Features

- High Performance, Low Power 32-bit AVR® Microcontroller
  - Compact Single-Cycle RISC Instruction Set Including DSP Instructions
  - Read-Modify-Write Instructions and Atomic Bit Manipulation
  - Performance
    - Up to 61 DMIPS Running at 48 MHz from Flash (1 Flash Wait State)
    - Up to 34 DMIPS Running at 24 MHz from Flash (0 Flash Wait State)
- Multi-Hierarchy Bus System
  - High-Performance Data Transfers on Separate Buses for Increased Performance
  - 7 Peripheral DMA Channels Improve Speed for Peripheral Communication
- Internal High-Speed Flash
  - 128 Kbytes, and 64 Kbytes Versions
  - Single-Cycle Access up to 24 MHz
  - Prefetch Buffer Optimizing Instruction Execution at Maximum Speed
  - 4 ms Page Programming Time and 8 ms Full-Chip Erase Time
  - 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
  - 16 Kbytes
- Interrupt Controller (INTC)
  - Autovectored Low Latency Interrupt Service with Programmable Priority
- External Interrupt Controller (EIC)
- System Functions
  - Power and Clock Manager
  - SleepWalking™ Power Saving Control
  - Internal System RC Oscillator (RCSYS)
  - 32 KHz Oscillator
  - Clock Failure Detection
  - One Multipurpose Oscillator and two Phase Locked Loop (PLL)
- 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
- Universal Serial Bus (USB)
  - Device 2.0 full speed and low speed
  - Flexible End-Point Configuration and Management
  - On-chip Transceivers Including Pull-Ups
- Three 16-bit Timer/Counter (TC) Channels
  - External Clock Inputs, PWM, Capture and Various Counting Capabilities
- 7 PWM Channels (PWMA)
  - 12-bit PWM up to 150 MHz Source Clock
- Three Universal Synchronous/Asynchronous Receiver/Transmitters (USART)
  - Independent Baudrate Generator, Support for SPI
  - Support for Hardware Handshaking
- One Master/Slave Serial Peripheral Interfaces (SPI) with Chip Select Signals
  - Up to 15 SPI Slaves can be Addressed

32-bit AVR® Microcontroller

ATUC128D3
ATUC64D3
ATUC128D4
ATUC64D4

Summary

32133D–11/2011

- One Master and One Slave Two-Wire Interfaces (TWI), 400 kbit/s I^2C-compatible
- One 8-channel Analog-To-Digital Converter (ADC)
- One Inter-IC Sound Controller (IISC) with Stereo Capabilities
- Autonomous Capacitive Touch Button (QTouch®) Capture
  - Up to 25 Touch Buttons
  - QWheel® and QSlide® Compatible
- QTouch® Library Support
  - Capacitive Touch Buttons, Sliders, and Wheels
  - QTouch® and QMatrix® Acquisition
  - Hardware assisted QTouch® Acquisition
- One Programmable Glue Logic Controller(GLOC) for General Purpose PCB Design
- On-Chip Non-Intrusive Debug System
  - Nexus Class 2+, Runtime Control
  - aWire™ Single-Pin Programming and Debug Interface Muxed with Reset Pin
  - 64-pin and 48-pin TQFP/QFN (51 and 35 GPIO Pins)
- Four High-Drive I/O Pins
- Single 3.3V Power Supply or Dual 1.8V-3.3V Power Supply

1. Description

The UC3D is a complete System-On-Chip microcontroller based on the AVR32UC RISC processor running at frequencies up to 48MHz. 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 fast and flexible interrupt controller for supporting modern operating systems and real-time operating systems.

Higher computation capability is achieved using a rich set of DSP instructions.

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

The Power Manager improves design flexibility and security. Power monitoring is supported by on-chip Power-On Reset (POR), and Brown-Out Detector (BOD). The device features several oscillators, such as Oscillator 0 (OSC0), 32 KHz Oscillator and system RC oscillator (RCSYS), and two Phase Lock Loop (PLL). Either of these oscillators/PLLs can be used as source for the system clock.

The Watchdog Timer (WDT) will reset the device unless it is periodically serviced by the software. This allows the device to recover from a condition that has caused the system to be unstable.

The Asynchronous Timer (AST) combined with the 32KHz crystal oscillator supports powerful real-time clock capabilities, with a maximum timeout of up to 136 years. The AST can operate in counter mode or calendar mode. The 32KHz crystal oscillator can operate in a 1- or 2-pin mode, trading pin usage and accuracy.

The Frequency Meter (FREQM) allows accurate measuring of a clock frequency by comparing it to a known reference clock.

The Full-Speed USB 2.0 Device interface supports several USB Classes at the same time thanks to the rich End-Point configuration.

The device includes three 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 Pulse Width Modulation controller (PWMA) provides 12-bit PWM channels which can be synchronized and controlled from a common timer. Seven PWM channels are available, enabling applications that require multiple PWM outputs, such as LCD backlight control. The PWM channels can operate independently, with duty cycles set independently from each other, or in interlinked mode, with multiple channels changed at the same time.

The UC3D also features many communication interfaces for communication intensive applications. In addition to standard serial interfaces like USART, SPI or TWI, USB is available. The USART supports different communication modes, like SPI mode.

A general purpose 8-channel ADC is provided; It features a fully configurable sequencer that handles many conversions. Window Mode allows each ADC channel to be used like a simple Analog Comparator.

The Inter-IC Sound controller (IISC) provides easy access to digital audio interfaces following I2S stereo standard.

The Capacitive Touch (CAT) module senses touch on external capacitive touch sensors, using the QTouch® technology. Capacitive touch sensors use no external mechanical components, unlike normal push buttons, and therefore demand less maintenance in the user application. The CAT module allows up to 25 touch sensors. One touch sensor can be configured to operate autonomously without software interaction,allowing wakeup from sleep modes when activated.

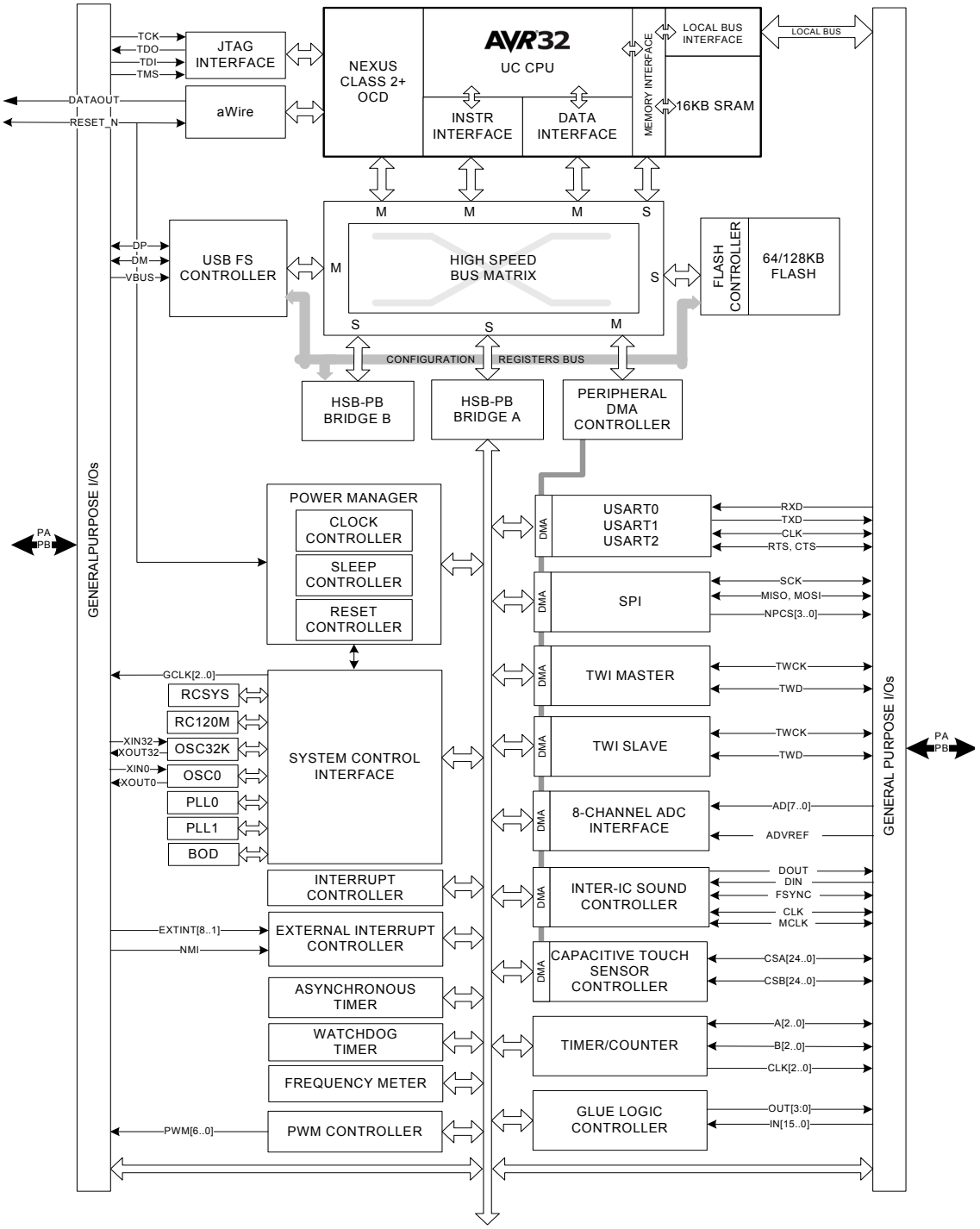
Atmel also offers the QTouch library for embedding capacitive touch buttons, sliders, and wheels functionality into AVR microcontrollers. The patented charge-transfer signal acquisition offers robust sensing and included fully debounced reporting of touch keys and includes Adjacent Key Suppression® (AKS®) 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.

The UC3D integrates a class 2+ Nexus 2.0 On-Chip Debug (OCD) System, with full-speed read/write memory access, in addition to basic runtime control. 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	ATUC128/64D3	ATUC128/64D4
Flash	128/64 KB	128/64KB
SRAM	16 KB	16 KB
Package	TQFP64, QFN64	TQFP48, QFN48
GPIO	51	35
FS USB Device	1	
Hi-drive pins	4	
External Interrupts	9	7
TWI Master/Slave	1/1	
USART	3	
Peripheral DMA Channels	7	
SPI	1	
Asynchronous Timers	1	
Timer/Counter Channels	3	
PWM channels	7	
Inter-IC Sound	1	
Frequency Meter	1	
Watchdog Timer	1	
Power Manager	1	
Oscillators	2x Phase Locked Loop 80-240 MHz (PLL) 1x Crystal Oscillator 0.4-20 MHz (OSC0) 1x Crystal Oscillator 32 KHz (OSC32K) 1x RC Oscillator 120 MHz (RC120M) 1x RC Oscillator 115 kHz (RCSYS)	
10-bit ADC channels	8	6
Capacitive Touch Sensor supported	25	17
Glue Logic Control Inputs/Outputs	16/4	14/4
JTAG	1	
aWire	1	
Max Frequency	48 MHz	

3. Package and Pinout

3.1 Package

The device pins are multiplexed with peripheral functions as described in Section 3.2.

