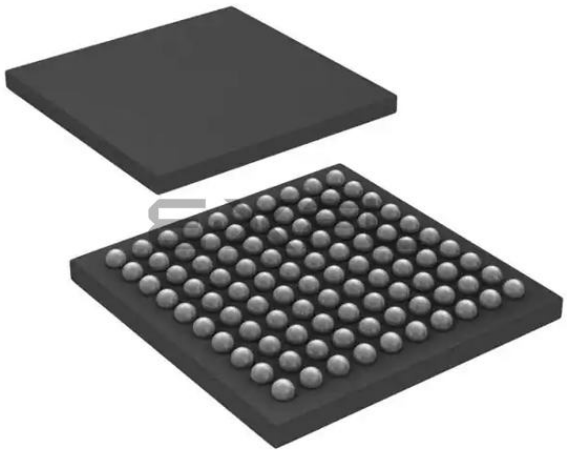


Welcome to E-XFL.COM

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	88
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	100-VFBGA
Supplier Device Package	100-VFBGA (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at32uc3a464-c1ur

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)

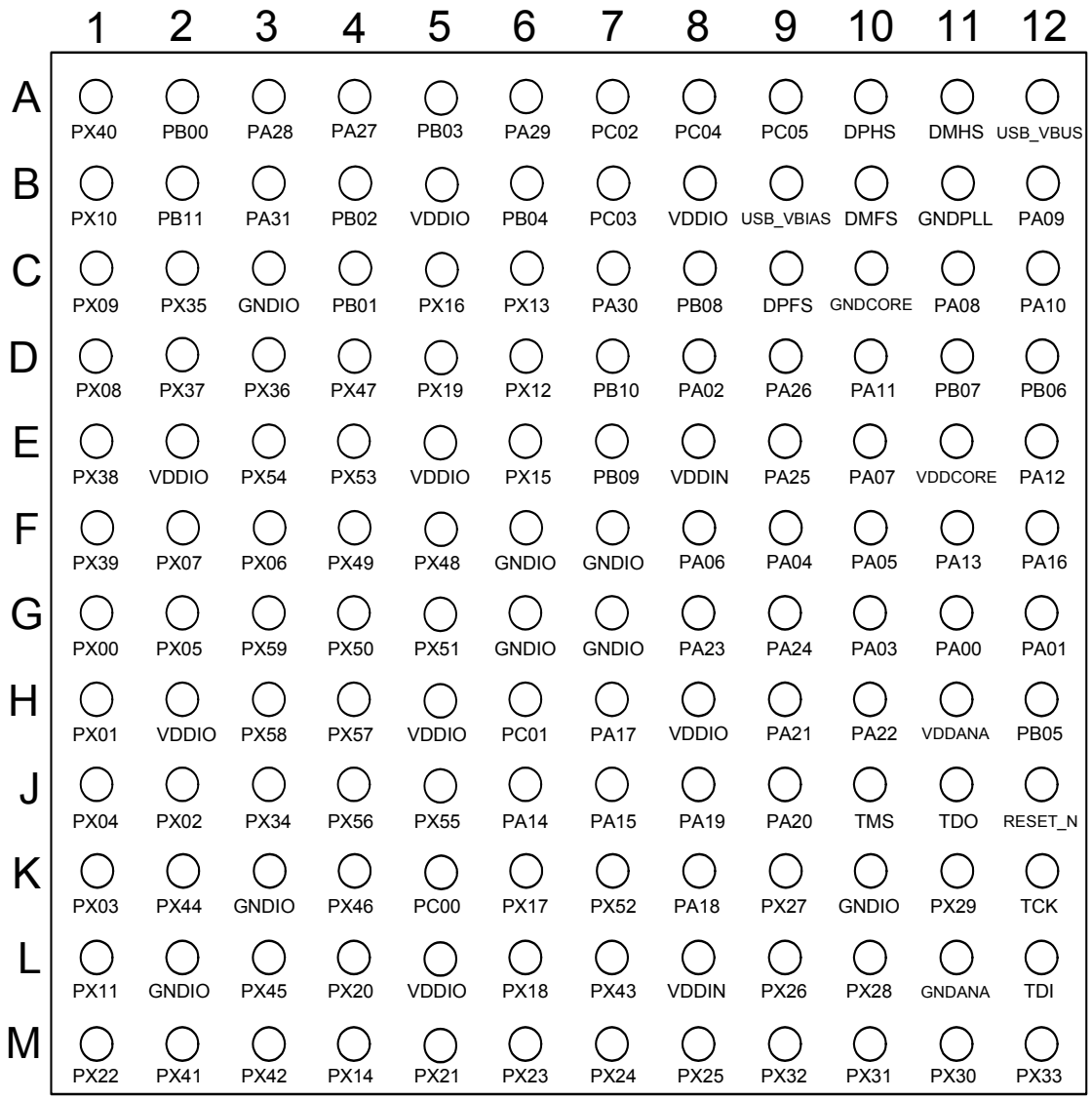
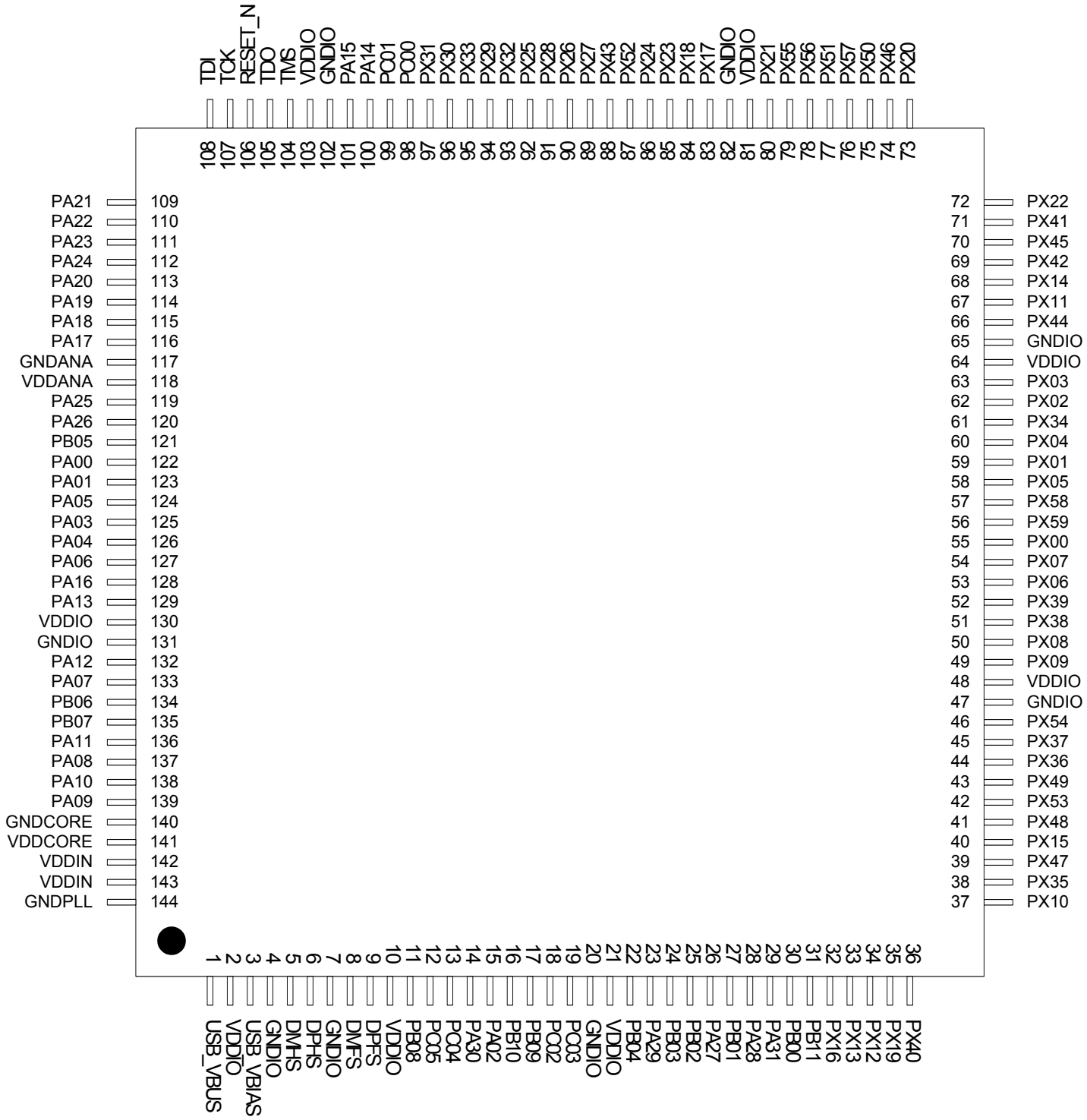


