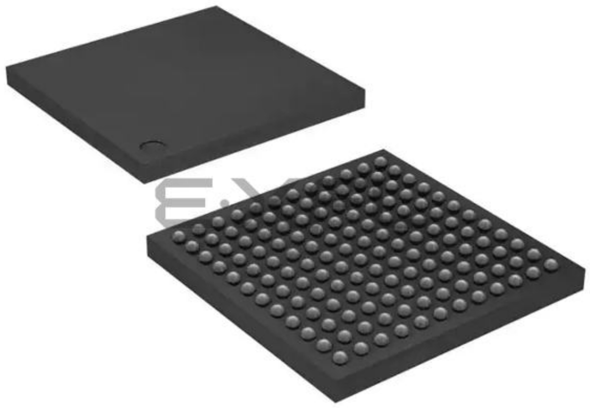
Microchip Technology - AT32UC3A364S-CTUT Datasheet



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

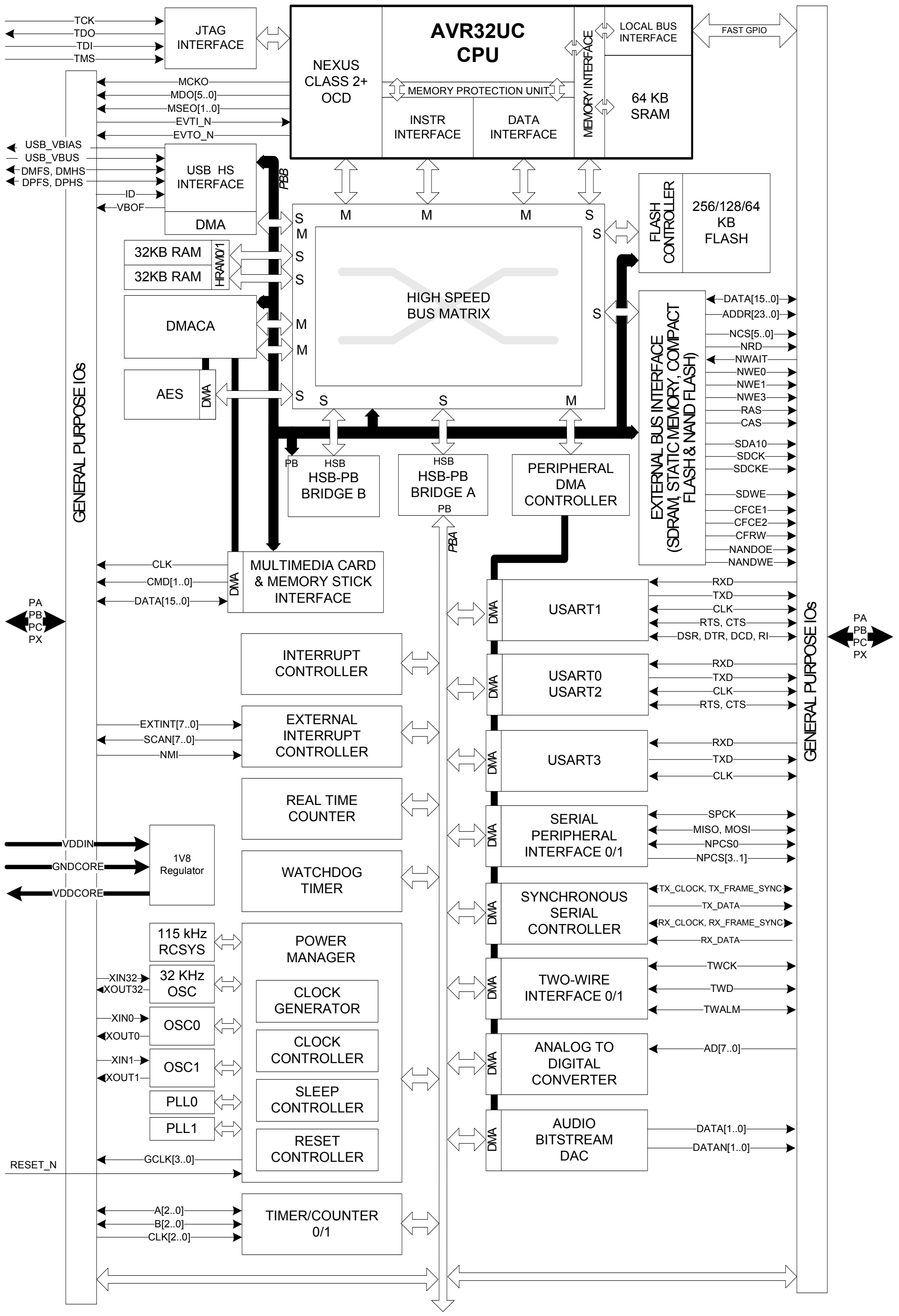
Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	AVR
Core Size	32-Bit Single-Core
Speed	66MHz
Connectivity	EBI/EMI, I²C, IrDA, Memory Card, SPI, SSC, UART/USART, USB OTG
Peripherals	Brown-out Detect/Reset, DMA, POR, WDT
Number of I/O	110
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128K x 8
Voltage - Supply (Vcc/Vdd)	1.75V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	144-TFBGA
Supplier Device Package	144-FFBGA (11x11)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at32uc3a364s-ctut

