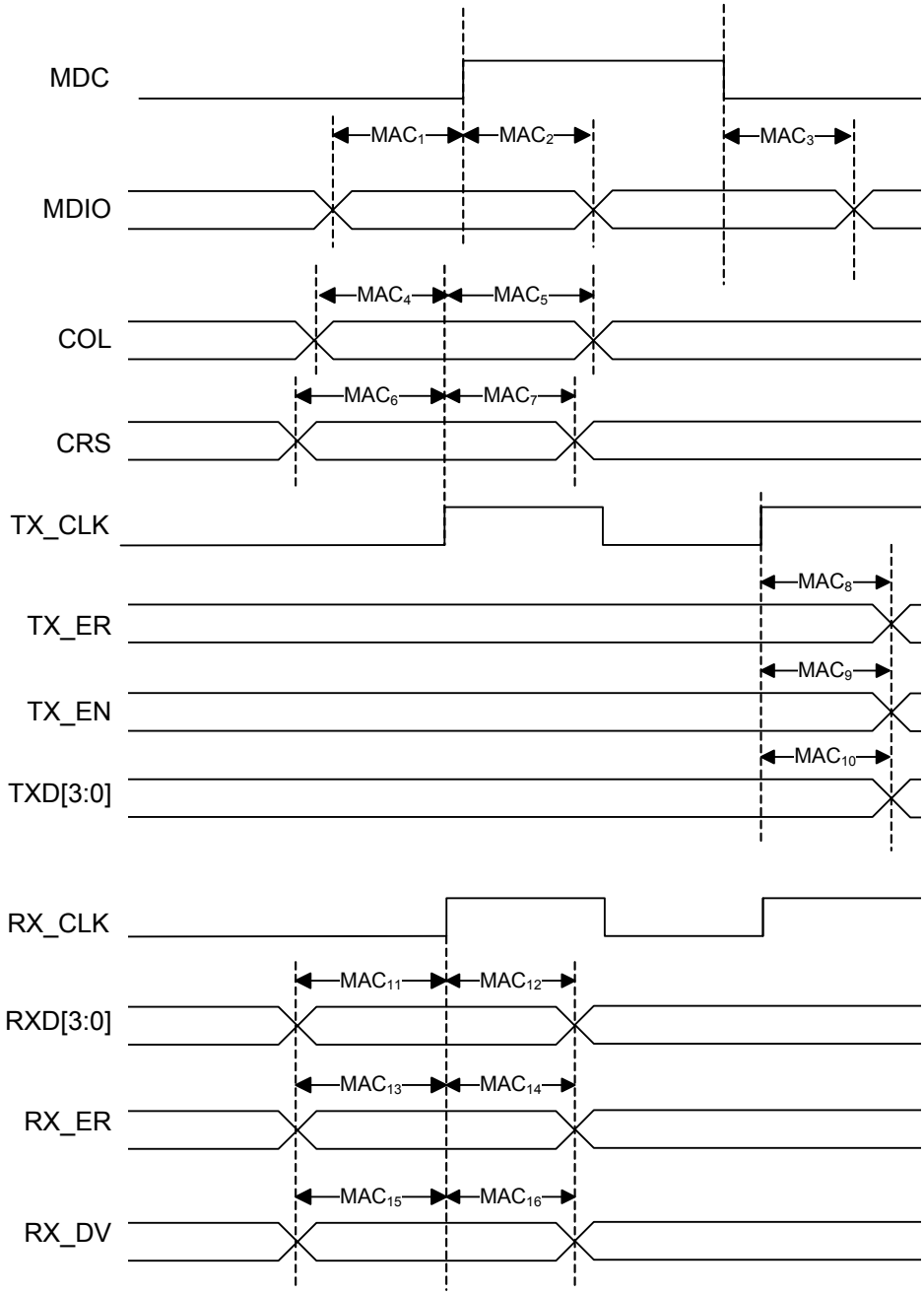
### 7.9.8 SDRAM Signals

**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 7-20.** Ethernet MAC MII 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.