Figure 3-1. TQFP48/QFN48 Pinout

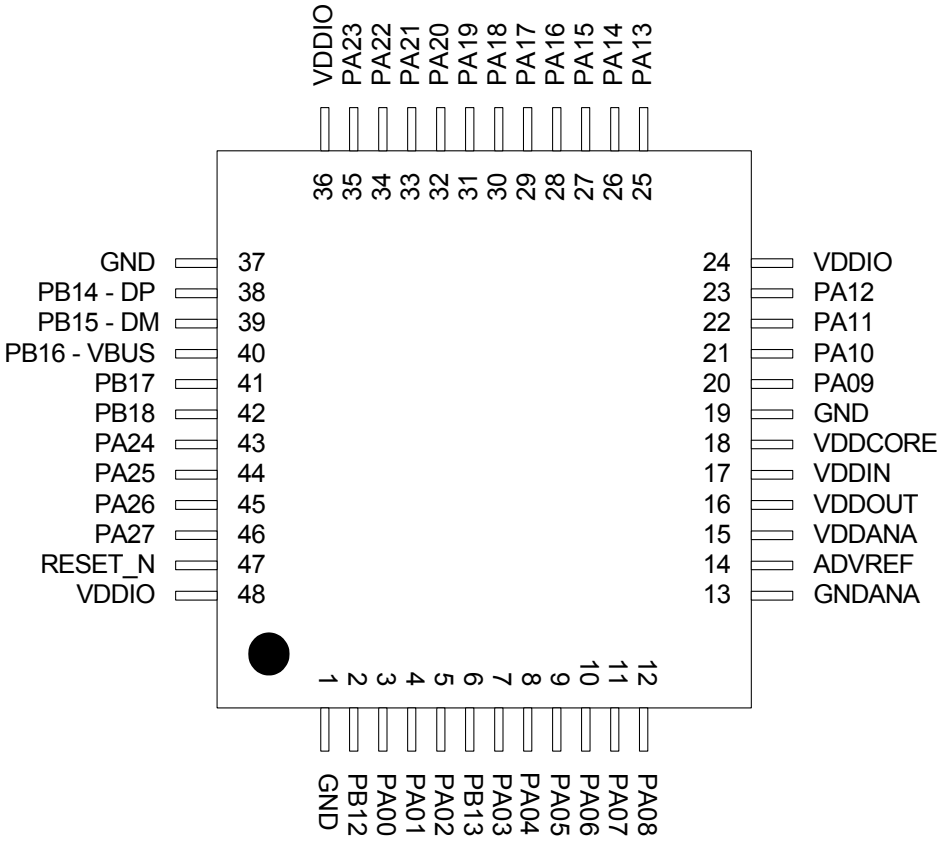
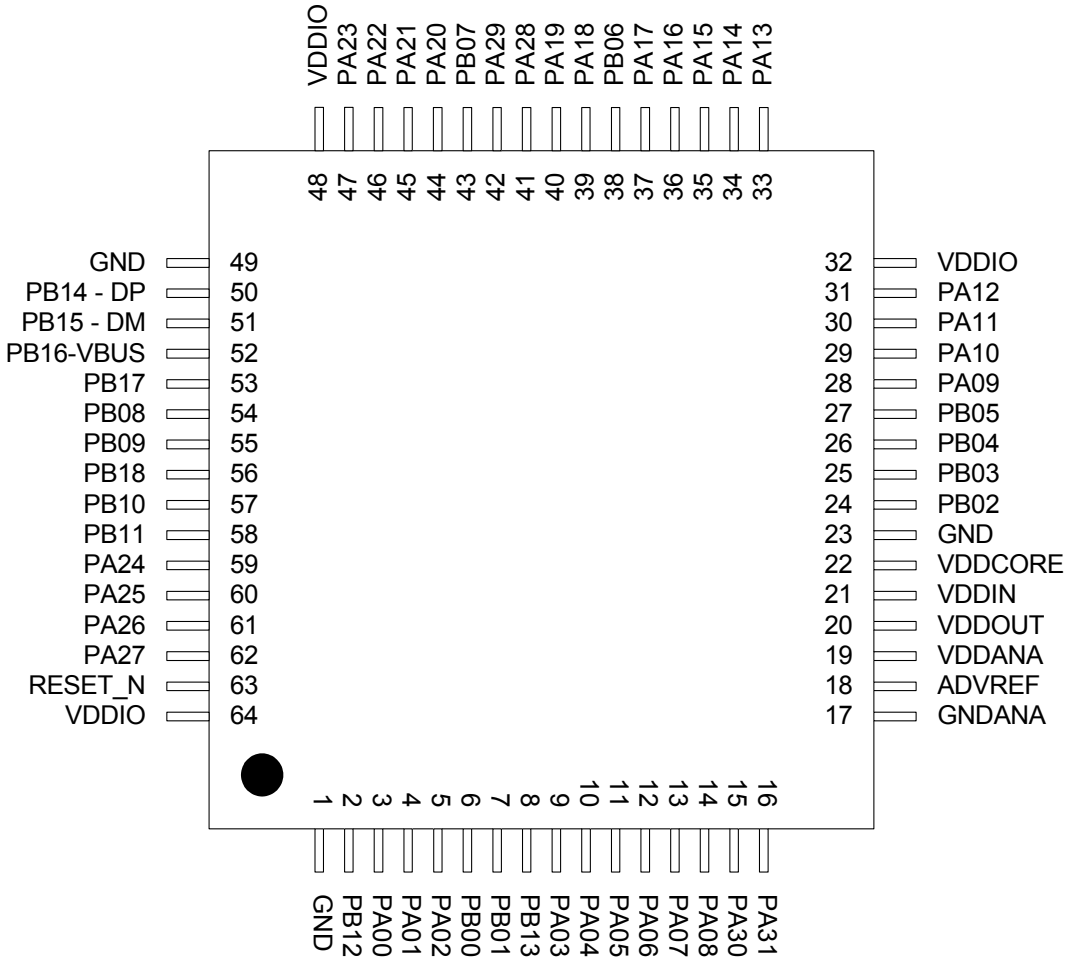


Figure 3-2. TQFP64/QFN64 Pinout



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

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. Multiplexed Signals on I/O Pins

48-pin Package	64-pin Package	PIN	GPIO	Supply	Pad Type	GPIO Function				Other Functions
						A	B	C	D	
3	3	PA00	0	VDDIO	Normal I/O	SPI - MISO	PWMA - PWMA[1]	GLOC - IN[0]	CAT - CSB[0]	JTAG-TDI
4	4	PA01	1	VDDIO	Normal I/O	SPI - MOSI	PWMA - PWMA[2]	GLOC - IN[1]	CAT - CSA[1]	JTAG-TDO
5	5	PA02	2	VDDIO	Normal I/O	SPI - SCK	PWMA - PWMA[3]	GLOC - IN[2]	CAT - CSB[1]	JTAG-TMS
7	9	PA03	3	VDDANA	Analog I/O	PKGANA - ADCIN0	SCIF - GCLK[0]	GLOC - IN[5]	CAT - CSB[2]	
8	10	PA04	4	VDDANA	Analog I/O	PKGANA - ADCIN1	SCIF - GCLK[1]	GLOC - IN[6]	CAT - CSA[3]	

Table 3-1. Multiplexed Signals on I/O Pins

48-pin Package	64-pin Package	PIN	GPIO	Supply	Pad Type	GPIO Function				Other Functions
						A	B	C	D	
9	11	PA05	5	VDDANA	Analog I/O	EIC - EXTINT[8]	PKGANA - ADCIN2	GLOC - OUT[1]	CAT - CSB[3]	
10	12	PA06	6	VDDANA	Analog I/O	EIC - EXTINT[1]	PKGANA - ADCIN3	GLOC - IN[7]	CAT - CSA[4]	
11	13	PA07	7	VDDANA	Analog I/O	PWMA - PWMA[0]	PKGANA - ADCIN4	GLOC - IN[8]	CAT - CSB[4]	
12	14	PA08	8	VDDANA	Analog I/O	PWMA - PWMA[1]	PKGANA - ADCIN5	GLOC - IN[9]	CAT - CSA[5]	
20	28	PA09	9	VDDIO	Normal I/O, 5V tolerant	TWIMS - TWCK	SPI - NPCS[2]	USART1 - CTS	CAT - CSB[5]	
21	29	PA10	10	VDDIO	Normal I/O, 5V tolerant	TWIMS - TWD	SPI - NPCS[3]	USART1 - RTS	CAT - CSA[6]	
22	30	PA11	11	VDDIO	Normal I/O	USART0 - RTS	TC - A2	PWMA - PWMA[0]	CAT - CSB[6]	OSC32 - XIN
23	31	PA12	12	VDDIO	Normal I/O	USART0 - CTS	TC - B2	PWMA - PWMA[1]	CAT - CSA[7]	OSC32 - XOUT
25	33	PA13	13	VDDIO	Normal I/O	EIC - EXTINT[0]	PWMA - PWMA[2]	USART0 - CLK	CAT - CSB[7]	
26	34	PA14	14	VDDIO	Normal I/O	SPI - MOSI	PWMA - PWMA[3]	EIC - EXTINT[2]	CAT - CSA[8]	
27	35	PA15	15	VDDIO	Normal I/O	SPI - SCK	PWMA - PWMA[4]	USART2 - CLK	CAT - CSB[8]	
28	36	PA16	16	VDDIO	Normal I/O	SPI - NPCS[0]	TC - CLK1	PWMA - PWMA[4]	CAT - CSA[9]	
29	37	PA17	17	VDDIO	Normal I/O	SPI - NPCS[1]	TC - CLK2	SPI - SCK	CAT - CSB[9]	
30	39	PA18	18	VDDIO	Normal I/O	USART0 - RXD	PWMA - PWMA[5]	SPI - MISO	CAT - CSA[10]	OSC0 - XIN
31	40	PA19	19	VDDIO	Normal I/O	USART0 - TXD	PWMA - PWMA[6]	SPI - MOSI	CAT - CSB[10]	OSC0 - XOUT
32	44	PA20	20	VDDIO	Normal I/O	USART1 - CLK	TC - CLK0	USART2 - RXD	CAT - CSA[11]	
33	45	PA21	21	VDDIO	Normal I/O	PWMA - PWMA[2]	TC - A1	USART2 - TXD	CAT - CSB[11]	
34	46	PA22	22	VDDIO	Normal I/O	PWMA - PWMA[6]	TC - B1	ADCIFD - EXTTRIG	CAT - CSA[12]	
35	47	PA23	23	VDDIO	Normal I/O	USART1 - TXD	SPI - NPCS[1]	EIC - EXTINT[3]	CAT - CSB[12]	
43	59	PA24	24	VDDIO	Normal I/O	USART1 - RXD	SPI - NPCS[0]	EIC - EXTINT[4]	CAT - CSB[15]	
44	60	PA25	25	VDDIO	Normal I/O	SPI - MISO	PWMA - PWMA[3]	EIC - EXTINT[5]	CAT - CSA[16]	
45	61	PA26	26	VDDIO	Normal I/O	IISC - IWS	USART2 - TXD	TC - A0	CAT - CSB[16]	
46	62	PA27	27	VDDIO	Normal I/O	IISC - ISCK	USART2 - RXD	TC - B0	CAT - CSA[0]	
	41	PA28	28	VDDIO	Normal I/O	USART0 - CLK	PWMA - PWMA[4]	SPI - MISO	CAT - CSB[21]	
	42	PA29	29	VDDIO	Normal I/O	TC - CLK0	TC - CLK1	SPI - MOSI	CAT - CSA[22]	
	15	PA30	30	VDDANA	Analog I/O	PKGANA - ADCIN6	EIC - EXTINT[6]	SCIF - GCLK[2]	CAT - CSA[18]	
	16	PA31	31	VDDANA	Analog I/O	PKGANA - ADCIN7	EIC - EXTINT[7]	PWMA - PWMA[6]	CAT - CSB[18]	
	6	PB00	32	VDDIO	Normal I/O	TC - A0	EIC - EXTINT[4]	USART2 - CTS	CAT - CSA[17]	
	7	PB01	33	VDDIO	Normal I/O	TC - B0	EIC - EXTINT[5]	USART2 - RTS	CAT - CSB[17]	
	24	PB02	34	VDDIO	Normal I/O	EIC - EXTINT[6]	TC - A1	USART1 - TXD	CAT - CSA[19]	
	25	PB03	35	VDDIO	Normal I/O	EIC - EXTINT[7]	TC - B1	USART1 - RXD	CAT - CSB[19]	
	26	PB04	36	VDDIO	Normal I/O	USART1 - CTS	SPI - NPCS[3]	TC - CLK2	CAT - CSA[20]	
	27	PB05	37	VDDIO	Normal I/O	USART1 - RTS	SPI - NPCS[2]	PWMA - PWMA[5]	CAT - CSB[20]	
	38	PB06	38	VDDIO	Normal I/O	IISC - ISCK	PWMA - PWMA[5]	GLOC - IN[15]	CAT - CSA[21]	
	43	PB07	39	VDDIO	Normal I/O	IISC - ISDI	EIC - EXTINT[2]	GLOC - IN[11]	CAT - CSB[22]	
	54	PB08	40	VDDIO	Normal I/O	IISC - IWS	EIC - EXTINT[0]	GLOC - IN[14]	CAT - CSA[23]	
	55	PB09	41	VDDIO	Normal I/O	IISC - ISCK	IISC - IMCK	GLOC - IN[3]	CAT - CSB[23]	
	57	PB10	42	VDDIO	Normal I/O	IISC - ISDO	TC - A2	USART0 - RXD	CAT - CSA[24]	
	58	PB11	43	VDDIO	Normal I/O	IISC - IWS	TC - B2	USART0 - TXD	CAT - CSB[24]	

Table 3-1. Multiplexed Signals on I/O Pins

48-pin Package	64-pin Package	PIN	GPIO	Supply	Pad Type	GPIO Function				Other Functions
						A	B	C	D	
2	2	PB12	44	VDDIO	Normal I/O	SPI - NPCS[0]	IISC - IMCK	GLOC - OUT[0]		JTAG-TCK
6	8	PB13	45	VDDIO	Normal I/O	CAT - SYNC	SCIF - GCLK[2]	GLOC - IN[4]	CAT - CSA[2]	
38	50	PB14	46	VDDIO	Normal I/O	USBC - DP	USART0 - RXD	GLOC - OUT[2]	CAT - CSA[13]	
39	51	PB15	47	VDDIO	Normal I/O	USBC - DM	USART0 - TXD	GLOC - IN[12]	CAT - CSB[13]	
40	52	PB16	48	VDDIO	Input only, 5V tolerant	USBC - VBUS		GLOC - IN[10]		USB-VBUS
41	53	PB17	49	VDDIO	Normal I/O	IISC - ISDO	USART0 - RTS	GLOC - IN[13]		
42	56	PB18	50	VDDIO	Normal I/O	IISC - ISDI	CAT - SYNC	GLOC - OUT[3]	CAT - CSA[15]	

See Section 4. for a description of the various peripheral signals.

Refer to "Electrical Characteristics" on page 37 for a description of the electrical properties of the pad types used.

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.

Table 3-2. Peripheral Functions

Function	Description
A	GPIO peripheral selection A
B	GPIO peripheral selection B
C	GPIO peripheral selection C
D	GPIO peripheral selection D

3.2.3 JTAG Port Connections

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

Table 3-3. JTAG Pinout

48-pin or 64-pin Package	Pin Name	JTAG Pin
2	PB12	TCK
5	PA02	TMS
4	PA01	TDO
3	PA00	TDI

3.2.4 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 3-4. Oscillator Pinout

48-pin Package	64-pin Package	Pin	Oscillator Function
30	39	PA18	XIN0
31	40	PA19	XOUT0
22	30	PA11	XIN32
23	31	PA12	XOUT32

3.2.5 Other Functions

The functions listed in Table 3-5 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 3-5. Other Functions

48-Pin Package	64-Pin Package	Pin	Function
47	63	RESET_N	aWire DATA
2	2	PB12	aWire DATAOUT

4. Signal Descriptions

The following table gives details on signal name classified by peripheral.