2. Overview

2.1 Block Diagram

Figure 2-1. Block Diagram

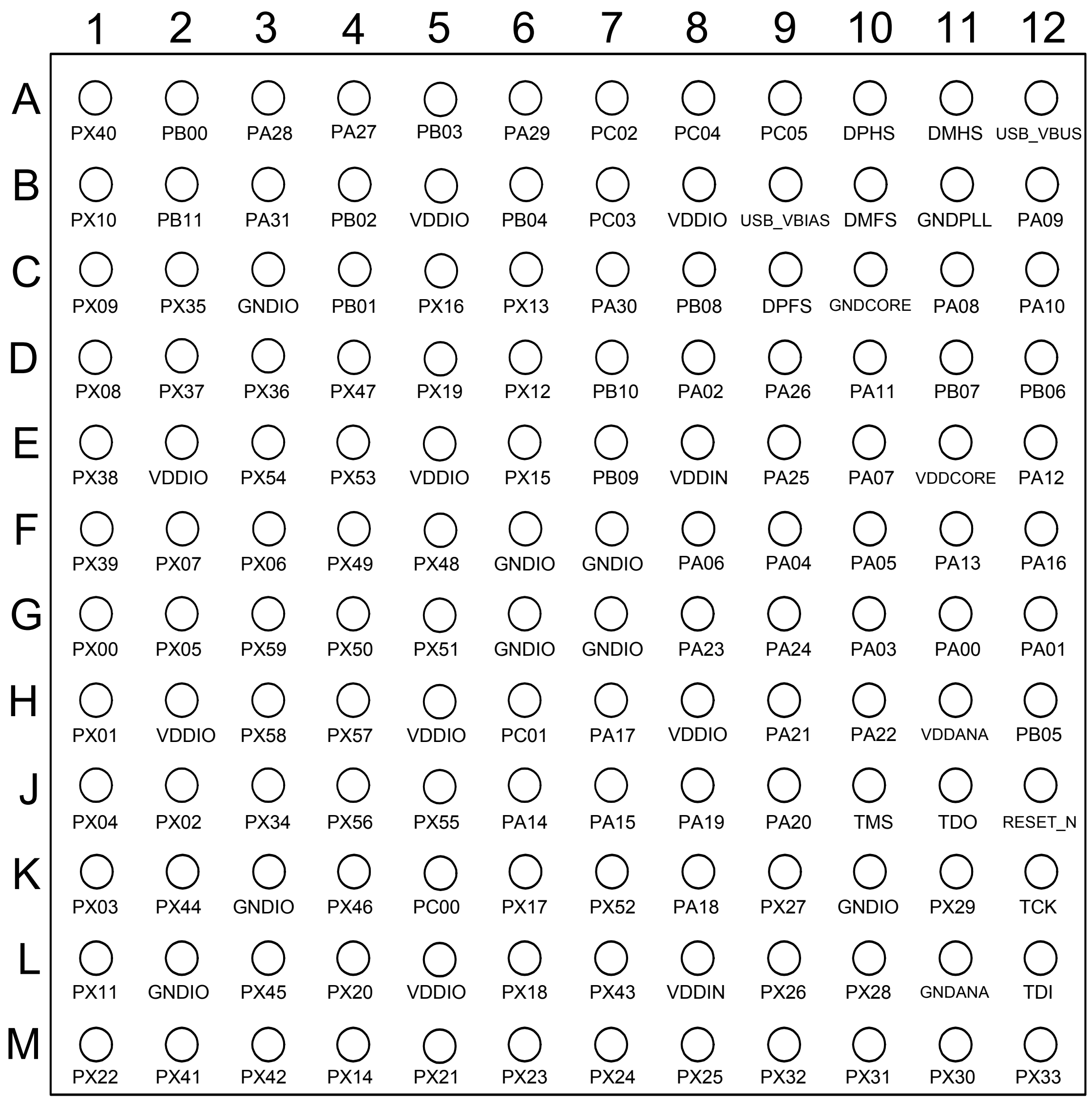


3. Package and Pinout

3.1 Package

The device pins are multiplexed with peripheral functions as described in the Peripheral Multiplexing on I/O Line section.

Figure 3-1. TFBGA144 Pinout (top view)



	1	2	3	4	5	6	7	8	9	10	11	12
A	PX40	PB00	PA28	PA27	PB03	PA29	PC02	PC04	PC05	DPHS	DMHS	USB_VBUS
B	PX10	PB11	PA31	PB02	VDDIO	PB04	PC03	VDDIO	USB_VBIAS	DMFS	GNDPLL	PA09
C	PX09	PX35	GNDIO	PB01	PX16	PX13	PA30	PB08	DPFS	GNDCORE	PA08	PA10
D	PX08	PX37	PX36	PX47	PX19	PX12	PB10	PA02	PA26	PA11	PB07	PB06
E	PX38	VDDIO	PX54	PX53	VDDIO	PX15	PB09	VDDIN	PA25	PA07	VDDCORE	PA12
F	PX39	PX07	PX06	PX49	PX48	GNDIO	GNDIO	PA06	PA04	PA05	PA13	PA16
G	PX00	PX05	PX59	PX50	PX51	GNDIO	GNDIO	PA23	PA24	PA03	PA00	PA01
H	PX01	VDDIO	PX58	PX57	VDDIO	PC01	PA17	VDDIO	PA21	PA22	VDDANA	PB05
J	PX04	PX02	PX34	PX56	PX55	PA14	PA15	PA19	PA20	TMS	TDO	RESET_N
K	PX03	PX44	GNDIO	PX46	PC00	PX17	PX52	PA18	PX27	GNDIO	PX29	TCK
L	PX11	GNDIO	PX45	PX20	VDDIO	PX18	PX43	VDDIN	PX26	PX28	GNDANA	TDI
M	PX22	PX41	PX42	PX14	PX21	PX23	PX24	PX25	PX32	PX31	PX30	PX33

