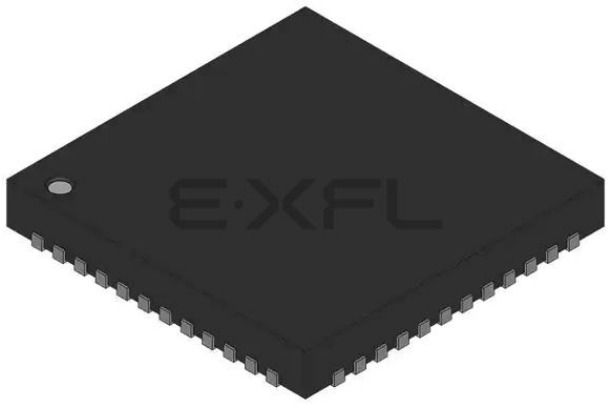


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	AVR
Core Size	32-Bit Single-Core
Speed	48MHz
Connectivity	I²C, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	35
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	1.65V ~ 3.6V
Data Converters	A/D 6x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-VFQFN Exposed Pad
Supplier Device Package	48-QFN (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/atmel/atuc128d4-z1ur

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

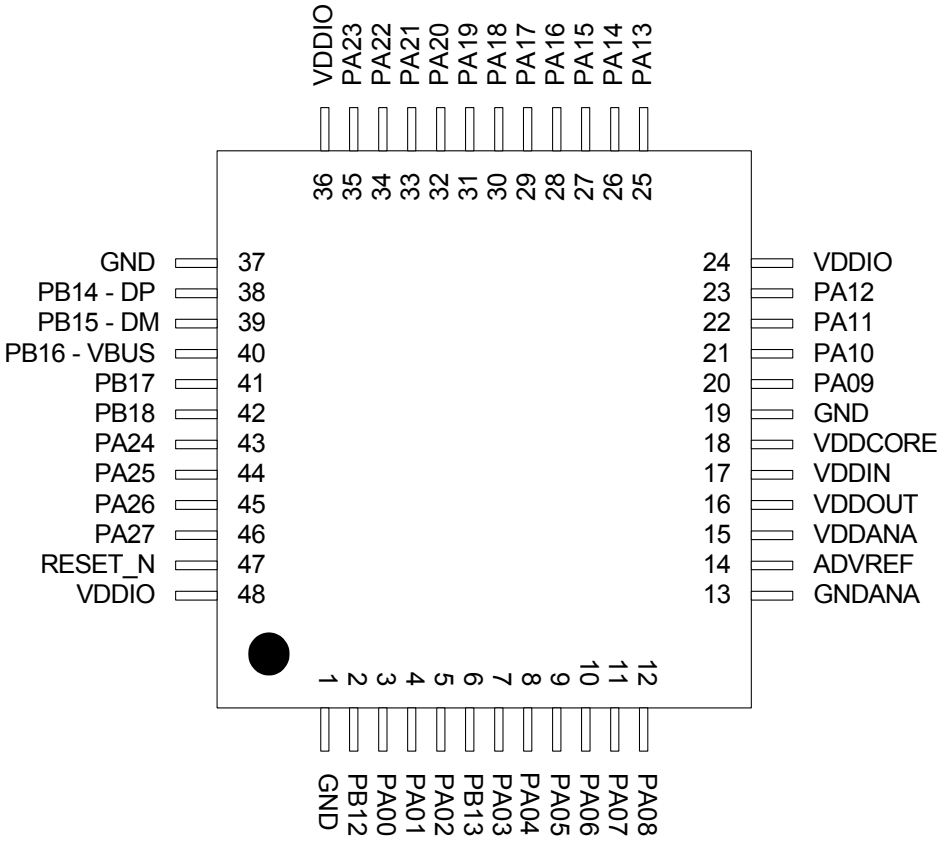
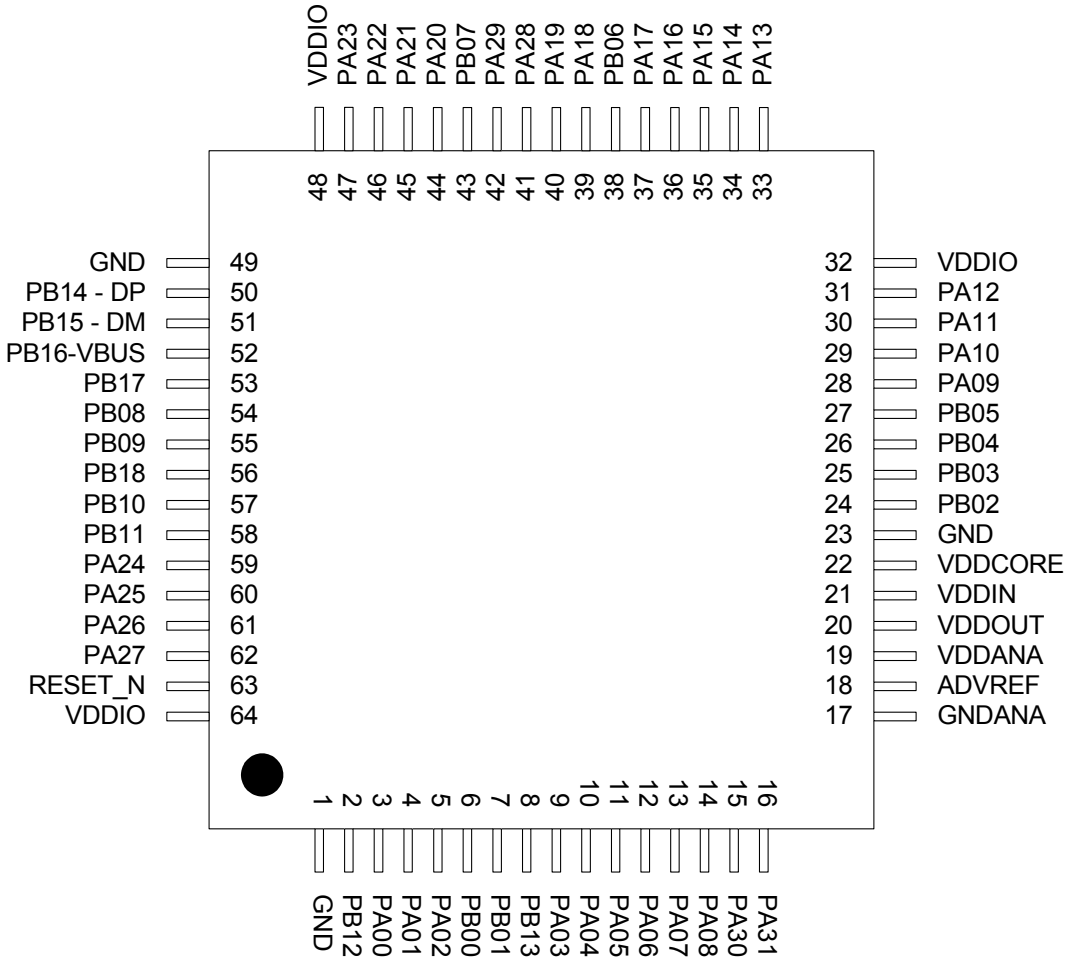