Table 4-1. Signal Descriptions List

Signal Name	Function	Type	Active Level	Comments
aWire - AW				
DATA	aWire data	I/O		
DATAOUT	aWire data output for 2-pin mode	I/O		
External Interrupt Controller - EIC				
NMI	Non-Maskable Interrupt	Input		
EXTINT8 - EXTINT1	External interrupt	Input		
JTAG module - JTAG				
TCK	Test Clock	Input		
TDI	Test Data In	Input		
TDO	Test Data Out	Output		
TMS	Test Mode Select	Input		
Power Manager - PM				
RESET_N	Reset	Input	Low	
Basic Pulse Width Modulation Controller - PWMA				
PWMA6 - PWMA0	PWMA channel waveforms	Output		
System Control Interface - SCIF				
GCLK2 - GCLK0	Generic clock	Output		
XIN0	Oscillator 0 XIN Pin	Analog		
XOUT0	Oscillator 0 XOUT Pin	Analog		
XIN32	32K Oscillator XIN Pin	Analog		
XOUT32	32K Oscillator XOUT Pin	Analog		
Serial Peripheral Interface - SPI				
MISO	Master In Slave Out	I/O		
MOSI	Master Out Slave In	I/O		
NPCS3 - NPCS0	SPI Peripheral Chip Select	I/O	Low	
SCK	Clock	I/O		
Timer/Counter - TC				
A0	Channel 0 Line A	I/O		
A1	Channel 1 Line A	I/O		

Table 4-1. Signal Descriptions List

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 Master- TWIM				
TWCK	Two-wire Serial Clock			
TWD	Two-wire Serial Data			
Two Wire Interface Slave- TWIS				
TWCK	Two-wire Serial Clock			
TWD	Two-wire Serial Data			
Universal Synchronous/Asynchronous Receiver/Transmitter - USART0/1/2				
CLK	Clock	I/O		
CTS	Clear To Send	Input	Low	
RTS	Request To Send	Output	Low	
RXD	Receive Data	Input		
TXD	Transmit Data	Output		
Universal Serial Bus 2.0 Full Speed Interface - USBC				
DM	DM for USB FS			
DP	DP for USB FS			
VBUS	VBUS			
IIS Controller - IISC				
IBCK	IIS Serial Clock	I/O		
ISDI	IIS Serial Data In	Input		
ISDO	IIS Serial Data Out	Output		
IWS	IIS Word Select	I/O		
IMCK	IIS Master Clock	Output		
Capacitive Touch Sensor - CAT				
CSA24 - CSA0	Capacitive Sensor Group A	I/O		
CSB24 - CSB0	Capacitive Sensor Group B	I/O		
SYNC	Synchronize signal	Input		
Glue Logic Controller - GLOC				
IN15 - IN0	Inputs to lookup tables	Input		
OUT3 - OUT0	Outputs from lookup tables	Output		
ADC controller interface - ADCIFD				

Table 4-1. Signal Descriptions List

EXTTRIG	ADCIFD EXTTRIG	Input		
AD7 - AD0	ADC Inputs	Analog		
Power				
VDDIO	Digital I/O Power Supply	Power Input		3.0 V to 3.6V.
VDDANA	Analog Power Supply	Power Input		3.0 V to 3.6V
ADVREF	Analog Reference Voltage	Power Input		2.6 V to 3.6 V
VDDCORE	Core Power Supply	Power Input		1.65 V to 1.95 V
VDDIN	Voltage Regulator Input	Power Input		3.0 V to 3.6V
VDDOUT	Voltage Regulator Output	Power Output		1.65 V to 1.95V
GNDANA	Analog Ground	Ground		
GND	Ground	Ground		
General Purpose I/O pin - GPIOA, GPIOB				
PA31 - PA00	General Purpose I/O Controller GPIO A	I/O		
PB18 - PB00	General Purpose I/O Controller GPIO B	I/O		

4.1 I/O Line Considerations

4.1.1 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. TDO pin is an output, driven at VDDIO, and has no pull-up resistor. These JTAG pins can be used as GPIO pins and muxed with peripherals when the JTAG is disabled.

4.1.2 RESET_N Pin

The RESET_N pin is a schmitt input and integrates a programmable pull-up resistor to VDDIO. As the product integrates a power-on reset detector, 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 the application.

4.1.3 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.

4.1.4 GPIO Pins

All the I/O lines integrate a pull-up resistor. Programming of this pull-up resistor is performed

independently for each I/O line through the GPIO Controller. After reset, I/O lines default as inputs with pull-up resistors disabled.

4.1.5 High drive pins

Four I/O lines can be used to drive twice current than other I/O capability (see Electrical Characteristics section).

48-pin Package	64-pin Package	Pin Name
32	44	PA20
33	45	PA21
34	46	PA22
35	47	PA23

4.2 Power Considerations

4.2.1 Power Supplies

The UC3D has several types of power supply pins:

- VDDIO: Powers Digital I/O lines. Voltage is 3.3V nominal.
- VDDIN: Powers the internal regulator. Voltage is 3.3V nominal.
- VDDCORE : Powers the internal core digital logic. Voltage is 1.8 V nominal.
- VDDANA: Powers the ADC and Analog I/O lines. Voltage is 3.3V nominal.

The ground pins GND is dedicated to VDDIO and VDDCORE. The ground pin for VDDANA is GNDANA.

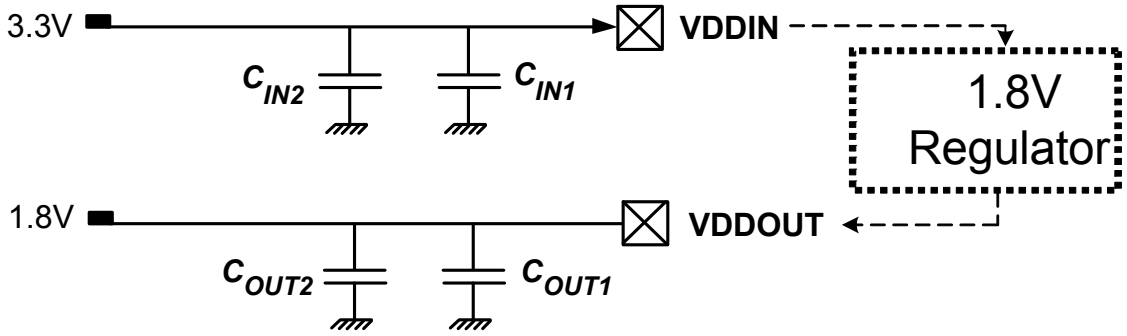
Refer to ”Electrical Characteristics” on page 37 for power consumption on the various supply pins.

4.2.2 Voltage Regulator

The UC3D embeds a voltage regulator that converts from 3.3V nominal to 1.8V with a load of up to 100 mA. The regulator is intended to supply the logic, memories, oscillators and PLLs. See Section 4.2.3 for regulator connection figures.

Adequate output supply decoupling is mandatory on VDDOUT to reduce ripple and avoid oscillations. The best way to achieve this is to use two capacitors in parallell between VDDOUT and GND as close to the chip as possible. Please refer to Section 8.9.1 for decoupling capacitors values and regulator characteristics. VDDOUT can be connected externally to the 1.8V domains to power external components.

Figure 4-1. Supply Decoupling



For decoupling recommendations for VDDIO, VDDANA and VDDCORE, please refer to the Schematic checklist.

4.2.3 Regulator Connection

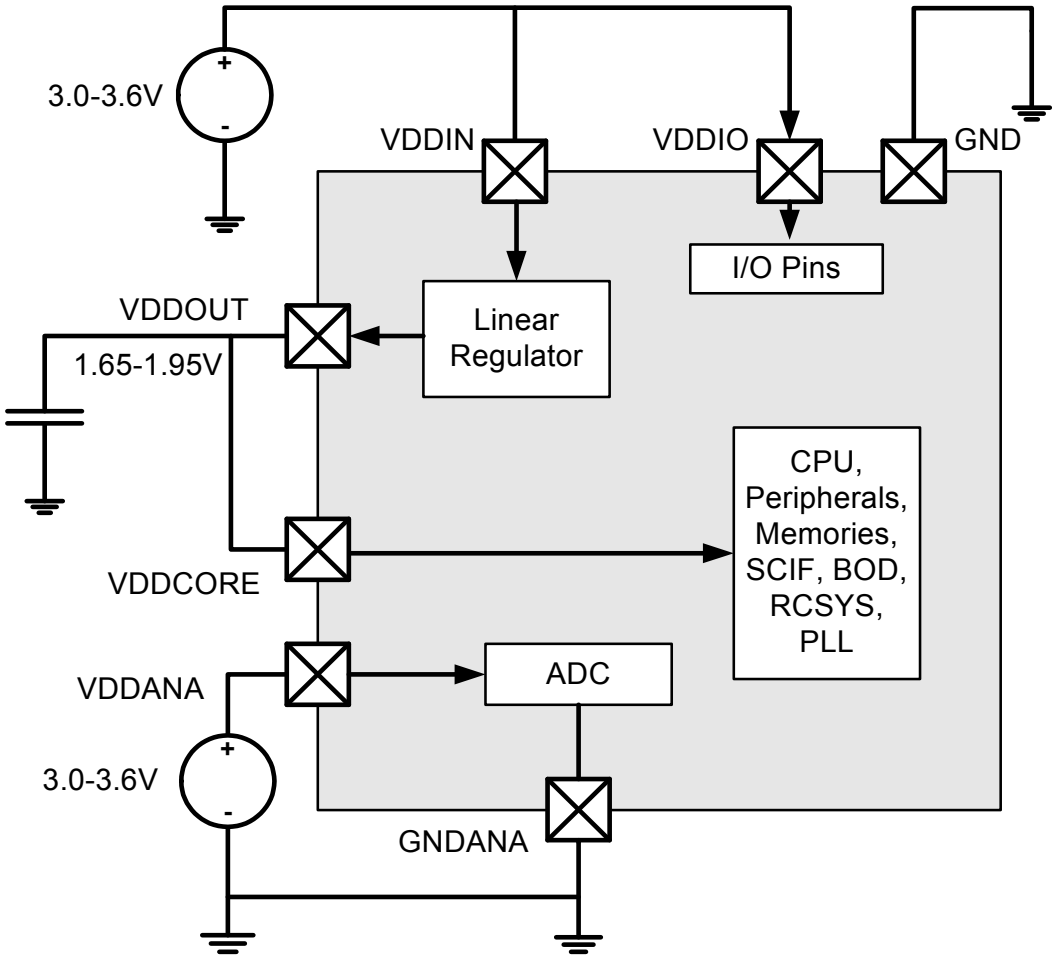
The UC3D supports two power supply configurations:

- 3.3V single supply mode
- 3.3V - 1.8V dual supply mode

4.2.3.1 3.3V Single Supply Mode

In 3.3V single supply mode the internal regulator is connected to the 3.3V source (VDDIN pin). The regulator output (VDDOUT) needs to be externally connected to VDDCORE pin to supply internal logic. Figure 4-2 shows the power schematics to be used for 3.3V single supply mode.

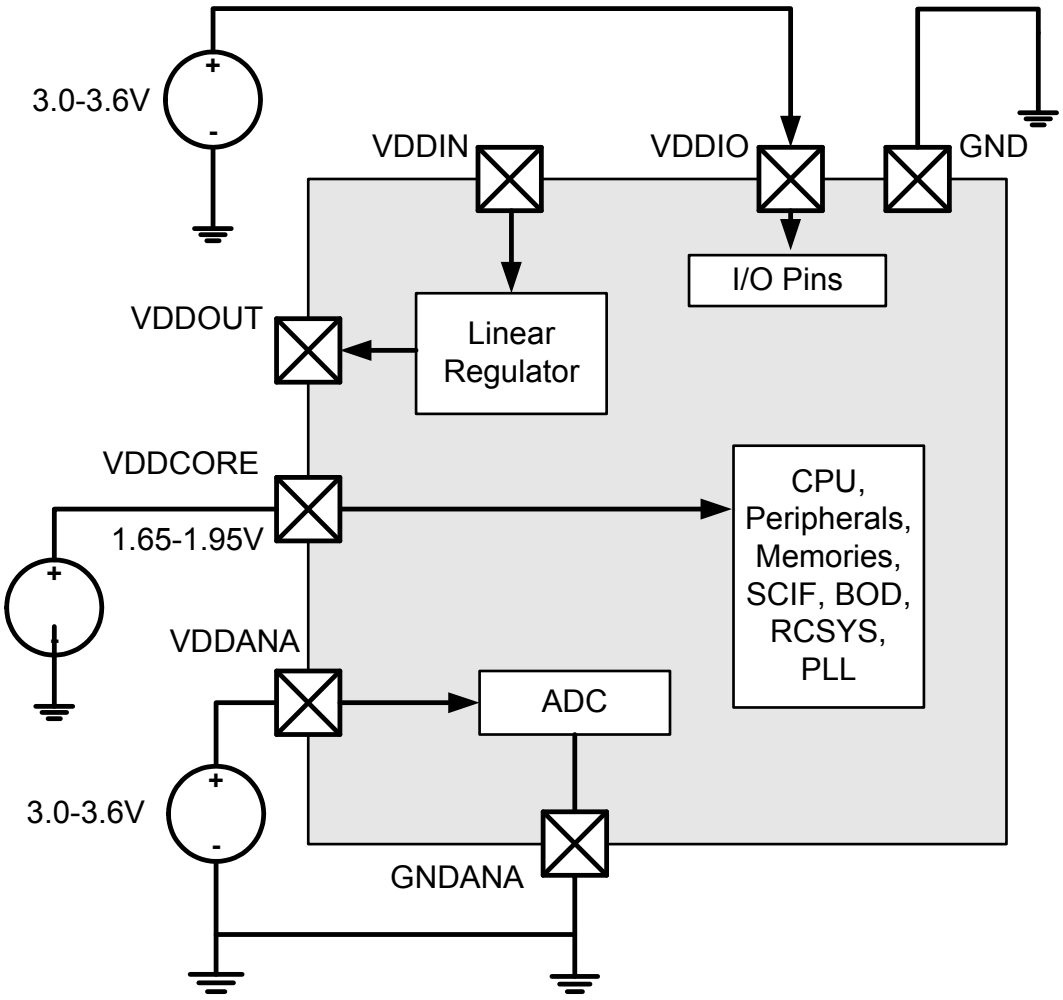
Figure 4-2. 3.3V Single Power Supply mode



4.2.3.2 *3.3V + 1.8V Dual Supply Mode*

In dual supply mode the internal regulator is not used (unconnected), VDDIO is powered by 3.3V supply and VDDCORE is powered by a 1.8V supply as shown in Figure 4-3.

