

Features

- High Performance, Low Power 32-bit AVR[®] Microcontroller
 - Compact Single-cycle RISC Instruction Set Including DSP Instruction Set
 - Built-in Floating-Point Processing Unit (FPU)
 - Read-Modify-Write Instructions and Atomic Bit Manipulation
 - Performing 1.49 DMIPS / MHz
 - Up to 91 DMIPS Running at 66 MHz from Flash (1 Wait-State)
 - Up to 49 DMIPS Running at 33 MHz from Flash (0 Wait-State)
 - Memory Protection Unit
- Multi-hierarchy Bus System
 - High-Performance Data Transfers on Separate Buses for Increased Performance
 - 16 Peripheral DMA Channels Improves Speed for Peripheral Communication
- Internal High-Speed Flash
 - 512 Kbytes, 256 Kbytes, 128 Kbytes, 64 Kbytes Versions
 - Single Cycle Access up to 33 MHz
 - FlashVault™ Technology Allows Pre-programmed Secure Library Support for End User Applications
 - Prefetch Buffer Optimizing Instruction Execution at Maximum Speed
 - 100,000 Write Cycles, 15-year Data Retention Capability
 - Flash Security Locks and User Defined Configuration Area
- Internal High-Speed SRAM, Single-Cycle Access at Full Speed
 - 64 Kbytes (512 KB and 256 KB Flash), 32 Kbytes (128 KB Flash), 16 Kbytes (64 KB Flash)
 - 4 Kbytes on the Multi-Layer Bus System (HSB RAM)
- External Memory Interface on AT32UC3C0 Derivatives
 - SDRAM / SRAM Compatible Memory Bus (16-bit Data and 24-bit Address Buses)
- Interrupt Controller
 - Autovectorized Low Latency Interrupt Service with Programmable Priority
- System Functions
 - Power and Clock Manager
 - Internal 115KHz (RCSYS) and 8MHz/1MHz (RC8M) RC Oscillators
 - One 32 KHz and Two Multipurpose Oscillators
 - Clock Failure detection
 - Two Phase-Lock-Loop (PLL) allowing Independent CPU Frequency from USB or CAN Frequency
- Windowed Watchdog Timer (WDT)
- Asynchronous Timer (AST) with Real-Time Clock Capability
 - Counter or Calendar Mode Supported
- Frequency Meter (FREQM) for Accurate Measuring of Clock Frequency
- Ethernet MAC 10/100 Mbps interface
 - 802.3 Ethernet Media Access Controller
 - Supports Media Independent Interface (MII) and Reduced MII (RMII)
- Universal Serial Bus (USB)
 - Device 2.0 and Embedded Host Low Speed and Full Speed
 - Flexible End-Point Configuration and Management with Dedicated DMA Channels
 - On-chip Transceivers Including Pull-Ups
- One 2-channel Controller Area Network (CAN)
 - CAN2A and CAN2B protocol compliant, with high-level mailbox system
 - Two independent channels, 16 Message Objects per Channel



32-bit AVR[®] Microcontroller

AT32UC3C0512C
AT32UC3C0256C
AT32UC3C0128C
AT32UC3C064C
AT32UC3C1512C
AT32UC3C1256C
AT32UC3C1128C
AT32UC3C164C
AT32UC3C2512C
AT32UC3C2256C
AT32UC3C2128C
AT32UC3C264C

Summary

NOTE: This is a summary document.
The complete document is available on
the Atmel website at www.atmel.com.

32117DS-AVR-01/12



- **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), 400kbit/s I²C-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

1. Description

The AT32UC3C is a complete System-On-Chip microcontroller based on the AVR32UC RISC processor running at frequencies up to 66 MHz. AVR32UC is a high-performance 32-bit RISC microprocessor core, designed for cost-sensitive embedded applications, with particular emphasis on low power consumption, high code density and high performance.

The processor implements a Memory Protection Unit (MPU) and a fast and flexible interrupt controller for supporting modern operating systems and real-time operating systems. Using the Secure Access Unit (SAU) together with the MPU provides the required security and integrity.

Higher computation capabilities are achievable either using a rich set of DSP instructions or using the floating-point instructions.

The AT32UC3C incorporates on-chip Flash and SRAM memories for secure and fast access. For applications requiring additional memory, an external memory interface is provided on AT32UC3C0 derivatives.

The Memory Direct Memory Access controller (MDMA) enables transfers of block of data from memories to memories without processor involvement.

The Peripheral Direct Memory Access (PDCA) controller enables data transfers between peripherals and memories without processor involvement. The PDCA drastically reduces processing overhead when transferring continuous and large data streams.

The AT32UC3C incorporates on-chip Flash and SRAM memories for secure and fast access. The FlashVault technology allows secure libraries to be programmed into the device. The secure libraries can be executed while the CPU is in Secure State, but not read by non-secure software in the device. The device can thus be shipped to end customers, who are able to program their own code into the device, accessing the secure libraries, without any risk of compromising the proprietary secure code.

The Power Manager improves design flexibility and security. Power monitoring is supported by on-chip Power-On Reset (POR), Brown-Out Detectors (BOD18, BOD33, BOD50). The CPU runs from the on-chip RC oscillators, the PLLs, or the Multipurpose Oscillators. The Asynchronous Timer (AST) combined with the 32 KHz oscillator keeps track of the time. The AST can operate in counter or calendar mode.

The device includes six identical 16-bit Timer/Counter (TC) channels. Each channel can be independently programmed to perform frequency measurement, event counting, interval measurement, pulse generation, delay timing, and pulse width modulation.

The PWM module provides four channels with many configuration options including polarity, edge alignment and waveform non overlap control. The PWM channels can operate independently, with duty cycles set independently from each other, or in interlinked mode, with multiple channels updated at the same time. It also includes safety feature with fault inputs and the ability to lock the PWM configuration registers and the PWM pin assignment.

The AT32UC3C also features many communication interfaces for communication intensive applications. In addition to standard serial interfaces like UART, SPI or TWI, other interfaces like flexible CAN, USB and Ethernet MAC are available. The USART supports different communication modes, like SPI mode and LIN mode.

The Inter-IC Sound Controller (I2SC) provides a 5-bit wide, bidirectional, synchronous, digital audio link with off-chip audio devices. The controller is compliant with the I2S bus specification.

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

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

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

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

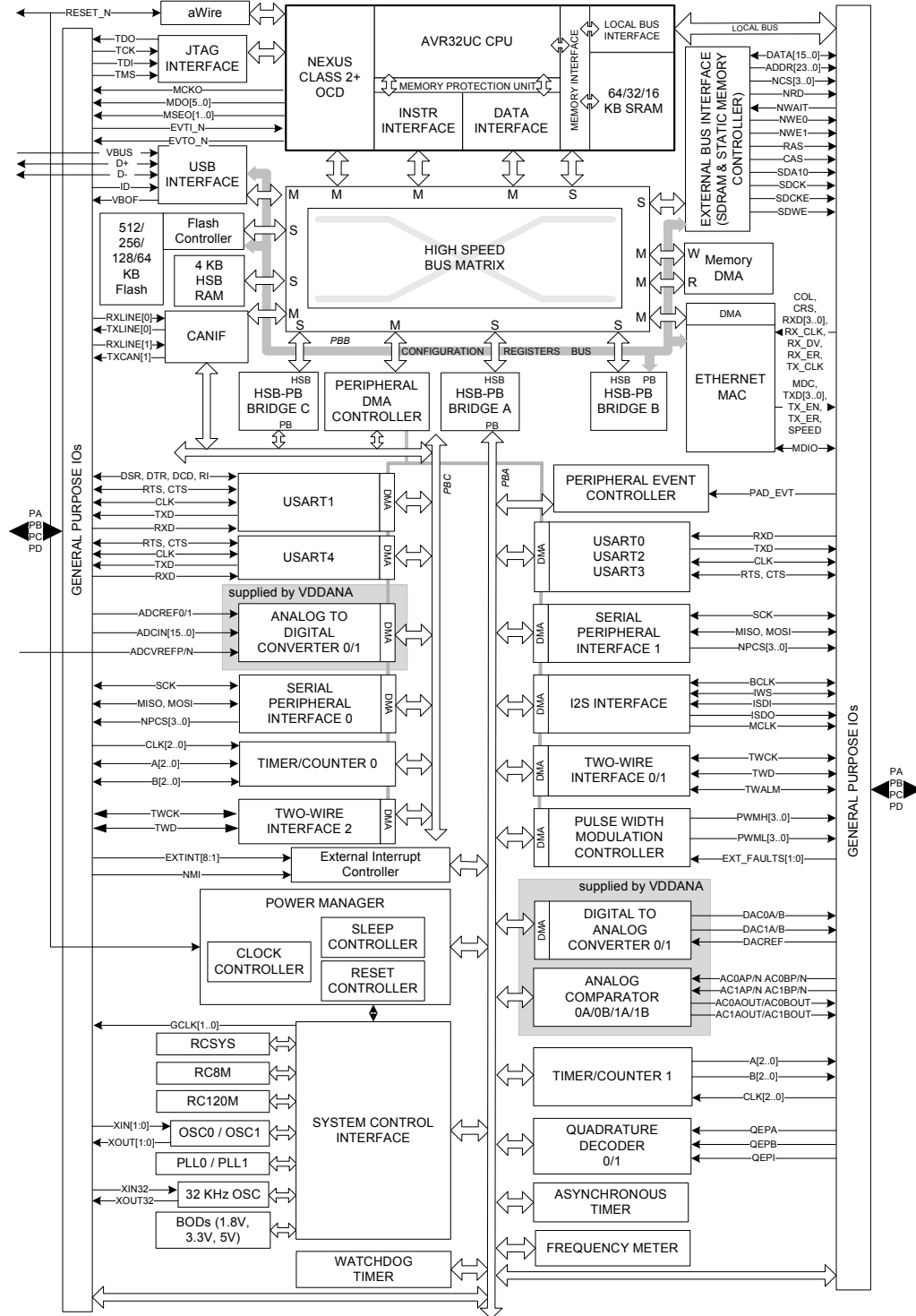
Atmel offers the QTouch library for embedding capacitive touch buttons, sliders, and wheels functionality into AVR microcontrollers. The patented charge-transfer signal acquisition offers robust sensing and included fully debounced reporting of touch keys and includes Adjacent Key Suppression[®] (AKS[®])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

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

2. Overview

2.1 Block diagram

Figure 2-1. Block diagram



2.2 Configuration Summary

Table 2-1. Configuration Summary

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

Table 2-1. Configuration Summary

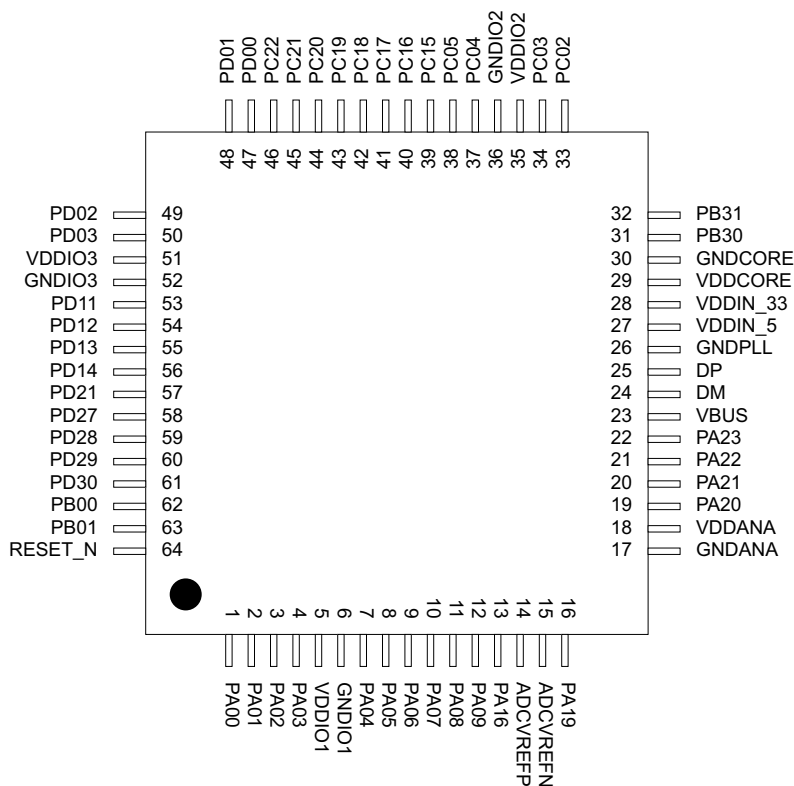
| Feature | AT32UC3C0512C/ AT32UC3C0256C/ AT32UC3C0128C/ AT32UC3C064C | AT32UC3C1512C/ AT32UC3C1256C/ AT32UC3C1128C/ AT32UC3C164C | AT32UC3C2512C/ AT32UC3C2256C/ AT32UC3C2128C/ AT32UC3C264C |
|--------------------|--|--|--|
| Analog Comparators | 4 | 4 | 2 |
| JTAG | 1 | | |
| aWire | 1 | | |
| Max Frequency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

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.

Figure 3-2. TQFP100 Pinout

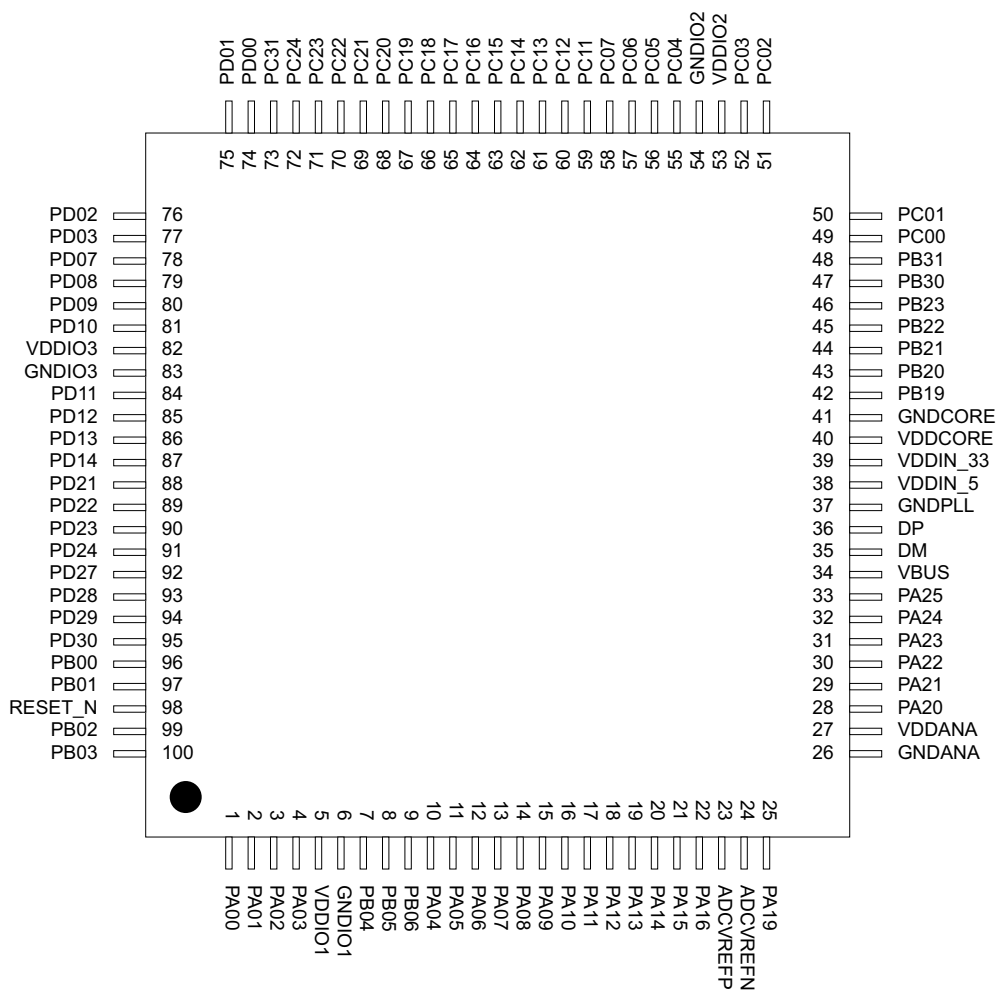
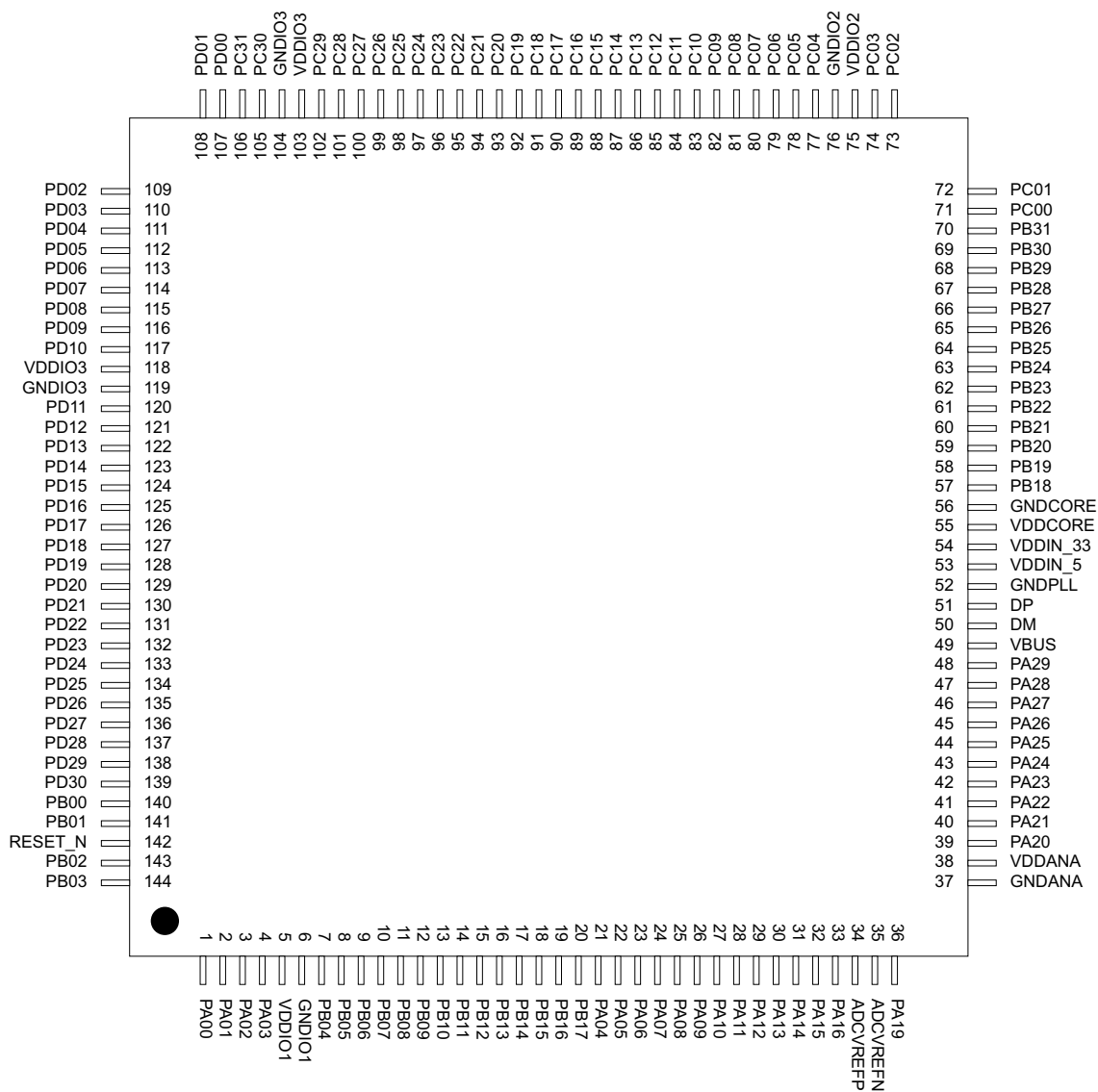


Figure 3-3. LQFP144 Pinout



3.2 Peripheral Multiplexing on I/O lines

3.2.1 Multiplexed Signals

Each GPIO line can be assigned to one of the peripheral functions. The following table describes the peripheral signals multiplexed to the GPIO lines.

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 | | |
| 1 | 1 | 1 | PA00 | 0 | VDDIO1 | x1/x2 | | CANIF - TXLINE[1] | | | | | | |
| 2 | 2 | 2 | PA01 | 1 | VDDIO1 | x1/x2 | | CANIF - RXLINE[1] | PEVC - PAD_EVT [0] | | | | | |
| 3 | 3 | 3 | PA02 | 2 | VDDIO1 | x1/x2 | SCIF - GCLK[0] | | PEVC - PAD_EVT [1] | | | | | |
| 4 | 4 | 4 | PA03 | 3 | VDDIO1 | x1/x2 | SCIF - GCLK[1] | EIC - EXTINT[1] | | | | | | |
| 7 | 10 | 21 | PA04 | 4 | VDDANA | x1/x2 | ADCIN0 | USBC - ID | ACIFA0 - ACAOUT | | | | | |
| 8 | 11 | 22 | PA05 | 5 | VDDANA | x1/x2 | ADCIN1 | USBC - VBOF | ACIFA0 - ACBOUT | | | | | |
| 9 | 12 | 23 | PA06 | 6 | VDDANA | x1/x2 | ADCIN2 | AC1AP1 | PEVC - PAD_EVT [2] | | | | | |
| 10 | 13 | 24 | PA07 | 7 | VDDANA | x1/x2 | ADCIN3 | AC1AN1 | PEVC - PAD_EVT [3] | | | | | |
| 11 | 14 | 25 | PA08 | 8 | VDDANA | x1/x2 | ADCIN4 | AC1BP1 | EIC - EXTINT[2] | | | | | |
| 12 | 15 | 26 | PA09 | 9 | VDDANA | x1/x2 | ADCIN5 | AC1BN1 | | | | | | |
| | 16 | 27 | PA10 | 10 | VDDANA | x1/x2 | ADCIN6 | EIC - EXTINT[4] | PEVC - PAD_EVT [13] | | | | | |
| | 17 | 28 | PA11 | 11 | VDDANA | x1/x2 | ADCIN7 | ADCREFP1 | PEVC - PAD_EVT [14] | | | | | |
| | 18 | 29 | PA12 | 12 | VDDANA | x1/x2 | AC1AP0 | SPI0 - NPSC[0] | AC1AP0 or DAC1A | | | | | |
| | 19 | 30 | PA13 | 13 | VDDANA | x1/x2 | AC1AN0 | SPI0 - NPSC[1] | ADCIN15 | | | | | |
| | 20 | 31 | PA14 | 14 | VDDANA | x1/x2 | AC1BP0 | SPI1 - NPSC[0] | | | | | | |
| | 21 | 32 | PA15 | 15 | VDDANA | x1/x2 | AC1BN0 | SPI1 - NPSC[1] | AC1BN0 or DAC1B | | | | | |
| 13 | 22 | 33 | PA16 | 16 | VDDANA | x1/x2 | ADCREFP0 | | DACREF | | | | | |
| 14 | 23 | 34 | ADC REFP | | | | | | | | | | | |
| 15 | 24 | 35 | ADC REFN | | | | | | | | | | | |