Figure 3-2. LQFP144 Pinout



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 micro-architectures, 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.

Table 4-3. System Registers (Continued)

Reg #	Address	Name	Function
26	104	JAVA_LV3	Unused in AVR32UC
27	108	JAVA_LV4	Unused in AVR32UC
28	112	JAVA_LV5	Unused in AVR32UC
29	116	JAVA_LV6	Unused in AVR32UC
30	120	JAVA_LV7	Unused in AVR32UC
31	124	JTBA	Unused in AVR32UC
32	128	JBCR	Unused in AVR32UC
33-63	132-252	Reserved	Reserved for future use
64	256	CONFIG0	Configuration register 0
65	260	CONFIG1	Configuration register 1
66	264	COUNT	Cycle Counter register
67	268	COMPARE	Compare register
68	272	TLBEHI	Unused in AVR32UC
69	276	TLBELO	Unused in AVR32UC
70	280	PTBR	Unused in AVR32UC
71	284	TLBEAR	Unused in AVR32UC
72	288	MMUCR	Unused in AVR32UC
73	292	TLBARLO	Unused in AVR32UC
74	296	TLBARHI	Unused in AVR32UC
75	300	PCCNT	Unused in AVR32UC
76	304	PCNT0	Unused in AVR32UC
77	308	PCNT1	Unused in AVR32UC
78	312	PCCR	Unused in AVR32UC
79	316	BEAR	Bus Error Address Register
80	320	MPUAR0	MPU Address Register region 0
81	324	MPUAR1	MPU Address Register region 1
82	328	MPUAR2	MPU Address Register region 2
83	332	MPUAR3	MPU Address Register region 3
84	336	MPUAR4	MPU Address Register region 4
85	340	MPUAR5	MPU Address Register region 5
86	344	MPUAR6	MPU Address Register region 6
87	348	MPUAR7	MPU Address Register region 7
88	352	MPUPSR0	MPU Privilege Select Register region 0
89	356	MPUPSR1	MPU Privilege Select Register region 1
90	360	MPUPSR2	MPU Privilege Select Register region 2
91	364	MPUPSR3	MPU Privilege Select Register region 3