Figure 4-3. 3.3V + 1.8V Dual Power Supply Mode.



4.2.4 Power-up Sequence

4.2.4.1 *Maximum Rise Rate*

To avoid risk of latch-up, the rise rate of the power supplies must not exceed the values described in Supply Characteristics table in the Electrical Characteristics chapter.

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

4.2.4.2 *Minimum Rise Rate*

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

See Supply Characteristics table in the Electrical Characteristics chapter 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, one of the following configuration can be used:

- A logic “0” value is applied during power-up on pin RESET_N until VDDIN rises above 1.2V.

5. 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, and instruction set is presented. For further details, see the *AVR32 Architecture Manual* and the *AVR32UC Technical Reference Manual*.

5.1 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**

5.2 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.

5.3 The AVR32UC CPU

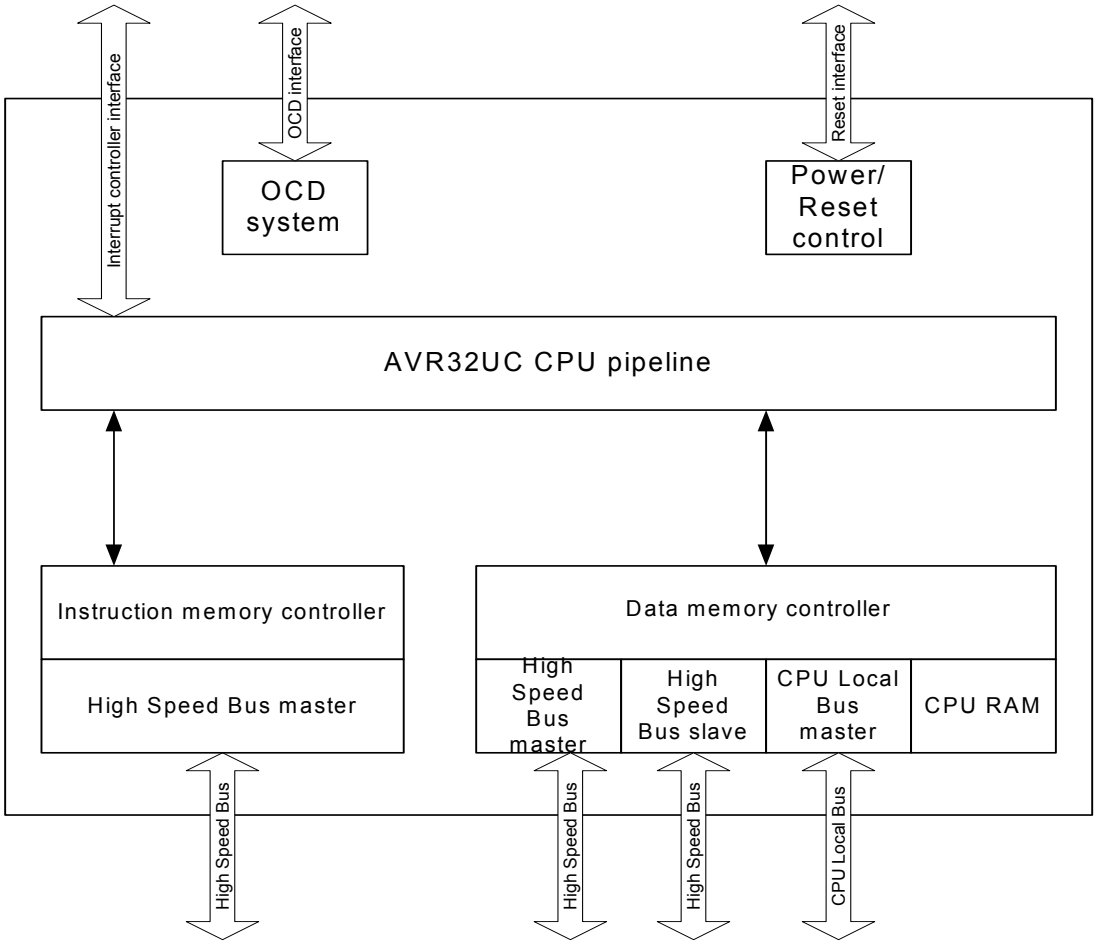
The AVR32UC CPU targets low- and medium-performance applications, and provides an advanced On-Chip Debug (OCD) system, and no caches. Java acceleration hardware is not implemented.

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 5-1 on page 22 displays the contents of AVR32UC.

Figure 5-1. Overview of the AVR32UC CPU



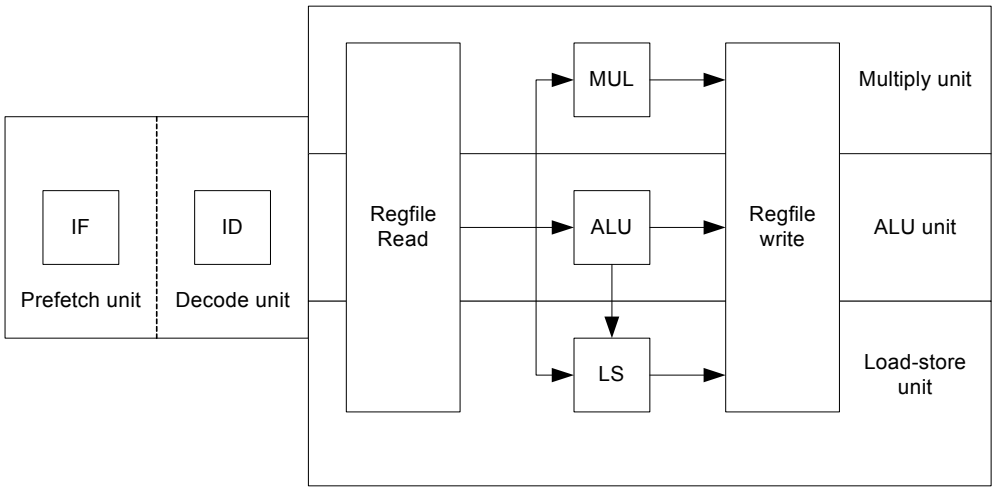
5.3.1 Pipeline Overview

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 5-2 on page 23 shows an overview of the AVR32UC pipeline stages.

Figure 5-2. The AVR32UC Pipeline



5.3.2 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.

5.3.2.1 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.

5.3.2.2 Java Support

AVR32UC does not provide Java hardware acceleration.

5.3.2.3 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 5-1. Instructions with Unaligned Reference Support

Instruction	Supported Alignment
ld.d	Word
st.d	Word

5.3.2.4 *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

5.3.2.5 *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.

5.4 Programming Model

5.4.1 Register File Configuration

The AVR32UC register file is shown below.

Figure 5-3. 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
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_SEC
R12	R12	R12	R12	R12	R12	R12	R12	R12
R11	R11	R11	R11	R11	R11	R11	R11	R11
R10	R10	R10	R10	R10	R10	R10	R10	R10
R9	R9	R9	R9	R9	R9	R9	R9	R9
R8	R8	R8	R8	R8	R8	R8	R8	R8
R7	R7	R7	R7	R7	R7	R7	R7	R7
R6	R6	R6	R6	R6	R6	R6	R6	R6
R5	R5	R5	R5	R5	R5	R5	R5	R5
R4	R4	R4	R4	R4	R4	R4	R4	R4
R3	R3	R3	R3	R3	R3	R3	R3	R3
R2	R2	R2	R2	R2	R2	R2	R2	R2
R1	R1	R1	R1	R1	R1	R1	R1	R1
R0	R0	R0	R0	R0	R0	R0	R0	R0
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

5.4.2 Status Register Configuration

The Status Register (SR) is split into two halfwords, one upper and one lower, see Figure 5-4 and Figure 5-5. 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 5-4. The Status Register High Halfword

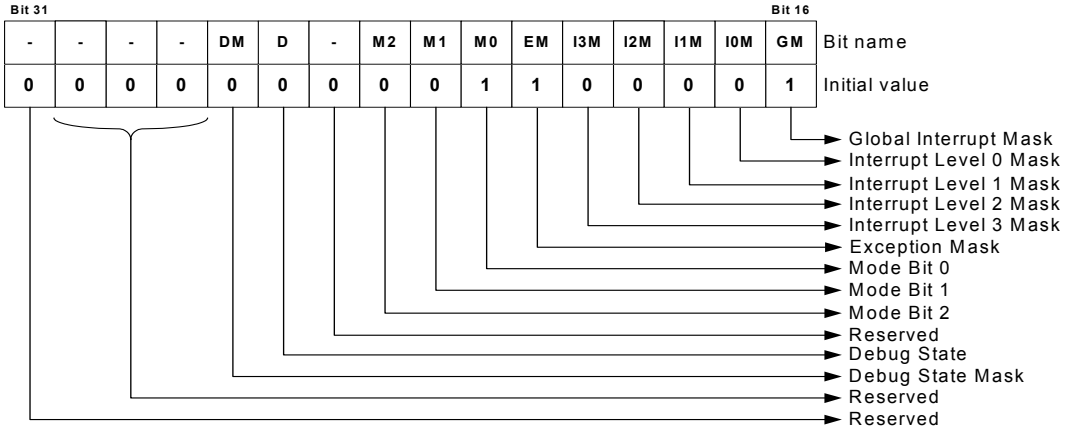
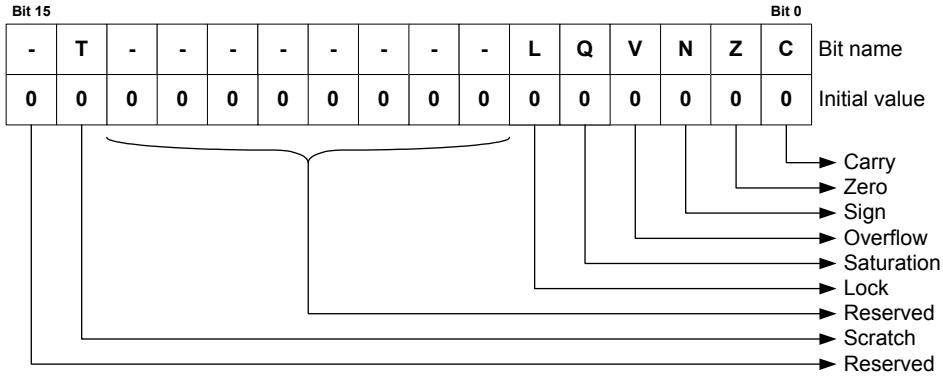


Figure 5-5. The Status Register Low Halfword



5.4.3 Processor States

5.4.3.1 Normal RISC State

The AVR32 processor supports several different execution contexts as shown in Table 5-2.

Table 5-2. 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.

5.4.3.2 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.

5.4.4 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 5-3. 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
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

Table 5-3. System Registers (Continued)

Reg #	Address	Name	Function
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
90-102	360-408	Reserved	Reserved for future use
103-111	412-444	Reserved	Reserved for future use
112-191	448-764	Reserved	Reserved for future use
192-255	768-1020	IMPL	IMPLEMENTATION DEFINED

5.5 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 5-4 on page 32. 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.

5.5.1 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.

5.5.2 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 responsability 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 5-4 on page 32, 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.

5.5.3 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.

5.5.4 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.

5.5.5 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.

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.

If several events occur on the same instruction, they are handled in a prioritized way. The priority ordering is presented in Table 5-4 on page 32. 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 5-4 on page 32. Some of the exceptions are unused in AVR32UC since it has no MMU, coprocessor interface, or floating-point unit.