Table 3-6. Signal Description List

Signal Name	Function	Type	Active Level	Comments
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 - TWI0, TWI1				
TWCK	Serial Clock	I/O		
TWD	Serial Data	I/O		
TWALM	SMBALERT signal	I/O		
Universal Synchronous Asynchronous Receiver Transmitter - USART0, USART1, USART2, USART3				
CLK	Clock	I/O		
CTS	Clear To Send	Input		
DCD	Data Carrier Detect			Only USART1
DSR	Data Set Ready			Only USART1
DTR	Data Terminal Ready			Only USART1
RI	Ring Indicator			Only USART1
RTS	Request To Send	Output		
RXD	Receive Data	Input		
TXD	Transmit Data	Output		
Analog to Digital Converter - ADC				
AD0 - AD7	Analog input pins	Analog input		
Audio Bitstream DAC (ABDAC)				
DATA0-DATA1	D/A Data out	Output		
DATAN0-DATAN1	D/A Data inverted out	Output		
Universal Serial Bus Device - USB				
DMFS	USB Full Speed Data -	Analog		
DPFS	USB Full Speed Data +	Analog		

3.5 Power Considerations

3.5.1 Power Supplies

The AT32UC3A3 has several types of power supply pins:

- **VDDIO: Powers I/O lines. Voltage is 3.3V nominal**
- **VDDANA: Powers the ADC. Voltage is 3.3V nominal**
- **VDDIN: Input voltage for the voltage regulator. Voltage is 3.3V nominal**
- **VDDCORE: Output voltage from regulator for filtering purpose and provides the supply to the core, memories, and peripherals. Voltage is 1.8V nominal**

The ground pin GNDCORE is common to VDDCORE and VDDIN. The ground pin for VDDANA is GNDANA. The ground pins for VDDIO are GNDIO.

Refer to Electrical Characteristics chapter for power consumption on the various supply pins.

3.5.2 Voltage Regulator

The AT32UC3A3 embeds a voltage regulator that converts from 3.3V to 1.8V with a load of up to 100 mA. The regulator takes its input voltage from VDDIN, and supplies the output voltage on VDDCORE and powers the core, memories and peripherals.

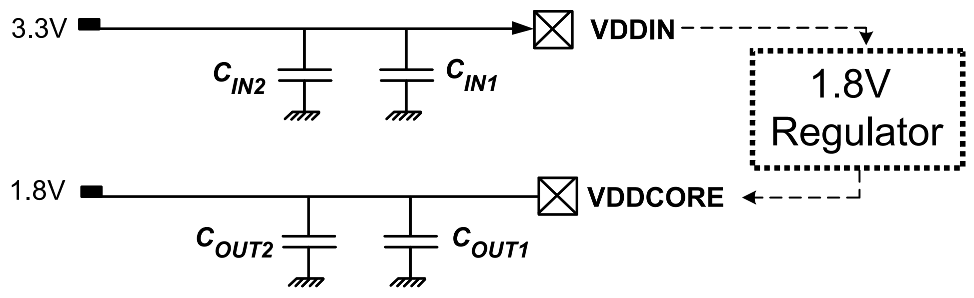
Adequate output supply decoupling is mandatory for VDDCORE to reduce ripple and avoid oscillations.

The best way to achieve this is to use two capacitors in parallel between VDDCORE and GNDCORE:

- One external 470pF (or 1nF) NPO capacitor (COUT1) should be connected as close to the chip as possible.
- One external 2.2µF (or 3.3µF) X7R capacitor (COUT2).

Adequate input supply decoupling is mandatory for VDDIN in order to improve startup stability and reduce source voltage drop.

The input decoupling capacitor should be placed close to the chip, e.g., two capacitors can be used in parallel (1nF NPO and 4.7µF X7R).



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

4. Processor and Architecture

Rev: 1.4.2.0

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

4.1 Features

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

4.2 AVR32 Architecture

AVR32 is a 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.