Table 5-2. Peripheral Address Mapping

0xFFFF0C00	PM	Power Manager - PM
0xFFFF0D00	RTC	Real Time Counter - RTC
0xFFFF0D30	WDT	Watchdog Timer - WDT
0xFFFF0D80	EIC	External Interrupt Controller - EIC
0xFFFF1000	GPIO	General Purpose Input/Output Controller - GPIO
0xFFFF1400	USART0	Universal Synchronous/Asynchronous Receiver/Transmitter - USART0
0xFFFF1800	USART1	Universal Synchronous/Asynchronous Receiver/Transmitter - USART1
0xFFFF1C00	USART2	Universal Synchronous/Asynchronous Receiver/Transmitter - USART2
0xFFFF2000	USART3	Universal Synchronous/Asynchronous Receiver/Transmitter - USART3
0xFFFF2400	SPI0	Serial Peripheral Interface - SPI0
0xFFFF2800	SPI1	Serial Peripheral Interface - SPI1
0xFFFF2C00	TWIM0	Two-wire Master Interface - TWIM0
0xFFFF3000	TWIM1	Two-wire Master Interface - TWIM1
0xFFFF3400	SSC	Synchronous Serial Controller - SSC
0xFFFF3800	TC0	Timer/Counter - TC0
0xFFFF3C00	ADC	Analog to Digital Converter - ADC
0xFFFF4000	ABDAC	Audio Bitstream DAC - ABDAC
0xFFFF4400	TC1	Timer/Counter - TC1

Table 5-2. Peripheral Address Mapping

0xFFFF5000	TWIS0	Two-wire Slave Interface - TWIS0
0xFFFF5400	TWIS1	Two-wire Slave Interface - TWIS1

5.4 CPU Local Bus Mapping

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

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

The following GPIO registers are mapped on the local bus:

Table 5-3. Local Bus Mapped GPIO Registers

Port	Register	Mode	Local Bus Address	Access
0	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
1	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

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

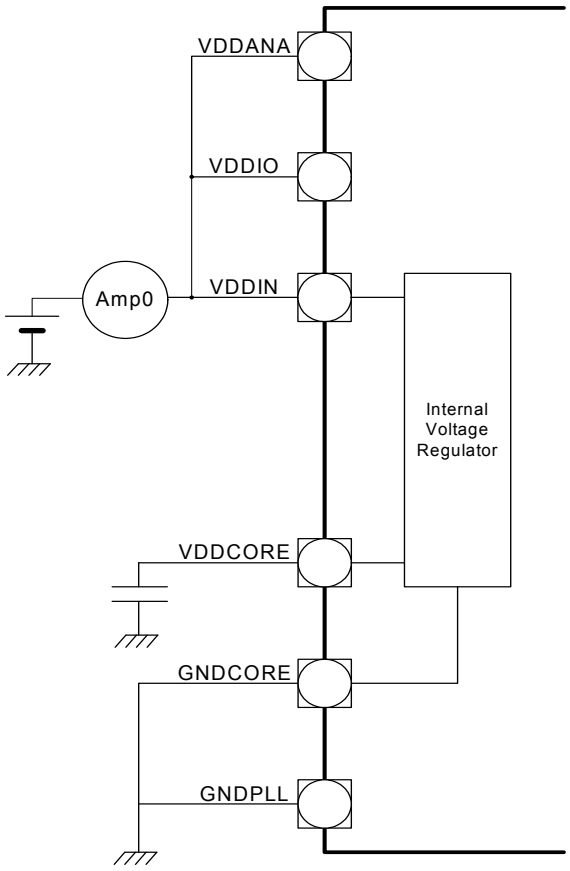
Symbol	Parameter	Conditions		Min.	Typ.	Max.	Unit
V_{VDDIO}	DC Supply Peripheral I/Os			3.0		3.6	V
V_{VDDANA}	DC Analog Supply			3.0		3.6	V
V_{IL}	Input Low-level Voltage	All I/O pins except TWCK, TWD, RESET_N, TCK, TDI		-0.3		+0.8	V
		TWCK, TWD		V_{VDDIO} x0.7		V_{VDDIO} +0.5	V
		RESET_N, TCK, TDI		+0.8V			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

7.6 Power Consumption

The values in Table 7-12 and Table 7-13 on page 50 are measured values of power consumption with operating conditions as follows:

- V_{DDIO} = 3.3V
- T_A = 25°C
- I/Os are configured in input, pull-up enabled.