Table 5-4. 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			
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			
14	EVBA+0x18			
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			
25	EVBA+0x70			
26	EVBA+0x3C			
27	EVBA+0x40			
28	EVBA+0x44			

6. Memories

6.1 Embedded Memories

- **Internal High-Speed Flash**
  - **128Kbytes (ATUC128D)**
  - **64Kbytes (ATUC64D)**
    - **0 Wait State Access at up to 24 MHz in Worst Case Conditions**
    - **1 Wait State Access at up to 48 MHz in Worst Case Conditions**
    - **Pipelined Flash Architecture, allowing burst reads from sequential Flash locations, hiding penalty of 1 wait state access**
    - **100 000 Write Cycles, 15-year Data Retention Capability**
    - **4ms Page Programming Time, 8 ms Chip Erase Time**
    - **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**
  - **16Kbytes**

6.2 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 6-1. UC3D Physical Memory Map

Device		Embedded SRAM	Embedded Flash	HSB-PB Bridge A	HSB-PB Bridge B
Start Address		0x0000_0000	0x8000_0000	0xFFFF_0000	0xFFFE_0000
Size	ATUC128D	16 Kbytes	128 Kbytes	64 Kbytes	64 Kbytes
	ATUC64D	16 Kbytes	64 Kbytes	64 Kbytes	64 Kbytes

6.3 Peripheral Address Map

Table 6-2. Peripheral Address Mapping

Address		Peripheral Name
0xFFFE0000	USBC	USB 2.0 Interface - USBC
0xFFFE1000	HMATRIX	HSB Matrix - HMATRIX
0xFFFE1400	FLASHCDW	Flash Controller - FLASHCDW
0xFFFF0000	PDCA	Peripheral DMA Controller - PDCA
0xFFFF1000	INTC	Interrupt controller - INTC

Table 6-2. Peripheral Address Mapping

0xFFFF1400	PM	Power Manager - PM
0xFFFF1800	AST	Asynchronous Timer - AST
0xFFFF1C00	WDT	Watchdog Timer - WDT
0xFFFF2000	EIC	External Interrupt Controller - EIC
0xFFFF2800	GPIO	General Purpose Input/Output Controller - GPIO
0xFFFF3000	USART0	Universal Synchronous/Asynchronous Receiver/Transmitter - USART0
0xFFFF3400	USART1	Universal Synchronous/Asynchronous Receiver/Transmitter - USART1
0xFFFF3800	USART2	Universal Synchronous/Asynchronous Receiver/Transmitter - USART2
0xFFFF3C00	SPI	Serial Peripheral Interface - SPI
0xFFFF4000	TWIM	Two-wire Master Interface - TWIM
0xFFFF4400	TWIS	Two-wire Slave Interface - TWIS
0xFFFF4800	PWMA	Pulse Width Modulation Controller - PWMA
0xFFFF4C00	IISC	Inter-IC Sound (I2S) Controller - IISC
0xFFFF5000	TC	Timer/Counter - TC
0xFFFF5400	ADCIFD	ADC controller interface - ADCIFD
0xFFFF5800	SCIF	System Control Interface - SCIF
0xFFFF5C00	FREQM	Frequency Meter - FREQM
0xFFFF6000	CAT	Capacitive Touch Module - CAT

Table 6-2. Peripheral Address Mapping

0xFFFF6400	GLOC	Glue Logic Controller - GLOC
0xFFFF6800	AW	aWire - AW

6.4 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 6-3. 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	Write-only
		SET	0x40000144	Write-only
		CLEAR	0x40000148	Write-only
		TOGGLE	0x4000014C	Write-only
	Output Value Register (OVR)	WRITE	0x40000150	Write-only
		SET	0x40000154	Write-only
		CLEAR	0x40000158	Write-only
		TOGGLE	0x4000015C	Write-only
	Pin Value Register (PVR)	-	0x40000160	Read-only

7. Boot Sequence

This chapter summarizes the boot sequence of the UC3D. The behavior after power-up is controlled by the Power Manager. For specific details, refer to the Power Manager chapter.

7.1 Starting of Clocks

After power-up, the device will be held in a reset state by the Power-On Reset circuitry for a short time to allow the power to stabilize throughout the device. After reset, the device will use the System RC Oscillator (RCSYS) as clock source.

On system start-up, 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 System RC Oscillator.

7.2 Fetching of Initial Instructions

After reset has been released, the AVR32UC CPU starts fetching instructions from the reset address, which is 0x80000000. This address points to the first address in the internal Flash.

The code read from the internal Flash is free to configure the system to divide the frequency of the clock routed to some of the peripherals, and to gate the clocks to unused peripherals.

8. Electrical Characteristics

8.1 Disclaimer

All values in this chapter are preliminary and subject to change without further notice.

8.2 Absolute Maximum Ratings*

Table 8-1. Absolute Maximum Ratings

Operating temperature ..... -40°C to +85°C
Storage temperature ..... -60°C to +150°C
Voltage on input pins (except for 5V pins) with respect to ground .....-0.3V to V_{VDD}[2]+0.3V
Voltage on 5V tolerant[1] pins with respect to ground .....-0.3V to 5.5V
Total DC output current on all I/O pins - VDDIO 152mA
Total DC output current on all I/O pins - VDDANA..... 152mA
Maximum operating voltage VDDCORE..... 1.95V
Maximum operating voltage VDDIO, VDDIN 3.6V

*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. 5V tolerant pins, see Section 3.2 "Peripheral Multiplexing on I/O lines" on page 8

2. V_{VDD} corresponds to either V_{VDDIN} or V_{VDDIO}, depending on the supply for the pin. Refer to Section 3.2 on page 8 for details.

8.3 Supply Characteristics

The following characteristics are applicable to the operating temperature range: T_A = -40°C to 85°C, unless otherwise specified and are certified for a junction temperature up to T_J = 100°C.

Table 8-2. Supply Characteristics

Symbol	Parameter	Voltage		
		Min	Max	Unit
V_{VDDIO}	DC supply peripheral I/Os	3.0	3.6	V
V_{VDDIN}	DC supply internal regulator, 3.3V single supply mode	3.0	3.6	V
$V_{VDDCORE}$	DC supply core	1.65	1.95	V
V_{VDDANA}	Analog supply voltage	3.0	3.6	V
$V_{ADVREFP}$	Analog reference voltage	2.6	V_{VDDANA}	V

8.4 Maximum Clock Frequencies

These parameters are given in the following conditions:

- $V_{VDDCORE}$ = 1.65 to 1.95V

- Temperature = -40°C to 85°C

Table 8-3. Clock Frequencies

Symbol	Parameter	Conditions	Min	Max	Units
f_{CPU}	CPU clock frequency			48	MHz
f_{PBA}	PBA clock frequency			48	MHz
f_{PBB}	PBB clock frequency			48	MHz
f_{GCLK0}	GCLK0 clock frequency	GLOC, GCLK0 pin		48	MHz
f_{GCLK1}	GCLK1 clock frequency	GCLK1 pin		48	MHz
f_{GCLK2}	GCLK2 clock frequency	GCLK2 pin		48	MHz
f_{GCLK3}	GCLK3 clock frequency	USB		48	MHz
f_{GCLK4}	GCLK4 clock frequency	PWMA		150	MHz
f_{GCLK5}	GCLK5 clock frequency	IISC		48	MHz
f_{GCLK6}	GCLK6 clock frequency	AST		80	MHz
f_{GCLK8}	GCLK8 clock frequency	ADCIFB		48	MHz

8.5 Power Consumption

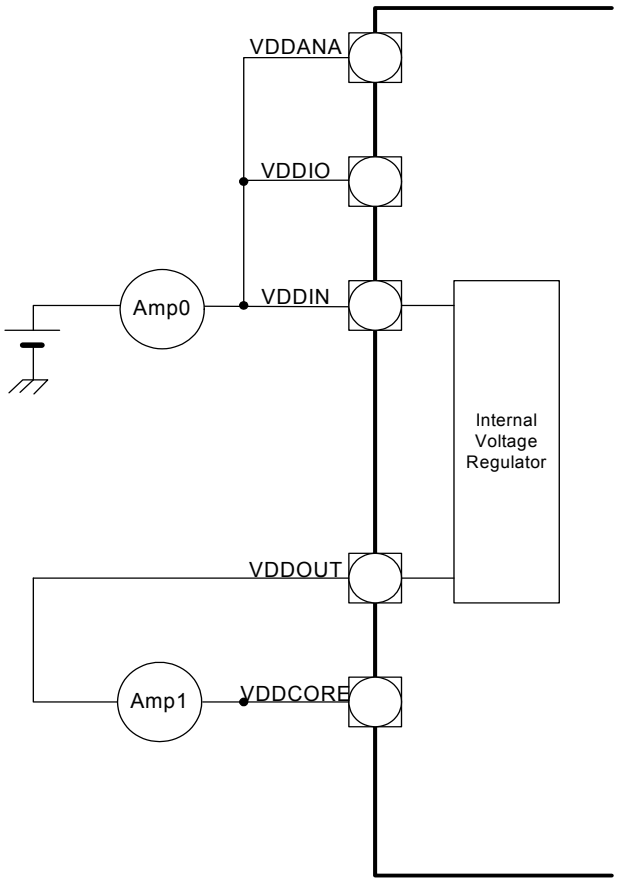
The values in Table 8-4 are measured values of power consumption under the following conditions, except where noted:

- Operating conditions internal core supply (Figure 8-1) - this is the default configuration
  - V_{VDDIN} = 3.3V
  - $V_{VDDCORE}$ = 1.8V, supplied by the internal regulator
  - Corresponds to the 3.3V supply mode with 1.8 V regulated I/O lines, please refer to the Supply and Startup Considerations section for more details
  - The following peripheral clocks running
- T_A = 25°C
- Oscillators
  - OSC0 running (external clock)as reference
  - PLL running at 48MHz with OSC0 as reference
- Clocks
  - PLL used as main clock source
  - CPU, HSB, and PBB clocks undivided
  - PBA clock divided by 4
  - The following peripheral clocks running
    - PM, SCIF, AST, FLASHCDW, PBA bridge
  - All other peripheral clocks stopped
- I/Os are inactive with internal pull-up

Table 8-4. Power Consumption for Different Operating Modes

Mode	Conditions		Consumption Typ	Unit
Active	- CPU running a recursive Fibonacci Algorithm from flash and clocked from PLL0 at f MHz. - Voltage regulator is on. - XIN0: external clock. - All peripheral clocks activated with a division by 4. - GPIOs are inactive with internal pull-up, JTAG unconnected with external pull-up and Input pins are connected to GND		0.3105xf(MHz) + 0.2707	mA/MHz
	Same conditions at 48MHz		15.17	mA
Idle	See Active mode conditions		0.1165xf(MHz) + 0.1457	mA/MHz
	Same conditions at 48MHz		5.74	mA
Frozen	See Active mode conditions		0.0718xf(MHz) + 0.0903	mA/MHz
	Same conditions at 48MHz		3.54	mA
Standby	See Active mode conditions		0.0409xf(MHz) + 0.0935	mA/MHz
	Same conditions at 48MHz		2.06	mA
Stop	- CPU running in sleep mode - XIN0, Xin1 and XIN32 are stopped. - All peripheral clocks are desactived. - GPIOs are inactive with internal pull-up, JTAG unconnected with external pull-up and Input pins are connected to GND.	Voltage Regulator On	60	µA
		Voltage Regulator Off	51	µA
Deepstop	See Stop mode conditions	Voltage Regulator On	26	µA
		Voltage Regulator Off	17	µA
Static	See Stop mode conditions	Voltage Regulator On	13	µA
		Voltage Regulator Off	3.5	µA

Figure 8-1. Measurement Schematic, External Core Supply



8.5.1 Peripheral Power Consumption

The values in Table 8-5 are measured values of power consumption under the following conditions.

- Operating conditions external core supply (Figure 8-1)
  - V_{VDDIN} = 3.3V
  - $V_{VDDCORE}$ = 1.8V, supplied by the internal regulator
  - Corresponds to the 3.3V + 1.8V dual supply mode , please refer to the Supply and Startup Considerations section for more details
- T_A = 25°C
- Oscillators
  - OSC0 on external clock running
  - PLL running at 48MHz with OSC0 as reference
- Clocks
  - OSC0 external clock used as main clock source
  - CPU, HSB, and PB clocks undivided

- I/Os are inactive with internal pull-up

Consumption idle is the added current consumption when turning the module clock on and the module is uninitialized. 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 8-5. Typical Current Consumption by Peripheral

Peripheral	Typ Consumption Active	Unit
ADCIFD[1]	3.6	µA/MHz
AST	4.5	
AW USART	9.8	
CAT	14	
EIC	2.3	
FREQM	1.1	
GLOC	1.3	
GPIO	10.6	
IISC	4.7	
PWMA	5.6	
SPI	6.3	
TC	7.3	
TWIM	4.5	
TWIS	2.8	
USART	3.9	
WDT	1.8	

Notes: 1. Includes the current consumption on VDDANA and ADVREFP.

8.6 I/O Pin Characteristics

Table 8-6. Normal I/O Pin Characteristics[1]

Symbol	Parameter	Condition		Min	Typ	Max	Units
R_{PULLUP}	Pull-up resistance			9	15	25	kOhm
V_{IL}	Input low-level voltage	V_{VDD} = 3.0V	[3]	-0.3		+0.8	V
			[4]	-0.3		+0.4	V
V_{IH}	Input high-level voltage	V_{VDD} = 3.6V	[3]	+2		V_{VDD} + 0.3	V
			[4]	+1.6		V_{VDD} + 0.3	V
V_{OL}	Output low-level voltage	V_{VDD} = 3.0V, I_{OL} = 4mA				0.4	V
V_{OH}	Output high-level voltage	V_{VDD} = 3.0V, I_{OH} = 4mA		V_{VDD} - 0.4			V

Table 8-6. Normal I/O Pin Characteristics[1]

Symbol	Parameter	Condition		Min	Typ	Max	Units
I_{OL}	Output low-level current	V_{VDD} = 3.0V	[5]			4	mA
			[6]			8	mA
I_{OH}	Output high-level current	V_{VDD} = 3.0V	[5]			4	mA
			[6]			8	mA
F_{MAX}	Output frequency[2]	V_{VDD} = 3.0V, load = 10 pF	[5]			195	MHz
			[6]			348	MHz
		V_{VDD} = 3.0V, load = 30 pF	[5]			78	MHz
			[6]			149	MHz
t_{RISE}	Rise time[2]	V_{VDD} = 3.0V, load = 10 pF	[5]			2.21	ns
			[6]			1.26	ns
		V_{VDD} = 3.0V, load = 30 pF	[5]			5.45	ns
			[6]			2.88	ns
t_{FALL}	Fall time[2]	V_{VDD} = 3.0V, load = 10 pF	[5]			2.57	ns
			[6]			1.44	ns
		V_{VDD} = 3.0V, load = 30 pF	[5]			6.41	ns
			[6]			3.35	ns
I_{LEAK}	Input leakage current	Pull-up resistors disabled				1	µA
C_{IN}	Input capacitance,	[7]			2		pF
		PA09, PA10			16.5		pF
		PA11, PA12, PA18, PA19			18.5		pF
		PB14, PB15			5		pF

Notes:
1. V_{VDD} corresponds to either V_{VDDIN} or V_{VDDIO}, depending on the supply for the pin. Refer to Section 3.2 on page 8 for details.
2. 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.
3. This applies to all normal drive pads except PB13, PB17 and PB18.
4. This applies to PB13, PB17 and PB18 pads only.
5. This applies to all normal drive pad except PA00, PA01, PA02, PA03, PA04, PA05, PA06, PA07, PA08, PA09, PA10, PA11, PA12, PA13, PA18, PA19, PA27, PA30, PA31, PB13, PB16 and RESET_N.
6. This applies to PA00, PA01, PA02, PA03, PA04, PA05, PA06, PA07, PA08, PA09, PA10, PA11, PA12, PA13, PA18, PA19, PA27, PA30, PA31, PB13, PB16 and RESET_N pads only.
7. This applies to all normal drive pads except PA09, PA10, PA11, PA12, PA18, PA19, PB14, PB15.