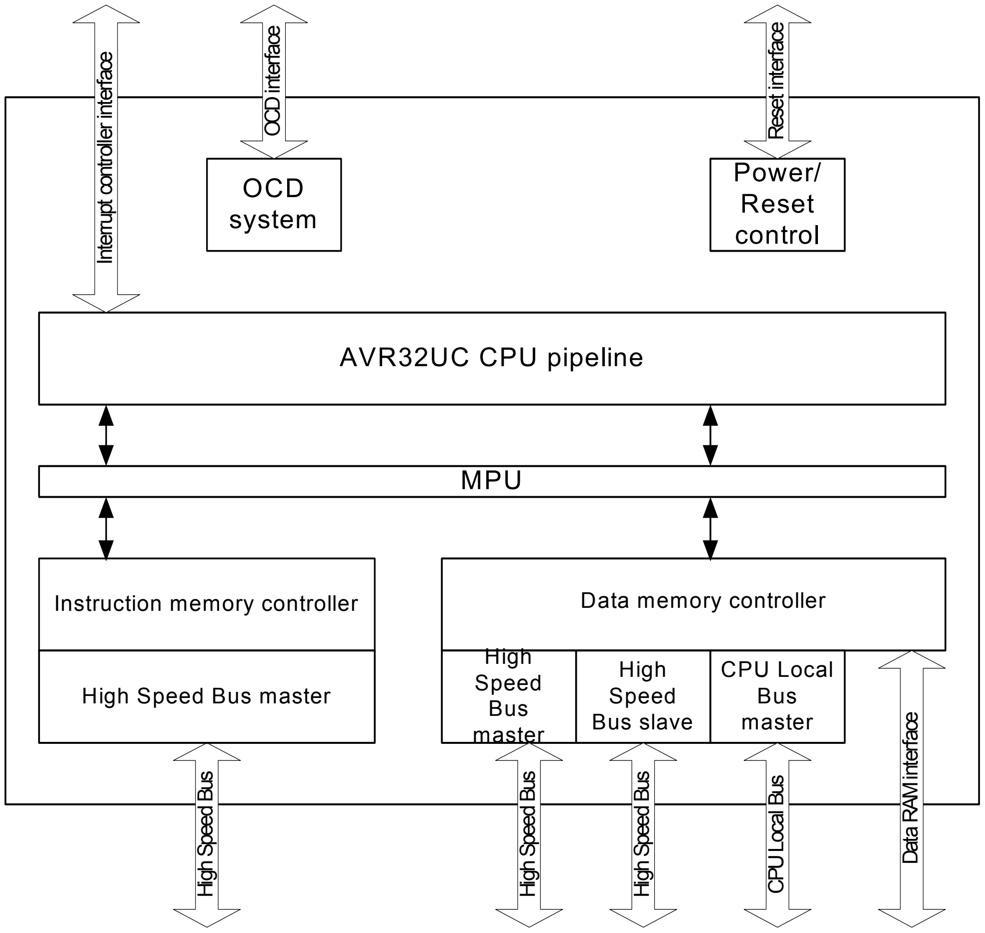


Figure 4-1. Overview of the AVR32UC CPU

4.3.1 Pipeline Overview

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

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

Figure 4-2 on page 24 shows an overview of the AVR32UC pipeline stages.

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

Debug state can be entered as described in the *AVR32UC Technical Reference Manual*.

Debug state is exited by the *retd* instruction.

4.4.4 System Registers

The system registers are placed outside of the virtual memory space, and are only accessible using the privileged *mfsr* and *mtsr* instructions. The table below lists the system registers specified in the AVR32 architecture, some of which are unused in AVR32UC. The programmer is responsible for maintaining correct sequencing of any instructions following a *mtsr* instruction. For detail on the system registers, refer to the *AVR32UC Technical Reference Manual*.

Table 4-3. System Registers

Reg #	Address	Name	Function
0	0	SR	Status Register
1	4	EVBA	Exception Vector Base Address
2	8	ACBA	Application Call Base Address
3	12	CPUCR	CPU Control Register
4	16	ECR	Exception Cause Register
5	20	RSR_SUP	Unused in AVR32UC
6	24	RSR_INT0	Unused in AVR32UC
7	28	RSR_INT1	Unused in AVR32UC
8	32	RSR_INT2	Unused in AVR32UC
9	36	RSR_INT3	Unused in AVR32UC
10	40	RSR_EX	Unused in AVR32UC
11	44	RSR_NMI	Unused in AVR32UC
12	48	RSR_DBG	Return Status Register for Debug mode
13	52	RAR_SUP	Unused in AVR32UC
14	56	RAR_INT0	Unused in AVR32UC
15	60	RAR_INT1	Unused in AVR32UC
16	64	RAR_INT2	Unused in AVR32UC
17	68	RAR_INT3	Unused in AVR32UC
18	72	RAR_EX	Unused in AVR32UC
19	76	RAR_NMI	Unused in AVR32UC
20	80	RAR_DBG	Return Address Register for Debug mode
21	84	JECR	Unused in AVR32UC
22	88	JOSP	Unused in AVR32UC
23	92	JAVA_LV0	Unused in AVR32UC
24	96	JAVA_LV1	Unused in AVR32UC
25	100	JAVA_LV2	Unused in AVR32UC

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

4.5.2 Exceptions and Interrupt Requests

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

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

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

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

4.5.3 Supervisor Calls

The AVR32 instruction set provides a supervisor mode call instruction. The *scall* instruction is designed so that privileged routines can be called from any context. This facilitates sharing of code between different execution modes. The *scall* mechanism is designed so that a minimal execution cycle overhead is experienced when performing supervisor routine calls from time-critical event handlers.

The *scall* instruction behaves differently depending on which mode it is called from. The behaviour is detailed in the instruction set reference. In order to allow the *scall* routine to return to the correct context, a return from supervisor call instruction, *rets*, is implemented. In the AVR32UC CPU, *scall* and *rets* uses the system stack to store the return address and the status register.

4.5.4 Debug Requests

The AVR32 architecture defines a dedicated Debug mode. When a debug request is received by the core, Debug mode is entered. Entry into Debug mode can be masked by the DM bit in the

5. Memories

5.1 Embedded Memories

- **Internal High-Speed Flash**
  - **256KBytes (AT32UC3A3256/S)**
  - **128Kbytes (AT32UC3A3128/S)**
  - **64Kbytes (AT32UC3A364/S)**
    - **0 wait state access at up to 42MHz in worst case conditions**
    - **1 wait state access at up to 84MHz in worst case conditions**
    - **Pipelined Flash architecture, allowing burst reads from sequential Flash locations, hiding penalty of 1 wait state access**
    - **Pipelined Flash architecture typically reduces the cycle penalty of 1 wait state operation to only 15% compared to 0 wait state operation**
    - **100 000 write cycles, 15-year data retention capability**
    - **Sector lock capabilities, Bootloader protection, Security Bit**
    - **32 Fuses, Erased During Chip Erase**
    - **User page for data to be preserved during Chip Erase**
- **Internal High-Speed SRAM**
  - **64KBytes, Single-cycle access at full speed on CPU Local Bus and accessible through the High Speed Bud (HSB) matrix**
  - **2x32KBytes, accessible independently through the High Speed Bud (HSB) matrix**

5.2 Physical Memory Map

The System Bus is implemented as a bus matrix. All system bus addresses are fixed, and they are never remapped in any way, not even in boot.