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 | |
| 16 | 25 | 36 | PA19 | 19 | VDDANA | x1/x2 | ADCIN8 | EIC - EXTINT[1] | | | | | |
| 19 | 28 | 39 | PA20 | 20 | VDDANA | x1/x2 | ADCIN9 | AC0AP0 | AC0AP0 or DAC0A | | | | |
| 20 | 29 | 40 | PA21 | 21 | VDDANA | x1/x2 | ADCIN10 | AC0BN0 | AC0BN0 or DAC0B | | | | |
| 21 | 30 | 41 | PA22 | 22 | VDDANA | x1/x2 | ADCIN11 | AC0AN0 | PEVC - PAD_EVT [4] | | | MACB - SPEED | |
| 22 | 31 | 42 | PA23 | 23 | VDDANA | x1/x2 | ADCIN12 | AC0BP0 | PEVC - PAD_EVT [5] | | | MACB - WOL | |
| | 32 | 43 | PA24 | 24 | VDDANA | x1/x2 | ADCIN13 | SPI1 - NPCS[2] | | | | | |
| | 33 | 44 | PA25 | 25 | VDDANA | x1/x2 | ADCIN14 | SPI1 - NPCS[3] | EIC - EXTINT[0] | | | | |
| | | 45 | PA26 | 26 | VDDANA | x1/x2 | AC0AP1 | EIC - EXTINT[1] | | | | | |
| | | 46 | PA27 | 27 | VDDANA | x1/x2 | AC0AN1 | EIC - EXTINT[2] | | | | | |
| | | 47 | PA28 | 28 | VDDANA | x1/x2 | AC0BP1 | EIC - EXTINT[3] | | | | | |
| | | 48 | PA29 | 29 | VDDANA | x1/x2 | AC0BN1 | EIC - EXTINT[0] | | | | | |
| 62 | 96 | 140 | PB00 | 32 | VDDIO1 | x1 | USART0 - CLK | CANIF - RXLINE[1] | EIC - EXTINT[8] | PEVC - PAD_EVT [10] | | | |
| 63 | 97 | 141 | PB01 | 33 | VDDIO1 | x1 | | CANIF - TXLINE[1] | | PEVC - PAD_EVT [11] | | | |
| | 99 | 143 | PB02 | 34 | VDDIO1 | x1 | | USBC - ID | PEVC - PAD_EVT [6] | TC1 - A1 | | | |
| | 100 | 144 | PB03 | 35 | VDDIO1 | x1 | | USBC - VBOF | PEVC - PAD_EVT [7] | | | | |
| | 7 | 7 | PB04 | 36 | VDDIO1 | x1/x2 | SPI1 - MOSI | CANIF - RXLINE[0] | QDEC1 - QEPI | | | MACB - TXD[2] | |
| | 8 | 8 | PB05 | 37 | VDDIO1 | x1/x2 | SPI1 - MISO | CANIF - TXLINE[0] | PEVC - PAD_EVT [12] | USART3 - CLK | | MACB - TXD[3] | |
| | 9 | 9 | PB06 | 38 | VDDIO1 | x2/x4 | SPI1 - SCK | | QDEC1 - QEPA | USART1 - CLK | | MACB - TX_ER | |
| | | 10 | PB07 | 39 | VDDIO1 | x1/x2 | SPI1 - NPCS[0] | EIC - EXTINT[2] | QDEC1 - QEPB | | | MACB - RX_DV | |
| | | 11 | PB08 | 40 | VDDIO1 | x1/x2 | SPI1 - NPCS[1] | PEVC - PAD_EVT [1] | PWM - PWML[0] | | | MACB - RXD[0] | |
| | | 12 | PB09 | 41 | VDDIO1 | x1/x2 | SPI1 - NPCS[2] | | PWM - PWMH[0] | | | MACB - RXD[1] | |
| | | 13 | PB10 | 42 | VDDIO1 | x1/x2 | USART1 - DTR | SPI0 - MOSI | PWM - PWML[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 |
| | | 14 | PB11 | 43 | VDDIO1 | x1/x2 | USART1 - DSR | SPI0 - MISO | PWM - PWMH[1] | | | |
| | | 15 | PB12 | 44 | VDDIO1 | x1/x2 | USART1 - DCD | SPI0 - SCK | PWM - PWML[2] | | | |
| | | 16 | PB13 | 45 | VDDIO1 | x1/x2 | USART1 - RI | SPI0 - NPCS[0] | PWM - PWMH[2] | | MACB - RX_ER | |
| | | 17 | PB14 | 46 | VDDIO1 | x1/x2 | USART1 - RTS | SPI0 - NPCS[1] | PWM - PWML[3] | | MACB - MDC | |
| | | 18 | PB15 | 47 | VDDIO1 | x1/x2 | USART1 - CTS | USART1 - CLK | PWM - PWMH[3] | | MACB - MDIO | |
| | | 19 | PB16 | 48 | VDDIO1 | x1/x2 | USART1 - RXD | SPI0 - NPCS[2] | PWM - EXT_FAULTS[0] | | CANIF - RXLINE[0] | |
| | | 20 | PB17 | 49 | VDDIO1 | x1/x2 | USART1 - TXD | SPI0 - NPCS[3] | PWM - EXT_FAULTS[1] | | CANIF - TXLINE[0] | |
| | | 57 | PB18 | 50 | VDDIO2 | x1/x2 | TC0 - CLK2 | | EIC - EXTINT[4] | | | |
| | 42 | 58 | PB19 | 51 | VDDIO2 | x1/x2 | TC0 - A0 | SPI1 - MOSI | IISC - ISDO | | MACB - CRS | |
| | 43 | 59 | PB20 | 52 | VDDIO2 | x1/x2 | TC0 - B0 | SPI1 - MISO | IISC - ISDI | ACIFA1 - ACAOUT | MACB - COL | |
| | 44 | 60 | PB21 | 53 | VDDIO2 | x2/x4 | TC0 - CLK1 | SPI1 - SCK | IISC - IMCK | ACIFA1 - ACBOUT | MACB - RXD[2] | |
| | 45 | 61 | PB22 | 54 | VDDIO2 | x1/x2 | TC0 - A1 | SPI1 - NPCS[3] | IISC - ISCK | SCIF - GCLK[0] | MACB - RXD[3] | |
| | 46 | 62 | PB23 | 55 | VDDIO2 | x1/x2 | TC0 - B1 | SPI1 - NPCS[2] | IISC - IWS | SCIF - GCLK[1] | MACB - RX_CLK | |
| | | 63 | PB24 | 56 | VDDIO2 | x1/x2 | TC0 - CLK0 | SPI1 - NPCS[1] | | | | |
| | | 64 | PB25 | 57 | VDDIO2 | x1/x2 | TC0 - A2 | SPI1 - NPCS[0] | PEVC - PAD_EVT [8] | | | |
| | | 65 | PB26 | 58 | VDDIO2 | x2/x4 | TC0 - B2 | SPI1 - SCK | PEVC - PAD_EVT [9] | | MACB - TX_EN | |
| | | 66 | PB27 | 59 | VDDIO2 | x1/x2 | QDEC0 - QEPA | SPI1 - MISO | PEVC - PAD_EVT [10] | TC1 - CLK0 | MACB - TXD[0] | |
| | | 67 | PB28 | 60 | VDDIO2 | x1/x2 | QDEC0 - QEPB | SPI1 - MOSI | PEVC - PAD_EVT [11] | TC1 - B0 | MACB - TXD[1] | |
| | | 68 | PB29 | 61 | VDDIO2 | x1/x2 | QDEC0 - QEPI | SPI0 - NPCS[0] | PEVC - PAD_EVT [12] | TC1 - A0 | | |
| 31 | 47 | 69 | PB30 | 62 | VDDIO2 | x1 | | | | | | |
| 32 | 48 | 70 | PB31 | 63 | VDDIO2 | x1 | | | | | | |
| | 49 | 71 | PC00 | 64 | VDDIO2 | x1/x2 | USBC - ID | SPI0 - NPCS[1] | USART2 - CTS | TC1 - B2 | CANIF - TXLINE[1] | |
| | 50 | 72 | PC01 | 65 | VDDIO2 | x1/x2 | USBC - VBOF | SPI0 - NPCS[2] | USART2 - RTS | TC1 - A2 | CANIF - RXLINE[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 |
| 33 | 51 | 73 | PC02 | 66 | VDDIO2 | x1 | TWIMS0 - TWD | SPIO - NPCS[3] | USART2 - RXD | TC1 - CLK1 | MACB - MDC | |
| 34 | 52 | 74 | PC03 | 67 | VDDIO2 | x1 | TWIMS0 - TWCK | EIC - EXTINT[1] | USART2 - TXD | TC1 - B1 | MACB - MDIO | |
| 37 | 55 | 77 | PC04 | 68 | VDDIO2 | x1 | TWIMS1 - TWD | EIC - EXTINT[3] | USART2 - TXD | TC0 - B1 | | |
| 38 | 56 | 78 | PC05 | 69 | VDDIO2 | x1 | TWIMS1 - TWCK | EIC - EXTINT[4] | USART2 - RXD | TC0 - A2 | | |
| | 57 | 79 | PC06 | 70 | VDDIO2 | x1 | PEVC - PAD_EVT [15] | USART2 - CLK | USART2 - CTS | TC0 - CLK2 | TWIMS2 - TWD | TWIMS0 - TWALM |
| | 58 | 80 | PC07 | 71 | VDDIO2 | x1 | PEVC - PAD_EVT [2] | EBI - NCS[3] | USART2 - RTS | TC0 - B2 | TWIMS2 - TWCK | TWIMS1 - TWALM |
| | | 81 | PC08 | 72 | VDDIO2 | x1/x2 | PEVC - PAD_EVT [13] | SPI1 - NPCS[1] | EBI - NCS[0] | | USART4 - TXD | |
| | | 82 | PC09 | 73 | VDDIO2 | x1/x2 | PEVC - PAD_EVT [14] | SPI1 - NPCS[2] | EBI - ADDR[23] | | USART4 - RXD | |
| | | 83 | PC10 | 74 | VDDIO2 | x1/x2 | PEVC - PAD_EVT [15] | SPI1 - NPCS[3] | EBI - ADDR[22] | | | |
| | 59 | 84 | PC11 | 75 | VDDIO2 | x1/x2 | PWM - PWMH[3] | CANIF - RXLINE[1] | EBI - ADDR[21] | TC0 - CLK0 | | |
| | 60 | 85 | PC12 | 76 | VDDIO2 | x1/x2 | PWM - PWML[3] | CANIF - TXLINE[1] | EBI - ADDR[20] | USART2 - CLK | | |
| | 61 | 86 | PC13 | 77 | VDDIO2 | x1/x2 | PWM - PWMH[2] | EIC - EXTINT[7] | | USART0 - RTS | | |
| | 62 | 87 | PC14 | 78 | VDDIO2 | x1/x2 | PWM - PWML[2] | USART0 - CLK | EBI - SDCKE | USART0 - CTS | | |
| 39 | 63 | 88 | PC15 | 79 | VDDIO2 | x1/x2 | PWM - PWMH[1] | SPIO - NPCS[0] | EBI - SDWE | USART0 - RXD | CANIF - RXLINE[1] | |
| 40 | 64 | 89 | PC16 | 80 | VDDIO2 | x1/x2 | PWM - PWML[1] | SPIO - NPCS[1] | EBI - CAS | USART0 - TXD | CANIF - TXLINE[1] | |
| 41 | 65 | 90 | PC17 | 81 | VDDIO2 | x1/x2 | PWM - PWMH[0] | SPIO - NPCS[2] | EBI - RAS | IISC - ISDO | | USART3 - TXD |
| 42 | 66 | 91 | PC18 | 82 | VDDIO2 | x1/x2 | PWM - PWML[0] | EIC - EXTINT[5] | EBI - SDA10 | IISC - ISDI | | USART3 - RXD |
| 43 | 67 | 92 | PC19 | 83 | VDDIO3 | x1/x2 | PWM - PWML[2] | SCIF - GCLK[0] | EBI - DATA[0] | IISC - IMCK | | USART3 - CTS |
| 44 | 68 | 93 | PC20 | 84 | VDDIO3 | x1/x2 | PWM - PWMH[2] | SCIF - GCLK[1] | EBI - DATA[1] | IISC - ISCK | | USART3 - RTS |
| 45 | 69 | 94 | PC21 | 85 | VDDIO3 | x1/x2 | PWM - EXT_FAULTS[0] | CANIF - RXLINE[0] | EBI - DATA[2] | IISC - IWS | | |
| 46 | 70 | 95 | PC22 | 86 | VDDIO3 | x1/x2 | PWM - EXT_FAULTS[1] | CANIF - TXLINE[0] | EBI - DATA[3] | | USART3 - CLK | |
| | 71 | 96 | PC23 | 87 | VDDIO3 | x1/x2 | QDEC1 - QEPB | CANIF - RXLINE[1] | EBI - DATA[4] | PEVC - PAD_EVT [3] | | |



Table 3-1. 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 3-1. GPIO Controller Function Multiplexing

| TQFP / QFN 64 | TQFP 100 | LQFP 144 | PIN | GPIO | Supply | Pin Type (1) | GPIO function | | | | | |
|---------------|----------|----------|------|------|--------|--------------|---------------|-------------------|----------------|----------------|---------------|---|
| | | | | | | | A | B | C | D | E | F |
| | | 124 | PD15 | 111 | VDDIO3 | x1/x2 | TC0 - A0 | USART3 - TXD | EBI - ADDR[11] | | | |
| | | 125 | PD16 | 112 | VDDIO3 | x1/x2 | TC0 - B0 | USART3 - RXD | EBI - ADDR[12] | | | |
| | | 126 | PD17 | 113 | VDDIO3 | x1/x2 | TC0 - A1 | USART3 - CTS | EBI - ADDR[13] | USART3 - CLK | | |
| | | 127 | PD18 | 114 | VDDIO3 | x1/x2 | TC0 - B1 | USART3 - RTS | EBI - ADDR[14] | | | |
| | | 128 | PD19 | 115 | VDDIO3 | x1/x2 | TC0 - A2 | | EBI - ADDR[15] | | | |
| | | 129 | PD20 | 116 | VDDIO3 | x1/x2 | TC0 - B2 | | EBI - ADDR[16] | | | |
| 57 | 88 | 130 | PD21 | 117 | VDDIO3 | x1/x2 | USART3 - TXD | EIC - EXTINT[0] | EBI - ADDR[17] | QDEC1 - QEPI | | |
| | 89 | 131 | PD22 | 118 | VDDIO1 | x1/x2 | USART3 - RXD | TC0 - A2 | EBI - ADDR[18] | SCIF - GCLK[0] | | |
| | 90 | 132 | PD23 | 119 | VDDIO1 | x1/x2 | USART3 - CTS | USART3 - CLK | EBI - ADDR[19] | QDEC1 - QEPA | | |
| | 91 | 133 | PD24 | 120 | VDDIO1 | x1/x2 | USART3 - RTS | EIC - EXTINT[8] | EBI - NWE1 | QDEC1 - QEPB | | |
| | | 134 | PD25 | 121 | VDDIO1 | x1/x2 | TC0 - CLK0 | USBC - ID | EBI - NWE0 | | USART4 - CLK | |
| | | 135 | PD26 | 122 | VDDIO1 | x1/x2 | TC0 - CLK1 | USBC - VBOF | EBI - NRD | | | |
| 58 | 92 | 136 | PD27 | 123 | VDDIO1 | x1/x2 | USART0 - TXD | CANIF - RXLINE[0] | EBI - NCS[1] | TC0 - A0 | MACB - RX_ER | |
| 59 | 93 | 137 | PD28 | 124 | VDDIO1 | x1/x2 | USART0 - RXD | CANIF - TXLINE[0] | EBI - NCS[2] | TC0 - B0 | MACB - RX_DV | |
| 60 | 94 | 138 | PD29 | 125 | VDDIO1 | x1/x2 | USART0 - CTS | EIC - EXTINT[6] | USART0 - CLK | TC0 - CLK0 | MACB - TX_CLK | |
| 61 | 95 | 139 | PD30 | 126 | VDDIO1 | x1/x2 | USART0 - RTS | EIC - EXTINT[3] | EBI - NWAIT | TC0 - A1 | MACB - TX_EN | |

Note: 1. Refer to "Electrical Characteristics" on page 50 for a description of the electrical properties of the pin types used.
See Section 3.3 for a description of the various peripheral signals.

3.2.2 Peripheral Functions

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

Table 3-2. Peripheral Functions

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

Table 3-2. Peripheral Functions

| Function | Description |
|-----------------------|------------------------------|
| 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,

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

Table 3-5. Nexus OCD AUX port connections

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

3.2.6 Other Functions

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

Table 3-6. Other Functions

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

3.3 Signals Description

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

Table 3-7. Signal Description List

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

Table 3-7. Signal Description List

| Signal Name | Function | Type | Active Level | Comments |
|---|---|------------------------|--------------|---|
| VDDIN_5 | 1.8V Voltage Regulator Input | Power Input | | Power Supply: 4.5V to 5.5V or 3.0V to 3.6 V |
| VDDIN_33 | USB I/O power supply | Power Output/ Input | | Capacitor Connection for the 3.3V voltage regulator or power supply: 3.0V to 3.6 V |
| VDDCORE | 1.8V Voltage Regulator Output | Power output | | Capacitor Connection for the 1.8V voltage regulator |
| GNDIO1 GNDIO2 GNDIO3 | I/O Ground | Ground | | |
| GNDANA | Analog Ground | Ground | | |
| GNDCORE | Ground of the core | Ground | | |
| GNDPLL | Ground of the PLLs | Ground | | |
| Analog Comparator Interface - ACIFA0/1 | | | | |
| AC0AN1/AC0AN0 | Negative inputs for comparator AC0A | Analog | | |
| AC0AP1/AC0AP0 | Positive inputs for comparator AC0A | Analog | | |
| AC0BN1/AC0BN0 | Negative inputs for comparator AC0B | Analog | | |
| AC0BP1/AC0BP0 | Positive inputs for comparator AC0B | Analog | | |
| AC1AN1/AC1AN0 | Negative inputs for comparator AC1A | Analog | | |
| AC1AP1/AC1AP0 | Positive inputs for comparator AC1A | Analog | | |
| AC1BN1/AC1BN0 | Negative inputs for comparator AC1B | Analog | | |
| AC1BP1/AC1BP0 | Positive inputs for comparator AC1B | Analog | | |
| ACAOUT/ACBOUT | analog comparator outputs | output | | |
| ADC Interface - ADCIFA | | | | |
| ADCIN[15:0] | ADC input pins | Analog | | |
| ADCREFO | Analog positive reference 0 voltage input | Analog | | |
| ADCREF1 | Analog positive reference 1 voltage input | Analog | | |
| ADCVREFP | Analog positive reference connected to external capacitor | Analog | | |

Table 3-7. Signal Description List

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

Table 3-7. Signal Description List

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

Table 3-7. Signal Description List

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

Table 3-7. Signal Description List

| Signal Name | Function | Type | Active Level | Comments |
|---|--------------------------------|--------|--------------|------------------------|
| NPCS[3:0] | SPI Peripheral Chip Select | I/O | Low | |
| SCK | Clock | Output | | |
| Timer/Counter - TC0, TC1 | | | | |
| A0 | Channel 0 Line A | I/O | | |
| A1 | Channel 1 Line A | I/O | | |
| A2 | Channel 2 Line A | I/O | | |
| B0 | Channel 0 Line B | I/O | | |
| B1 | Channel 1 Line B | I/O | | |
| B2 | Channel 2 Line B | I/O | | |
| CLK0 | Channel 0 External Clock Input | Input | | |
| CLK1 | Channel 1 External Clock Input | Input | | |
| CLK2 | Channel 2 External Clock Input | Input | | |
| Two-wire Interface - TWIMS0, TWIMS1, TWIMS2 | | | | |
| TWALM | SMBus SMBALERT | I/O | Low | Only on TWIMS0, TWIMS1 |
| TWCK | Serial Clock | I/O | | |
| TWD | Serial Data | I/O | | |
| Universal Synchronous Asynchronous Receiver Transmitter - USART0, USART1, USART2, USART3, USART4 | | | | |
| CLK | Clock | I/O | | |
| CTS | Clear To Send | Input | Low | |
| DCD | Data Carrier Detect | Input | Low | Only USART1 |
| DSR | Data Set Ready | Input | Low | Only USART1 |
| DTR | Data Terminal Ready | Output | Low | Only USART1 |
| RI | Ring Indicator | Input | Low | Only USART1 |
| RTS | Request To Send | Output | Low | |
| RXD | Receive Data | Input | | |
| TXD | Transmit Data | Output | | |
| Universal Serial Bus Device - USB | | | | |
| DM | USB Device Port Data - | Analog | | |

Table 3-7. Signal Description List

| Signal Name | Function | Type | Active Level | Comments |
|-------------|---|--------------|--------------|----------|
| DP | USB Device Port Data + | Analog | | |
| VBUS | USB VBUS Monitor and OTG Negotiation | Analog Input | | |
| ID | ID Pin of the USB Bus | Input | | |
| VBOF | USB VBUS On/off: bus power control port | output | | |

3.4 I/O Line Considerations

3.4.1 JTAG pins

The JTAG is enabled if TCK is low while the RESET_N pin is released. The TCK, TMS, and TDI pins have pull-up resistors when JTAG is enabled. The TCK pin always have pull-up enabled during reset. The TDO pin is an output, driven at VDDIO1, and has no pull-up resistor. The JTAG pins can be used as GPIO pins and muxed with peripherals when the JTAG is disabled. Please refer to [Section 3.2.4](#) for the JTAG port connections.

3.4.2 RESET_N pin

The RESET_N pin integrates a pull-up resistor to VDDIO1. As the product integrates a power-on reset cell, the RESET_N pin can be left unconnected in case no reset from the system needs to be applied to the product.

The RESET_N pin is also used for the aWire debug protocol. When the pin is used for debugging, it must not be driven by external circuitry.

3.4.3 TWI pins

When these pins are used for TWI, the pins are open-drain outputs with slew-rate limitation and inputs with inputs with spike-filtering. When used as GPIO-pins or used for other peripherals, the pins have the same characteristics as GPIO pins.

3.4.4 GPIO pins

All I/O lines integrate programmable pull-up and pull-down resistors. Most I/O lines integrate drive strength control, see [Table 3-1](#). Programming of this pull-up and pull-down resistor or this drive strength is performed independently for each I/O line through the GPIO Controllers.

After reset, I/O lines default as inputs with pull-up/pull-down resistors disabled. After reset, output drive strength is configured to the lowest value to reduce global EMI of the device.

When the I/O line is configured as analog function (ADC I/O, AC inputs, DAC I/O), the pull-up and pull-down resistors are automatically disabled.

4. Processor and Architecture

Rev: 2.1.2.0

This chapter gives an overview of the AVR32UC CPU. AVR32UC is an implementation of the AVR32 architecture. A summary of the programming model, instruction set, and MPU is presented. For further details, see the *AVR32 Architecture Manual* and the *AVR32UC Technical Reference Manual*.

4.1 Features

- **32-bit load/store AVR32A RISC architecture**
 - 15 general-purpose 32-bit registers
 - 32-bit Stack Pointer, Program Counter and Link Register reside in register file
 - Fully orthogonal instruction set
 - Privileged and unprivileged modes enabling efficient and secure operating systems
 - Innovative instruction set together with variable instruction length ensuring industry leading code density
 - DSP extension with saturating arithmetic, and a wide variety of multiply instructions
- **3-stage pipeline allowing one instruction per clock cycle for most instructions**
 - Byte, halfword, word, and double word memory access
 - Multiple interrupt priority levels
- **MPU allows for operating systems with memory protection**
- **FPU enables hardware accelerated floating point calculations**
- **Secure State for supporting FlashVault technology**

4.2 AVR32 Architecture

AVR32 is a new, high-performance 32-bit RISC microprocessor architecture, designed for cost-sensitive embedded applications, with particular emphasis on low power consumption and high code density. In addition, the instruction set architecture has been tuned to allow a variety of microarchitectures, enabling the AVR32 to be implemented as low-, mid-, or high-performance processors. AVR32 extends the AVR family into the world of 32- and 64-bit applications.

Through a quantitative approach, a large set of industry recognized benchmarks has been compiled and analyzed to achieve the best code density in its class. In addition to lowering the memory requirements, a compact code size also contributes to the core's low power characteristics. The processor supports byte and halfword data types without penalty in code size and performance.

Memory load and store operations are provided for byte, halfword, word, and double word data with automatic sign- or zero extension of halfword and byte data. The C-compiler is closely linked to the architecture and is able to exploit code optimization features, both for size and speed.

In order to reduce code size to a minimum, some instructions have multiple addressing modes. As an example, instructions with immediates often have a compact format with a smaller immediate, and an extended format with a larger immediate. In this way, the compiler is able to use the format giving the smallest code size.

Another feature of the instruction set is that frequently used instructions, like add, have a compact format with two operands as well as an extended format with three operands. The larger format increases performance, allowing an addition and a data move in the same instruction in a

single cycle. Load and store instructions have several different formats in order to reduce code size and speed up execution.

The register file is organized as sixteen 32-bit registers and includes the Program Counter, the Link Register, and the Stack Pointer. In addition, register R12 is designed to hold return values from function calls and is used implicitly by some instructions.

4.3 The AVR32UC CPU

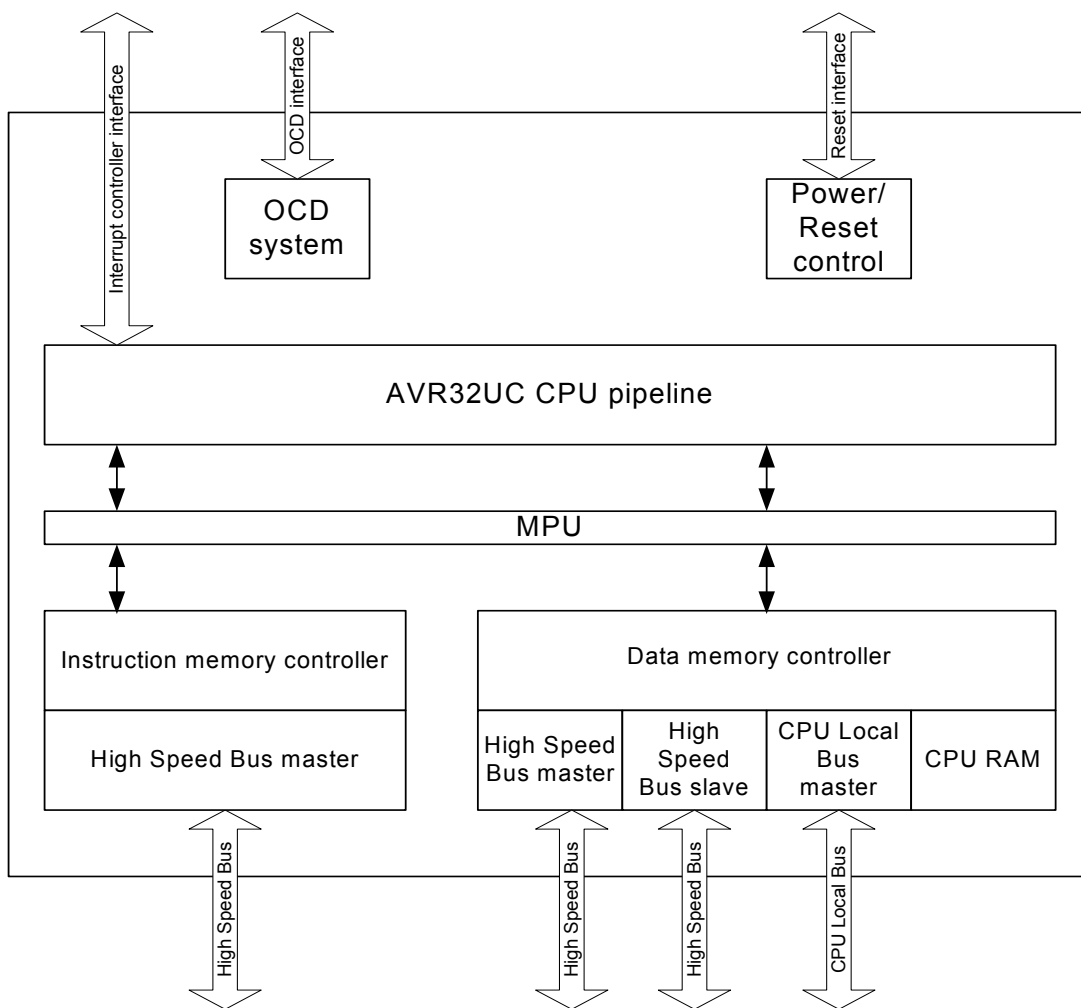
The AVR32UC CPU targets low- and medium-performance applications, and provides an advanced On-Chip Debug (OCD) system, no caches, and a Memory Protection Unit (MPU). A hardware Floating Point Unit (FPU) is also provided through the coprocessor instruction space. Java acceleration hardware is not implemented.