Table 8-7. High-drive I/O Pin Characteristics[1]

Symbol	Parameter	Condition	Min	Typ	Max	Units
R_{PULLUP}	Pull-up resistance		9	15	25	kOhm
V_{IL}	Input low-level voltage	V_{VDD} = 3.0V	-0.3		+0.8	V
V_{IH}	Input high-level voltage	V_{VDD} = 3.6V	+2		V_{VDD} + 0.3	V
V_{OL}	Output low-level voltage	V_{VDD} = 3.0V, I_{OL} = 6mA			0.4	V

Table 8-7. High-drive I/O Pin Characteristics[1]

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{OH}	Output high-level voltage	V_{VDD} = 3.0V, I_{OH} = 6mA	V_{VDD} - 0.4			V
I_{OL}	Output low-level current	V_{VDD} = 3.0V			16	mA
I_{OH}	Output high-level current	V_{VDD} = 3.0V			16	mA
F_{MAX}	Output frequency	V_{VDD} = 3.0V, load = 10 pF			471	MHz
		V_{VDD} = 3.0V, load = 30 pF			249	MHz
t_{RISE}	Rise time, all High-drive I/O pins	V_{VDD} = 3.0V, load = 10 pF			0.86	ns
		V_{VDD} = 3.0V, load = 30 pF			1.70	ns
t_{FALL}	Fall time	V_{VDD} = 3.0V, load = 10 pF			1.06	ns
		V_{VDD} = 3.0V, load = 30 pF			2.01	ns
I_{LEAK}	Input leakage current	Pull-up resistors disabled			1	µA
C_{IN}	Input capacitance,	TQFP48 package		2		pF

Notes: 1. V_{VDD} corresponds to either V_{VDDIN} or V_{VDDIO}, depending on the supply for the pin. Refer to Section 3.2 on page 8 for details.
2. 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 8-8. PB14-DP, PB15-DM Pins Characteristics

Symbol	Parameter	Condition	Min	Typ	Max	Units
R_{PULLUP}	Pull-up resistance		50	100	150	kOhm

Table 8-9. PB16-VBUS Pin Characteristics[1]

Symbol	Parameter	Condition	Min	Typ	Max	Units
R_{PULLUP}[3]	Pull-up resistance					kOhm
V_{IL}	Input low-level voltage	V_{VDD} = 3.0V	-0.3		+0.8	V
V_{IH}	Input high-level voltage	V_{VDD} = 3.6V	+2		V_{VDD} + 0.3	V
I_{LEAK}	Input leakage current	5.5V, pull-up resistors disabled			1	µA
C_{IN}	Input capacitance	48 pin packages		0.6		pF

Notes: 1. V_{VDD} corresponds to either V_{VDDIN} or V_{VDDIO}, depending on the supply for the pin. Refer to Section 3.2 on page 8 for details.
2. 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.
3. PB16-VBUS pad has no pull-up resistance

8.7 Oscillator Characteristics

8.7.1 Oscillator 0 (OSC0) Characteristics

8.7.1.1 Digital Clock Characteristics

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

Table 8-10. Digital Clock Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Units
f_{CPXIN}	XIN clock frequency				50	MHz
t_{CPXIN}	XIN clock duty cycle		40		60	%

8.7.1.2 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 8-2. 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:

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

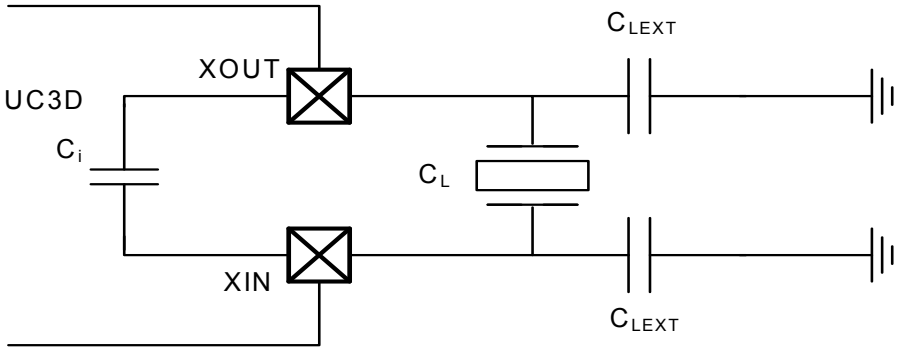
where C_{PCB} is the capacitance of the PCB.

Table 8-11. Crystal Oscillator Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$1/(t_{CPMAIN})$	Crystal oscillator frequency		0.4		20	MHz
C_L	Crystal load capacitance		6		18	pF
C_i	Internal equivalent load capacitance			1.7		pF
$t_{STARTUP}$	Startup time	400 KHz Resonator SCIF.OSCCTRL.GAIN = 0[1]		198		µs
		2 MHz Quartz SCIF.OSCCTRL.GAIN = 0[1]		4666		
		8 MHz Quartz SCIF.OSCCTRL.GAIN = 1[1]		975		
		12 MHz Quartz SCIF.OSCCTRL.GAIN = 2[1]		615		
		16 MHz Quartz SCIF.OSCCTRL.GAIN = 2[1]		1106		
		20 MHz Quartz SCIF.OSCCTRL.GAIN = 3[1]		1109		

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

Figure 8-2. Oscillator Connection



8.7.2 32KHz Crystal Oscillator (OSC32K) Characteristics

8.7.2.1 Digital Clock Characteristics

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

Table 8-12. Digital Clock Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Units
f_{CPXIN}	XIN32 clock frequency			32.768	5000	KHz
t_{CPXIN}	XIN32 clock duty cycle		40		60	%

Figure 8-2 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 8-13. 32 KHz Crystal Oscillator Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$1/(t_{CP32KHz})$	Crystal oscillator frequency			32.768	5000	KHz
t_{ST}	Startup time	R_S = 50kOhm, C_L = 9pF		2		s
C_L	Crystal load capacitance		6		15	pF
C_i	Internal equivalent load capacitance			1.4		pF

8.7.3 Phase Locked Loop (PLL) Characteristics

Table 8-14. Phase Lock Loop Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
F_{OUT}	VCO Output Frequency		80		240	MHz
F_{IN}	Input Frequency		4		16	MHz
I_{PLL}	Current Consumption	Active mode F_{VCO}@80 MHz		240		µA
		Active mode F_{VCO}@240 MHz		600		
$t_{STARTUP}$	Startup time, from enabling the PLL until the PLL is locked	Wide Bandwith mode disabled		15		µs
		Wide Bandwith mode enabled		45		

8.7.4 120 MHz RC Oscillator (RC120M) Characteristics

Table 8-15. Internal 120 MHz 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.

8.7.5 System RC Oscillator (RCSYS) Characteristics

Table 8-16. System RC Oscillator Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{OUT}	Output frequency	Calibrated point Ta = 85°C	110	115.2	116	kHz
		Ta = 25°C	105	109	115	kHz
		Ta = -40°C	100	104	108	kHz

8.8 Flash Characteristics

Table 8-17 gives the device maximum operating frequency depending on the number of flash wait states and the flash read mode. The FSW bit in the FLASHCDW FSR register controls the number of wait states used when accessing the flash memory.

Table 8-17. Maximum Operating Frequency

Flash Wait States	Maximum Operating Frequency
1	48 MHz
0	24 MHz

Table 8-18. Flash Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
T_{FPP}	Page programming time	f_{CLK_HSB}= 48MHz		5		ms
T_{FPE}	Page erase time			5		
T_{FFP}	Fuse programming time			1		
T_{FEA}	Full chip erase time (EA)			6		
T_{FCE}	JTAG chip erase time (CHIP_ERASE)	f_{CLK_HSB}= 115kHz		310		

Table 8-19. 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)		10k			cycles
t_{RET}	Data retention		15			years

8.9 Analog Characteristics

8.9.1 Voltage Regulator Characteristics

8.9.1.1 Electrical Characteristics

Table 8-20. Electrical Characteristics

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{VDDIN}	Input voltage range		3	3.3	3.6	V
$V_{VDDCORE}$	Output voltage	V_{VDDIN} >= 3V	1.75	1.8	1.85	V
	Output voltage accuracy	I_{OUT} = 0.1mA to 100mA, V_{VDDIN}>3V		2		%
I_{OUT}	DC output current	V_{VDDIN}=3.3V			100	mA
I_{VREG}	Static current of internal regulator	Low power mode		10		µA

8.9.1.2 Decoupling Requirements

Table 8-21. Decoupling Requirements

Symbol	Parameter	Condition	Typ	Techno.	Units
C_{IN1}	Input regulator capacitor 1		1	NPO	nF

Table 8-21. Decoupling Requirements

Symbol	Parameter	Condition	Typ	Techno.	Units
C_{IN2}	Input regulator capacitor 2		4.7	X7R	nF
C_{OUT1}	Output regulator capacitor 1		470	NPO	nF
C_{OUT2}	Output regulator capacitor 2		2.2	X7R	µF

8.9.2 ADC Characteristics

Table 8-22. Channel Conversion Time and ADC Clock

Parameter	Conditions	Min.	Typ.	Max.	Unit
ADC Clock Frequency	10-bit resolution mode			5	MHz
	8-bit resolution mode			8	MHz
Startup Time	Return from Idle Mode			20	µs
Track and Hold Acquisition Time		600			ns
Conversion Time	ADC Clock = 5 MHz			2	µs
	ADC Clock = 8 MHz			1.25	µs
Throughput Rate	ADC Clock = 5 MHz			384 (1)	kSPS
	ADC Clock = 8 MHz			533 (2)	kSPS

1. Corresponds to 13 clock cycles: 3 clock cycles for track and hold acquisition time and 10 clock cycles for conversion.
2. Corresponds to 15 clock cycles: 5 clock cycles for track and hold acquisition time and 10 clock cycles for conversion.

Table 8-23. ADC Power Consumption

Parameter	Conditions	Min.	Typ.	Max.	Unit
Current Consumption on VDDANA (1)	On 13 samples with ADC clock = 5 MHz			1.25	mA

1. Including internal reference input current

Table 8-24. Analog Inputs

Parameter	Conditions	Min.	Typ.	Max.	Unit
Input Voltage Range		0		VDDANA	V
Input Leakage Current				1	µA
Input Capacitance			7		pF
Input Resistance			370	810	Ohm

Table 8-25. Transfer Characteristics in 8-bit mode

Parameter	Conditions	Min.	Typ.	Max.	Unit
Resolution			8		Bit
Absolute Accuracy	ADC Clock = 5 MHz			0.8	LSB
	ADC Clock = 8 MHz			1.5	LSB
Integral Non-linearity	ADC Clock = 5 MHz		0.35	0.5	LSB
	ADC Clock = 8 MHz		0.5	1.0	LSB

Table 8-25. Transfer Characteristics in 8-bit mode

Parameter	Conditions	Min.	Typ.	Max.	Unit
Differential Non-linearity	ADC Clock = 5 MHz		0.3	0.5	LSB
	ADC Clock = 8 MHz		0.5	1.0	LSB
Offset Error	ADC Clock = 5 MHz	-0.5		0.5	LSB
Gain Error	ADC Clock = 5 MHz	-0.5		0.5	LSB

Table 8-26. Transfer Characteristics in 10-bit mode

Parameter	Conditions	Min.	Typ.	Max.	Unit
Resolution			10		Bit
Absolute Accuracy	ADC Clock = 5 MHz			3	LSB
Integral Non-linearity	ADC Clock = 5 MHz		1.5	2	LSB
Differential Non-linearity	ADC Clock = 5 MHz		1	2	LSB
	ADC Clock = 2.5 MHz		0.6	1	LSB
Offset Error	ADC Clock = 5 MHz	-2		2	LSB
Gain Error	ADC Clock = 5 MHz	-2		2	LSB

8.9.3 BOD

The values in Table 8-27 describe the values of the BODLEVEL in the flash General Purpose Fuse register.