Note that AVR32 UC CPU uses unsegmented translation, as described in the AVR32UC Technical Architecture Manual.

The 32-bit physical address space is mapped as follows:

Table 5-1. AT32UC3A3A4 Physical Memory Map

Device	Start Address	Size	Size	Size
		AT32UC3A3256S AT32UC3A3256 AT32UC3A4256S AT32UC3A4256	AT32UC3A3128S AT32UC3A3128 AT32UC3A4128S AT32UC3A4128	AT32UC3A364S AT32UC3A364 AT32UC3A464S AT32UC3A464
Embedded CPU SRAM	0x00000000	64KByte	64KByte	64KByte
Embedded Flash	0x80000000	256KByte	128KByte	64KByte
EBI SRAM CS0	0xC0000000	16MByte	16MByte	16MByte
EBI SRAM CS2	0xC8000000	16MByte	16MByte	16MByte
EBI SRAM CS3	0xCC000000	16MByte	16MByte	16MByte
EBI SRAM CS4	0xD8000000	16MByte	16MByte	16MByte
EBI SRAM CS5	0xDC000000	16MByte	16MByte	16MByte
EBI SRAM CS1 /SDRAM CS0	0xD0000000	128MByte	128MByte	128MByte
USB Data	0xE0000000	64KByte	64KByte	64KByte

Table 5-3. Local Bus Mapped GPIO Registers

Port	Register	Mode	Local Bus Address	Access
2	Output Driver Enable Register (ODER)	WRITE	0x40000240	Write-only
		SET	0x40000244	Write-only
		CLEAR	0x40000248	Write-only
		TOGGLE	0x4000024C	Write-only
	Output Value Register (OVR)	WRITE	0x40000250	Write-only
		SET	0x40000254	Write-only
		CLEAR	0x40000258	Write-only
		TOGGLE	0x4000025C	Write-only
	Pin Value Register (PVR)	-	0x40000260	Read-only
3	Output Driver Enable Register (ODER)	WRITE	0x40000340	Write-only
		SET	0x40000344	Write-only
		CLEAR	0x40000348	Write-only
		TOGGLE	0x4000034C	Write-only
	Output Value Register (OVR)	WRITE	0x40000350	Write-only
		SET	0x40000354	Write-only
		CLEAR	0x40000358	Write-only
		TOGGLE	0x4000035C	Write-only
	Pin Value Register (PVR)	-	0x40000360	Read-only

7.2 DC 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 toT_J = 100°C.

Table 7-1. DC Characteristics

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Unit
VVDDIO	DC Supply Peripheral I/Os			3.0		3.6	V
VVDDANA	DC Analog Supply			3.0		3.6	V
V_{IL}	Input Low-level Voltage	All I/O pins except TWCK, TWD, RESET_N, TCK, TDI		-0.3		+0.8	V
		TWCK, TWD		V_{VDDIO} x0.7		V_{VDDIO} +0.5	V
		RESET_N, TCK, TDI		+0.8V			V
V_{IH}	Input High-level Voltage	All I/O pins except TWCK, TWD		2.0		3.6	V
		TWCK, TWD					V
V_{OL}	Output Low-level Voltage	I_{OL} = -2mA for Pin drive x1 I_{OL} = -4mA for Pin drive x2 I_{OL} = -8mA for Pin drive x3				0.4	V
V_{OH}	Output High-level Voltage	I_{OH} = 2mA for Pin drive x1 I_{OH} = 4mA for Pin drive x2 I_{OH} = 8mA for Pin drive x3		V_{VDDIO} -0.4			V
I_{LEAK}	Input Leakage Current	Pullup resistors disabled			0.05	1	µA
C_{IN}	Input Capacitance				7		pF
R_{PULLUP}	Pull-up Resistance	All I/O pins except RESET_N, TCK, TDI, TMS		9	15	25	KΩ
		RESET_N, TCK, TDI, TMS		5		25	KΩ
I_O	Output Current Pin drive 1x Pin drive 2x Pin drive 3x					2.0 4.0 8.0	mA
I_{SC}	Static Current	On V_{VDDIN} = 3.3V, CPU in static mode	T_A = 25°C		30		µA
			T_A = 85°C		175		µA