Figure 3-2. TQFP64/QFN64 Pinout



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 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.

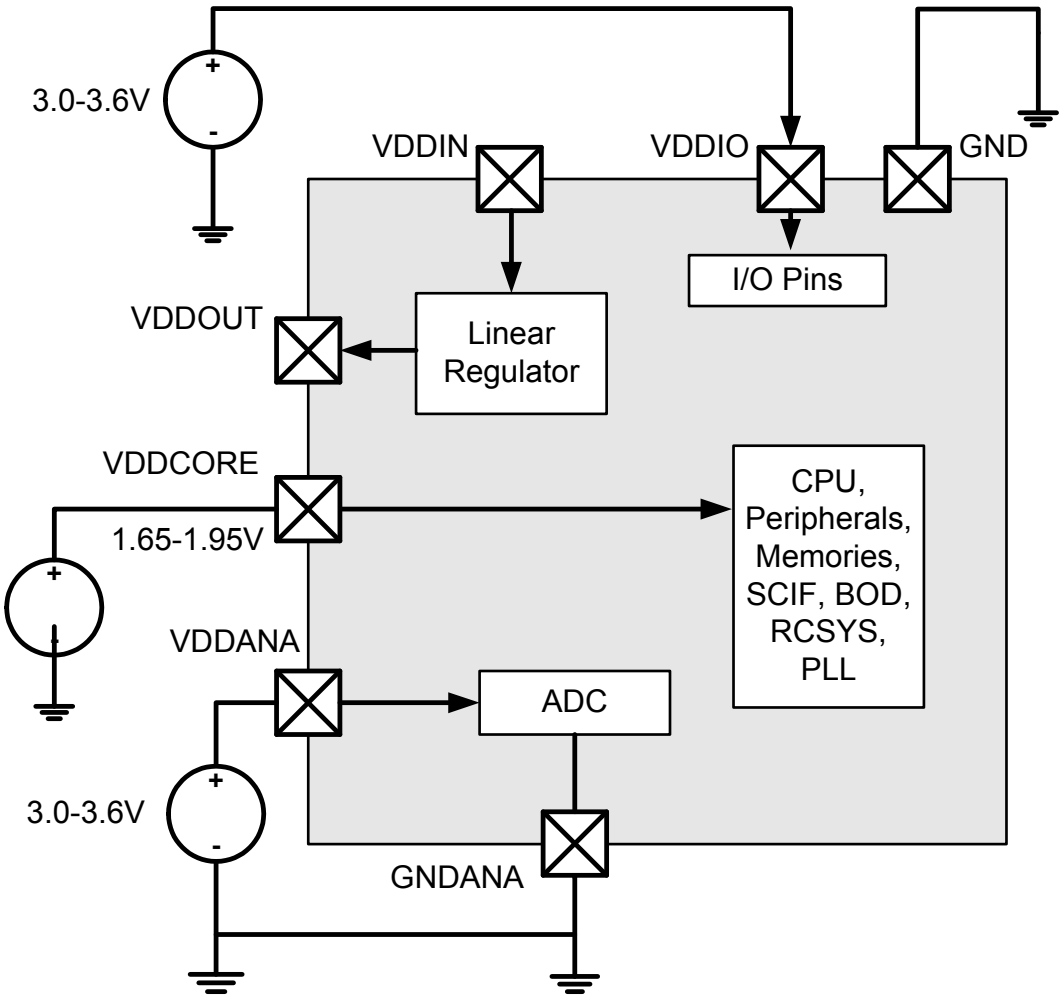
4.1.4 GPIO Pins

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

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.

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.

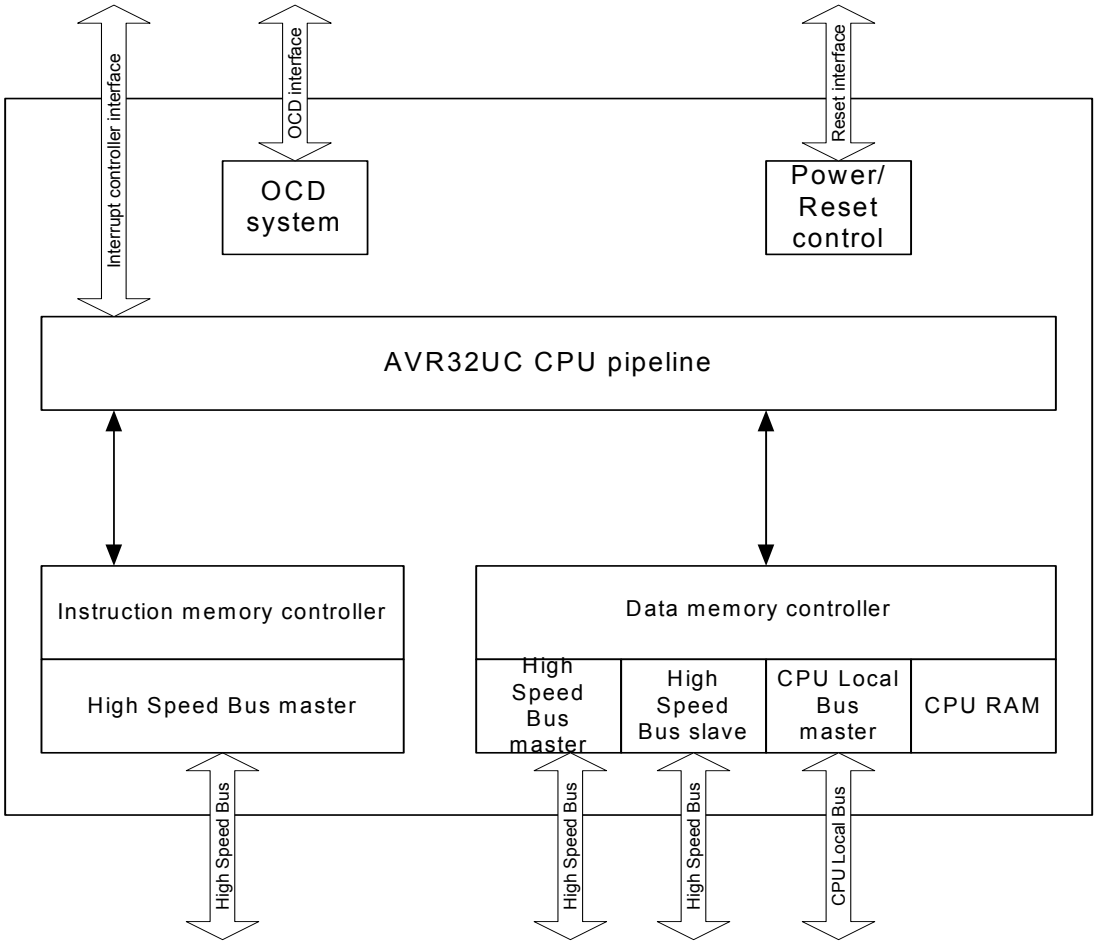
Through a quantitative approach, a large set of industry recognized benchmarks has been compiled and analyzed to achieve the best code density in its class. In addition to lowering the memory requirements, a compact code size also contributes to the core's low power characteristics. The processor supports 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.