# 9. Ordering Information

**Table 9-1.** Ordering Information

| Device | Ordering Code | Carrier Type | Package | Temperature Operating Range |
|---|---|---|---|---|
| **AT32UC3C0512C** | AT32UC3C0512C-ALZT | Tray | LQFP 144 | Automotive (-40°C to 125°C) |
| | AT32UC3C0512C-ALZR | Tape & Reel | | |
| **AT32UC3C1512C** | AT32UC3C1512C-AZT | Tray | TQFP 100 | |
| | AT32UC3C1512C-AZR | Tape & Reel | | |
| **AT32UC3C1256C** | AT32UC3C1256C-AZT | Tray | TQFP 100 | |
| | AT32UC3C1256C-AZR | Tape & Reel | | |
| **AT32UC3C2512C** | AT32UC3C2512C-A2ZT | Tray | TQFP 64 | |
| | AT32UC3C2512C-A2ZR | Tape & Reel | | |
| **AT32UC3C2512C** | AT32UC3C2512C-Z2ZT | Tray | QFN 64 | |
| | AT32UC3C2512C-Z2ZR | Tape & Reel | | |
| **AT32UC3C2256C** | AT32UC3C2256C-A2ZT | Tray | TQFP 64 | |
| | AT32UC3C2256C-A2ZR | Tape & Reel | | |
| **AT32UC3C2256C** | AT32UC3C2256C-Z2ZT | Tray | QFN 64 | |
| | AT32UC3C2256C-Z2ZR | Tape & Reel | | |
| **AT32UC3C2128C** | AT32UC3C2128C-A2ZT | Tray | TQFP 64 | |
| | AT32UC3C2128C-A2ZR | Tape & Reel | | |
| **AT32UC3C2128C** | AT32UC3C2128C-Z2ZT | Tray | QFN 64 | |
| | AT32UC3C2128C-Z2ZR | Tape & Reel | | |

2 **Disabling SPI has no effect on the SR.TDRE bit**
Disabling SPI has no effect on the SR.TDRE bit whereas the write data command is filtered when SPI is disabled. Writing to TDR when SPI is disabled will not clear SR.TDRE. If SPI is disabled during a PDCA transfer, the PDCA will continue to write data to TDR until its buffer is empty, and this data will be lost.
**Fix/Workaround**
Disable the PDCA, add two NOPs, and disable the SPI. To continue the transfer, enable the SPI and PDCA.

3 **SPI disable does not work in SLAVE mode**
SPI disable does not work in SLAVE mode.
**Fix/Workaround**
Read the last received data, then perform a software reset by writing a one to the Software Reset bit in the Control Register (CR.SWRST).

4 **SPI bad serial clock generation on 2nd chip_select when SCBR=1, CPOL=1, and NCPHA=0**
When multiple chip selects (CS) are in use, if one of the baudrates equal 1 while one (CSRn.SCBR=1) of the others do not equal 1, and CSRn.CPOL=1 and CSRn.NCPHA=0, then an additional pulse will be generated on SCK.
**Fix/Workaround**
When multiple CS are in use, if one of the baudrates equals 1, the others must also equal 1 if CSRn.CPOL=1 and CSRn.NCPHA=0.

### 10.2.8 TC

1 **Channel chaining skips first pulse for upper channel**
When chaining two channels using the Block Mode Register, the first pulse of the clock between the channels is skipped.
**Fix/Workaround**
Configure the lower channel with RA = 0x1 and RC = 0x2 to produce a dummy clock cycle for the upper channel. After the dummy cycle has been generated, indicated by the SR.CPCS bit, reconfigure the RA and RC registers for the lower channel with the real values.

### 10.2.9 TWIM

1 **SMBALERT bit may be set after reset**
For TWIM0 and TWIM1 modules, the SMBus Alert (SMBALERT) bit in the Status Register (SR) might be erroneously set after system reset.
**Fix/Workaround**
After system reset, clear the SR.SMBALERT bit before commencing any TWI transfer.

For TWIM2 module, the SMBus Alert (SMBALERT) is not implemented but the bit in the Status Register (SR) is erroneously set once TWIM2 is enabled.
**Fix/Workaround**
None.

2 **TWIM TWALM polarity is wrong**
The TWALM signal in the TWIM is active high instead of active low.
**Fix/Workaround**
Use an external inverter to invert the signal going into the TWIM. When using both TWIM and TWIS on the same pins, the TWALM cannot be used.