7.2.1 I/O Pin Output Level Typical Characteristics

Figure 7-1. I/O Pin drive x2 Output Low Level Voltage (VOL) vs. Source Current

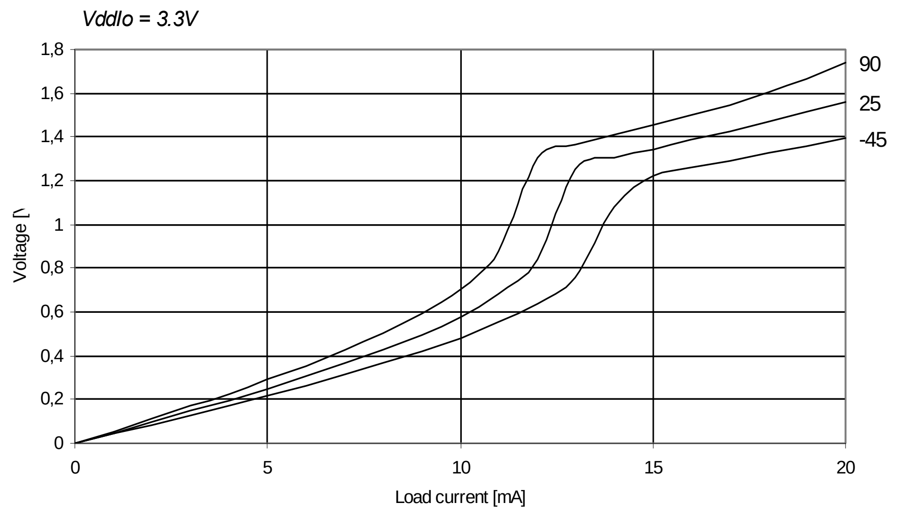
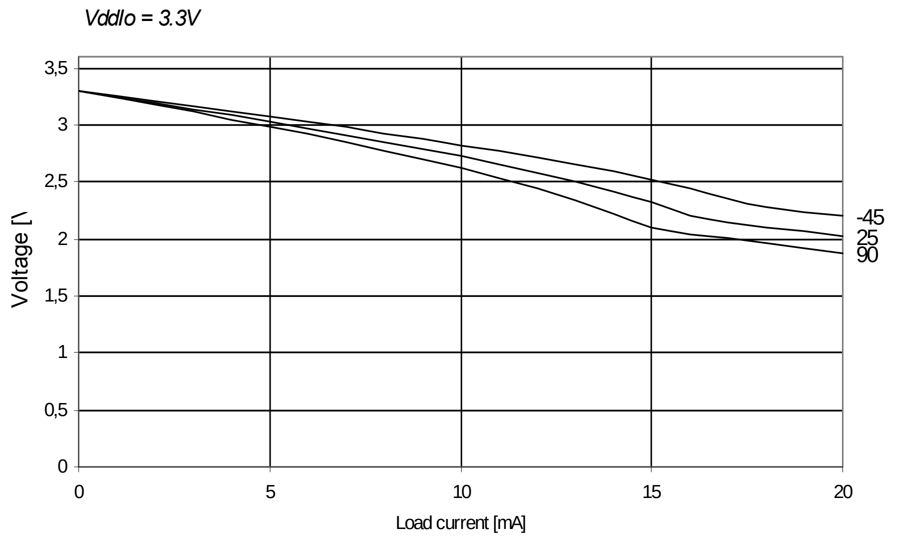


Figure 7-2. I/O Pin drive x2 Output High Level Voltage (VOH) vs. Source Current



7.3 I/O pin Characteristics

These parameters are given in the following conditions:

- V_{DDCORE} = 1.8V
- V_{DDIO} = 3.3V
- Ambient Temperature = 25°C

Table 7-9. BOD Timing

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
T_{BOD}	Minimum time with VDDCORE < VBOD to detect power failure	Falling VDDCORE from 1.8V to 1.1V		300	800	ns

7.5.3 Reset Sequence

Table 7-10. Electrical Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
V_{DDRR}	VDDIN/VDDIO rise rate to ensure power-on-reset		0.8			V/ms
V_{POR+}	Rising threshold voltage: voltage up to which device is kept under reset by POR on rising VDDIN	Rising VDDIN: $V_{RESTART}$ -> V_{POR+}		2.7		V
V_{POR-}	Falling threshold voltage: voltage when POR resets device on falling VDDIN	Falling VDDIN: 3.3V -> V_{POR-}		2.7		V
$V_{RESTART}$	On falling VDDIN, voltage must go down to this value before supply can rise again to ensure reset signal is released at V_{POR+}	Falling VDDIN: 3.3V -> $V_{RESTART}$			0.2	V
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

7.7 System Clock Characteristics

These parameters are given in the following conditions:

- V_{DDCORE} = 1.8V

7.7.1 CPU/HSB Clock Characteristics

Table 7-14. Core Clock Waveform Parameters

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
$1/(t_{CPCPU})$	CPU Clock Frequency	-40°C < Ambient Temperature < 70°C			84	MHz
$1/(t_{CPCPU})$	CPU Clock Frequency	-40°C < Ambient Temperature < 85°C			66	MHz

7.7.2 PBA Clock Characteristics

Table 7-15. PBA Clock Waveform Parameters

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
$1/(t_{CPPBA})$	PBA Clock Frequency	-40°C < Ambient Temperature < 70°C			84	MHz
$1/(t_{CPPBA})$	PBA Clock Frequency	-40°C < Ambient Temperature < 85°C			66	MHz

7.7.3 PBB Clock Characteristics

Table 7-16. PBB Clock Waveform Parameters

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
$1/(t_{CPPBB})$	PBB Clock Frequency	-40°C < Ambient Temperature < 70°C			84	MHz
$1/(t_{CPPBB})$	PBB Clock Frequency	-40°C < Ambient Temperature < 85°C			66	MHz

Table 7-34. SMC Write Signals with No Hold Settings (NWE Controlled only)

Symbol	Parameter	Min.	Unit
SMC_{43}	Data Out Valid before NWE Rising	(nwe pulse length - 1) * t_{CPSMC} - 1.2	ns
SMC_{44}	Data Out Valid after NWE Rising	5	ns
SMC_{45}	NWE Pulse Width	nwe pulse length * t_{CPSMC} - 0.9	ns