Figure 7-6. Measurement Setup



These figures represent the power consumption measured on the power supplies

Table 7-13. Typical Cuurent Consumption by Peripheral

Peripheral	Typ.	Unit
ADC	7	µA/MHz
AES	80	
ABDAC	10	
DMACA	70	
EBI	23	
EIC	0.5	
GPIO	37	
INTC	3	
MCI	40	
MSI	10	
PDCA	20	
SDRAM	5	
SMC	9	
SPI	6	
SSC	10	
RTC	5	
TC	8	
TWIM	2	
TWIS	2	
USART	10	
USBB	90	
WDT	2	

Table 7-26. 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

7.10 USB Transceiver Characteristics

7.10.1 Electrical Characteristics

Table 7-27. 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
R_{BIAS}	VBIAS External Resistor (1)	±1%		6810		Ω
C_{BIAS}	VBIAS External Capcitor			10		pF

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

7.10.2 Static Power Consumption

Table 7-28. Static Power Consumption

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
I_{BIAS}	Bias current consumption on VBG				1	µA
$I_{VDDUTMI}$	HS Transceiver and I/O current consumption				8	µA
	FS/HS Transceiver and I/O current consumption	If cable is connected, add 200µA (typical) due to Pull-up/Pull-down current consumption			3	µA

7.10.3 Dynamic Power Consumption

Table 7-29. Dynamic Power Consumption

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
I_{BIAS}	Bias current consumption on VBG			0.7	0.8	mA

Table 7-29. Dynamic Power Consumption

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
$I_{VDDUTMI}$	HS Transceiver current consumption	HS transmission		47	60	mA
	HS Transceiver current consumption	HS reception		18	27	mA
	FS/HS Transceiver current consumption	FS transmission 0m cable (1)		4	6	mA
	FS/HS Transceiver current consumption	FS transmission 5m cable		26	30	mA
	FS/HS Transceiver current consumption	FS reception		3	4.5	mA

1. Including 1 mA due to Pull-up/Pull-down current consumption.

34.5.5 USB High Speed Design Guidelines

In order to facilitate hardware design, Atmel provides an application note on www.atmel.com.

Table 7-36. SDRAM Clock Signal

Symbol	Parameter	Conditions	Min.	Max.	Unit
$SDRAMC_{13}$	Bank Change before SDCK Rising Edge		6.3		ns
$SDRAMC_{14}$	Bank Change after SDCK Rising Edge		2.4		ns
$SDRAMC_{15}$	CAS Low before SDCK Rising Edge		7.4		ns
$SDRAMC_{16}$	CAS High after SDCK Rising Edge		1.9		ns
$SDRAMC_{17}$	DQM Change before SDCK Rising Edge		6.4		ns
$SDRAMC_{18}$	DQM Change after SDCK Rising Edge		2.2		ns
$SDRAMC_{19}$	D0-D15 in Setup before SDCK Rising Edge		9		ns
$SDRAMC_{20}$	D0-D15 in Hold after SDCK Rising Edge		0		ns
$SDRAMC_{23}$	SDWE Low before SDCK Rising Edge		7.6		ns
$SDRAMC_{24}$	SDWE High after SDCK Rising Edge		1.8		ns
$SDRAMC_{25}$	D0-D15 Out Valid before SDCK Rising Edge		7.1		ns
$SDRAMC_{26}$	D0-D15 Out Valid after SDCK Rising Edge		1.5		ns