AVR32UC provides three memory interfaces, one High Speed Bus master for instruction fetch, one High Speed Bus master for data access, and one High Speed Bus slave interface allowing other bus masters to access data RAMs internal to the CPU. Keeping data RAMs internal to the CPU allows fast access to the RAMs, reduces latency, and guarantees deterministic timing. Also, power consumption is reduced by not needing a full High Speed Bus access for memory accesses. A dedicated data RAM interface is provided for communicating with the internal data RAMs.

A local bus interface is provided for connecting the CPU to device-specific high-speed systems, such as floating-point units and I/O controller ports. This local bus has to be enabled by writing a one to the LOCEN bit in the CPUCR system register. The local bus is able to transfer data between the CPU and the local bus slave in a single clock cycle. The local bus has a dedicated memory range allocated to it, and data transfers are performed using regular load and store instructions. Details on which devices that are mapped into the local bus space is given in the CPU Local Bus section in the Memories chapter.

[Figure 4-1 on page 27](#) displays the contents of AVR32UC.

Figure 4-1. Overview of the AVR32UC CPU



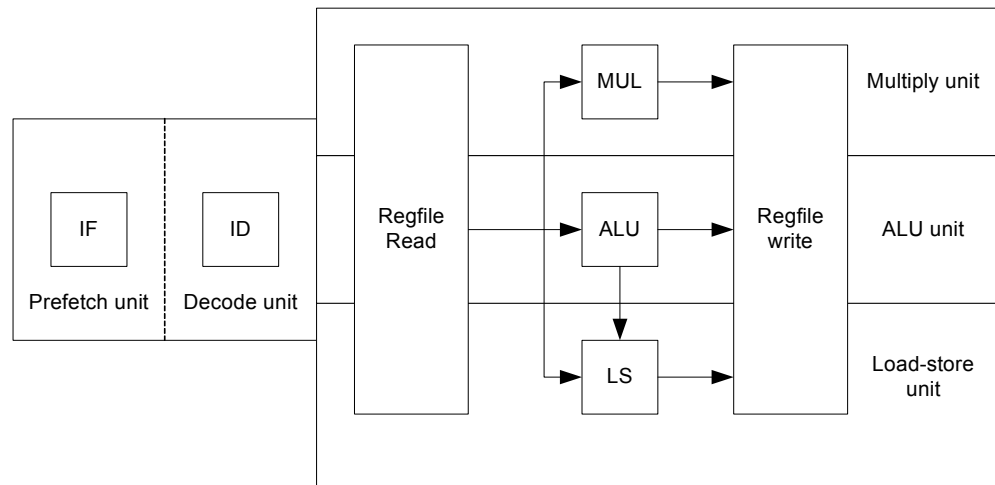
4.3.1 Pipeline Overview

AVR32UC has three pipeline stages, Instruction Fetch (IF), Instruction Decode (ID), and Instruction Execute (EX). The EX stage is split into three parallel subsections, one arithmetic/logic (ALU) section, one multiply (MUL) section, and one load/store (LS) section.

Instructions are issued and complete in order. Certain operations require several clock cycles to complete, and in this case, the instruction resides in the ID and EX stages for the required number of clock cycles. Since there is only three pipeline stages, no internal data forwarding is required, and no data dependencies can arise in the pipeline.

Figure 4-2 on page 28 shows an overview of the AVR32UC pipeline stages.

Figure 4-2. The AVR32UC Pipeline



4.3.2 AVR32A Microarchitecture Compliance

AVR32UC implements an AVR32A microarchitecture. The AVR32A microarchitecture is targeted at cost-sensitive, lower-end applications like smaller microcontrollers. This microarchitecture does not provide dedicated hardware registers for shadowing of register file registers in interrupt contexts. Additionally, it does not provide hardware registers for the return address registers and return status registers. Instead, all this information is stored on the system stack. This saves chip area at the expense of slower interrupt handling.

4.3.2.1 Interrupt Handling

Upon interrupt initiation, registers R8-R12 are automatically pushed to the system stack. These registers are pushed regardless of the priority level of the pending interrupt. The return address and status register are also automatically pushed to stack. The interrupt handler can therefore use R8-R12 freely. Upon interrupt completion, the old R8-R12 registers and status register are restored, and execution continues at the return address stored popped from stack.

The stack is also used to store the status register and return address for exceptions and *scall*. Executing the *rete* or *rets* instruction at the completion of an exception or system call will pop this status register and continue execution at the popped return address.

4.3.2.2 Java Support

AVR32UC does not provide Java hardware acceleration.

4.3.2.3 Floating Point Support

A fused multiply-accumulate Floating Point Unit (FPU), performing a multiply and accumulate as a single operation with no intermediate rounding, thereby increasing precision is provided. The floating point hardware conforms to the requirements of the C standard, which is based on the IEEE 754 floating point standard.

4.3.2.4 Memory Protection

The MPU allows the user to check all memory accesses for privilege violations. If an access is attempted to an illegal memory address, the access is aborted and an exception is taken. The MPU in AVR32UC is specified in the AVR32UC Technical Reference manual.

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.

4.4 Programming Model

4.4.1 Register File Configuration

The AVR32UC register file is shown below.

Figure 4-3. The AVR32UC Register File

| Application | | Supervisor | | INT0 | | INT1 | | INT2 | | INT3 | | Exception | | NMI | | Secure | |
|-------------|--------|------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|-----------|--------|--------|--------|--------|--------|
| Bit 31 | Bit 0 | Bit 31 | Bit 0 | Bit 31 | Bit 0 | Bit 31 | Bit 0 | Bit 31 | Bit 0 | Bit 31 | Bit 0 | Bit 31 | Bit 0 | Bit 31 | Bit 0 | Bit 31 | Bit 0 |
| PC | PC | PC | PC | PC | PC | PC | PC | PC | PC | PC | PC | PC | PC | PC | PC | PC | PC |
| LR | LR | LR | LR | LR | LR | LR | LR | LR | LR | LR | LR | LR | LR | LR | LR | LR | LR |
| SP_APP | SP_SYS | SP_SYS | SP_SYS | SP_SYS | SP_SYS | SP_SYS | SP_SYS | SP_SYS | SP_SYS | SP_SYS | SP_SYS | SP_SYS | SP_SYS | SP_SYS | SP_SYS | SP_SEC | SP_SEC |
| R12 | R12 | R12 | R12 | R12 | R12 | R12 | R12 | R12 | R12 | R12 | R12 | R12 | R12 | R12 | R12 | R12 | R12 |
| R11 | R11 | R11 | R11 | R11 | R11 | R11 | R11 | R11 | R11 | R11 | R11 | R11 | R11 | R11 | R11 | R11 | R11 |
| R10 | R10 | R10 | R10 | R10 | R10 | R10 | R10 | R10 | R10 | R10 | R10 | R10 | R10 | R10 | R10 | R10 | R10 |
| R9 | R9 | R9 | R9 | R9 | R9 | R9 | R9 | R9 | R9 | R9 | R9 | R9 | R9 | R9 | R9 | R9 | R9 |
| R8 | R8 | R8 | R8 | R8 | R8 | R8 | R8 | R8 | R8 | R8 | R8 | R8 | R8 | R8 | R8 | R8 | R8 |
| R7 | R7 | R7 | R7 | R7 | R7 | R7 | R7 | R7 | R7 | R7 | R7 | R7 | R7 | R7 | R7 | R7 | R7 |
| R6 | R6 | R6 | R6 | R6 | R6 | R6 | R6 | R6 | R6 | R6 | R6 | R6 | R6 | R6 | R6 | R6 | R6 |
| R5 | R5 | R5 | R5 | R5 | R5 | R5 | R5 | R5 | R5 | R5 | R5 | R5 | R5 | R5 | R5 | R5 | R5 |
| R4 | R4 | R4 | R4 | R4 | R4 | R4 | R4 | R4 | R4 | R4 | R4 | R4 | R4 | R4 | R4 | R4 | R4 |
| R3 | R3 | R3 | R3 | R3 | R3 | R3 | R3 | R3 | R3 | R3 | R3 | R3 | R3 | R3 | R3 | R3 | R3 |
| R2 | R2 | R2 | R2 | R2 | R2 | R2 | R2 | R2 | R2 | R2 | R2 | R2 | R2 | R2 | R2 | R2 | R2 |
| R1 | R1 | R1 | R1 | R1 | R1 | R1 | R1 | R1 | R1 | R1 | R1 | R1 | R1 | R1 | R1 | R1 | R1 |
| R0 | R0 | R0 | R0 | R0 | R0 | R0 | R0 | R0 | R0 | R0 | R0 | R0 | R0 | R0 | R0 | R0 | R0 |
| SR | SR | SR | SR | SR | SR | SR | SR | SR | SR | SR | SR | SR | SR | SR | SR | SR | SR |

| |
|-----------|
| SS_STATUS |
| SS_ADRF |
| SS_ADRR |
| SS_ADR0 |
| SS_ADR1 |
| SS_SP_SYS |
| SS_SP_APP |
| SS_RAR |
| SS_RSR |

4.4.2 Status Register Configuration

The Status Register (SR) is split into two halfwords, one upper and one lower, see [Figure 4-4](#) and [Figure 4-5](#). The lower word contains the C, Z, N, V, and Q condition code flags and the R, T, and L bits, while the upper halfword contains information about the mode and state the processor executes in. Refer to the *AVR32 Architecture Manual* for details.

Figure 4-4. The Status Register High Halfword

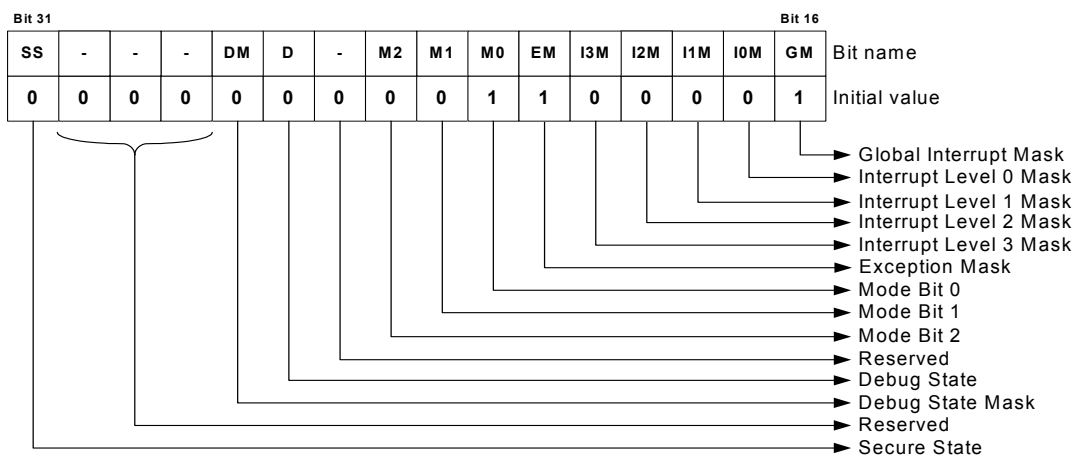


Figure 4-5. The Status Register Low Halfword



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 |

Table 4-3. System Registers (Continued)

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

Table 4-3. System Registers (Continued)

| Reg # | Address | Name | Function |
|---------|----------|-----------|---|
| 90 | 360 | MPUPSR2 | MPU Privilege Select Register region 2 |
| 91 | 364 | MPUPSR3 | MPU Privilege Select Register region 3 |
| 92 | 368 | MPUPSR4 | MPU Privilege Select Register region 4 |
| 93 | 372 | MPUPSR5 | MPU Privilege Select Register region 5 |
| 94 | 376 | MPUPSR6 | MPU Privilege Select Register region 6 |
| 95 | 380 | MPUPSR7 | MPU Privilege Select Register region 7 |
| 96 | 384 | MPUCRA | Unused in this version of AVR32UC |
| 97 | 388 | MPUCRB | Unused in this version of AVR32UC |
| 98 | 392 | MPUBRA | Unused in this version of AVR32UC |
| 99 | 396 | MPUBRB | Unused in this version of AVR32UC |
| 100 | 400 | MPUAPRA | MPU Access Permission Register A |
| 101 | 404 | MPUAPRB | MPU Access Permission Register B |
| 102 | 408 | MPUCR | MPU Control Register |
| 103 | 412 | SS_STATUS | Secure State Status Register |
| 104 | 416 | SS_ADRF | Secure State Address Flash Register |
| 105 | 420 | SS_ADRR | Secure State Address RAM Register |
| 106 | 424 | SS_ADR0 | Secure State Address 0 Register |
| 107 | 428 | SS_ADR1 | Secure State Address 1 Register |
| 108 | 432 | SS_SP_SYS | Secure State Stack Pointer System Register |
| 109 | 436 | SS_SP_APP | Secure State Stack Pointer Application Register |
| 110 | 440 | SS_RAR | Secure State Return Address Register |
| 111 | 444 | SS_RSR | Secure State Return Status Register |
| 112-191 | 448-764 | Reserved | Reserved for future use |
| 192-255 | 768-1020 | IMPL | IMPLEMENTATION DEFINED |

4.5 Exceptions and Interrupts

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

When an event occurs, the execution of the instruction stream is halted, and execution is passed to an event handler at an address specified in [Table 4-4 on page 38](#). Most of the handlers are placed sequentially in the code space starting at the address specified by EVBA, with four bytes between each handler. This gives ample space for a jump instruction to be placed there, jumping to the event routine itself. A few critical handlers have larger spacing between them, allowing the entire event routine to be placed directly at the address specified by the EVBA-relative offset generated by hardware. All interrupt sources have autovectored interrupt service routine (ISR) addresses. This allows the interrupt controller to directly specify the ISR address as an address

relative to EVBA. The autovector offset has 14 address bits, giving an offset of maximum 16384 bytes. The target address of the event handler is calculated as $(EVBA | event_handler_offset)$, not $(EVBA + event_handler_offset)$, so EVBA and exception code segments must be set up appropriately. The same mechanisms are used to service all different types of events, including interrupt requests, yielding a uniform event handling scheme.

An interrupt controller does the priority handling of the interrupts and provides the autovector offset to the CPU.

4.5.1 System Stack Issues

Event handling in AVR32UC uses the system stack pointed to by the system stack pointer, SP_SYS, for pushing and popping R8-R12, LR, status register, and return address. Since event code may be timing-critical, SP_SYS should point to memory addresses in the IRAM section, since the timing of accesses to this memory section is both fast and deterministic.

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

4.5.2 Exceptions and Interrupt Requests

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

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

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

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

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

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

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

Table 4-4. Priority and Handler Addresses for Events

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

5. Memories

5.1 Embedded Memories

- Internal High-Speed Flash (See [Table 5-1 on page 40](#))
 - 512 Kbytes
 - 256 Kbytes
 - 128 Kbytes
 - 64 Kbytes
 - 0 Wait State Access at up to 33 MHz in Worst Case Conditions
 - 1 Wait State Access at up to 66 MHz in Worst Case Conditions
 - Pipelined Flash Architecture, allowing burst reads from sequential Flash locations, hiding penalty of 1 wait state access
 - Pipelined Flash Architecture typically reduces the cycle penalty of 1 wait state operation to only 15% compared to 0 wait state operation
 - 100 000 Write Cycles, 15-year Data Retention Capability
 - Sector Lock Capabilities, Bootloader Protection, Security Bit
 - 32 Fuses, Erased During Chip Erase
 - User Page For Data To Be Preserved During Chip Erase
- Internal High-Speed SRAM, Single-cycle access at full speed (See [Table 5-1 on page 40](#))
 - 64 Kbytes
 - 32 Kbytes
 - 16 Kbytes
- Supplementary Internal High-Speed System SRAM (HSB RAM), Single-cycle access at full speed
 - Memory space available on System Bus for peripherals data.
 - 4 Kbytes

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 C2512C | C0256C | C1256C C2256C | C0128C | C1128C C2128C | C064C | C164C C264C |
| Embedded SRAM | 0x0000_0000 | 64 KB | 64 KB | 64 KB | 64 KB | 32 KB | 32 KB | 16 KB | 16 KB |
| Embedded Flash | 0x8000_0000 | 512 KB | 512 KB | 256 KB | 256 KB | 128 KB | 128 KB | 64 KB | 64 KB |
| SAU | 0x9000_0000 | 1 KB | 1 KB | 1 KB | 1 KB | 1 KB | 1 KB | 1 KB | 1 KB |
| HSB SRAM | 0xA000_0000 | 4 KB | 4 KB | 4 KB | 4 KB | 4 KB | 4 KB | 4 KB | 4 KB |
| EBI SRAM CS0 | 0xC000_0000 | 16 MB | - | 16 MB | - | 16 MB | - | 16 MB | - |
| EBI SRAM CS2 | 0xC800_0000 | 16 MB | - | 16 MB | - | 16 MB | - | 16 MB | - |
| EBI SRAM CS3 | 0xCC00_0000 | 16 MB | - | 16 MB | - | 16 MB | - | 16 MB | - |
| EBI SRAM /SDRAM CS1 | 0xD000_0000 | 128 MB | - | 128 MB | - | 128 MB | - | 128 MB | - |
| HSB-PB Bridge C | 0xFFFD_0000 | 64 KB | 64 KB | 64 KB | 64 KB | 64 KB | 64 KB | 64 KB | 64 KB |
| HSB-PB Bridge B | 0xFFFE_0000 | 64 KB | 64 KB | 64 KB | 64 KB | 64 KB | 64 KB | 64 KB | 64 KB |
| HSB-PB Bridge A | 0xFFFF_0000 | 64 KB | 64 KB | 64 KB | 64 KB | 64 KB | 64 KB | 64 KB | 64 KB |

Table 5-2. Flash Memory Parameters

| Part Number | Flash Size (FLASH_PW) | Number of pages (FLASH_P) | Page size (FLASH_W) |
|---|-----------------------|---------------------------|---------------------|
| AT32UC3C0512C AT32UC3C1512C AT32UC3C2512C | 512 Kbytes | 1024 | 128 words |
| AT32UC3C0256C AT32UC3C1256C AT32UC3C2256C | 256 Kbytes | 512 | 128 words |
| AT32UC3C0128C AT32UC3C1128C AT32UC3C2128C | 128 Kbytes | 256 | 128 words |
| AT32UC3C064C AT32UC3C164C AT32UC3C264C | 64 Kbytes | 128 | 128 words |

5.3 Peripheral Address Map

Table 5-3. Peripheral Address Mapping

| Address | Peripheral Name |
|-------------|---|
| 0xFFFFD0000 | PDCA Peripheral DMA Controller - PDCA |
| 0xFFFFD1000 | MDMA Memory DMA - MDMA |
| 0xFFFFD1400 | USART1 Universal Synchronous/Asynchronous Receiver/Transmitter - USART1 |
| 0xFFFFD1800 | SPI0 Serial Peripheral Interface - SPI0 |
| 0xFFFFD1C00 | CANIF Control Area Network interface - CANIF |
| 0xFFFFD2000 | TC0 Timer/Counter - TC0 |
| 0xFFFFD2400 | ADCIFA ADC controller interface with Touch Screen functionality - ADCIFA |
| 0xFFFFD2800 | USART4 Universal Synchronous/Asynchronous Receiver/Transmitter - USART4 |
| 0xFFFFD2C00 | TWIM2 Two-wire Master Interface - TWIM2 |
| 0xFFFFD3000 | TWIS2 Two-wire Slave Interface - TWIS2 |

Table 5-3. Peripheral Address Mapping

| | | |
|------------|---------|--|
| 0xFFFE0000 | HFLASHC | Flash Controller - HFLASHC |
| 0xFFFE1000 | USBC | USB 2.0 OTG Interface - USBC |
| 0xFFFE2000 | HMATRIX | HSB Matrix - HMATRIX |
| 0xFFFE2400 | SAU | Secure Access Unit - SAU |
| 0xFFFE2800 | SMC | Static Memory Controller - SMC |
| 0xFFFE2C00 | SDRAMC | SDRAM Controller - SDRAMC |
| 0xFFFE3000 | MACB | Ethernet MAC - MACB |
| 0xFFFF0000 | INTC | Interrupt controller - INTC |
| 0xFFFF0400 | PM | Power Manager - PM |
| 0xFFFF0800 | SCIF | System Control Interface - SCIF |
| 0xFFFF0C00 | AST | Asynchronous Timer - AST |
| 0xFFFF1000 | WDT | Watchdog Timer - WDT |
| 0xFFFF1400 | EIC | External Interrupt Controller - EIC |
| 0xFFFF1800 | FREQM | Frequency Meter - FREQM |
| 0xFFFF2000 | GPIO | General Purpose Input/Output Controller - GPIO |
| 0xFFFF2800 | USART0 | Universal Synchronous/Asynchronous Receiver/Transmitter - USART0 |
| 0xFFFF2C00 | USART2 | Universal Synchronous/Asynchronous Receiver/Transmitter - USART2 |
| 0xFFFF3000 | USART3 | Universal Synchronous/Asynchronous Receiver/Transmitter - USART3 |
| 0xFFFF3400 | SPI1 | Serial Peripheral Interface - SPI1 |

Table 5-3. Peripheral Address Mapping

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

5.4 CPU Local Bus Mapping

Some of the registers in the GPIO module are mapped onto the CPU local bus, in addition to being mapped on the Peripheral Bus. These registers can therefore be reached both by accesses on the Peripheral Bus, and by accesses on the local bus.

Mapping these registers on the local bus allows cycle-deterministic toggling of GPIO pins since the CPU and GPIO are the only modules connected to this bus. Also, since the local bus runs at CPU speed, one write or read operation can be performed per clock cycle to the local bus-mapped GPIO registers.

The following GPIO registers are mapped on the local bus:

Table 5-4. Local bus mapped GPIO registers

| Port | Register | Mode | Local Bus Address | Access |
|-----------------------------|--------------------------------------|--------------------------------------|-------------------|------------|
| A | Output Driver Enable Register (ODER) | WRITE | 0x40000040 | Write-only |
| | | SET | 0x40000044 | Write-only |
| | | CLEAR | 0x40000048 | Write-only |
| | | TOGGLE | 0x4000004C | Write-only |
| | Output Value Register (OVR) | WRITE | 0x40000050 | Write-only |
| | | SET | 0x40000054 | Write-only |
| | | CLEAR | 0x40000058 | Write-only |
| | | TOGGLE | 0x4000005C | Write-only |
| | Pin Value Register (PVR) | - | 0x40000060 | Read-only |
| | B | Output Driver Enable Register (ODER) | WRITE | 0x40000140 |
| 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 |
| | 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 |

Table 5-4. Local bus mapped GPIO registers

| Port | Register | Mode | Local Bus Address | Access |
|------|--------------------------------------|--------|-------------------|------------|
| 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 |

6. Supply and Startup Considerations

6.1 Supply Considerations

6.1.1 Power Supplies

The AT32UC3C has several types of power supply pins:

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

See "[Electrical Characteristics](#)" on page 50 for power consumption on the various supply pins.

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

6.1.2 Voltage Regulators

The AT32UC3C embeds two voltage regulators:

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

6.1.3 Regulators Connection

The AT32UC3C supports two power supply configurations.