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.

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

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

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

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.

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

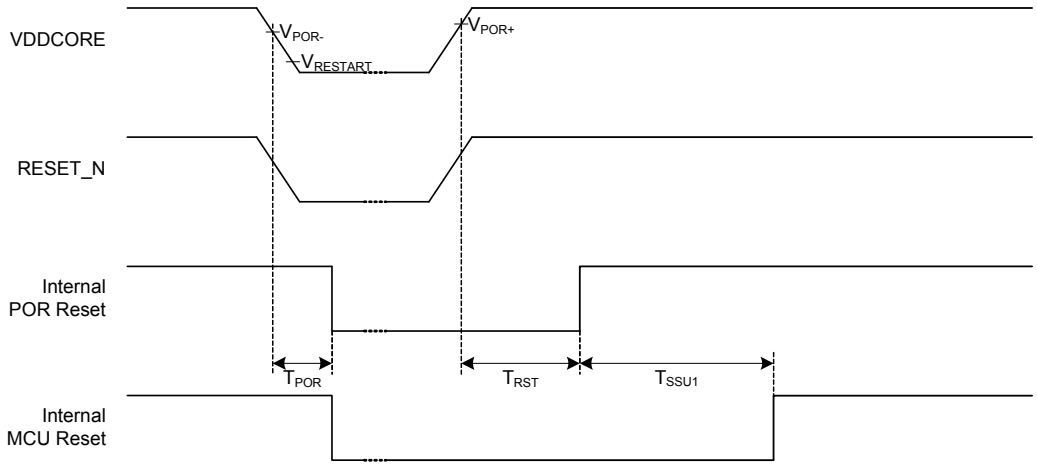
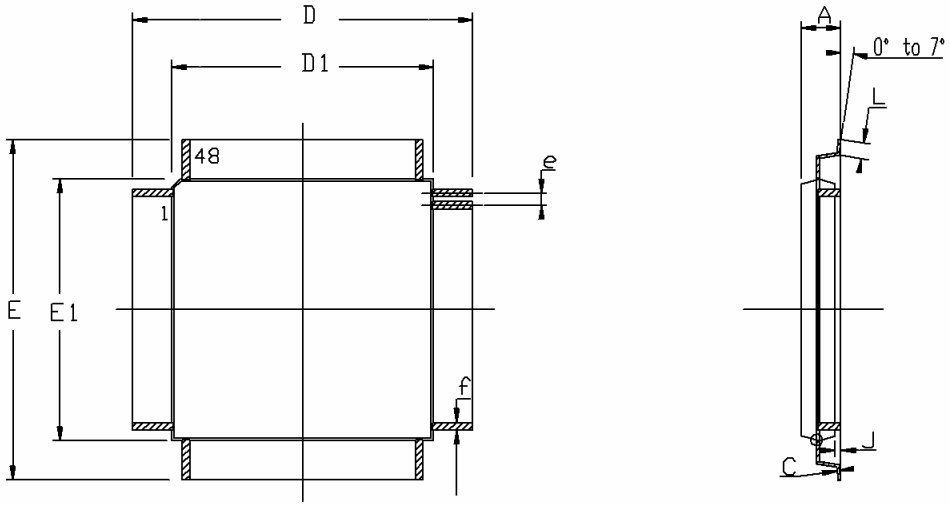