Table 8-27. BODLEVEL Values

BODLEVEL Value	Min	Typ	Max	Units
000000b (00)		1.44		V
010111b (23)		1.52		V
011111b (31)		1.61		V
100111b (39)		1.71		V

Table 8-28. BOD Characteristics

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{HYST}	BOD hysteresis	T=25C°		10		mV
t_{DET}	Detection time	Time with VDDCORE < BODLEVEL necessary to generate a reset signal		1		µs
I_{BOD}	Current consumption			16		µA
$t_{STARTUP}$	Startup time			5		µs

8.9.4 Reset Sequence

Table 8-29. Electrical Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
V_{DDRR}	VDDCORE rise rate to ensure power-on-reset		2.5			V/ms
V_{DDFR}	VDDCORE fall rate to ensure power-on-reset		0.01		400	V/ms
V_{POR+}	Rising threshold voltage: voltage up to which device is kept under reset by POR on rising VDDCORE	Rising VDDCORE: $V_{RESTART}$ -> V_{POR+}	1.4	1.55	1.65	V
V_{POR-}	Falling threshold voltage: voltage when POR resets device on falling VDDCORE	Falling VDDCORE: 1.8V -> V_{POR+}	1.2	1.3	1.4	V
$V_{RESTART}$	On falling VDDCORE, voltage must go down to this value before supply can rise again to ensure reset signal is released at V_{POR+}	Falling VDDCORE: 1.8V -> $V_{RESTART}$	-0.1		0.5	V
T_{POR}	Minimum time with VDDCORE < V_{POR-}	Falling VDDCORE: 1.8V -> 1.1V		15		µs
T_{RST}	Time for reset signal to be propagated to system			200	400	µs
T_{SSU1}	Time for Cold System Startup: Time for CPU to fetch its first instruction (RCosc not calibrated)		480		960	µs
T_{SSU2}	Time for Hot System Startup: Time for CPU to fetch its first instruction (RCosc calibrated)			420		µs

Figure 8-3. MCU Cold Start-Up RESET_N tied to VDDIN

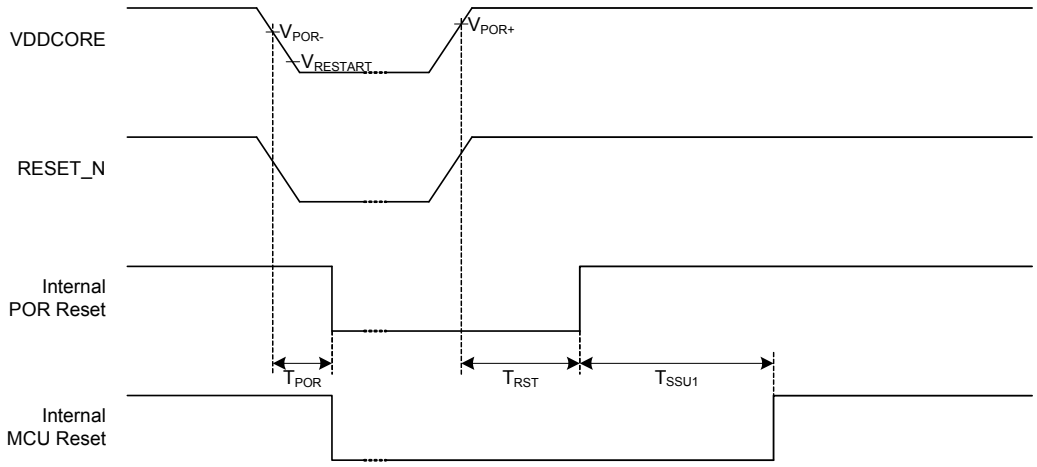


Figure 8-4. MCU Cold Start-Up RESET_N Externally Driven

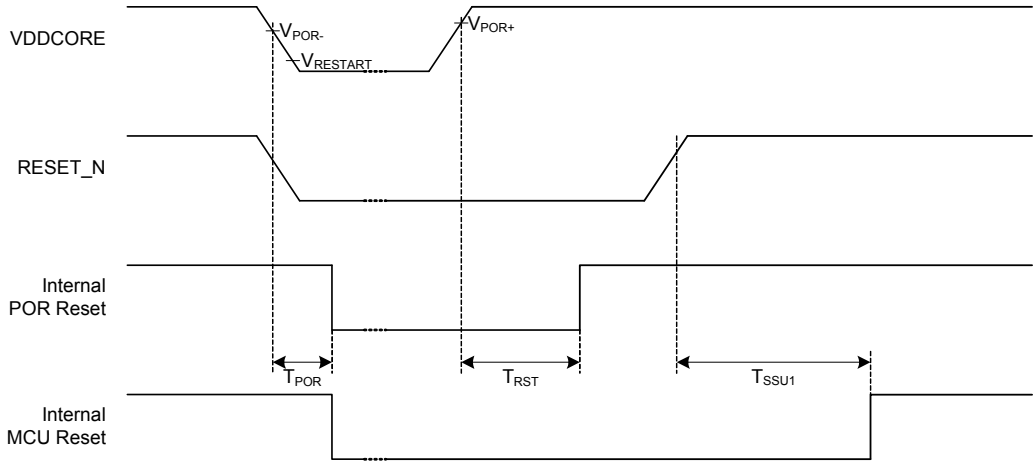
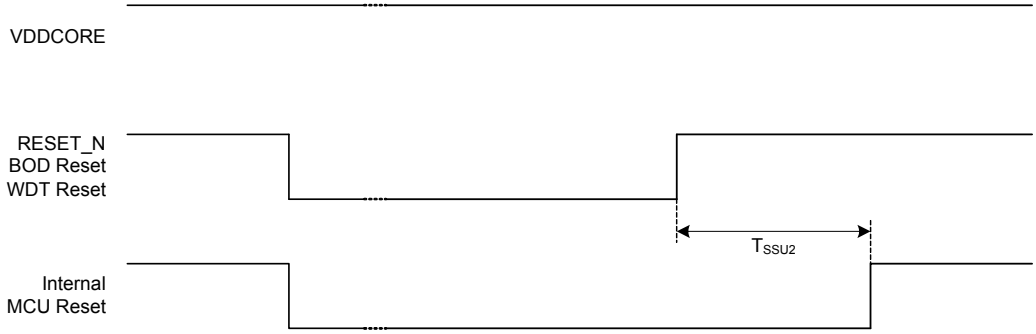


Figure 8-5. MCU Hot Start-Up

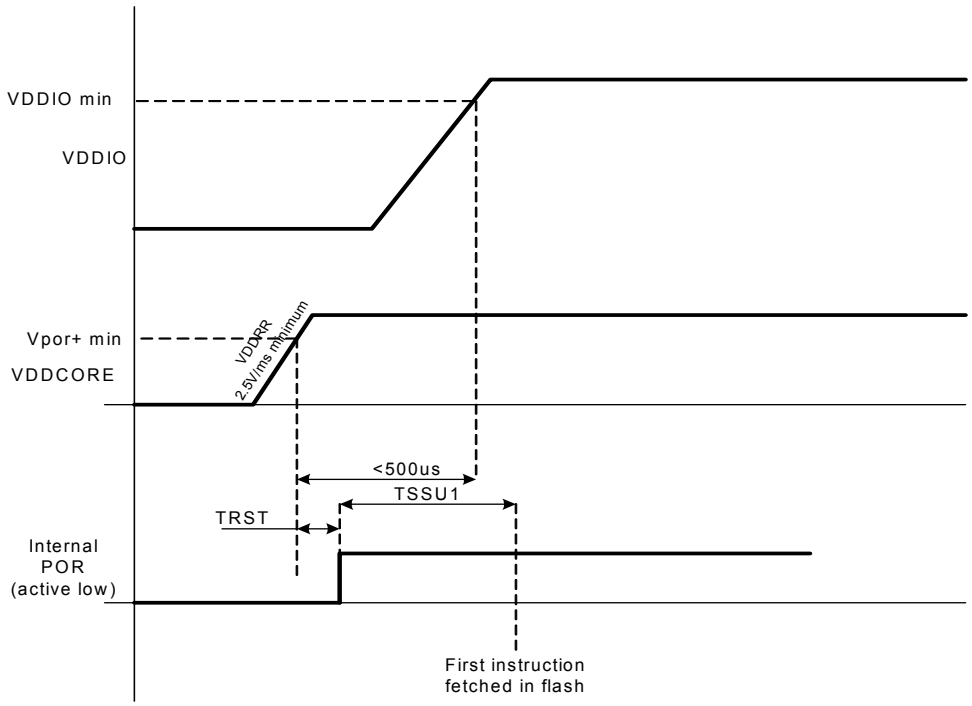


In dual supply configuration, the power up sequence must be carefully managed to ensure a safe startup of the device in all conditions.

The power up sequence must ensure that the internal logic is safely powered when the internal reset (Power On Reset) is released and that the internal Flash logic is safely powered when the CPU fetch the first instructions.

Therefore VDDCORE rise rate (VDDRR) must be equal or superior to 2.5V/ms and VDDIO must reach VDDIO mini value before 500 us (< TRST + TSSU1) after VDDCORE has reached V_{POR+} min value.

Figure 8-6. Dual Supply Configuration



8.9.5 RESET_N Characteristics

Table 8-30. RESET_N Waveform Parameters

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
t_{RESET}	RESET_N minimum pulse width		10			ns

8.10 USB Transceiver Characteristics

8.10.1 Electrical Characteristics

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.

9. Mechanical Characteristics

9.1 Thermal Considerations

9.1.1 Thermal Data

Table 9-1 summarizes the thermal resistance data depending on the package.

Table 9-1. Thermal Resistance Data

Symbol	Parameter	Condition	Package	Typ	Unit
θ_{JA}	Junction-to-ambient thermal resistance	Still Air	TQFP48	65.1	°C/W
θ_{JC}	Junction-to-case thermal resistance		TQFP48	23.4	
θ_{JA}	Junction-to-ambient thermal resistance	Still Air	QFN48	29.2	°C/W
θ_{JC}	Junction-to-case thermal resistance		QFN48	2.7	
θ_{JA}	Junction-to-ambient thermal resistance	Still Air	TQFP64	63.1	°C/W
θ_{JC}	Junction-to-case thermal resistance		TQFP64	23.0	
θ_{JA}	Junction-to-ambient thermal resistance	Still Air	QFN64	26.9	°C/W
θ_{JC}	Junction-to-case thermal resistance		QFN64	2.7	

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

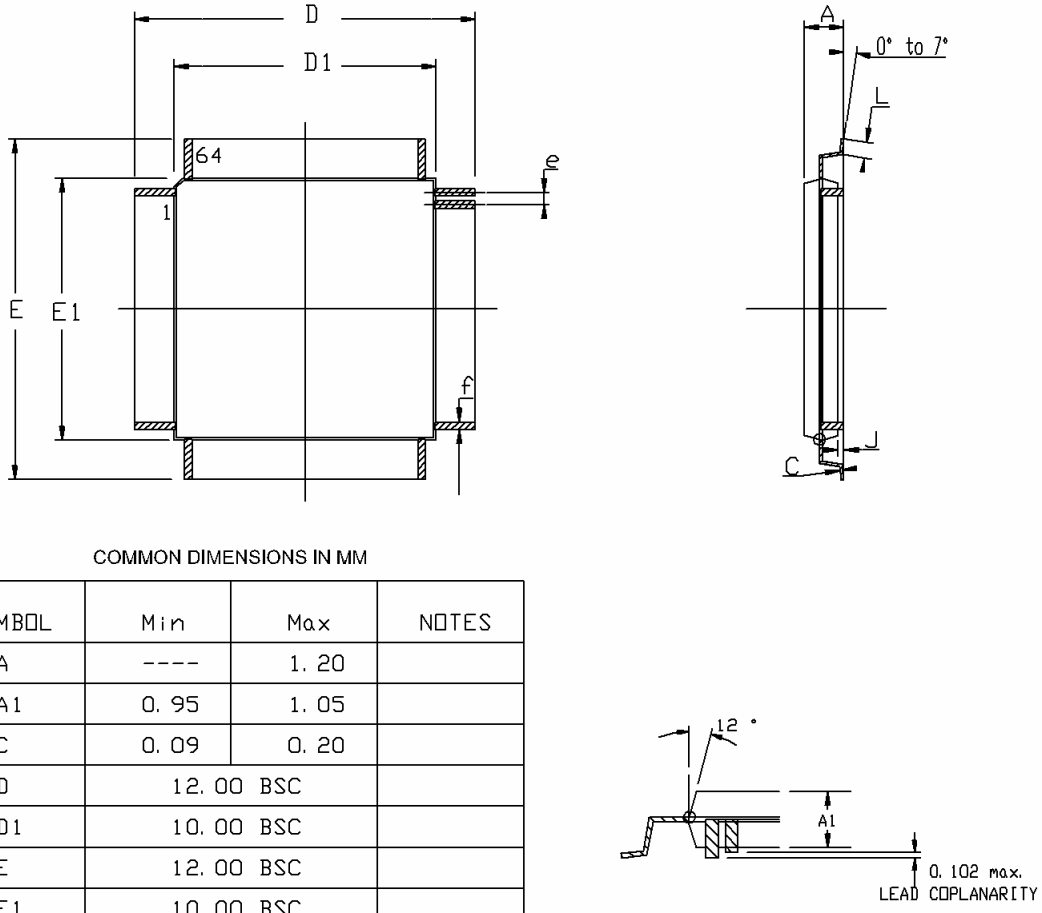
where:

- θ_{JA} = package thermal resistance, Junction-to-ambient (°C/W), provided in Table 9-1.
- θ_{JC} = package thermal resistance, Junction-to-case thermal resistance (°C/W), provided in Table 9-1.
- $\theta_{HEAT\ SINK}$ = cooling device thermal resistance (°C/W), provided in the device datasheet.
- P_D = device power consumption (W) estimated from data provided in the Section 8.5 on page 38.
- T_A = ambient temperature (°C).