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

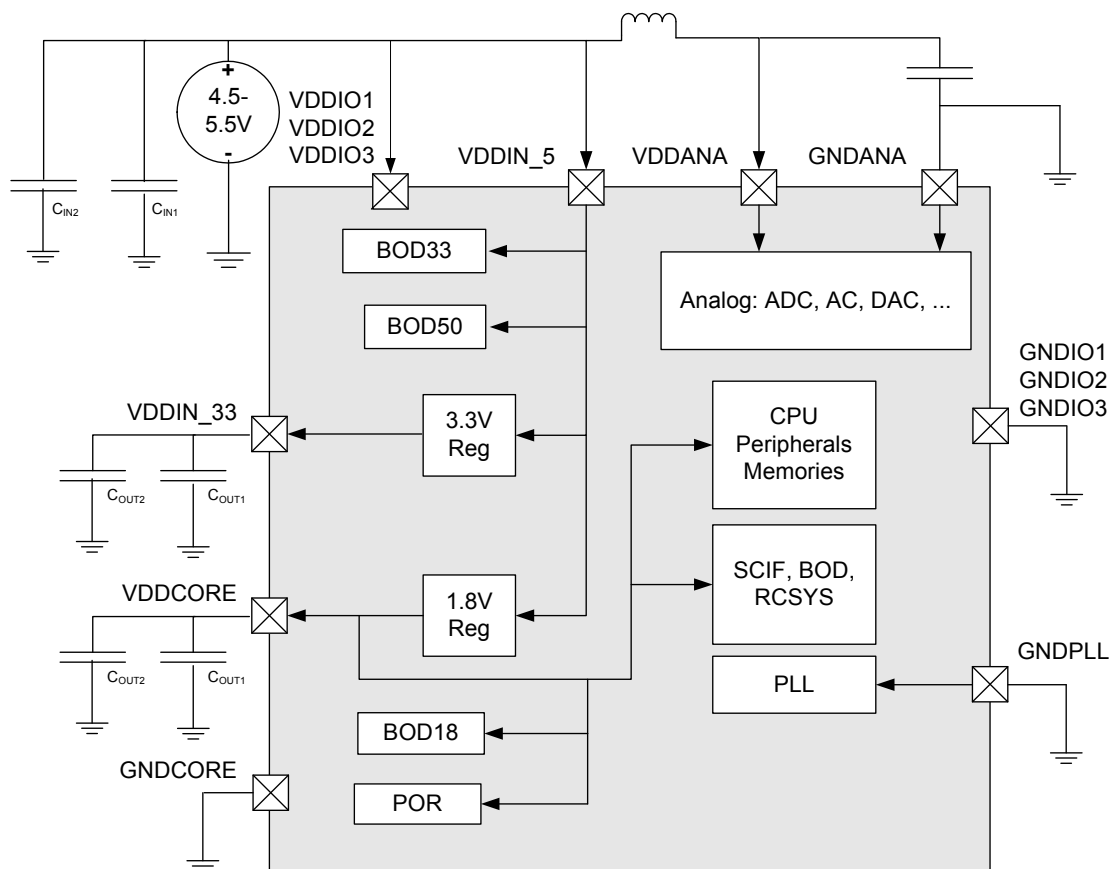
6.1.3.1 5V Single Supply Mode

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

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

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

Figure 6-1. 5V Single Power Supply mode



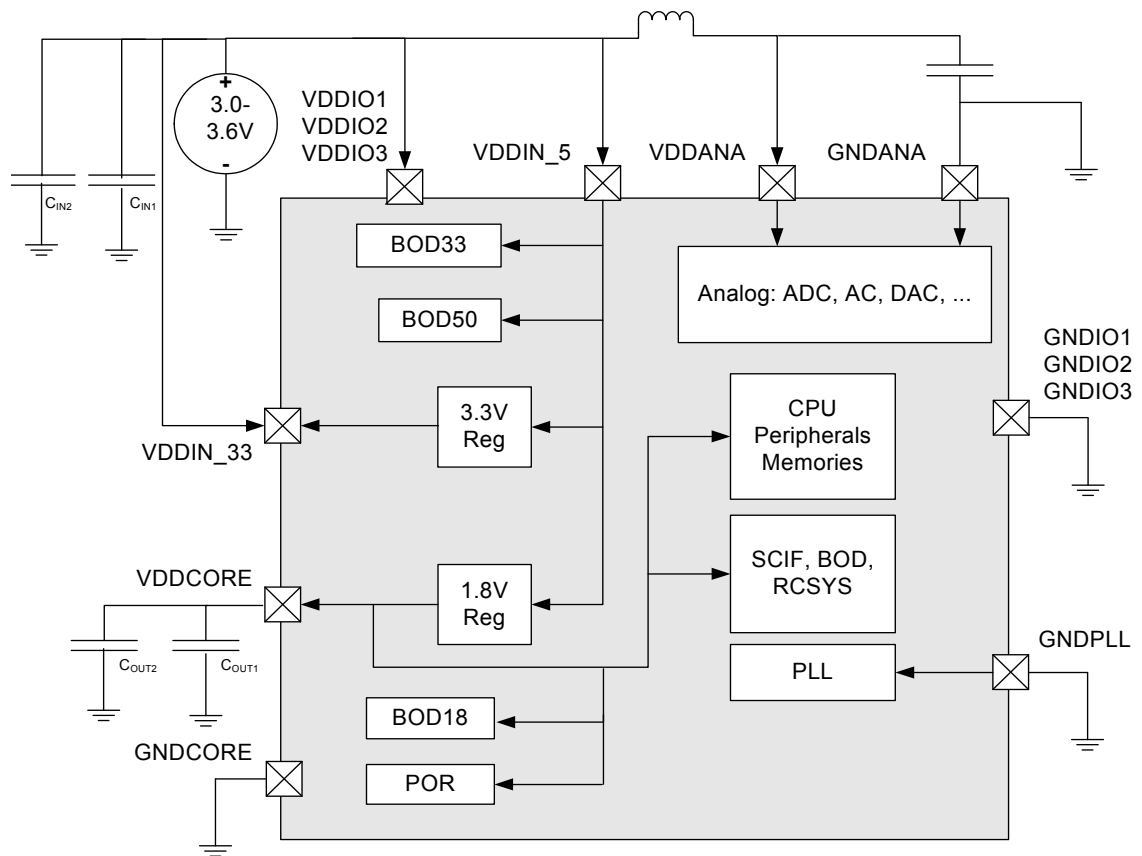
6.1.3.2 3.3V Single Supply Mode

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

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

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

Figure 6-2. 3 Single Power Supply Mode



6.1.4 Power-up Sequence

6.1.4.1 Maximum Rise Rate

To avoid risk of latch-up, the rise rate of the power supplies must not exceed the values described in [Table 7-2 on page 51](#).

Recommended order for power supplies is also described in this table.

6.1.4.2 Minimum Rise Rate

The integrated Power-Reset circuitry monitoring the powering supply requires a minimum rise rate for the VDDIN_5 power supply.

See [Table 7-2 on page 51](#) for the minimum rise rate value.

If the application can not ensure that the minimum rise rate condition for the VDDIN power supply is met, the following configuration can be used:

- A logic “0” value is applied during power-up on pin RESET_N until:
 - VDDIN_5 rises above 4.5V in 5V single supply mode.
 - VDDIN_33 rises above 3V in 3.3V single supply mode.

6.2 Startup Considerations

This chapter summarizes the boot sequence of the AT32UC3C. The behavior after power-up is controlled by the Power Manager. For specific details, refer to the Power Manager chapter.

6.2.1 Starting of clocks

At power-up, the BOD33 and the BOD18 are enabled. The device will be held in a reset state by the power-up circuitry, until the VDDIN_33 (resp. VDDCORE) has reached the reset threshold of the BOD33 (resp BOD18). Refer to the Electrical Characteristics for the BOD thresholds. Once the power has stabilized, the device will use the System RC Oscillator (RCSYS, 115KHz typical frequency) as clock source. The BOD18 and BOD33 are kept enabled or are disabled according to the fuse settings (See the Fuse Setting section in the Flash Controller chapter).

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 receive a clock with the same frequency as the internal RC Oscillator.

6.2.2 Fetching of initial instructions

After reset has been released, the AVR32UC CPU starts fetching instructions from the reset address, which is 0x8000_0000. This address points to the first address in the internal Flash.

The internal Flash uses VDDIO voltage during read and write operations. It is recommended to use the BOD33 to monitor this voltage and make sure the VDDIO is above the minimum level (3.0V).

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.

7. Electrical Characteristics

7.1 Absolute Maximum Ratings*

| | |
|---|--------------------------------|
| Operating temperature..... | -40°C to +85°C |
| Storage temperature..... | -60°C to +150°C |
| Voltage on any pin except DM/DP/VBUS with respect to ground | -0.3V to $V_{VDD}^{(1)}$ +0.3V |
| Voltage on DM/DP with respect to ground..... | -0.3V to +3.6V |
| Voltage on VBUS with respect to ground..... | -0.3V to +5.5V |
| Maximum operating voltage (VDDIN_5) | 5.5V |
| Maximum operating voltage (VDDIO1, VDDIO2, VDDIO3, VDDANA)..... | 5.5V |
| Maximum operating voltage (VDDIN_33) | 3.6V |
| Total DC output current on all I/O pins- VDDIO1 | 120 mA |
| Total DC output current on all I/O pins- VDDIO2 | 120 mA |
| Total DC output current on all I/O pins- VDDIO3 | 120 mA |
| Total DC output current on all I/O pins- VDDANA..... | 120 mA |

*NOTICE: Stresses beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Notes: 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.

7.2 Supply Characteristics

The following characteristics are applicable to the operating temperature range: $T_A = -40^\circ\text{C}$ to 85°C , unless otherwise specified and are valid for a junction temperature up to $T_J = 100^\circ\text{C}$. Please refer to [Section 6. "Supply and Startup Considerations" on page 46](#).

Table 7-1. Supply Characteristics

| Symbol | Parameter | Condition | Voltage | | Unit |
|--|--|------------------|---------|-----|------|
| | | | Min | Max | |
| V_{VDDIN_5} | DC supply internal regulators | 3V range | 3.0 | 3.6 | V |
| | | 5V range | 4.5 | 5.5 | |
| V_{VDDIN_33} | DC supply USB I/O | only in 3V range | 3.0 | 3.6 | V |
| V_{VDDANA} | DC supply peripheral I/O and analog part | 3V range | 3.0 | 3.6 | V |
| | | 5V range | 4.5 | 5.5 | |
| V_{VDDIO1} V_{VDDIO2} V_{VDDIO2} | DC supply peripheral I/O | 3V range | 3.0 | 3.6 | V |
| 5V range | | 4.5 | 5.5 | | |

Table 7-2. Supply Rise Rates and Order

| Symbol | Parameter | Rise Rate | | |
|---|--|-----------|-----------|---|
| | | Min | Max | Comment |
| V _{VDDIN_5} | DC supply internal 3.3V regulator | 0.01 V/ms | 1.25 V/us | |
| V _{VDDIN_33} | DC supply internal 1.8V regulator | 0.01 V/ms | 1.25 V/us | |
| V _{VDDIO1} V _{VDDIO2} V _{VDDIO3} | DC supply peripheral I/O | 0.01 V/ms | 1.25 V/us | Rise after or at the same time as VDDIN_5, VDDIN_33 |
| V _{VDDANA} | DC supply peripheral I/O and analog part | 0.01 V/ms | 1.25 V/us | Rise after or at the same time as VDDIN_5, VDDIN_33 |

7.3 Maximum Clock Frequencies

These parameters are given in the following conditions:

- V_{VDDCORE} > 1.85V
- Temperature = -40°C to 85°C

Table 7-3. Clock Frequencies

| Symbol | Parameter | Conditions | Min | Max | Units |
|---------------------|------------------------|-------------------------|-----|--------------------|-------|
| f _{CPU} | CPU clock frequency | | | 66 | MHz |
| f _{PBA} | PBA clock frequency | | | 66 | MHz |
| f _{PBB} | PBB clock frequency | | | 66 | MHz |
| f _{PBC} | PBC clock frequency | | | 66 | MHz |
| f _{GCLK0} | GCLK0 clock frequency | Generic clock for USBC | | 50 ⁽¹⁾ | MHz |
| f _{GCLK1} | GCLK1 clock frequency | Generic clock for CANIF | | 66 ⁽¹⁾ | MHz |
| f _{GCLK2} | GCLK2 clock frequency | Generic clock for AST | | 80 ⁽¹⁾ | MHz |
| f _{GCLK4} | GCLK4 clock frequency | Generic clock for PWM | | 133 ⁽¹⁾ | MHz |
| f _{GCLK11} | GCLK11 clock frequency | Generic clock for IISC | | 50 ⁽¹⁾ | MHz |

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.

7.4 Power Consumption

The values in [Table 7-4](#) are measured values of power consumption under the following conditions, except where noted:

- Operating conditions core supply ([Figure 7-1](#))
 - V_{VDDIN_5} = V_{VDDIN_33} = 3.3V
 - V_{VDDCORE} = 1.85V, supplied by the internal regulator
 - V_{VDDIO1} = V_{VDDIO2} = V_{VDDIO3} = 3.3V
 - V_{VDDANA} = 3.3V

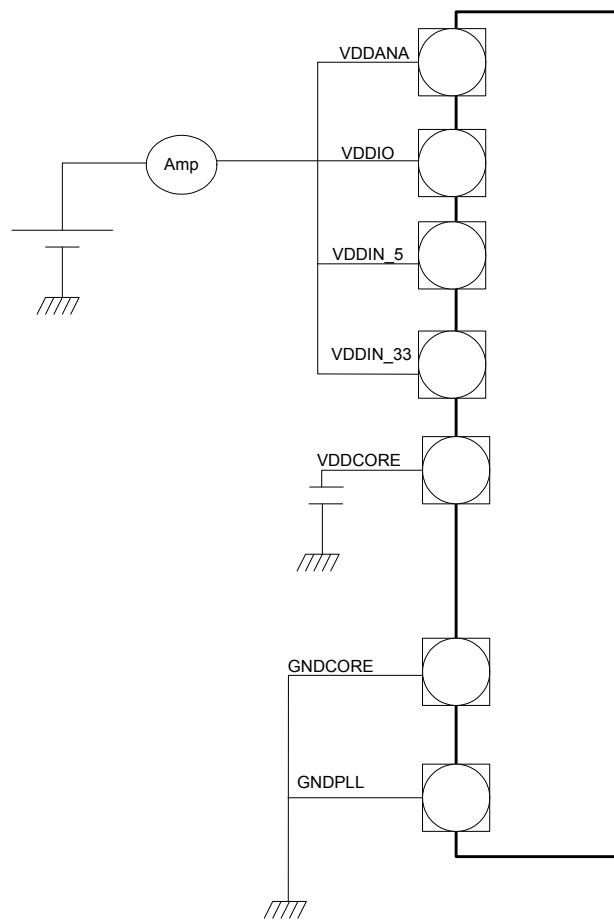
- Internal 3.3V regulator is off
- $T_A = 25^\circ\text{C}$
- I/Os are configured as inputs, with internal pull-up enabled.
- Oscillators
 - OSC0/1 (crystal oscillator) stopped
 - OSC32K (32KHz crystal oscillator) stopped
 - PLL0 running
 - PLL1 stopped
- Clocks
 - External clock on XIN0 as main clock source (10MHz)
 - CPU, HSB, and PBB clocks undivided
 - PBA, PBC clock divided by 4
 - All peripheral clocks running

Table 7-4. Power Consumption for Different Operating Modes

| Mode | Conditions | Measured on | Consumption Typ | Unit |
|------------------------|---|-------------|-----------------|--------------------------|
| Active ⁽¹⁾ | CPU running a recursive Fibonacci algorithm | Amp | 512 | $\mu\text{A}/\text{MHz}$ |
| Idle ⁽¹⁾ | | | 258 | |
| Frozen ⁽¹⁾ | | | 106 | |
| Standby ⁽¹⁾ | | | 48 | |
| Stop | | | 73 | μA |
| DeepStop | | | 43 | |
| Static | OSC32K and AST running | | 32 | |
| | AST and OSC32K stopped | | 31 | |

Note: 1. These numbers are valid for the measured condition only and must not be extrapolated to other frequencies.

Figure 7-1. Measurement Schematic



7.4.1 Peripheral Power Consumption

The values in [Table 7-5](#) are measured values of power consumption under the following conditions.

- Operating conditions core supply ([Figure 7-1](#))
 - $V_{VDDIN_5} = V_{VDDIN_33} = 3.3V$
 - $V_{VDDCORE} = 1.85V$, supplied by the internal regulator
 - $V_{VDDIO1} = V_{VDDIO2} = V_{VDDIO3} = 3.3V$
 - $V_{VDDANA} = 3.3V$
 - Internal 3.3V regulator is off.
- $T_A = 25^\circ C$
- I/Os are configured as inputs, with internal pull-up enabled.
- Oscillators
 - OSC0/1 (crystal oscillator) stopped
 - OSC32K (32KHz crystal oscillator) stopped
 - PLL0 running

- 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.

7.5 I/O Pin Characteristics

Table 7-6. Normal I/O Pin Characteristics⁽¹⁾

| Symbol | Parameter | Condition | Min | Typ | Max | Units | |
|-----------------------|---------------------------------|---|--|-----|------------------------|-------|-----|
| R _{PULLUP} | Pull-up resistance | V _{VDD} = 3V | 5 | | 26 | kOhm | |
| | | V _{VDD} = 5V | 5 | | 16 | kOhm | |
| R _{PULLDOWN} | Pull-down resistance | | 2 | | 16 | kOhm | |
| V _{IL} | Input low-level voltage | V _{VDD} = 3V | | | 0.3*V _{VDDIO} | V | |
| | | V _{VDD} = 4.5V | | | 0.3*V _{VDDIO} | | |
| V _{IH} | Input high-level voltage | V _{VDD} = 3.6V | 0.7*V _{VDDIO} | | | V | |
| | | V _{VDD} = 5.5V | 0.7*V _{VDDIO} | | | | |
| V _{OL} | Output low-level voltage | I _{OL} = -3.5mA, pin drive x1 ⁽²⁾ | | | 0.45 | V | |
| | | I _{OL} = -7mA, pin drive x2 ⁽²⁾ | | | | | |
| | | I _{OL} = -14mA, pin drive x4 ⁽²⁾ | | | | | |
| V _{OH} | Output high-level voltage | I _{OH} = 3.5mA, pin drive x1 ⁽²⁾ | V _{VDD} - 0.8 | | | V | |
| | | I _{OH} = 7mA, pin drive x2 ⁽²⁾ | | | | | |
| | | I _{OH} = 14mA, pin drive x4 ⁽²⁾ | | | | | |
| f _{MAX} | Output frequency ⁽³⁾ | V _{VDD} = 3.0V | load = 10pF, pin drive x1 ⁽²⁾ | | | 35 | MHz |
| | | | load = 10pF, pin drive x2 ⁽²⁾ | | | 55 | |
| | | | load = 10pF, pin drive x4 ⁽²⁾ | | | 70 | |
| | | | load = 30pF, pin drive x1 ⁽²⁾ | | | 15 | |
| | | | load = 30pF, pin drive x2 ⁽²⁾ | | | 30 | |
| | | | load = 30pF, pin drive x4 ⁽²⁾ | | | 45 | |
| | | V _{VDD} = 4.5V | load = 10pF, pin drive x1 ⁽²⁾ | | | 50 | |
| | | | load = 10pF, pin drive x2 ⁽²⁾ | | | 80 | |
| | | | load = 10pF, pin drive x4 ⁽²⁾ | | | 95 | |
| | | | load = 30pF, pin drive x1 ⁽²⁾ | | | 25 | |
| | | | load = 30pF, pin drive x2 ⁽²⁾ | | | 40 | |
| | | | load = 30pF, pin drive x4 ⁽²⁾ | | | 65 | |

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⁾ | | | 7.7 | ns | |
| | | | load = 10pF, pin drive x2 ⁽²⁾ | | | 3.4 | | |
| | | | load = 10pF, pin drive x4 ⁽²⁾ | | | 1.9 | | |
| | | | load = 30pF, pin drive x1 ⁽²⁾ | | | 16 | | |
| | | | load = 30pF, pin drive x2 ⁽²⁾ | | | 7.5 | | |
| | | | load = 30pF, pin drive x4 ⁽²⁾ | | | 3.8 | | |
| | | $V_{VDD} = 4.5V$ | load = 10pF, pin drive x1 ⁽²⁾ | | | | | 5.3 |
| | | | load = 10pF, pin drive x2 ⁽²⁾ | | | | | 2.4 |
| | | | load = 10pF, pin drive x4 ⁽²⁾ | | | | | 1.3 |
| | | | load = 30pF, pin drive x1 ⁽²⁾ | | | | | 11.1 |
| | | | load = 30pF, pin drive x2 ⁽²⁾ | | | | | 5.2 |
| | | | load = 30pF, pin drive x4 ⁽²⁾ | | | | | 2.7 |
| t_{FALL} | Fall time ⁽³⁾ | $V_{VDD} = 3.0V$ | load = 10pF, pin drive x1 ⁽²⁾ | | | 7.6 | ns | |
| | | | load = 10pF, pin drive x2 ⁽²⁾ | | | 3.5 | | |
| | | | load = 10pF, pin drive x4 ⁽²⁾ | | | 1.9 | | |
| | | | load = 30pF, pin drive x1 ⁽²⁾ | | | 15.8 | | |
| | | | load = 30pF, pin drive x2 ⁽²⁾ | | | 7.3 | | |
| | | | load = 30pF, pin drive x4 ⁽²⁾ | | | 3.8 | | |
| | | $V_{VDD} = 4.5V$ | load = 10pF, pin drive x1 ⁽²⁾ | | | | | 5.2 |
| | | | load = 10pF, pin drive x2 ⁽²⁾ | | | | | 2.4 |
| | | | load = 10pF, pin drive x4 ⁽²⁾ | | | | | 1.4 |
| | | | load = 30pF, pin drive x1 ⁽²⁾ | | | | | 10.9 |
| | | | load = 30pF, pin drive x2 ⁽²⁾ | | | | | 5.1 |
| | | | load = 30pF, pin drive x4 ⁽²⁾ | | | | | 2.7 |
| I_{LEAK} | Input leakage current | Pull-up resistors disabled | | | 1.0 | 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:
- 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.
 - 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.
 - 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.



7.6 Oscillator Characteristics

7.6.1 Oscillator (OSC0 and OSC1) Characteristics

7.6.1.1 Digital Clock Characteristics

The following table describes the characteristics for the oscillator when a digital clock is applied on XIN0 or XIN1.

Table 7-7. Digital Clock Characteristics

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|-------------|---------------------------|------------|------------------------|-----|------------------------|-------|
| f_{CPXIN} | XIN clock frequency | | | | 50 | MHz |
| t_{CPXIN} | XIN clock period | | 20 | | | ns |
| t_{CHXIN} | XIN clock high half-priod | | $0.4 \times t_{CPXIN}$ | | $0.6 \times t_{CPXIN}$ | ns |
| t_{CLXIN} | XIN clock low half-priod | | $0.4 \times t_{CPXIN}$ | | $0.6 \times t_{CPXIN}$ | ns |
| C_{IN} | XIN input capacitance | | | 2 | | pF |

7.6.1.2 Crystal Oscillator Characteristics

The following table describes the characteristics for the oscillator when a crystal is connected between XIN and XOUT as shown in [Figure 7-2](#). The user must choose a crystal oscillator where the crystal load capacitance C_L is within the range given in the table. The exact value of C_L can be found in the crystal datasheet. The capacitance of the external capacitors (C_{LEXT}) can then be computed as follows:

$$C_{LEXT} = 2(C_L - C_i) - C_{PCB}$$

where C_{PCB} is the capacitance of the PCB and C_i is the internal equivalent load capacitance.

Figure 7-2. Oscillator Connection

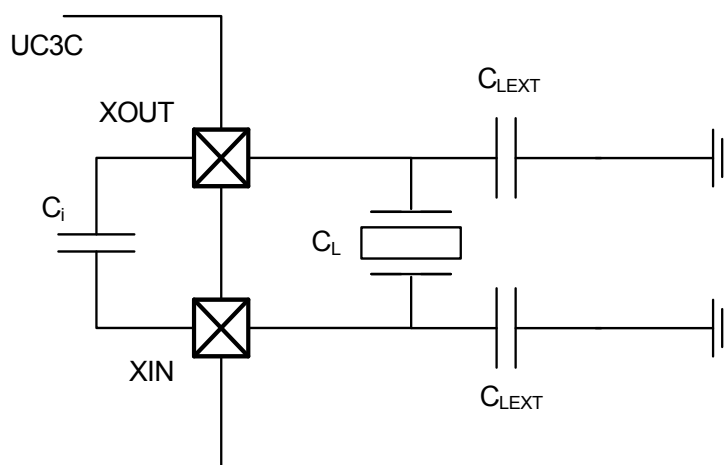


Table 7-8. Crystal Oscillator Characteristics

| Symbol | Parameter | Conditions | Min | Typ | Max | Unit |
|---------------|--------------------------------------|--|-----|------|-----|------|
| f_{OUT} | Crystal oscillator frequency | | 0.4 | | 20 | MHz |
| C_i | Internal equivalent load capacitance | | | 1.7 | | pF |
| $t_{STARTUP}$ | Startup time | $f_{OUT} = 8\text{MHz}$ SCIF.OSCCTRL.GAIN = 1 ⁽¹⁾ | | 975 | | us |
| | | $f_{OUT} = 16\text{MHz}$ SCIF.OSCCTRL.GAIN = 2 ⁽¹⁾ | | 1100 | | us |

Notes: 1. Please refer to the SCIF chapter for details.

7.6.2 32KHz Crystal Oscillator (OSC32K) Characteristics

7.6.2.1 Digital Clock Characteristics

The following table describes the characteristics for the oscillator when a digital clock is applied on XIN32.