Figure 9-2. TQFP-48 package drawing



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	

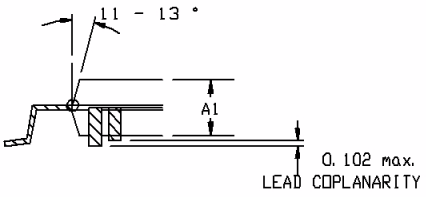


Table 9-5. Device and Package Maximum Weight

Weight	100 mg

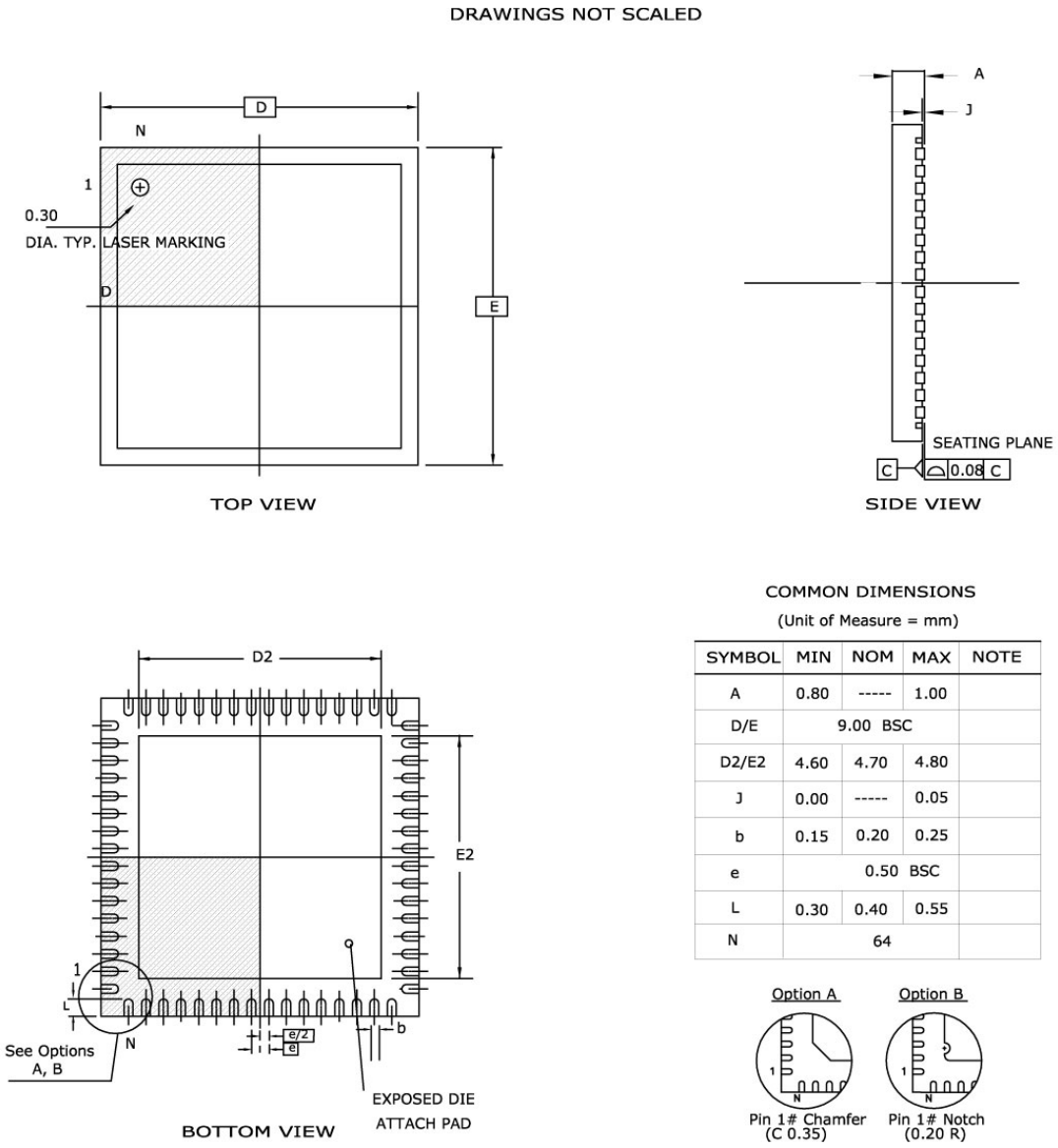
Table 9-6. Package Characteristics

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

Table 9-7. Package Reference

JEDEC Drawing Reference	MS-026
JESD97 Classification	e3

Figure 9-4. 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

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		

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.