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

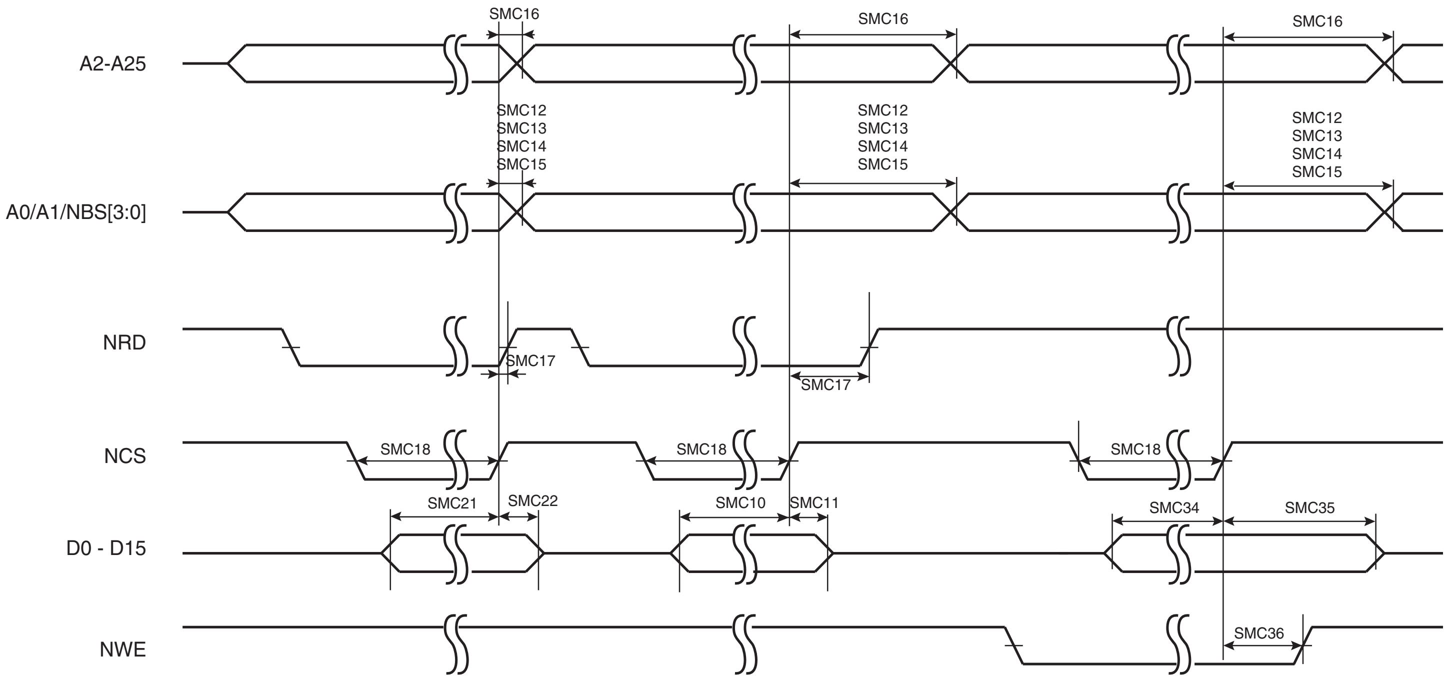
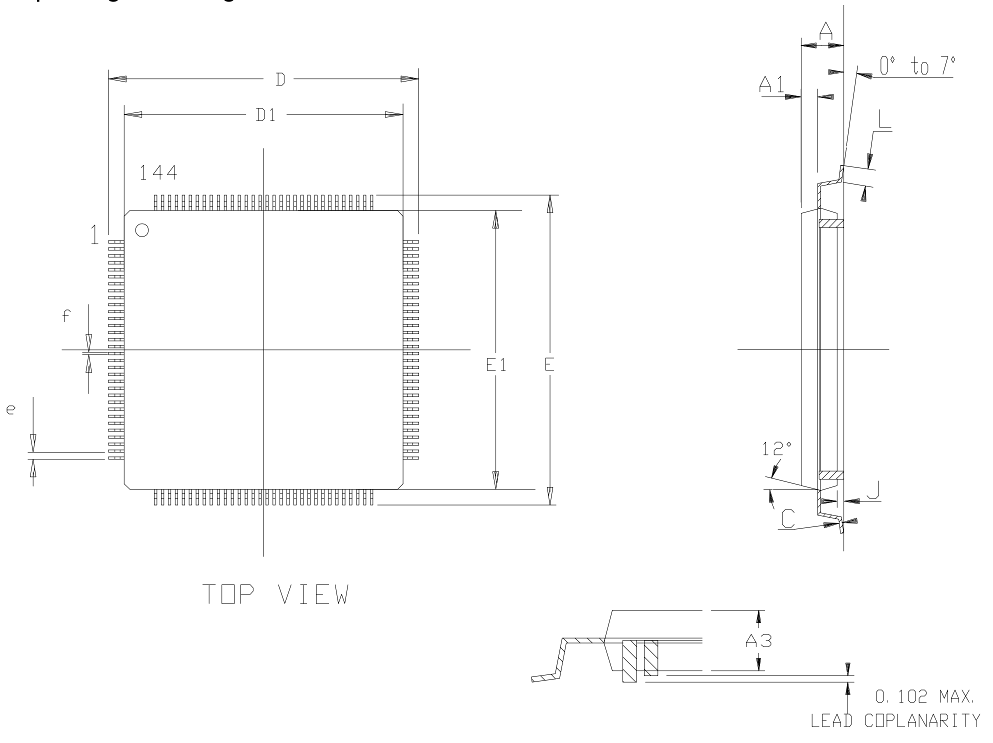