Table 7-9. Digital 32KHz Clock Characteristics

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|-------------|-----------------------------|------------|------------------------|--------|------------------------|-------|
| f_{CPXIN} | XIN32 clock frequency | | | 32.768 | 5000 | KHz |
| t_{CPXIN} | XIN32 clock period | | 200 | | | ns |
| t_{CHXIN} | XIN32 clock high half-priod | | $0.4 \times t_{CPXIN}$ | | $0.6 \times t_{CPXIN}$ | ns |
| t_{CLXIN} | XIN32 clock low half-priod | | $0.4 \times t_{CPXIN}$ | | $0.6 \times t_{CPXIN}$ | ns |
| C_{IN} | XIN32 input capacitance | | | 2 | | pF |

7.6.2.2 Crystal Oscillator Characteristics

Figure 7-2 and the equation above also applies to the 32KHz oscillator connection. The user must choose a crystal oscillator where the crystal load capacitance C_L is within the range given in the table. The exact value of C_L can then be found in the crystal datasheet..

Table 7-10. 32 KHz Crystal Oscillator Characteristics

| Symbol | Parameter | Conditions | Min | Typ | Max | Unit |
|---------------|--------------------------------------|--|-----|--------|-----|------|
| f_{OUT} | Crystal oscillator frequency | | | 32 768 | | Hz |
| $t_{STARTUP}$ | Startup time | $R_S = 50 \text{ kOhm}, C_L = 12.5 \text{ pF}$ | | 2 | | s |
| C_L | Crystal load capacitance | | 6 | | 15 | pF |
| C_i | Internal equivalent load capacitance | | | 1.4 | | pF |

7.6.3 Phase Lock Loop (PLL0 and PLL1) Characteristics

Table 7-11. PLL Characteristics

| Symbol | Parameter | Conditions | Min | Typ | Max | Unit |
|---------------|---|--|-----|-----|-----|---------------|
| f_{VCO} | Output frequency | | 80 | | 240 | MHz |
| f_{IN} | Input frequency | | 4 | | 16 | MHz |
| I_{PLL} | Current consumption | Active mode, $f_{VCO} = 80\text{MHz}$ | | 250 | | μA |
| | | Active mode, $f_{VCO} = 240\text{MHz}$ | | 600 | | |
| $t_{STARTUP}$ | Startup time, from enabling the PLL until the PLL is locked | Wide Bandwidth mode disabled | | 15 | | μs |
| | | Wide Bandwidth mode enabled | | 45 | | |

7.6.4 120MHz RC Oscillator (RC120M) Characteristics

Table 7-12. Internal 120MHz RC Oscillator Characteristics

| Symbol | Parameter | Conditions | Min | Typ | Max | Unit |
|---------------|---------------------------------|------------|-----|------|-----|---------------|
| f_{OUT} | Output frequency ⁽¹⁾ | | 88 | 120 | 152 | MHz |
| I_{RC120M} | Current consumption | | | 1.85 | | mA |
| $t_{STARTUP}$ | Startup time | | | 3 | | μs |

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.

7.6.5 System RC Oscillator (RCSYS) Characteristics

Table 7-13. System RC Oscillator Characteristics

| Symbol | Parameter | Conditions | Min | Typ | Max | Unit |
|-----------|------------------|--|-----|-------|-----|------|
| f_{OUT} | Output frequency | Calibrated at $T_A = 85^\circ\text{C}$ | 110 | 115.2 | 120 | kHz |
| | | $T_A = 25^\circ\text{C}$ | 105 | 109 | 115 | |
| | | $T_A = -40^\circ\text{C}$ | 100 | 104 | 108 | |

7.6.6 8MHz/1MHz RC Oscillator (RC8M) Characteristics

Table 7-14. 8MHz/1MHz RC Oscillator Characteristics

| Symbol | Parameter | Conditions | Min | Typ | Max | Unit |
|---------------|------------------|--|-------|-----|-------|---------------|
| f_{OUT} | Output frequency | SCIF.RCCR8.FREQMODE = 0 ⁽¹⁾ | 7.6 | 8 | 8.4 | MHz |
| | | SCIF.RCCR8.FREQMODE = 1 ⁽¹⁾ | 0.955 | 1 | 1.045 | |
| $t_{STARTUP}$ | Startup time | | | | 20 | μs |

Notes: 1. Please refer to the SCIF chapter for details.

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 | 33MHz |
| 1 | 2 cycles | 66MHz |

Table 7-16. Flash Characteristics

| Symbol | Parameter | Conditions | Min | Typ | Max | Unit |
|-----------|-----------------------------------|--------------------------------|-----|-----|-----|------|
| t_{FPP} | Page programming time | $f_{CLK_HSB} = 66\text{MHz}$ | | 4.3 | | ms |
| t_{FPE} | Page erase time | | | 4.3 | | |
| t_{FFP} | Fuse programming time | | | 0.6 | | |
| t_{FEA} | Full chip erase time (EA) | | | 4.9 | | |
| t_{FCE} | JTAG chip erase time (CHIP_ERASE) | $f_{CLK_HSB} = 115\text{kHz}$ | | 640 | | |

Table 7-17. Flash Endurance and Data Retention

| Symbol | Parameter | Conditions | Min | Typ | Max | Unit |
|--------------|---|------------|------|-----|-----|--------|
| N_{FARRAY} | Array endurance (write/page) | | 100k | | | cycles |
| N_{FFUSE} | General Purpose fuses endurance (write/bit) | | 1k | | | cycles |
| t_{RET} | Data retention | | 15 | | | years |

7.8 Analog Characteristics

7.8.1 1.8V Voltage Regulator Characteristics

Table 7-18. 1.8V Voltage Regulator Electrical Characteristics

| Symbol | Parameter | Condition | Min | Typ | Max | Units |
|----------------------|----------------------------------|-----------|-----|------|-----|-------|
| V _{VDDIN_5} | Input voltage range | 5V range | 4.5 | | 5.5 | V |
| | | 3V range | 3.0 | | 3.6 | |
| V _{VDDCORE} | Output voltage, calibrated value | | | 1.85 | | V |
| I _{OUT} | DC output current | | | | 80 | mA |

Table 7-19. 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 |

7.8.2 3.3V Voltage Regulator Characteristics

Table 7-20. 3.3V Voltage Regulator Electrical Characteristics

| Symbol | Parameter | Condition | Min | Typ | Max | Units |
|-----------------------|----------------------------------|----------------|-----|-----|-----|-------|
| V _{VDDIN_5} | Input voltage range | | 4.5 | | 5.5 | V |
| V _{VDDIN_33} | Output voltage, calibrated value | | | 3.4 | | V |
| I _{OUT} | DC output current | | | | 35 | mA |
| I _{VREG} | Static current of regulator | Low power mode | | 10 | | μA |

7.8.3 1.8V Brown Out Detector (BOD18) Characteristics

The values in [Table 7-21](#) describe the values of the BOD.LEVEL in the SCIF module.

Table 7-21. BODLEVEL Values

| BODLEVEL Value | Parameter | Min | Max | Units |
|----------------|--------------------------------|------|------|-------|
| 0 | | 1.29 | 1.58 | V |
| 20 | | 1.36 | 1.63 | |
| 26 | threshold at power-up sequence | 1.42 | 1.69 | |
| 28 | | 1.43 | 1.72 | |
| 32 | | 1.48 | 1.77 | |
| 36 | | 1.53 | 1.82 | |
| 40 | | 1.56 | 1.88 | |

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 | | 2.21 | 2.55 | V |
| 22 | | 2.30 | 2.64 | |
| 27 | | 2.39 | 2.74 | |
| 31 | threshold at power-up sequence | 2.46 | 2.82 | |
| 33 | | 2.50 | 2.86 | |
| 39 | | 2.60 | 2.98 | |
| 44 | | 2.69 | 3.08 | |
| 49 | | 2.78 | 3.18 | |
| 53 | | 2.85 | 3.27 | |
| 60 | | 2.98 | 3.41 | |

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 | | 3.20 | 3.65 | V |
| 25 | | 3.42 | 3.92 | |
| 35 | | 3.68 | 4.22 | |
| 44 | | 3.91 | 4.48 | |
| 53 | | 4.15 | 4.74 | |
| 61 | | 4.36 | 4.97 | |

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 |
|---------------|-------------------------------|---|-----|------------------|------------------|-------|
| $V_{ADCREFO}$ | ADCREFO 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 applied 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 | k Ω |
| | | ADC used with S/H | | | 4.6 | |

Figure 7-3. ADC input

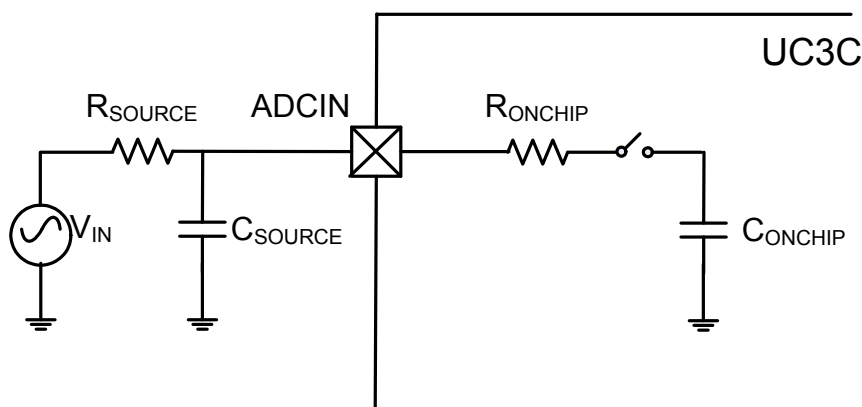


Table 7-31. ADC Transfer Characteristics 12-bit Resolution Mode⁽¹⁾

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|--------|----------------------------|---------------------------|-----|-----|-----|-------|
| RES | Resolution | Differential mode, | | | 12 | Bit |
| INL | Integral Non-Linearity | $V_{VDDANA} = 3V,$ | | | 5 | LSB |
| DNL | Differential Non-Linearity | $V_{ADCREFO} = 1V,$ | | | 3 | LSB |
| | Offset error | $ADCFIA.SEQCFGn.SRES = 0$ | -7 | | 7 | mV |
| | Gain error | $(F_{adc} = 1.2MHz)$ | -20 | | 20 | mV |

Table 7-31. ADC Transfer Characteristics (Continued) 12-bit Resolution Mode⁽¹⁾

| Symbol | Parameter | Conditions | Min | Typ | Max | Units | |
|--------|----------------------------|--|-----|-----|-----|-------|----|
| RES | Resolution | Differential mode, $V_{VDDANA} = 5V$, $V_{ADCREFO} = 3V$, ADCFIA.SEQCFGn.SRES = 0 ($F_{adc} = 1.5MHz$) | | | 12 | Bit | |
| INL | Integral Non-Linearity | | | | 4 | LSB | |
| DNL | Differential Non-Linearity | | | | 3 | LSB | |
| | Offset error | | | -15 | | 15 | mV |
| | Gain error | | | -25 | | 25 | mV |

Note: 1. The measures are done without any I/O activity on VDDANA/GNDANA power domain.

Table 7-32. ADC Transfer Characteristics 10-bit Resolution Mode⁽¹⁾

| Symbol | Parameter | Conditions | Min | Typ | Max | Units | |
|--------|----------------------------|--|-----|-----|------|-------|----|
| RES | Resolution | Differential mode, $V_{VDDANA} = 3V$, $V_{ADCREFO} = 1V$, ADCFIA.SEQCFGn.SRES = 1 ($F_{adc} = 1.5MHz$) | | | 10 | Bit | |
| INL | Integral Non-Linearity | | | | 1.25 | LSB | |
| DNL | Differential Non-Linearity | | | | 1 | LSB | |
| | Offset error | | | -10 | | 10 | mV |
| | Gain error | | | -20 | | 20 | mV |
| RES | Resolution | Differential mode, $V_{VDDANA} = 5V$, $V_{ADCREFO} = 3V$, ADCFIA.SEQCFGn.SRES = 1 ($F_{adc} = 1.5MHz$) | | | 10 | Bit | |
| INL | Integral Non-Linearity | | | | 1.25 | LSB | |
| DNL | Differential Non-Linearity | | | | 1 | LSB | |
| | Offset error | | | -15 | | 15 | mV |
| | Gain error | | | -20 | | 20 | mV |

Note: 1. The measures are done without any I/O activity on VDDANA/GNDANA power domain.

Table 7-33. ADC Transfer Characteristics 8-bit Resolution Mode⁽¹⁾

| Symbol | Parameter | Conditions | Min | Typ | Max | Units | |
|--------|----------------------------|--|-----|-----|------|-------|----|
| RES | Resolution | Differential mode, $V_{VDDANA} = 3V$, $V_{ADCREFO} = 1V$, ADCFIA.SEQCFGn.SRES = 2 ($F_{adc} = 1.5MHz$) | | | 8 | Bit | |
| INL | Integral Non-Linearity | | | | 0.3 | LSB | |
| DNL | Differential Non-Linearity | | | | 0.25 | LSB | |
| | Offset error | | | -10 | | 10 | mV |
| | Gain error | | | -20 | | 20 | mV |
| RES | Resolution | Differential mode, $V_{VDDANA} = 5V$, $V_{ADCREFO} = 3V$, ADCFIA.SEQCFGn.SRES = 2 ($F_{adc} = 1.5MHz$) | | | 8 | Bit | |
| INL | Integral Non-Linearity | | | | 0.2 | LSB | |
| DNL | Differential Non-Linearity | | | | 0.2 | LSB | |
| | Offset error | | | -20 | | 20 | mV |
| | Gain error | | | -20 | | 20 | mV |

Note: 1. The measures are done without any I/O activity on VDDANA/GNDANA power domain.

Table 7-34. ADC and S/H Transfer Characteristics 12-bit Resolution Mode and S/H gain = 1⁽¹⁾

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|--------|----------------------------|--|-----|-----|-----|-------|
| RES | Resolution | Differential mode, | | | 12 | Bit |
| INL | Integral Non-Linearity | $V_{VDDANA} = 3V,$ | | | 5 | LSB |
| DNL | Differential Non-Linearity | $V_{ADCREFO} = 1V,$ | | | 4 | LSB |
| | Offset error | ADCFIA.SEQCFGn.SRES = 0, S/H gain = 1 | -5 | | 5 | mV |
| | Gain error | ($F_{adc} = 1.2MHz$) | -20 | | 20 | mV |
| RES | Resolution | Differential mode, | | | 12 | Bit |
| INL | Integral Non-Linearity | $V_{VDDANA} = 5V,$ | | | 5 | LSB |
| DNL | Differential Non-Linearity | $V_{ADCREFO} = 3V,$ | | | 3 | LSB |
| | Offset error | ADCFIA.SEQCFGn.SRES = 0, S/H gain = 1 | -10 | | 10 | mV |
| | Gain error | ($F_{adc} = 1.5MHz$) | -20 | | 20 | mV |

Note: 1. The measures are done without any I/O activity on VDDANA/GNDANA power domain.

Table 7-35. ADC and S/H Transfer Characteristics 12-bit Resolution Mode and S/H gain from 1 to 8⁽¹⁾

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|--------|----------------------------|--|-----|-----|-----|-------|
| RES | Resolution | Differential mode, | | | 12 | Bit |
| INL | Integral Non-Linearity | $V_{VDDANA} = 3V,$ | | | 25 | LSB |
| DNL | Differential Non-Linearity | $V_{ADCREFO} = 1V,$ | | | 25 | LSB |
| | Offset error | ADCFIA.SEQCFGn.SRES = 0, S/H gain from 1 to 8 | -10 | | 10 | mV |
| | Gain error | ($F_{adc} = 1.2MHz$) | -20 | | 20 | mV |
| RES | Resolution | Differential mode, | | | 12 | Bit |
| INL | Integral Non-Linearity | $V_{VDDANA} = 5V,$ | | | 9 | LSB |
| DNL | Differential Non-Linearity | $V_{ADCREFO} = 3V,$ | | | 10 | LSB |
| | Offset error | ADCFIA.SEQCFGn.SRES = 0, S/H gain from 1 to 8 | -15 | | 15 | mV |
| | Gain error | ($F_{adc} = 1.5MHz$) | -20 | | 20 | mV |

Note: 1. The measures are done without any I/O activity on VDDANA/GNDANA power domain

Table 7-36. ADC and S/H Transfer Characteristics 10-bit Resolution Mode and S/H gain from 1 to 16⁽¹⁾

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|--------|----------------------------|---|-----|-----|-----|-------|
| RES | Resolution | Differential mode, | | | 10 | Bit |
| INL | Integral Non-Linearity | $V_{VDDANA} = 3V,$ | | | 3 | LSB |
| DNL | Differential Non-Linearity | $V_{ADCREFO} = 1V,$ | | | 3 | LSB |
| | Offset error | ADCFIA.SEQCFGn.SRES = 1, S/H gain from 1 to 16 | -15 | | 15 | mV |
| | Gain error | ($F_{adc} = 1.5MHz$) | -20 | | 20 | 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, $V_{VDDANA} = 5V$, $V_{ADCREFO} = 3V$, ADCFIA.SEQCFGn.SRES = 1, S/H gain from 1 to 16 ($F_{adc} = 1.5MHz$) | | | 10 | Bit | |
| INL | Integral Non-Linearity | | | | 1.5 | LSB | |
| DNL | Differential Non-Linearity | | | | 1.5 | LSB | |
| | Offset error | | | -25 | | 25 | mV |
| | Gain error | | | -15 | | 15 | 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 resistance | | 2 | | | k Ω |

Figure 7-4. DAC output

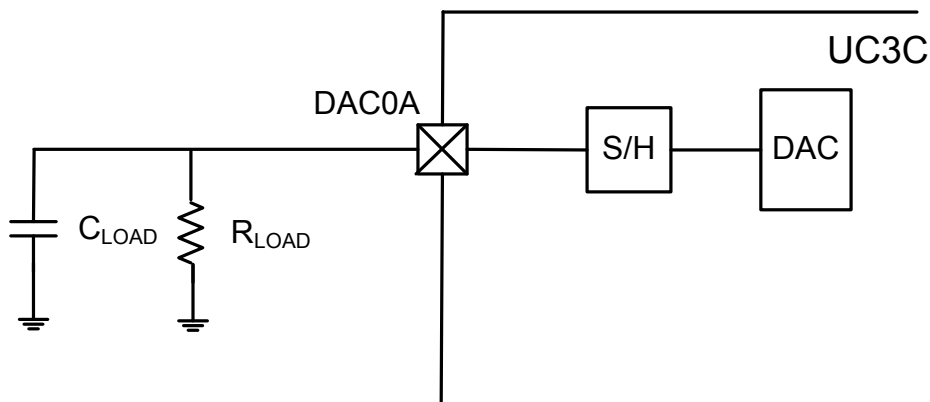


Table 7-40. Transfer Characteristics⁽¹⁾

| Symbol | Parameter | Conditions | Min | Typ | Max | Units | |
|--------|----------------------------|---|-----|-----|-----|-------|----|
| RES | Resolution | $V_{VDDANA} = 3V,$ $V_{DACREF} = 2V,$ One S/H | | | 12 | Bit | |
| INL | Integral Non-Linearity | | | 8 | | LSB | |
| DNL | Differential Non-linearity | | | | 6 | LSB | |
| | Offset error | | | -30 | | 30 | mV |
| | Gain error | | | -30 | | 30 | mV |
| RES | Resolution | $V_{VDDANA} = 5V,$ $V_{DACREF} = 3V,$ One S/H | | | 12 | Bit | |
| INL | Integral Non-Linearity | | | 12 | | LSB | |
| DNL | Differential Non-linearity | | | | 6 | 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.

7.8.8 Analog Comparator Characteristics

Table 7-41. Analog Comparator Characteristics⁽¹⁾

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|---------------|------------------------------|----------------------------------|-----|-----|--------------|-------|
| | Positive input voltage range | | 0 | | V_{VDDANA} | V |
| | Negative input voltage range | | 0 | | V_{VDDANA} | V |
| V_{OFFSET} | Offset | No hysteresis, Low Power mode | -29 | | 29 | mV |
| | | No hysteresis, High Speed mode | -16 | | 16 | mV |
| V_{HYST} | Hysteresis | Low hysteresis, Low Power mode | 7 | | 44 | mV |
| | | Low hysteresis, High Speed mode | 5 | | 34 | |
| | | High hysteresis, Low Power mode | 16 | | 102 | mV |
| | | High hysteresis, High Speed mode | 12 | | 69 | |
| t_{DELAY} | Propagation delay | Low Power mode | | | 2.9 | us |
| | | High Speed mode | | | 0.096 | |
| $t_{STARTUP}$ | Start-up time | | | | 20 | μs |

Note: 1. The measures are done without any I/O activity on VDDANA/GNDANA power domain.

Table 7-42. VDDANA scaled reference

| Symbol | Parameter | Min | Typ | Max | Units |
|---------------------|-------------------------------|-----|--------------------------------|-----|-------|
| SCF | ACIFA.SCFi.SCF range | 0 | | 32 | |
| V_{VDDANA} scaled | | | $(64 - SCF) * V_{VDDANA} / 65$ | | V |
| | V_{VDDANA} voltage accuracy | | | 3.2 | % |

7.8.9 USB Transceiver Characteristics

7.8.9.1 Electrical Characteristics

Table 7-43. Electrical Parameters

| Symbol | Parameter | Conditions | Min. | Typ. | Max. | Unit |
|-----------|--|--------------------------------------|------|------|------|------|
| R_{EXT} | Recommended external USB series resistor | In series with each USB pin with ±5% | | 39 | | Ω |

The USB on-chip buffers comply with the Universal Serial Bus (USB) v2.0 standard. All AC parameters related to these buffers can be found within the USB 2.0 electrical specifications.

7.9 Timing Characteristics

7.9.1 Startup, Reset, and Wake-up Timing

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

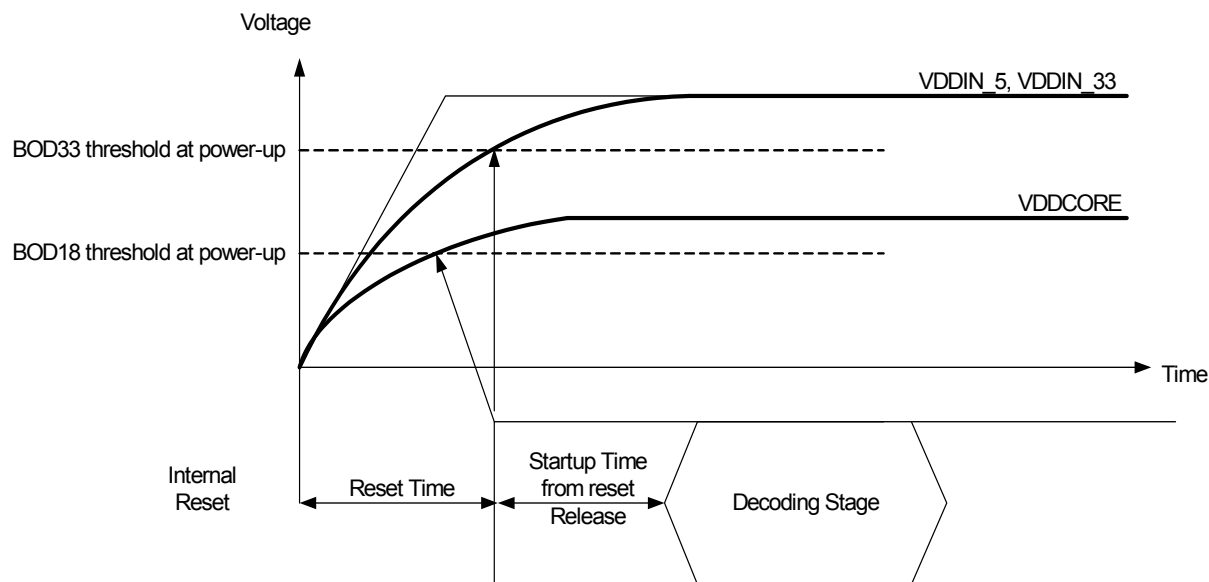
$$t = t_{CONST} + N_{CPU} \times t_{CPU}$$

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

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

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

Figure 7-5. Startup and Reset Time



7.9.2 RESET_N characteristics

Table 7-45. RESET_N Clock Waveform Parameters

| Symbol | Parameter | Condition | Min. | Typ. | Max. | Units |
|--------------------|------------------------------|-----------|------------------------|------|------|--------------|
| t _{RESET} | RESET_N minimum pulse length | | 2 * T _{RCSYS} | | | clock cycles |

7.9.3 USART in SPI Mode Timing

7.9.3.1 Master mode

Figure 7-6. USART in SPI Master Mode With (CPOL= CPHA= 0) or (CPOL= CPHA= 1)

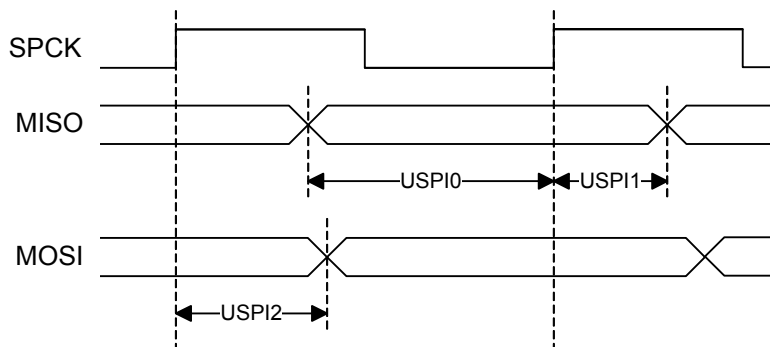


Figure 7-7. USART in SPI Master Mode With (CPOL= 0 and CPHA= 1) or (CPOL= 1 and CPHA= 0)

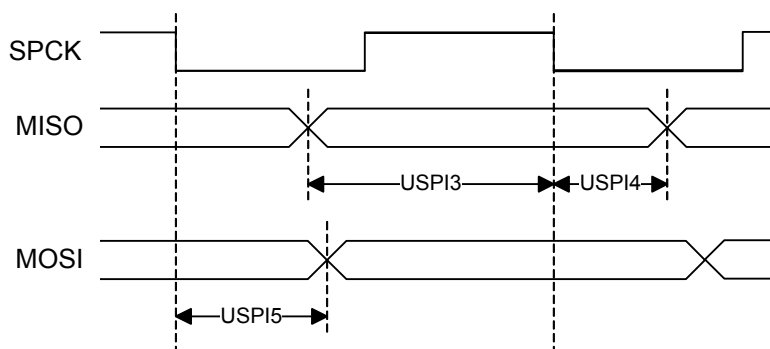


Table 7-46. USART in SPI Mode Timing, Master Mode⁽¹⁾

| Symbol | Parameter | Conditions | Min | Max | Units |
|--------|-----------------------------------|---------------------------|------------------------|------|-------|
| USPI0 | MISO setup time before SPCK rises | external capacitor = 40pF | 26+ $t_{SAMPLE}^{(2)}$ | | ns |
| USPI1 | MISO hold time after SPCK rises | | 0 | | ns |
| USPI2 | SPCK rising to MOSI delay | | | 11 | ns |
| USPI3 | MISO setup time before SPCK falls | | 26+ $t_{SAMPLE}^{(2)}$ | | ns |
| USPI4 | MISO hold time after SPCK falls | | 0 | | ns |
| USPI5 | SPCK falling to MOSI delay | | | 11.5 | 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:
$$t_{SAMPLE} = t_{SPCK} - \left(\left\lfloor \frac{t_{SPCK}}{2 \times t_{CLKUSART}} \right\rfloor \times t_{CLKUSART} \right)$$

Maximum SPI Frequency, Master Output

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

$$f_{SPCKMAX} = \text{MIN}(f_{PINMAX}, \frac{1}{SPI_{in}}, \frac{f_{CLKSPI} \times 2}{9})$$

Where SPI_{in} is the MOSI delay, USPI2 or USPI5 depending on CPOL and NCPHA. 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. f_{CLKSPI} is the maximum frequency of the CLK_SPI. Refer to the SPI chapter for a description of this clock.