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

9.2 Package Drawings

Figure 9-1. 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 9-2. Device and Package Maximum Weight

Weight	300 mg

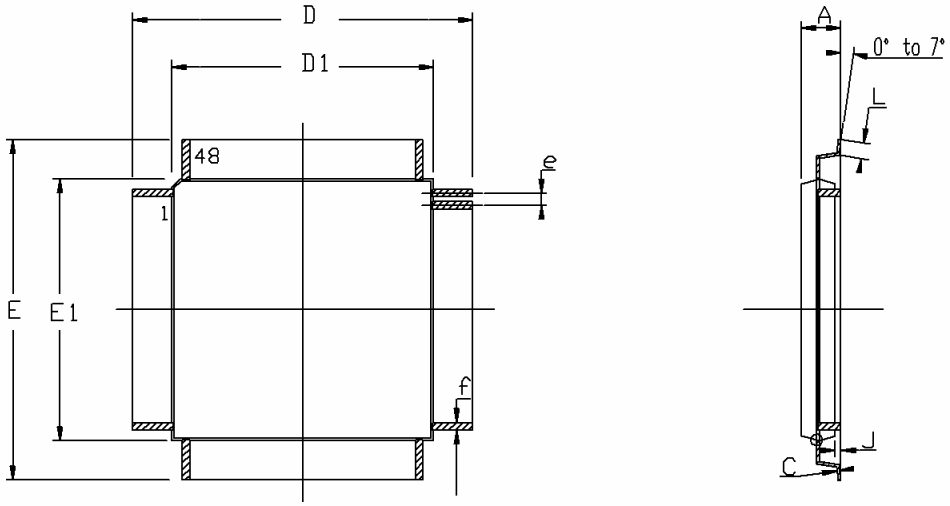
Table 9-3. Package Characteristics

Moisture Sensitivity Level	Jedec J-STD-20D-MSL3

Table 9-4. Package Reference

JEDEC Drawing Reference	MS-026
JESD97 Classification	e3

Figure 9-2. TQFP-48 package drawing



COMMON DIMENSIONS IN MM

SYMBOL	Min	Max	NOTES
A	----	1.20	
A1	0.95	1.05	
C	0.09	0.20	
D	9.00 BSC		
D1	7.00 BSC		
E	9.00 BSC		
E1	7.00 BSC		
J	0.05	0.15	
L	0.45	0.75	
e	0.50 BSC		
f	0.17	0.27	

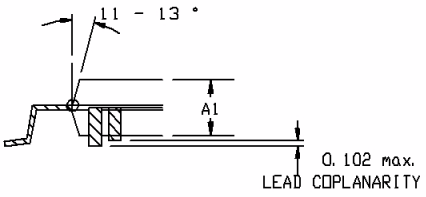


Table 9-5. Device and Package Maximum Weight

Weight	100 mg

Table 9-6. Package Characteristics

Moisture Sensitivity Level	Jedec J-STD-20D-MSL3

Table 9-7. Package Reference

JEDEC Drawing Reference	MS-026
JESD97 Classification	e3

Figure 9-3. QFN-48 Package Drawing

Table 9-8. Device and Package Maximum Weight

Weight	100 mg

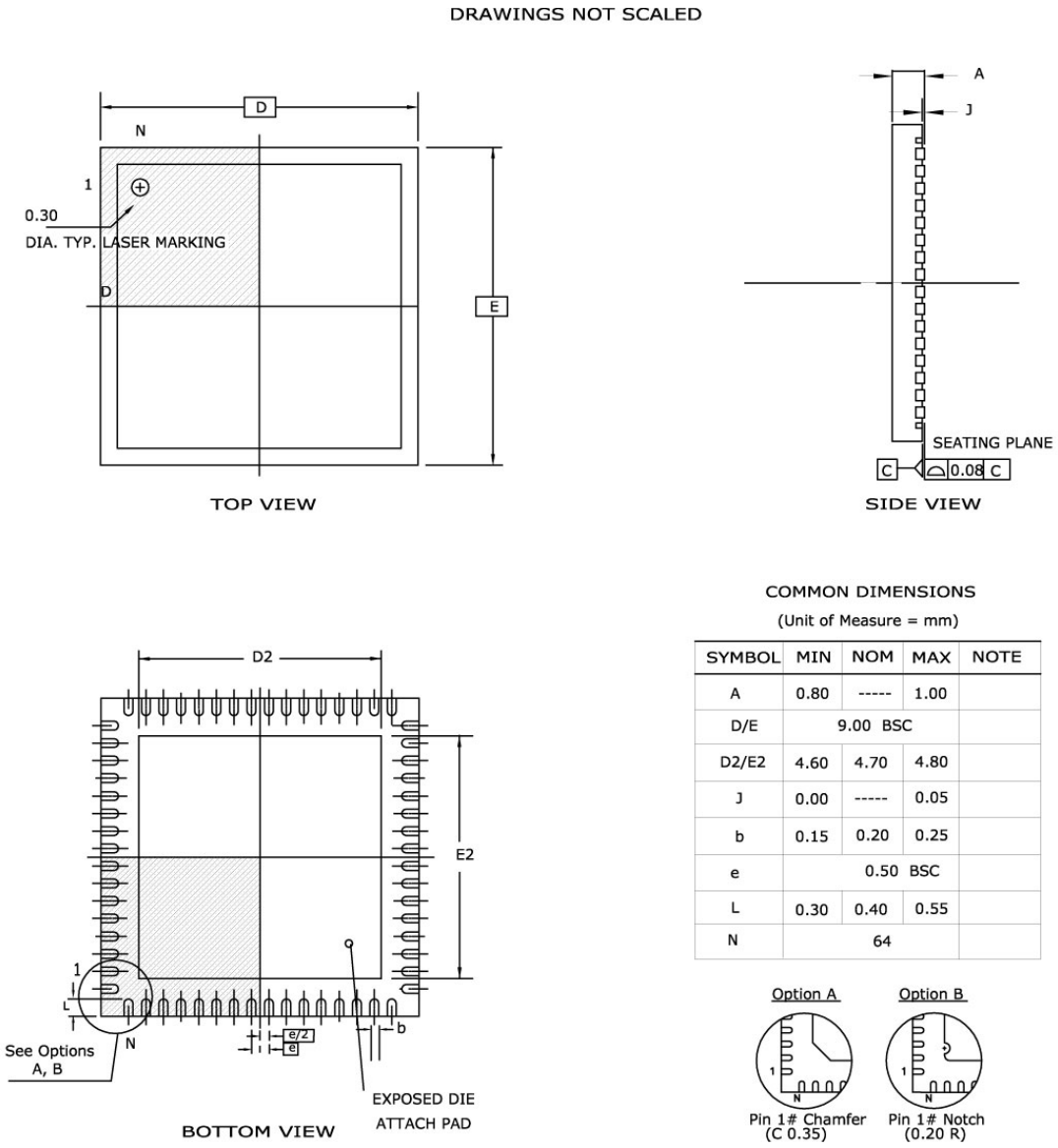
Table 9-9. Package Characteristics

Moisture Sensitivity Level	Jedec J-STD-20D-MSL3

Table 9-10. Package Reference

JEDEC Drawing Reference	M0-220
JESD97 Classification	e3

Figure 9-4. QFN-64 package drawing



SYMBOL	MIN	NOM	MAX	NOTE
A	0.80	-----	1.00	
D/E		9.00 BSC		
D2/E2	4.60	4.70	4.80	
J	0.00	-----	0.05	
b	0.15	0.20	0.25	
e		0.50 BSC		
L	0.30	0.40	0.55	
N		64		

Notes : 1. This drawing is for general information only. Refer to JEDEC Drawing MO-220, Variation VMMD-4, for proper dimensions, tolerances, datums, etc.
2. Dimension b applies to metallized terminal and is measured between 0.15mm and 0.30mm from the terminal tip.
If the terminal has the optical radius on the other end of the terminal, the dimension should not be measured in that radius area.

Table 9-11. Device and Package Maximum Weight

Weight	200 mg

Table 9-12. Package Characteristics

Moisture Sensitivity Level	Jedec J-STD-20D-MSL3

Table 9-13. Package Reference

JEDEC Drawing Reference	M0-220
JESD97 Classification	e3

9.3 Soldering Profile

Table 9-14 gives the recommended soldering profile from J-STD-20.

Table 9-14. Soldering Profile

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

A maximum of three reflow passes is allowed per component.

10. Ordering Information

Table 10-1. Ordering Information

Device	Ordering Code	Carrier Type	Package	Package Type	Temperature Operating Range
ATUC128D3	ATUC128D3-A2UT	Tray	TQFP 64	JESD97 Classification E3	Industrial (-40·C to 85·C)
	ATUC128D3-A2UR	Tape & Reel	TQFP 64		
	ATUC128D3-Z2UT	Tray	QFN 64		
	ATUC128D3-Z2UR	Tape & Reel	QFN 64		
ATUC128D4	ATUC128D4-AUT	Tray	TQFP 48		
	ATUC128D4-AUR	Tape & Reel	TQFP 48		
	ATUC128D4-Z1UT	Tray	QFN 48		
	ATUC128D4-Z1UR	Tape & Reel	QFN 48		
ATUC64D3	ATUC64D3-A2UT	Tray	TQFP 64	JESD97 Classification E3	Industrial (-40·C to 85·C)
	ATUC64D3-A2UR	Tape & Reel	TQFP 64		
	ATUC64D3-Z2UT	Tray	QFN 64		
	ATUC64D3-Z2UR	Tape & Reel	QFN 64		
ATUC64D4	ATUC64D4-AUT	Tray	TQFP 48		
	ATUC64D4-AUR	Tape & Reel	TQFP 48		
	ATUC64D4-Z1UT	Tray	QFN 48		
	ATUC64D4-Z1UR	Tape & Reel	QFN 48		

11. Errata

11.1 Rev. C

11.1.1 SPI

1. **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).

2. **PCS field in receive data register is inaccurate**
   The PCS field in the SPI_RDR register does not accurately indicate from which slave the received data is read.
   Fix/Workaround
   None.

3. **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.

4. **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.

5. **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.

6. **Timer Counter**

7. **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.

11.1.2 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.

11.1.3 PWMA

1. **The SR.READY bit cannot be cleared by writing to SCR.READY**
   The Ready bit in the Status Register will not be cleared when writing a one to the corresponding bit in the Status Clear register. The Ready bit will be cleared when the Busy bit is set.
   Fix/Workaround
   Disable the Ready interrupt in the interrupt handler when receiving the interrupt. When an operation that triggers the Busy/Ready bit is started, wait until the ready bit is low in the Status Register before enabling the interrupt.

11.2 Rev. B

11.2.1 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.

11.2.2 SPI

1. **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).

2. **PCS field in receive data register is inaccurate**
   The PCS field in the SPI_RDR register does not accurately indicate from which slave the received data is read.
   Fix/Workaround
   None.

3. **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.

4. **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.

5. **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.

6. **Timer Counter**

7. **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.

11.2.3 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.

11.2.4 PWMA

1. **The SR.READY bit cannot be cleared by writing to SCR.READY**
   The Ready bit in the Status Register will not be cleared when writing a one to the corresponding bit in the Status Clear register. The Ready bit will be cleared when the Busy bit is set.
   Fix/Workaround
   Disable the Ready interrupt in the interrupt handler when receiving the interrupt. When an operation that triggers the Busy/Ready bit is started, wait until the ready bit is low in the Status Register before enabling the interrupt.

11.3 Rev. A

11.3.1 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.

11.3.2 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 needs to wake the CPU up.
   Fix/Workaround
   Disable the TWIS or the AST before entering idle or frozen sleep mode.

3. **SPI**

4. **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).

5. **PCS field in receive data register is inaccurate**
   The PCS field in the SPI_RDR register does not accurately indicate from which slave the received data is read.
   Fix/Workaround
   None.

6. **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.

7. **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.

8. **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.

9. **I/O Pins**

10. **Current leakage through pads PA09, PA10 and PB16**
Pads PA09 (TWI), PA10 (TWI) and PB16 (USB VBUS) are not fully 5V tolerant. A leakage current can be observed when a 5V voltage is applied onto those pads inputs. Their behavior is normal at 3.3V
Fix/Workaround
None for pads PA09 and PA10. A voltage divider can be used for PB16 (VBUS) to bring the input voltage down into the 3.3V range.

11. **Current leakage through pads PB13, PB17 and PB18**
For applications in which UC3D is considered as a drop in replacement solution to UC3B, pads PB13, PB17 and PB18 can no longer be used as VDDCORE supply pins.Maintaining a 1.8V voltage on those inputs will however lead to a current over consumption through the pins.
Fix/Workaround
Do not connect PB13, PB17 and PB18 when using UC3D as a drop in replacement for a UC3B specific application.

12. **IO drive strength mismatch with UC3B specification for pads PA11, PA12, PA18 and PA19**
For applications in which UC3D is considered as a drop in replacement solution to UC3B, GPIOs PA11, PA12, PA18 and PA19 are not completely compatible in terms of drive strength. Those pads have a 8 mA current capability on UC3B, while this is limited to 4 mA in UC3D.
Fix/Workaround
None.

13. **WDT**

14. **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.

11.3.3 Timer Counter

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.

11.3.4 TWIS

1. **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.

2. **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.

3. **CAT**

4. **CAT module does not terminate QTouch burst on detect**
   The CAT module does not terminate a QTouch burst when the detection voltage is reached on the sense capacitor. This can cause the sense capacitor to be charged more than necessary. Depending on the dielectric absorption characteristics of the capacitor, this can lead to unstable measurements.
   Fix/Workaround
   Use the minimum possible value for the MAX field in the ATCFG1, TG0CFG1, and TG1CFG1 registers.

11.3.5 PWMA

1. **The SR.READY bit cannot be cleared by writing to SCR.READY**
   The Ready bit in the Status Register will not be cleared when writing a one to the corresponding bit in the Status Clear register. The Ready bit will be cleared when the Busy bit is set.
   Fix/Workaround
   Disable the Ready interrupt in the interrupt handler when receiving the interrupt. When an operation that triggers the Busy/Ready bit is started, wait until the ready bit is low in the Status Register before enabling the interrupt.

2.

12. 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.

12.1 Rev. A – 11/2009

1. Initial revision.

12.2 Rev. B – 04/2011

1. Minor.

12.3 Rev. C – 07/2011

1. Final revision.

12.4 Rev. D – 11/2011

1. Adding errata for silicon Revision C .
2. Fixed PLLOPT field description in SCIF chapter

Table of Contents

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