Figure 8-2. LQFP-144 package drawing



	MM			INCH		
	Min	Nom	Max	Min	Nom	Max
A	–	–	1. 60	–	–	. 063
C	0. 09	–	0. 20	. 004	–	. 008
A3	1. 35	1. 40	1. 45	. 053	. 055	. 057
D	21. 90	22. 00	22. 10	. 862	. 866	. 870
D1	19. 90	20. 00	20. 10	. 783	. 787	. 791
E	21. 90	22. 00	22. 10	. 862	. 866	. 870
E1	19. 90	20. 00	20. 10	. 783	. 787	. 791
J	0. 05	–	0. 15	. 002	–	. 006
L	0. 45	0. 60	0. 75	. 018	. 024	. 030
e	0. 50 BSC			. 0197 BSC		
f	0. 22 BSC			. 009 BSC		

Table 8-2. Device and Package Maximum Weight

1300	mg

Table 8-3. Package Characteristics

Moisture Sensitivity Level	MSL3

Table 8-4. Package Reference

JEDEC Drawing Reference	MS-026
JESD97 Classification	E3

Fix/Workaround
Set to 1b bit CORRS4 of the ECCHRS mode register (MD). In C-code: *((volatile int*) (0xFFFE2404))= 0x400.

DMACA data transfer fails when CTLx.SRC_TR_WIDTH is not equal to CTLx.DST_TR_WIDTH
Fix/Workaround
For any DMACA transfer make sure CTLx.SRC_TR_WIDTH = CTLx.DST_TR_WIDTH.

3.3V supply monitor is not available
FGPFRLO[30:29] are reserved and should not be used by the application.
Fix/Workaround
None.

Service access bus (SAB) can not access DMACA registers
Fix/Workaround
None.

10.2.2 Processor and Architecture

LDM instruction with PC in the register list and without ++ increments Rp
For LDM with PC in the register list: the instruction behaves as if the ++ field is always set, ie the pointer is always updated. This happens even if the ++ field is cleared. Specifically, the increment of the pointer is done in parallel with the testing of R12.
Fix/Workaround
None.

Hardware breakpoints may corrupt MAC results
Hardware breakpoints on MAC instructions may corrupt the destination register of the MAC instruction.
Fix/Workaround
Place breakpoints on earlier or later instructions.

When the main clock is RCSYS, TIMER_CLOCK5 is equal to PBA clock
When the main clock is generated from RCSYS, TIMER_CLOCK5 is equal to PBA Clock and not PBA Clock / 128.
Fix/Workaround
None.

10.2.3 MPU

Privilege violation when using interrupts in application mode with protected system stack
If the system stack is protected by the MPU and an interrupt occurs in application mode, an MPU DTLB exception will occur.
Fix/Workaround
Make a DTLB Protection (Write) exception handler which permits the interrupt request to be handled in privileged mode.

10.2.4 USB

UPCFGn.INTFRQ is irrelevant for isochronous pipe
As a consequence, isochronous IN and OUT tokens are sent every 1ms (Full Speed), or every 125uS (High Speed).

11.7 Rev. B – 08/09

1. Updated the datasheet with new device AT32UC3A4.

11.8 Rev. A – 03/09

1. Initial revision.

1 Description 3

2 Overview 4

2.1 Block Diagram 4

2.2 Configuration Summary 5

3 Package and Pinout 6

3.1 Package 6

3.2 Peripheral Multiplexing on I/O lines 9

3.3 Signal Descriptions 14

3.4 I/O Line Considerations 19

3.5 Power Considerations 20

4 Processor and Architecture 21

4.1 Features 21

4.2 AVR32 Architecture 21

4.3 The AVR32UC CPU 22

4.4 Programming Model 26

4.5 Exceptions and Interrupts 30

5 Memories 34

5.1 Embedded Memories 34

5.2 Physical Memory Map 34

5.3 Peripheral Address Map 35

5.4 CPU Local Bus Mapping 37

6 Boot Sequence 39

6.1 Starting of Clocks 39

6.2 Fetching of Initial Instructions 39

7 Electrical Characteristics 40

7.1 Absolute Maximum Ratings* 40

7.2 DC Characteristics 41

7.3 I/O pin Characteristics 42

7.4 Regulator characteristics 43

7.5 Analog characteristics 44

7.6 Power Consumption 48

7.7 System Clock Characteristics 51

7.8 Oscillator Characteristics 52

7.9 ADC Characteristics 54

7.10 USB Transceiver Characteristics 55
7.11 EBI Timings 57
7.12 JTAG Characteristics 63
7.13 SPI Characteristics 64
7.14 MCI 66
7.15 Flash Memory Characteristics 67

8 Mechanical Characteristics 68

8.1 Thermal Considerations 68
8.2 Package Drawings 69
8.3 Soldering Profile 72

9 Ordering Information 73

10 Errata 74

10.1 Rev. H 74
10.2 Rev. E 78
10.3 Rev. D 84

11 Datasheet Revision History 90

11.1 Rev. H– 10/12 90
11.2 Rev. G– 11/11 90
11.3 Rev. F – 08/11 90
11.4 Rev. E – 06/11 90
11.5 Rev. D – 04/11 90
11.6 Rev. C – 03/10 90
11.7 Rev. B – 08/09 91
11.8 Rev. A – 03/09 91