Maximum SPI Frequency, Master Input

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

$$f_{SPCKMAX} = \text{MIN}(\frac{1}{SPI_{in} + t_{VALID}}, \frac{f_{CLKSPI} \times 2}{9})$$

Where SPI_{in} is the MISO setup and hold time, USPI0 + USPI1 or USPI3 + USPI4 depending on CPOL and NCPHA. T_{VALID} is the SPI slave response time. Please refer to the SPI slave datasheet for $T_{VALID} \cdot f_{CLKSPI}$ is the maximum frequency of the CLK_SPI. Refer to the SPI chapter for a description of this clock.

7.9.3.2 Slave mode

Figure 7-8. USART in SPI Slave Mode With (CPOL= 0 and CPHA= 1) or (CPOL= 1 and CPHA= 0)

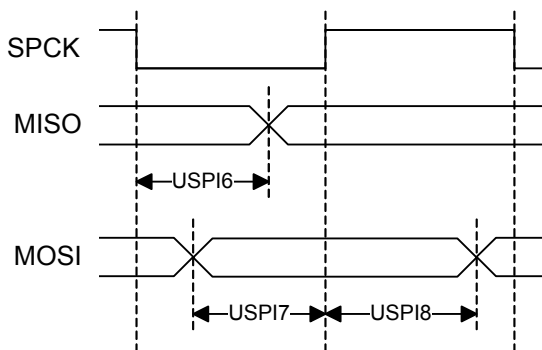


Figure 7-9. USART in SPI Slave Mode With (CPOL= CPHA= 0) or (CPOL= CPHA= 1)

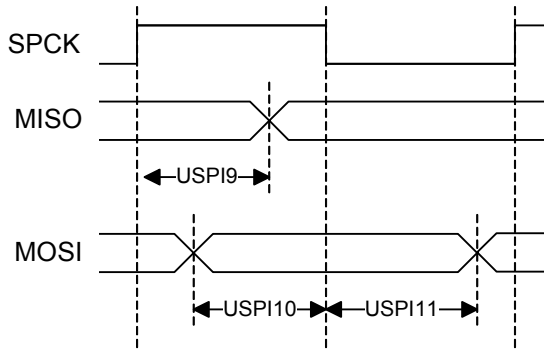


Figure 7-10. USART in SPI Slave Mode NPCS Timing

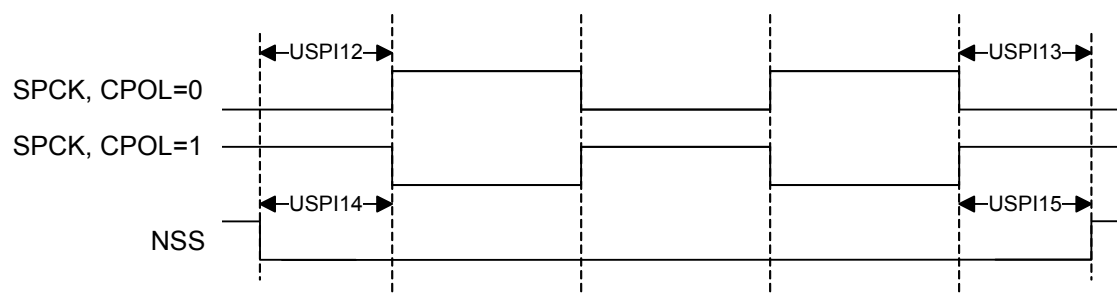


Table 7-47. USART in SPI mode Timing, Slave Mode⁽¹⁾

| Symbol | Parameter | Conditions | Min | Max | Units |
|--------|-----------------------------------|---------------------------|-------------------------------------|-----|-------|
| USPI6 | SPCK falling to MISO delay | external capacitor = 40pF | | 27 | 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 | | | 28 | 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 | | 33 | | ns |
| USPI13 | NSS hold time after SPCK falls | | 0 | | ns |
| USPI14 | NSS setup time before SPCK falls | | 33 | | 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: $t_{SAMPLE} = t_{SPCK} - \left(\frac{t_{SPCK}}{2 \times t_{CLK_USART}} \right) + \frac{1}{2} \times t_{CLK_USART}$

Maximum SPI Frequency, Slave Input Mode

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

$$f_{SPCKMAX} = \text{MIN}\left(\frac{f_{CLKSPI} \times 2}{9}, \frac{1}{SPI_{In}}\right)$$

Where SPI_{In} 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:

$$f_{SPCKMAX} = \text{MIN}\left(\frac{f_{CLKSPI} \times 2}{9}, f_{PINMAX}, \frac{1}{SPI_{In} + t_{SETUP}}\right)$$

Where SPI_{In} 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 $T_{SETUP} \cdot f_{CLKSPI}$ 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)

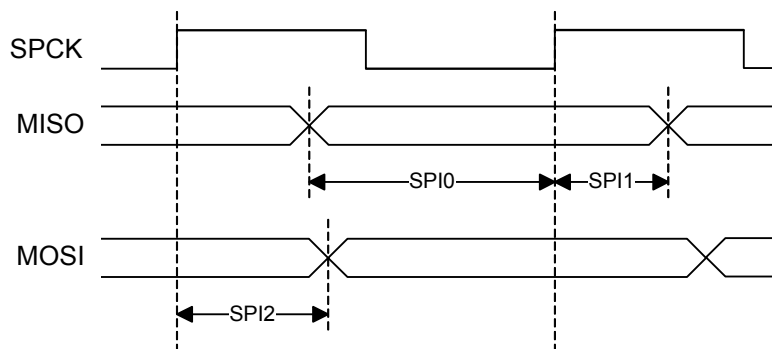


Figure 7-12. SPI Master Mode With (CPOL= 0 and NCPHA= 1) or (CPOL= 1 and NCPHA= 0)

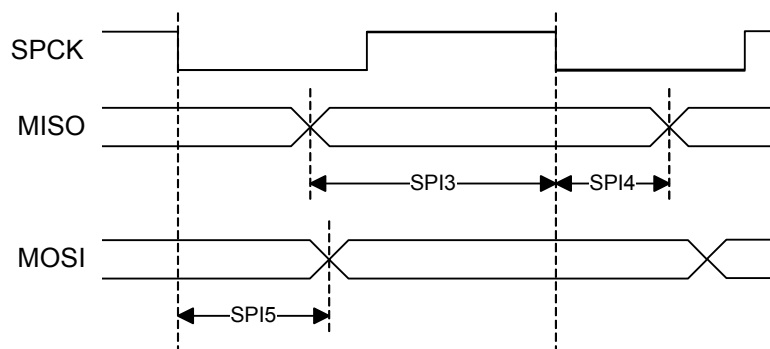


Table 7-48. SPI Timing, Master Mode⁽¹⁾

| Symbol | Parameter | Conditions | Min | Max | Units |
|--------|-----------------------------------|---------------------------|----------------------------------|------|-------|
| SPI0 | MISO setup time before SPCK rises | external capacitor = 40pF | 28.5 + (t _{CLK_SPI})/2 | | ns |
| SPI1 | MISO hold time after SPCK rises | | 0 | | ns |
| SPI2 | SPCK rising to MOSI delay | | | 10.5 | ns |
| SPI3 | MISO setup time before SPCK falls | | 28.5 + (t _{CLK_SPI})/2 | | ns |
| SPI4 | MISO hold time after SPCK falls | | 0 | | ns |
| SPI5 | SPCK falling to MOSI delay | | | 10.5 | 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.

Maximum SPI Frequency, Master Output

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

$$f_{SPCKMAX} = \text{MIN}(f_{PINMAX}, \frac{1}{SPI_n})$$

Where SPI_n is the MOSI delay, SPI2 or SPI5 depending on CPOL and NCPHA. 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.

Maximum SPI Frequency, Master Input

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

$$f_{SPCKMAX} = \frac{1}{SPI_n + t_{VALID}}$$

Where SPI_n is the MISO setup and hold time, SPI0 + SPI1 or SPI3 + SPI4 depending on CPOL and NCPHA. t_{VALID} is the SPI slave response time. Please refer to the SPI slave datasheet for t_{VALID} .

7.9.4.2 Slave mode

Figure 7-13. SPI Slave Mode With (CPOL= 0 and NCPHA= 1) or (CPOL= 1 and NCPHA= 0)

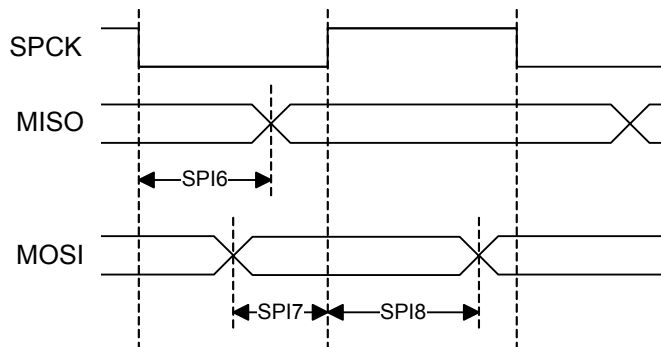


Figure 7-14. SPI Slave Mode With (CPOL= NCPHA= 0) or (CPOL= NCPHA= 1)

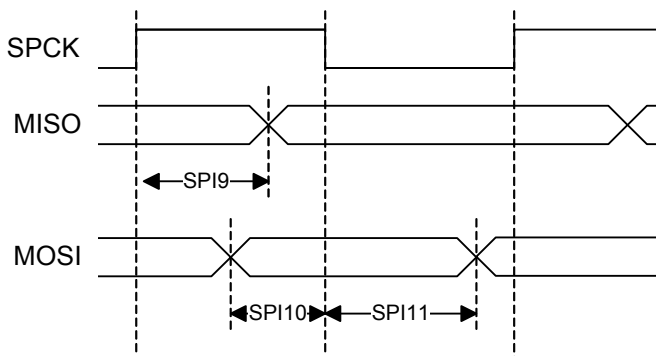


Figure 7-15. SPI Slave Mode NPCS Timing

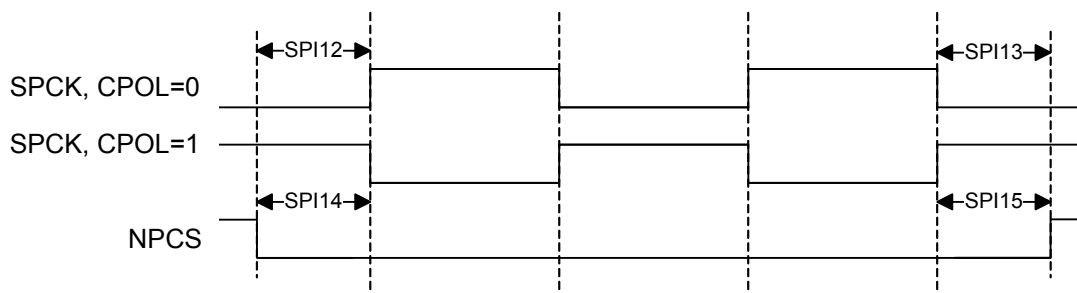


Table 7-49. SPI Timing, Slave Mode⁽¹⁾

| Symbol | Parameter | Conditions | Min | Max | Units |
|--------|-----------------------------------|---------------------------------|-----|-----|-------|
| SPI6 | SPCK falling to MISO delay | external capacitor = 40pF | | 29 | ns |
| SPI7 | MOSI setup time before SPCK rises | | 0 | | ns |
| SPI8 | MOSI hold time after SPCK rises | | 6.5 | | ns |
| SPI9 | SPCK rising to MISO delay | | | 30 | ns |
| SPI10 | MOSI setup time before SPCK falls | | 0 | | ns |
| SPI11 | MOSI hold time after SPCK falls | | 5 | | ns |
| SPI12 | NPCS setup time before SPCK rises | | 0 | | ns |
| SPI13 | NPCS hold time after SPCK falls | | 1.5 | | ns |
| SPI14 | NPCS setup time before SPCK falls | | 0 | | ns |
| SPI15 | NPCS hold time after SPCK rises | | 1.5 | | 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.

Maximum SPI Frequency, Slave Input Mode

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

$$f_{SPCKMAX} = \text{MIN}(f_{CLKSPI}, \frac{1}{SPI_{In}})$$

Where SPI_{In} is the MOSI setup and hold time, SPI7 + SPI8 or SPI10 + SPI11 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:

$$f_{SPCKMAX} = \text{MIN}(f_{PINMAX}, \frac{1}{SPI_{In} + t_{SETUP}})$$

Where SPI_{In} is the MISO delay, SPI6 or SPI9 depending on CPOL and NCPHA. t_{SETUP} is the SPI master setup time. Please refer to the SPI masterdatasheet for t_{SETUP} . 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.

7.9.5 TWIM/TWIS Timing

Figure 7-50 shows the TWI-bus timing requirements and the compliance of the device with them. Some of these requirements (t_r and t_f) are met by the device without requiring user intervention. Compliance with the other requirements (t_{HD-STA} , t_{SU-STA} , t_{SU-STO} , t_{HD-DAT} , $t_{SU-DAT-I2C}$, $t_{LOW-I2C}$, t_{HIGH} , and f_{TWCK}) requires user intervention through appropriate programming of the relevant

TWIM and TWIS user interface registers. Please refer to the TWIM and TWIS sections for more information.

Table 7-50. TWI-Bus Timing Requirements

| Symbol | Parameter | Mode | Minimum | | Maximum | | Unit |
|-------------------------|------------------------------|-------------------------|-------------------------|---------------------|-------------|-------------------------|------|
| | | | Requirement | Device | Requirement | Device | |
| t _r | TWCK and TWD rise time | Standard ⁽¹⁾ | - | | 1000 | | ns |
| | | Fast ⁽¹⁾ | 20 + 0.1 C _b | | 300 | | |
| t _f | TWCK and TWD fall time | Standard ⁽¹⁾ | - | | 300 | | ns |
| | | Fast ⁽¹⁾ | 20 + 0.1 C _b | | 300 | | |
| t _{HD-STA} | (Repeated) START hold time | Standard ⁽¹⁾ | 4.0 | t _{clkpb} | - | | μs |
| | | Fast ⁽¹⁾ | 0.6 | | | | |
| t _{SU-STA} | (Repeated) START set-up time | Standard ⁽¹⁾ | 4.7 | t _{clkpb} | - | | μs |
| | | Fast ⁽¹⁾ | 0.6 | | | | |
| t _{SU-STO} | STOP set-up time | Standard ⁽¹⁾ | 4.0 | 4t _{clkpb} | - | | μs |
| | | Fast ⁽¹⁾ | 0.6 | | | | |
| t _{HD-DAT} | Data hold time | Standard ⁽¹⁾ | 0.3 ⁽²⁾ | 2t _{clkpb} | 3.45 | ?? | μs |
| | | Fast ⁽¹⁾ | | | 0.9 | | |
| t _{SU-DAT-I2C} | Data set-up time | Standard ⁽¹⁾ | 250 | 2t _{clkpb} | - | | ns |
| | | Fast ⁽¹⁾ | 100 | | | | |
| t _{SU-DAT} | | - | - | t _{clkpb} | - | | - |
| t _{LOW-I2C} | TWCK LOW period | Standard ⁽¹⁾ | 4.7 | 4t _{clkpb} | - | | μs |
| | | Fast ⁽¹⁾ | 1.3 | | | | |
| t _{LOW} | | - | - | t _{clkpb} | - | | - |
| t _{HIGH} | TWCK HIGH period | Standard ⁽¹⁾ | 4.0 | 8t _{clkpb} | - | | μs |
| | | Fast ⁽¹⁾ | 0.6 | | | | |
| f _{TWCK} | TWCK frequency | Standard ⁽¹⁾ | - | | 100 | $\frac{1}{12t_{clkpb}}$ | kHz |
| | | Fast ⁽¹⁾ | | | 400 | | |

- Notes:
- Standard mode: f_{TWCK} ≤ 100 kHz ; fast mode: f_{TWCK} > 100 kHz .
 - A device must internally provide a hold time of at least 300 ns for TWD with reference to the falling edge of TWCK.

Notations:

C_b = total capacitance of one bus line in pF

t_{clkpb} = period of TWI peripheral bus clock

t_{prescaled} = period of TWI internal prescaled clock (see chapters on TWIM and TWIS)

The maximum t_{HD;DAT} has only to be met if the device does not stretch the LOW period (t_{LOW-I2C}) of TWCK.

7.9.6 JTAG Timing

Figure 7-16. JTAG Interface Signals

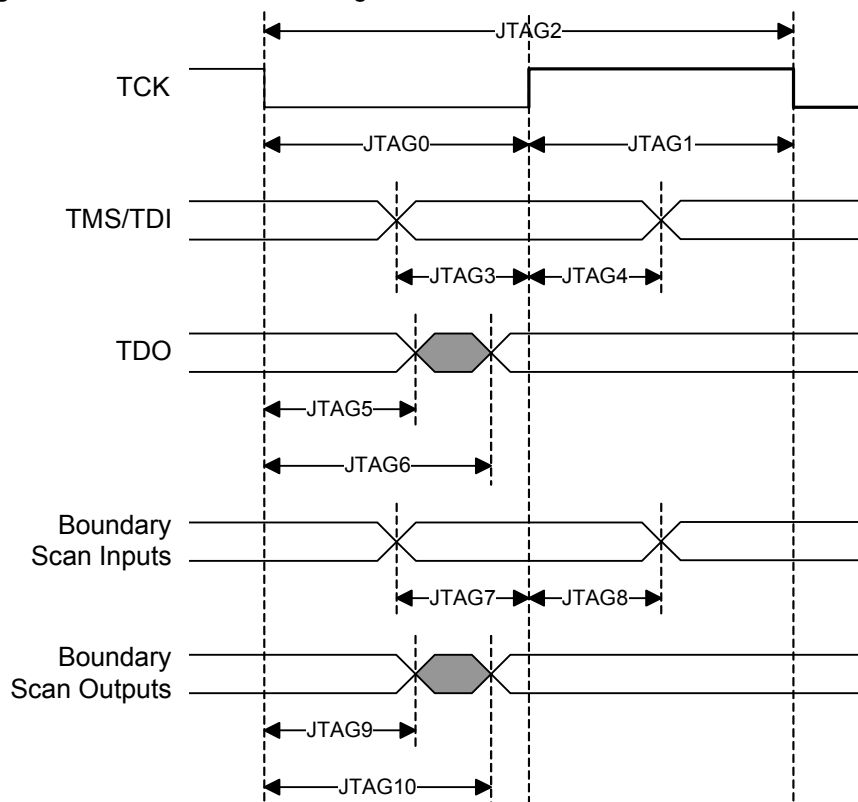


Table 7-51. JTAG Timings⁽¹⁾

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

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

7.9.7 EBI Timings

See EBI I/O lines description for more details.

Table 7-52. SMC Clock Signal.

| Symbol | Parameter | Max ⁽¹⁾ | Units |
|-----------------|--------------------------------|--------------------|-------|
| $1/(t_{CPSMC})$ | SMC Controller clock frequency | f_{cpu} | MHz |

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

Table 7-53. SMC Read Signals with Hold Settings⁽¹⁾

| Symbol | Parameter | Conditions | Min | Units |
|---------------------------------------|--|--|---|-------|
| NRD Controlled (READ_MODE = 1) | | | | |
| SMC ₁ | Data setup before NRD high | $V_{VDD} = 3.0V$, drive strength of the pads set to the lowest, external capacitor = 40pF | 32.5 | ns |
| SMC ₂ | Data hold after NRD high | | 0 | |
| SMC ₃ | NRD high to NBS0/A0 change ⁽²⁾ | | $nrd\ hold\ length * t_{CPSMC} - 1.5$ | |
| SMC ₄ | NRD high to NBS1 change ⁽²⁾ | | $nrd\ hold\ length * t_{CPSMC} - 0$ | |
| SMC ₅ | NRD high to NBS2/A1 change ⁽²⁾ | | $nrd\ hold\ length * t_{CPSMC} - 0$ | |
| SMC ₇ | NRD high to A2 - A25 change ⁽²⁾ | | $nrd\ hold\ length * t_{CPSMC} - 5.6$ | |
| SMC ₈ | NRD high to NCS inactive ⁽²⁾ | | $(nrd\ hold\ length - ncs\ rd\ hold\ length) * t_{CPSMC} - 1.3$ | |
| SMC ₉ | NRD pulse width | | $nrd\ pulse\ length * t_{CPSMC} - 0.6$ | |
| NRD Controlled (READ_MODE = 0) | | | | |
| SMC ₁₀ | Data setup before NCS high | $V_{VDD} = 3.0V$, drive strength of the pads set to the lowest, external capacitor = 40pF | 34.1 | ns |
| SMC ₁₁ | Data hold after NCS high | | 0 | |
| SMC ₁₂ | NCS high to NBS0/A0 change ⁽²⁾ | | $ncs\ rd\ hold\ length * t_{CPSMC} - 3$ | |
| SMC ₁₃ | NCS high to NBS0/A0 change ⁽²⁾ | | $ncs\ rd\ hold\ length * t_{CPSMC} - 2$ | |
| SMC ₁₄ | NCS high to NBS2/A1 change ⁽²⁾ | | $ncs\ rd\ hold\ length * t_{CPSMC} - 1.1$ | |
| SMC ₁₆ | NCS high to A2 - A25 change ⁽²⁾ | | $ncs\ rd\ hold\ length * t_{CPSMC} - 7.2$ | |
| SMC ₁₇ | NCS high to NRD inactive ⁽²⁾ | | $(ncs\ rd\ hold\ length - nrd\ hold\ length) * t_{CPSMC} - 2.2$ | |
| SMC ₁₈ | NCS pulse width | | $ncs\ rd\ pulse\ length * t_{CPSMC} - 3$ | |

Note: 1. These values are based on simulation and characterization of other AVR microcontrollers manufactured in the same process technology. These values are not covered by test limits in production.
 2. hold length = total cycle duration - setup duration - pulse duration. "hold length" is for "ncs rd hold length" or "nrd hold length".

Table 7-54. SMC Read Signals with no Hold Settings⁽¹⁾

| Symbol | Parameter | Conditions | Min | Units |
|---------------------------------------|----------------------------|--|------|-------|
| NRD Controlled (READ_MODE = 1) | | | | |
| SMC ₁₉ | Data setup before NRD high | $V_{VDD} = 3.0V$, drive strength of the pads set to the lowest, external capacitor = 40pF | 32.5 | ns |
| SMC ₂₀ | Data hold after NRD high | | 0 | |
| NRD Controlled (READ_MODE = 0) | | | | |
| SMC ₂₁ | Data setup before NCS high | $V_{VDD} = 3.0V$, drive strength of the pads set to the lowest, external capacitor = 40pF | 28.5 | ns |
| SMC ₂₂ | Data hold after NCS high | | 0 | |

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 7-55. SMC Write Signals with Hold Settings⁽¹⁾

| Symbol | Parameter | Conditions | Min | Units |
|---------------------------------------|--|---|--|-------|
| NRD Controlled (READ_MODE = 1) | | | | |
| SMC ₂₃ | Data Out valid before NWE high | $V_{VDD} = 3.0V$, drive strength of the pads set to the lowest, external capacitor = 40pF | $(nwe \text{ pulse length} - 1) * tcPSMC - 1.4$ | ns |
| SMC ₂₄ | Data Out valid after NWE high ⁽²⁾ | | $nwe \text{ pulse length} * tcPSMC - 4.7$ | |
| SMC ₂₅ | NWE high to NBS0/A0 change ⁽²⁾ | | $nwe \text{ pulse length} * tcPSMC - 2.7$ | |
| SMC ₂₉ | NWE high to NBS2/A1 change ⁽²⁾ | | $nwe \text{ pulse length} * tcPSMC - 0.7$ | |
| SMC ₃₁ | NWE high to A2 - A25 change ⁽²⁾ | | $nwe \text{ pulse length} * tcPSMC - 6.8$ | |
| SMC ₃₂ | NWE high to NCS inactive ⁽²⁾ | | $(nwe \text{ hold pulse} - ncs \text{ wr hold length}) * tcPSMC - 2.5$ | |
| SMC ₃₃ | NWE pulse width | | $nwe \text{ pulse length} * tcPSMC - 0.2$ | |
| NRD Controlled (READ_MODE = 0) | | | | |
| SMC ₃₄ | Data Out valid before NCS high | $V_{VDD} = 3.0V$, drive strength of the pads set to the lowest, external capacitor = 40pF | $(ncs \text{ wr pulse length} - 1) * tcPSMC - 2.2$ | ns |
| SMC ₃₅ | Data Out valid after NCS high ⁽²⁾ | | $ncs \text{ wr hold length} * tcPSMC - 5.1$ | |
| SMC ₃₆ | NCS high to NWE inactive ⁽²⁾ | | $(ncs \text{ wr hold length} - nwe \text{ hold length}) * tcPSMC - 2$ | |

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 wr hold length" or "nwe hold length"

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 | 8.7 | ns |
| SMC ₃₈ | NWE rising to NBS0/A0 valid | | 7.6 | |
| SMC ₄₀ | NWE rising to A1/NBS2 change | | 8.7 | |
| SMC ₄₂ | NWE rising to NCS rising | | 8.4 | |
| SMC ₄₃ | Data Out valid before NWE rising | | $(nwe\ pulse\ length - 1) * tc_{PSMC} - 1.2$ | |
| SMC ₄₄ | Data Out valid after NWE rising | | 8.4 | |
| SMC ₄₅ | NWE pulse width | | $nwe\ pulse\ length * tc_{PSMC} - 0$ | |

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

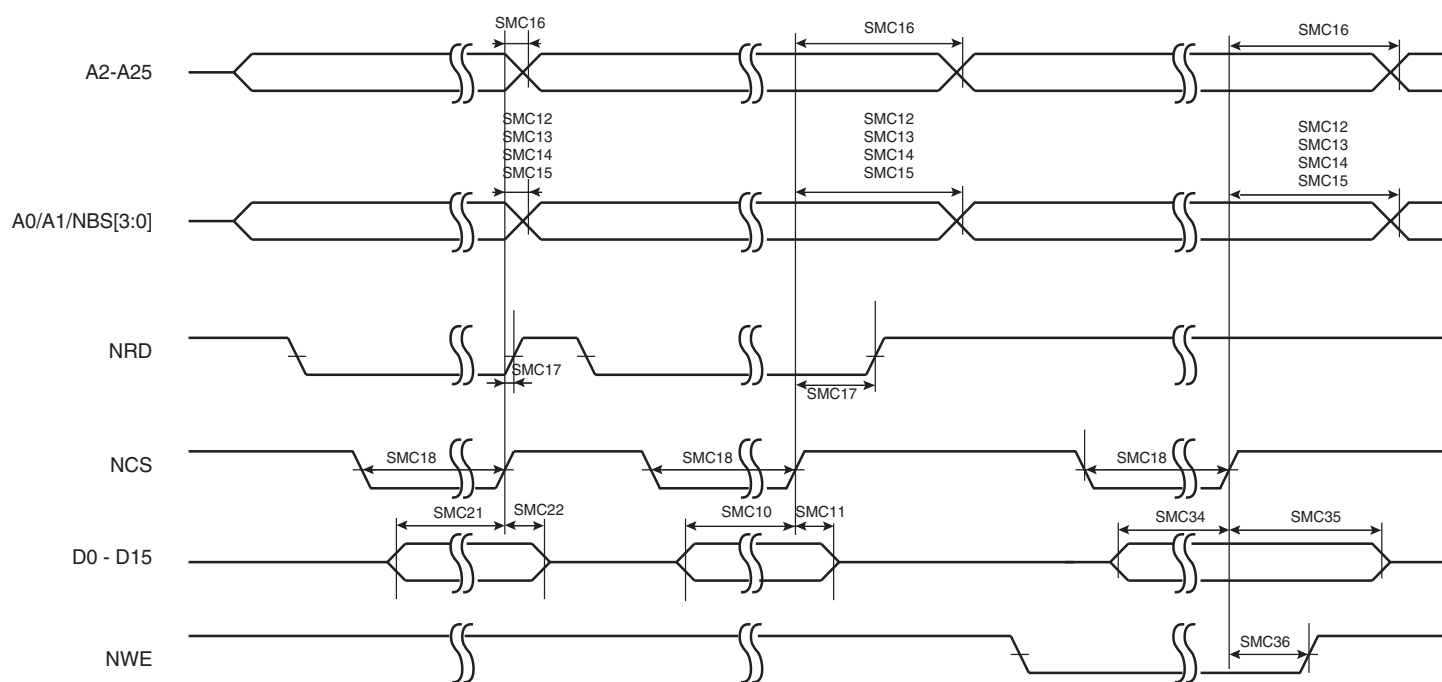
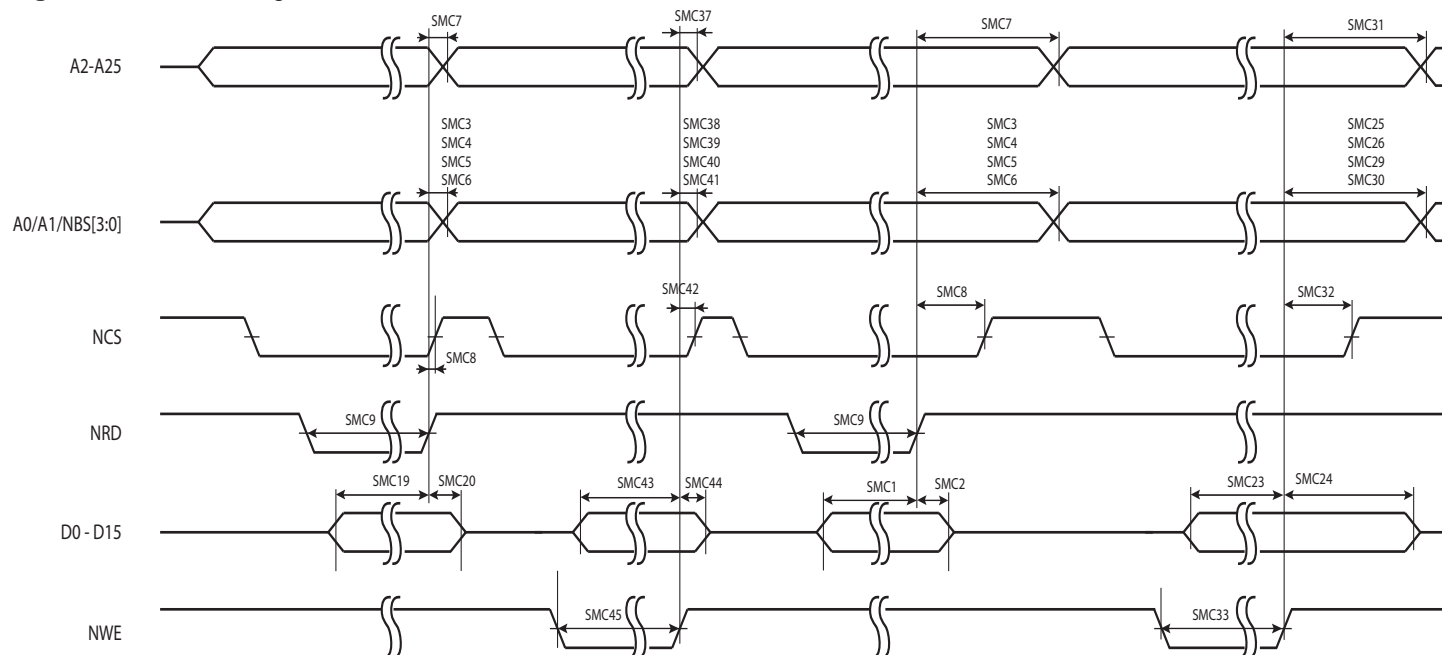


Figure 7-18. SMC Signals for NRD and NRW Controlled Accesses⁽¹⁾



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 |
|------------------------|----------------------------------|--------------------|-------|
| $1/(t_{\text{CPDCK}})$ | SDRAM Controller clock frequency | f_{cpu} | MHz |

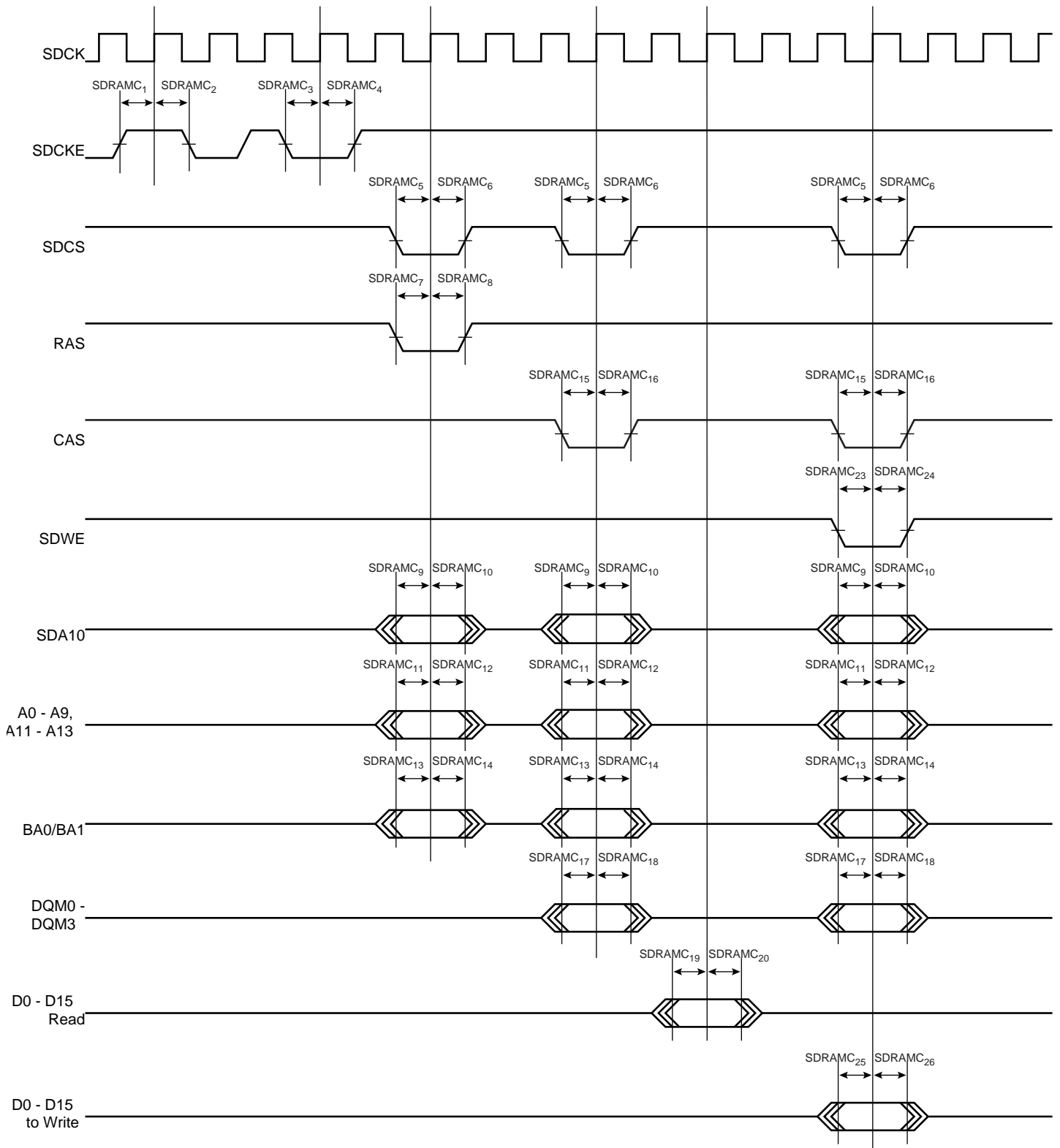
Note: 1. The maximum frequency of the SDRAMC interface is the same as the max frequency for the HSB.

Table 7-58. SDRAM Signal⁽¹⁾

| Symbol | Parameter | Conditions | Min | Units |
|----------------------|--|--|-----|-------|
| SDRAMC ₁ | SDCKE high before SDCK rising edge | $V_{DD} = 3.0V$, drive strength of the pads set to the highest, external capacitor = 40pF on SDRAM pins except 8 pF on SDCK pins | 5.6 | ns |
| SDRAMC ₂ | SDCKE low after SDCK rising edge | | 7.3 | |
| SDRAMC ₃ | SDCKE low before SDCK rising edge | | 6.8 | |
| SDRAMC ₄ | SDCKE high after SDCK rising edge | | 8.3 | |
| SDRAMC ₅ | SDCS low before SDCK rising edge | | 6.1 | |
| SDRAMC ₆ | SDCS high after SDCK rising edge | | 8.4 | |
| SDRAMC ₇ | RAS low before SDCK rising edge | | 7 | |
| SDRAMC ₈ | RAS high after SDCK rising edge | | 7.7 | |
| SDRAMC ₉ | SDA10 change before SDCK rising edge | | 6.4 | |
| SDRAMC ₁₀ | SDA10 change after SDCK rising edge | | 7.1 | |
| SDRAMC ₁₁ | Address change before SDCK rising edge | | 4.7 | |
| SDRAMC ₁₂ | Address change after SDCK rising edge | | 4.4 | |
| SDRAMC ₁₃ | Bank change before SDCK rising edge | | 6.2 | |
| SDRAMC ₁₄ | Bank change after SDCK rising edge | | 6.9 | |
| SDRAMC ₁₅ | CAS low before SDCK rising edge | | 6.6 | |
| SDRAMC ₁₆ | CAS high after SDCK rising edge | | 7.8 | |
| SDRAMC ₁₇ | DQM change before SDCK rising edge | | 6 | |
| SDRAMC ₁₈ | DQM change after SDCK rising edge | | 6.7 | |
| SDRAMC ₁₉ | D0-D15 in setup before SDCK rising edge | | 6.4 | |
| SDRAMC ₂₀ | D0-D15 in hold after SDCK rising edge | | 0 | |
| SDRAMC ₂₃ | SDWE low before SDCK rising edge | | 7 | |
| SDRAMC ₂₄ | SDWE high after SDCK rising edge | | 7.4 | |
| SDRAMC ₂₅ | D0-D15 Out valid before SDCK rising edge | | 5.2 | |
| SDRAMC ₂₆ | D0-D15 Out valid after SDCK rising edge | | 5.6 | |

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 7-19. SDRAMC Signals relative to SDCK.



7.9.9 MACB Characteristics

Table 7-59. Ethernet MAC Signals⁽¹⁾

| Symbol | Parameter | Conditions | Min. | Max. | Unit |
|------------------|--------------------------------|--|------|------|------|
| MAC ₁ | Setup for MDIO from MDC rising | V _{VDD} = 3.0V, drive strength of the pads set to the highest, external capacitor = 10pF on MACB pins | 0 | 2.5 | ns |
| MAC ₂ | Hold for MDIO from MDC rising | | 0 | 0.7 | ns |
| MAC ₃ | MDIO toggling from MDC falling | | 0 | 1.1 | ns |

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

Table 7-60. Ethernet MAC MII Specific Signals⁽¹⁾

| Symbol | Parameter | Conditions | Min. | Max. | Unit |
|-------------------|-----------------------------------|--|------|------|------|
| MAC ₄ | Setup for COL from TX_CLK rising | V _{VDD} = 3.0V, drive strength of the pads set to the highest, external capacitor = 10pF on MACB pins | 0 | | ns |
| MAC ₅ | Hold for COL from TX_CLK rising | | 0 | | ns |
| MAC ₆ | Setup for CRS from TX_CLK rising | | 0.5 | | ns |
| MAC ₇ | Hold for CRS from TX_CLK rising | | 0.5 | | ns |
| MAC ₈ | TX_ER toggling from TX_CLK rising | | 16.4 | 18.6 | ns |
| MAC ₉ | TX_EN toggling from TX_CLK rising | | 14.5 | 15.3 | ns |
| MAC ₁₀ | TXD toggling from TX_CLK rising | | 13.9 | 18.2 | ns |
| MAC ₁₁ | Setup for RXD from RX_CLK | | 1.3 | | ns |
| MAC ₁₂ | Hold for RXD from RX_CLK | | 1.8 | | ns |
| MAC ₁₃ | Setup for RX_ER from RX_CLK | | 3.4 | | ns |
| MAC ₁₄ | Hold for RX_ER from RX_CLK | | 0 | | ns |
| MAC ₁₅ | Setup for RX_DV from RX_CLK | | 0.7 | | ns |
| MAC ₁₆ | Hold for RX_DV from RX_CLK | | 1.3n | | ns |

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

Figure 7-20. Ethernet MAC MII Mode

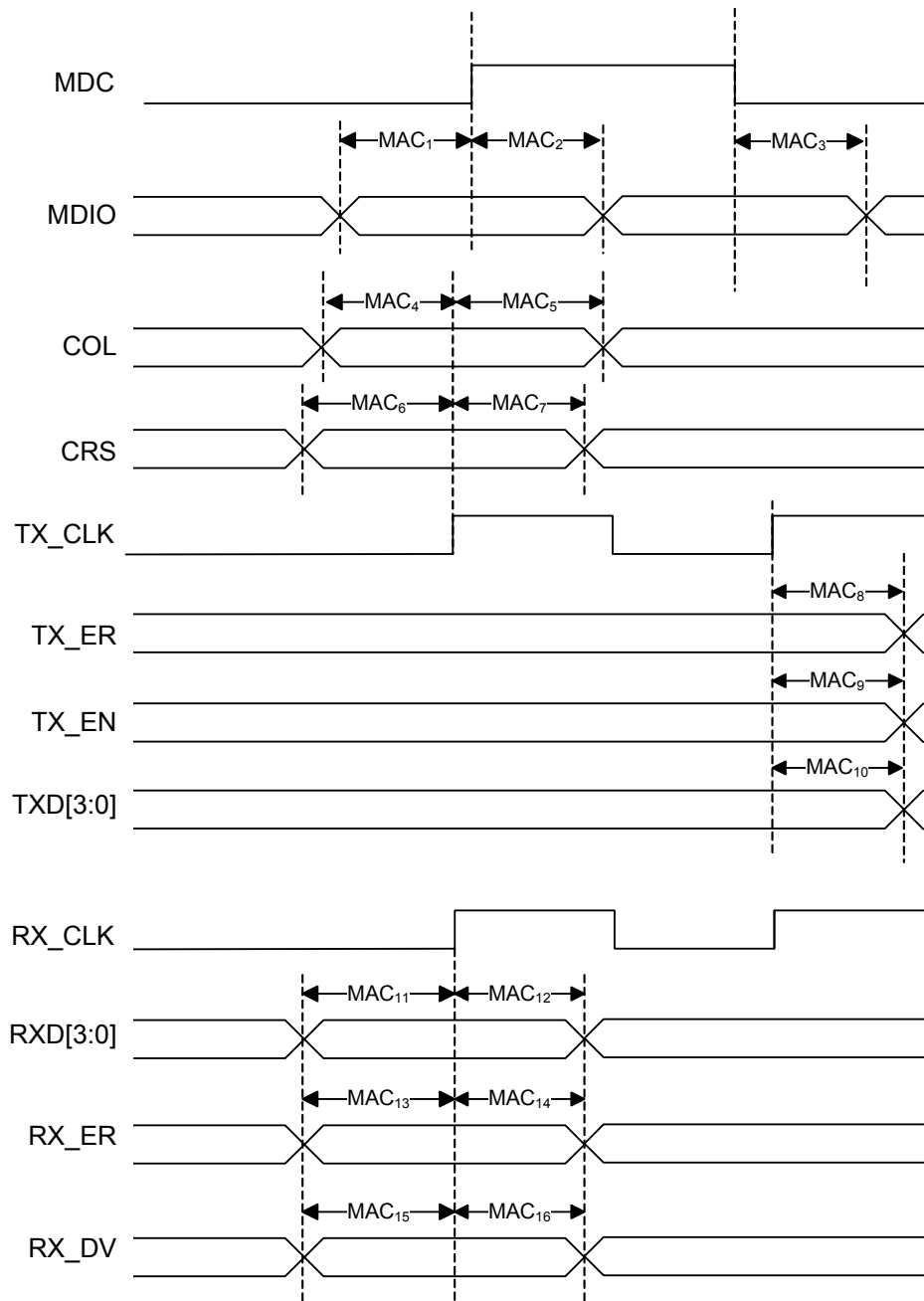
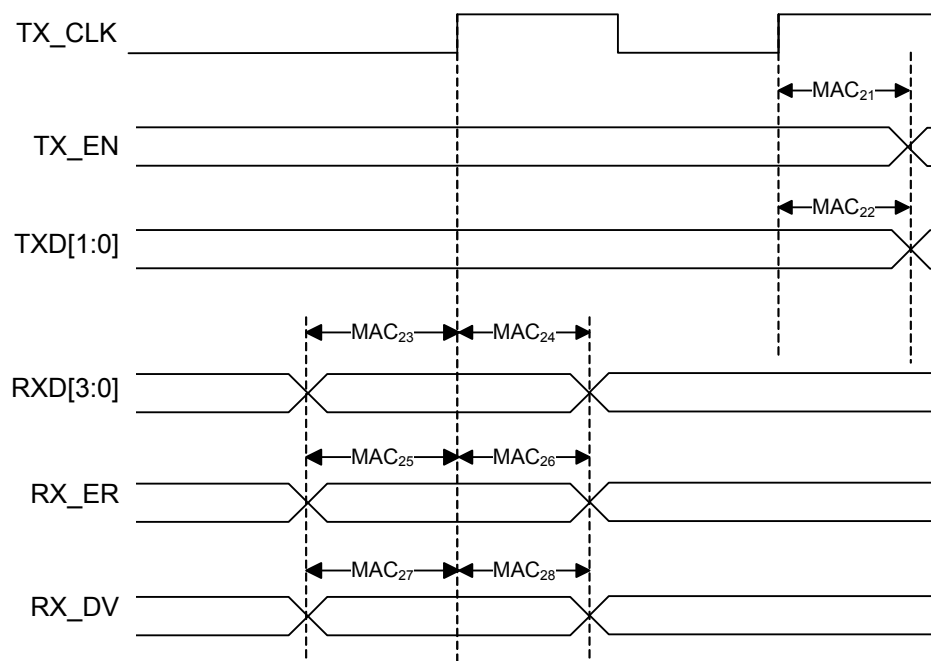


Table 7-61. Ethernet MAC RMII Specific Signals⁽¹⁾

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

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

Figure 7-21. Ethernet MAC RMII Mode



8. 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 |
|---------------|--|-------------|---------|------|------|
| θ_{JA} | Junction-to-ambient thermal resistance | No air flow | QFN64 | 20.0 | °C/W |
| θ_{JC} | Junction-to-case thermal resistance | | QFN64 | 0.8 | |
| θ_{JA} | Junction-to-ambient thermal resistance | No air flow | TQFP64 | 40.5 | °C/W |
| θ_{JC} | Junction-to-case thermal resistance | | TQFP64 | 8.7 | |
| θ_{JA} | Junction-to-ambient thermal resistance | No air flow | TQFP100 | 39.3 | °C/W |
| θ_{JC} | Junction-to-case thermal resistance | | TQFP100 | 8.5 | |
| θ_{JA} | Junction-to-ambient thermal resistance | No air flow | LQFP144 | 38.1 | °C/W |
| θ_{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. $T_J = T_A + (P_D \times (\theta_{HEATSINK} + \theta_{JC}))$

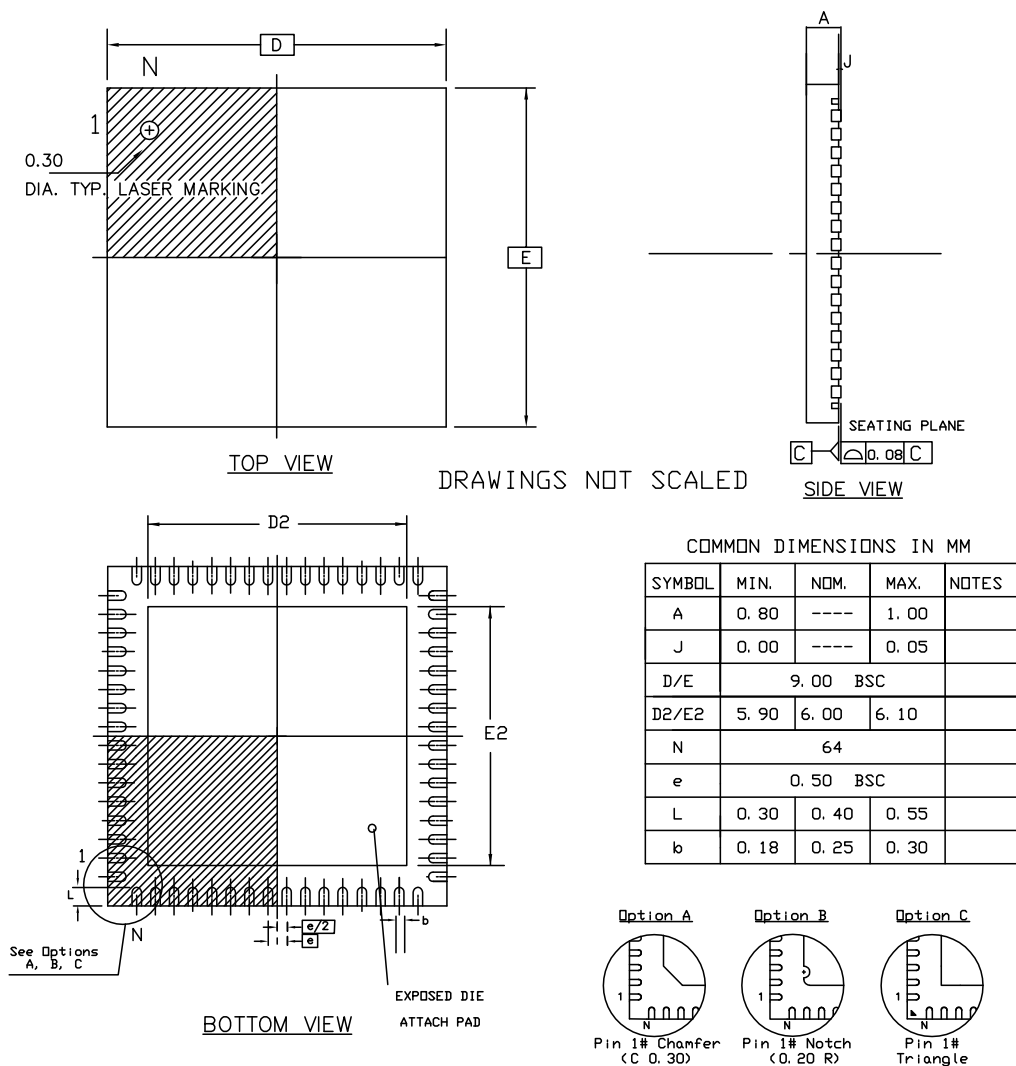
where:

- θ_{JA} = package thermal resistance, Junction-to-ambient (°C/W), provided in [Table 8-1 on page 90](#).
- θ_{JC} = package thermal resistance, Junction-to-case thermal resistance (°C/W), provided in [Table 8-1 on page 90](#).
- $\theta_{HEAT\ SINK}$ = 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 51.
- 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.

8.2 Package Drawings

Figure 8-1. QFN-64 package drawing



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

Table 8-2. Device and Package Maximum Weight

| | |
|-----|----|
| 200 | mg |
|-----|----|

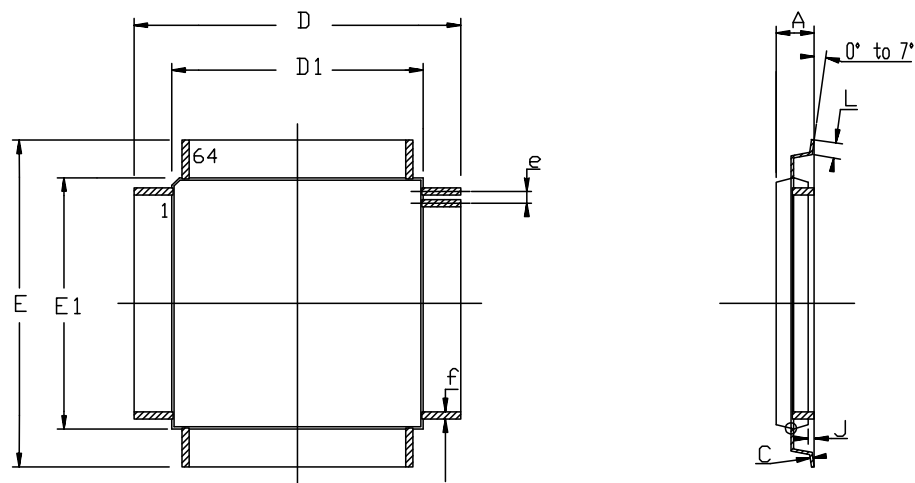
Table 8-3. Package Characteristics

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

Table 8-4. Package Reference

| | |
|-------------------------|--------|
| JEDEC Drawing Reference | MS-026 |
| JESD97 Classification | E3 |

Figure 8-2. TQFP-64 package drawing



COMMON DIMENSIONS IN MM

| SYMBOL | Min | Max | NOTES |
|--------|-----------|------|-------|
| A | ---- | 1.20 | |
| A1 | 0.95 | 1.05 | |
| C | 0.09 | 0.20 | |
| D | 12.00 BSC | | |
| D1 | 10.00 BSC | | |
| E | 12.00 BSC | | |
| E1 | 10.00 BSC | | |
| J | 0.05 | 0.15 | |
| L | 0.45 | 0.75 | |
| e | 0.50 BSC | | |
| f | 0.17 | 0.27 | |

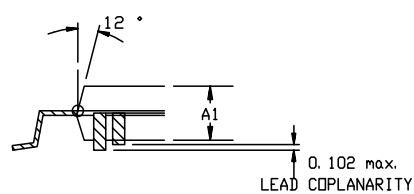


Table 8-5. Device and Package Maximum Weight

| | |
|-----|----|
| 300 | mg |
|-----|----|

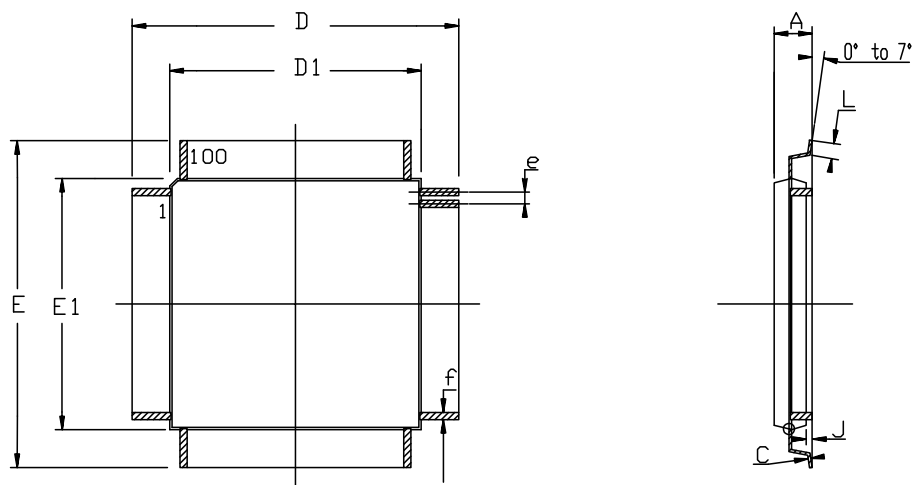
Table 8-6. Package Characteristics

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

Table 8-7. Package Reference

| | |
|-------------------------|--------|
| JEDEC Drawing Reference | MS-026 |
| JESD97 Classification | E3 |

Figure 8-3. TQFP-100 package drawing



COMMON DIMENSIONS IN MM

| SYMBOL | Min | Max | NOTES |
|--------|-----------|------|-------|
| A | ---- | 1.20 | |
| A1 | 0.95 | 1.05 | |
| C | 0.09 | 0.20 | |
| D | 16.00 BSC | | |
| D1 | 14.00 BSC | | |
| E | 16.00 BSC | | |
| E1 | 14.00 BSC | | |
| J | 0.05 | 0.15 | |
| L | 0.45 | 0.75 | |
| e | 0.50 BSC | | |
| f | 0.17 | 0.27 | |

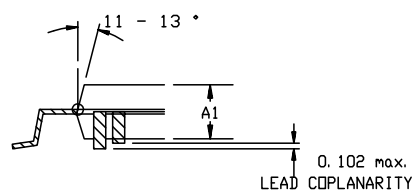


Table 8-8. Device and Package Maximum Weight

| | |
|-----|----|
| 500 | mg |
|-----|----|

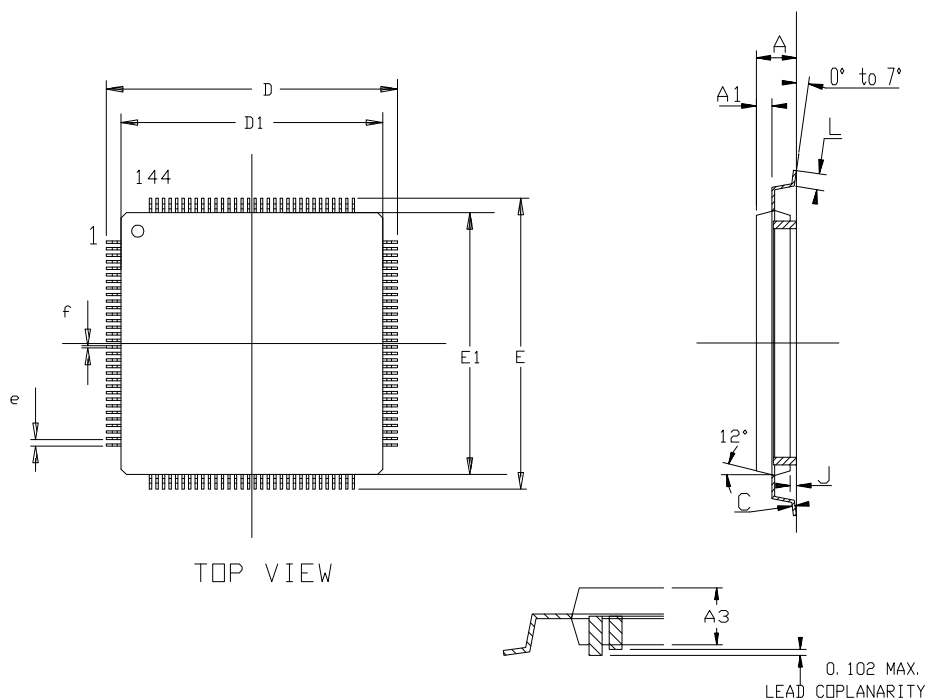
Table 8-9. Package Characteristics

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

Table 8-10. Package Reference

| | |
|-------------------------|--------|
| JEDEC Drawing Reference | MS-026 |
| JESD97 Classification | E3 |

Figure 8-4. LQFP-144 package drawing



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

Table 8-11. Device and Package Maximum Weight

| | |
|------|----|
| 1300 | mg |
|------|----|

Table 8-12. Package Characteristics

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

Table 8-13. Package Reference

| | |
|-------------------------|--------|
| JEDEC Drawing Reference | MS-026 |
| JESD97 Classification | E3 |

8.3 Soldering Profile

Table 8-14 gives the recommended soldering profile from J-STD-20.

Table 8-14. Soldering Profile

| Profile Feature | Green Package |
|--|--------------------------|
| Average Ramp-up Rate (217°C to Peak) | 3°C/sec |
| Preheat Temperature 175°C ±25°C | Min. 150 °C, Max. 200 °C |
| Temperature Maintained Above 217°C | 60-150 sec |
| Time within 5-C of Actual Peak Temperature | 30 sec |
| Peak Temperature Range | 260 °C |
| Ramp-down Rate | 6 °C/sec |
| Time 25-C to Peak Temperature | Max. 8 minutes |

Note: It is recommended to apply a soldering temperature higher than 250°C. A maximum of three reflow passes is allowed per component.

9. Ordering Information

Table 9-1. Ordering Information

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

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 0x10000000 before issuing the MEMORY_SPEED_REQUEST command and use this formula instead:

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

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.

10.1.5 SCIF

1 PLLCOUNT value larger than zero can cause PLEN glitch

Initializing the PLLCOUNT with a value greater than zero creates a glitch on the PLEN signal during asynchronous wake up.

Fix/Workaround

The lock-masking mechanism for the PLL should not be used.

The PLLCOUNT field of the PLL Control Register should always be written to zero.

2 PLL lock might not clear after disable

Under certain circumstances, the lock signal from the Phase Locked Loop (PLL) oscillator may not go back to zero after the PLL oscillator has been disabled. This can cause the propagation of clock signals with the wrong frequency to parts of the system that use the PLL clock.

Fix/Workaround

PLL must be turned off before entering STOP, DEEPSTOP or STATIC sleep modes. If PLL has been turned off, a delay of 30us must be observed after the PLL has been enabled again before the SCIF.PLL0LOCK bit can be used as a valid indication that the PLL is locked.

3 BOD33 reset locks the device

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

Fix/Workaround

Use an external BOD on VDDIN_33 or an external reset source.

10.1.6 SPI

1 SPI data transfer hangs with CSR0.CSAAT==1 and MR.MODFDIS==0

When CSR0.CSAAT==1 and mode fault detection is enabled (MR.MODFDIS==0), the SPI module will not start a data transfer.

Fix/Workaround

Disable mode fault detection by writing a one to MR.MODFDIS.

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.

10.1.7 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.

10.1.8 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.

10.1.9 TWIS

1 **Clearing the NAK bit before the BTF bit is set locks up the TWI bus**

When the TWIS is in transmit mode, clearing the NAK Received (NAK) bit of the Status Register (SR) before the end of the Acknowledge/Not Acknowledge cycle will cause the TWIS to attempt to continue transmitting data, thus locking up the bus.

Fix/Workaround

Clear SR.NAK only after the Byte Transfer Finished (BTF) bit of the same register has been set.

10.1.10 USBC

1 **UPINRQx.INRQ field is limited to 8-bits**

In Host mode, when using the UPINRQx.INRQ feature together with the multi-packet mode to launch a finite number of packet among multi-packet, the multi-packet size (located in the descriptor table) is limited to the UPINRQx.INRQ value multiply by the pipe size.

Fix/Workaround

UPINRQx.INRQ value shall be less than the number of configured multi-packet.

2 **In USB host mode, downstream resume feature does not work (UHCON.RESUME=1).**

Fix/Workaround

None.

3 In host mode, the disconnection during OUT transition is not supported

In USB host mode, a pipe can not work if the previous USB device disconnection has occurred during a USB transfer.

Fix/Workaround

Reset the USBC (USBCON.USB=0 and =1) after a device disconnection (UHINT.DDISCI).

4 In USB host mode, entering suspend mode can fail

In USB host mode, entering suspend mode can fail when UHCON.SOFE=0 is done just after a SOF reception (UHINT.HSOFI).

Fix/Workaround

Check that UHNUM.FLENHIGH is below 185 in Full speed and below 21 in Low speed before clearing UHCON.SOFE.

5 In USB host mode, entering suspend mode for low speed device can fail when the USB freeze (USBCON.FRZCLK=1) is done just after UHCON.SOFE=0.**Fix/Workaround**

When entering suspend mode (UHCON.SOFE is cleared), check that USBFSM.DRDSTATE is not equal to three before freezing the clock (USBCON.FRZCLK=1).

10.1.11 WDT**1 WDT Control Register does not have synchronization feedback**

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

Fix/Workaround

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

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

10.2 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 0x10000000 before issuing the MEMORY_SPEED_REQUEST command and use this formula instead:

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

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.



2 Requesting clocks in idle sleep modes will mask all other PB clocks than the requested

In idle or frozen sleep mode, all the PB clocks will be frozen if the TWIS or the AST need to wake the cpu up.

Fix/Workaround

Disable the TWIS or the AST before entering idle or frozen sleep mode.

3 TWIS may not wake the device from sleep mode

If the CPU is put to a sleep mode (except Idle and Frozen) directly after a TWI Start condition, the CPU may not wake upon a TWIS address match. The request is NACKed.

Fix/Workaround

When using the TWI address match to wake the device from sleep, do not switch to sleep modes deeper than Frozen. Another solution is to enable asynchronous EIC wake on the TWIS clock (TWCK) or TWIS data (TWD) pins, in order to wake the system up on bus events.

10.2.6 SCIF

1 PLLCOUNT value larger than zero can cause PLEN glitch

Initializing the PLLCOUNT with a value greater than zero creates a glitch on the PLEN signal during asynchronous wake up.

Fix/Workaround

The lock-masking mechanism for the PLL should not be used.

The PLLCOUNT field of the PLL Control Register should always be written to zero.

2 PLL lock might not clear after disable

Under certain circumstances, the lock signal from the Phase Locked Loop (PLL) oscillator may not go back to zero after the PLL oscillator has been disabled. This can cause the propagation of clock signals with the wrong frequency to parts of the system that use the PLL clock.

Fix/Workaround

PLL must be turned off before entering STOP, DEEPSTOP or STATIC sleep modes. If PLL has been turned off, a delay of 30us must be observed after the PLL has been enabled again before the SCIF.PLL0LOCK bit can be used as a valid indication that the PLL is locked.

3 BOD33 reset locks the device

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

Fix/Workaround

Use an external BOD on VDDIN_33 or an external reset source.

10.2.7 SPI

1 SPI data transfer hangs with CSR0.CSAAT==1 and MR.MODFDIS==0

When CSR0.CSAAT==1 and mode fault detection is enabled (MR.MODFDIS==0), the SPI module will not start a data transfer.

Fix/Workaround

Disable mode fault detection by writing a one to MR.MODFDIS.

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.

10.2.10 TWIS

- 1 **Clearing the NAK bit before the BTF bit is set locks up the TWI bus**
When the TWIS is in transmit mode, clearing the NAK Received (NAK) bit of the Status Register (SR) before the end of the Acknowledge/Not Acknowledge cycle will cause the TWIS to attempt to continue transmitting data, thus locking up the bus.
Fix/Workaround
Clear SR.NAK only after the Byte Transfer Finished (BTF) bit of the same register has been set.
- 2 **TWIS stretch on Address match error**
When the TWIS stretches TWCK due to a slave address match, it also holds TWD low for the same duration if it is to be receiving data. When TWIS releases TWCK, it releases TWD at the same time. This can cause a TWI timing violation.
Fix/Workaround
None.
- 3 **TWALM forced to GND**
The TWALM pin is forced to GND when the alternate function is selected and the TWIS module is enabled.
Fix/Workaround
None.

10.2.11 USBC

- 1 **UPINRQx.INRQ field is limited to 8-bits**
In Host mode, when using the UPINRQx.INRQ feature together with the multi-packet mode to launch a finite number of packet among multi-packet, the multi-packet size (located in the descriptor table) is limited to the UPINRQx.INRQ value multiply by the pipe size.
Fix/Workaround
UPINRQx.INRQ value shall be less than the number of configured multi-packet.
- 2 **In USB host mode, downstream resume feature does not work (UHCON.RESUME=1).**
Fix/Workaround
None.
- 3 **In host mode, the disconnection during OUT transition is not supported**
In USB host mode, a pipe can not work if the previous USB device disconnection has occurred during a USB transfer.
Fix/Workaround
Reset the USBC (USBCON.USB=0 and =1) after a device disconnection (UHINT.DDISCI).
- 4 **In USB host mode, entering suspend mode can fail**
In USB host mode, entering suspend mode can fail when UHCON.SOFE=0 is done just after a SOF reception (UHINT.HSOFI).
Fix/Workaround
Check that UHNUM.FLENHIGH is below 185 in Full speed and below 21 in Low speed before clearing UHCON.SOFE.
- 5 **In USB host mode, entering suspend mode for low speed device can fail when the USB freeze (USBCON.FRZCLK=1) is done just after UHCON.SOFE=0.**
Fix/Workaround
When entering suspend mode (UHCON.SOFE is cleared), check that USBFSM.DRDSTATE is not equal to three before freezing the clock (USBCON.FRZCLK=1).

10.2.12 WDT

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

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

Fix/Workaround

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

2 WDT Control Register does not have synchronization feedback

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

Fix/Workaround

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

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

11. Datasheet Revision History

Please note that the referring page numbers in this section are referred to this document. The referring revision in this section are referring to the document revision.

11.1 Rev. D – 01/12

- 1 Errata: Updated
- 2 PM: Clock Mask Table Updated
- 3 Fixed PLLOPT field description in SCIF chapter
- 4 MDMA: Swapped bit descriptions for IER and IDR
- 5 MACB: USRIO register description and bit descriptions for IMR/IDR/IER Updated
- 6 USBC: UPCON.PFREEZE and UPINRQn description Updated
- 7 ACIFA: Updated
- 8 ADCIFA: CFG.MUXSET, SSMQ description and conversion results section Updated
- 9 DACIFB: Calibration section Updated
- 10 Electrical Characteristics: ADCREFP/ADCREFN added

11.2 Rev. C – 08/11

- 1 Electrical Characteristics Updated:
 - I/O Pins characteristics
 - 8MHz/1MHz RC Oscillator (RC8M) characteristics
 - 1.8V Voltage Regulator characteristics
 - 3.3V Voltage Regulator characteristics
 - 1.8VBrown Out Detector (BOD18) characteristics
 - 3.3VBrown Out Detector (BOD33) characteristics
 - 5VBrown Out Detector (BOD50) characteristics
 - Analog to Digital Converter (ADC) and sample and hold (S/DH) Characteristics
 - Analog Comparator characteristics
- 2 Errata: Updated
- 3 TWIS: Updated

11.3 Rev. B – 03/11

- 1 Package and pinout: Added supply column. Updated peripheral functions
- 2 Supply and Startup Considerations: Updated I/O lines power
- 3 PM: Added AWEN description
- 4 SCIF: Added VREGCR register

- 5 AST: Updated digital tuner formula
- 6 SDRAMC: cleaned-up SDCS/NCS names. Added VERSION register
- 7 SAU: Updated SR.IDLE
- 8 USART: Updated
- 9 CANIF: Updated address map figure
- 10 USBC: Updated
- 11 DACIFB: Updated
- 12 Programming and Debugging: Added JTAG Data Registers section
- 13 Electrical Characteristics: Updated
- 14 Ordering Information: Updated
- 15 Errata: Updated

11.4 Rev. A – 10/10

- 1 Initial revision

Table of Contents

| | | |
|----------|---|-----------|
| 1 | <i>Description</i> | 3 |
| 2 | <i>Overview</i> | 5 |
| | 2.1 Block diagram | 5 |
| | 2.2 Configuration Summary | 6 |
| 3 | <i>Package and Pinout</i> | 8 |
| | 3.1 Package | 8 |
| | 3.2 Peripheral Multiplexing on I/O lines | 11 |
| | 3.3 Signals Description | 18 |
| | 3.4 I/O Line Considerations | 24 |
| 4 | <i>Processor and Architecture</i> | 25 |
| | 4.1 Features | 25 |
| | 4.2 AVR32 Architecture | 25 |
| | 4.3 The AVR32UC CPU | 26 |
| | 4.4 Programming Model | 30 |
| | 4.5 Exceptions and Interrupts | 34 |
| 5 | <i>Memories</i> | 39 |
| | 5.1 Embedded Memories | 39 |
| | 5.2 Physical Memory Map | 40 |
| | 5.3 Peripheral Address Map | 41 |
| | 5.4 CPU Local Bus Mapping | 43 |
| 6 | <i>Supply and Startup Considerations</i> | 46 |
| | 6.1 Supply Considerations | 46 |
| | 6.2 Startup Considerations | 49 |
| 7 | <i>Electrical Characteristics</i> | 50 |
| | 7.1 Absolute Maximum Ratings* | 50 |
| | 7.2 Supply Characteristics | 50 |
| | 7.3 Maximum Clock Frequencies | 51 |
| | 7.4 Power Consumption | 51 |
| | 7.5 I/O Pin Characteristics | 55 |
| | 7.6 Oscillator Characteristics | 57 |
| | 7.7 Flash Characteristics | 60 |
| | 7.8 Analog Characteristics | 61 |

7.9 Timing Characteristics70

8 Mechanical Characteristics 90

8.1 Thermal Considerations90

8.2 Package Drawings91

8.3 Soldering Profile95

9 Ordering Information 96

10 Errata 97

10.1 rev E97

10.2 rev D101

11 Datasheet Revision History 106

11.1 Rev. D – 01/12106

11.2 Rev. C – 08/11106

11.3 Rev. B – 03/11106

11.4 Rev. A – 10/10107



Headquarters

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

International

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

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

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

Product Contact

Web Site
www.atmel.com

Technical Support
avr32@atmel.com

Sales Contact
www.atmel.com/contacts

Literature Requests
www.atmel.com/literature

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. **EXCEPT AS SET FORTH IN ATMEL'S TERMS AND CONDITIONS OF SALE LOCATED ON ATMEL'S WEB SITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.** Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

© 2012 Atmel Corporation. All rights reserved. Atmel®, Atmel logo and combinations thereof, AVR® and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.