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

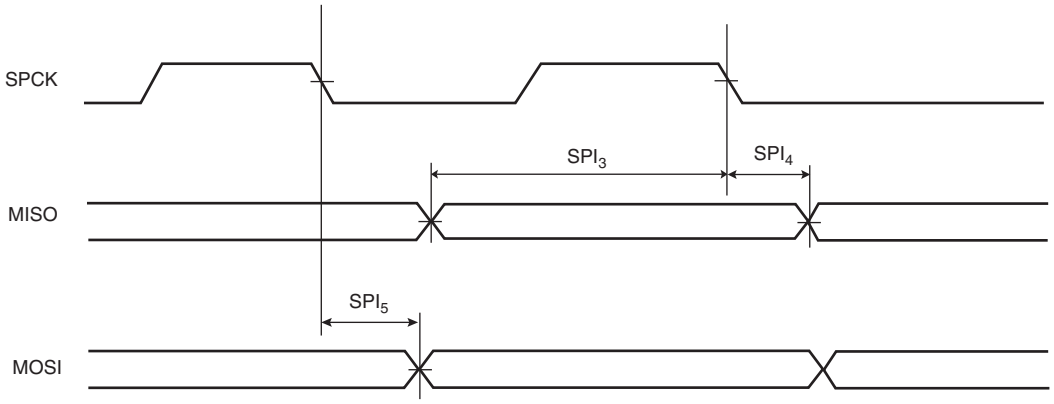


Figure 7-13. SPI Slave mode with (CPOL= 0 and NCPHA= 1) or (CPOL= 1 and NCPHA= 0)

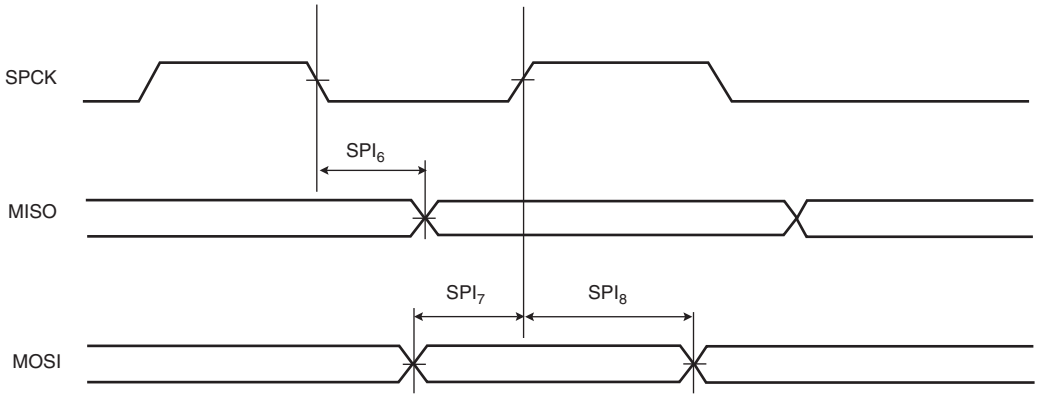


Figure 7-14. SPI Slave mode with (CPOL= NCPHA= 0) or (CPOL= NCPHA= 1)

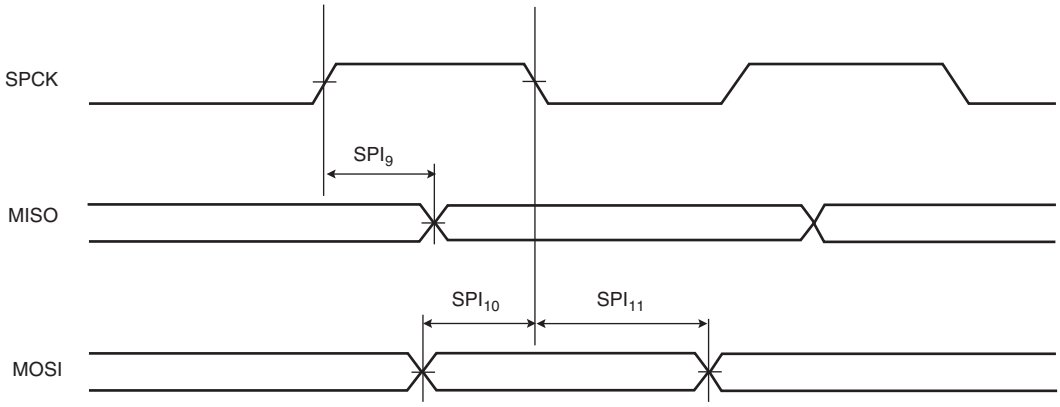
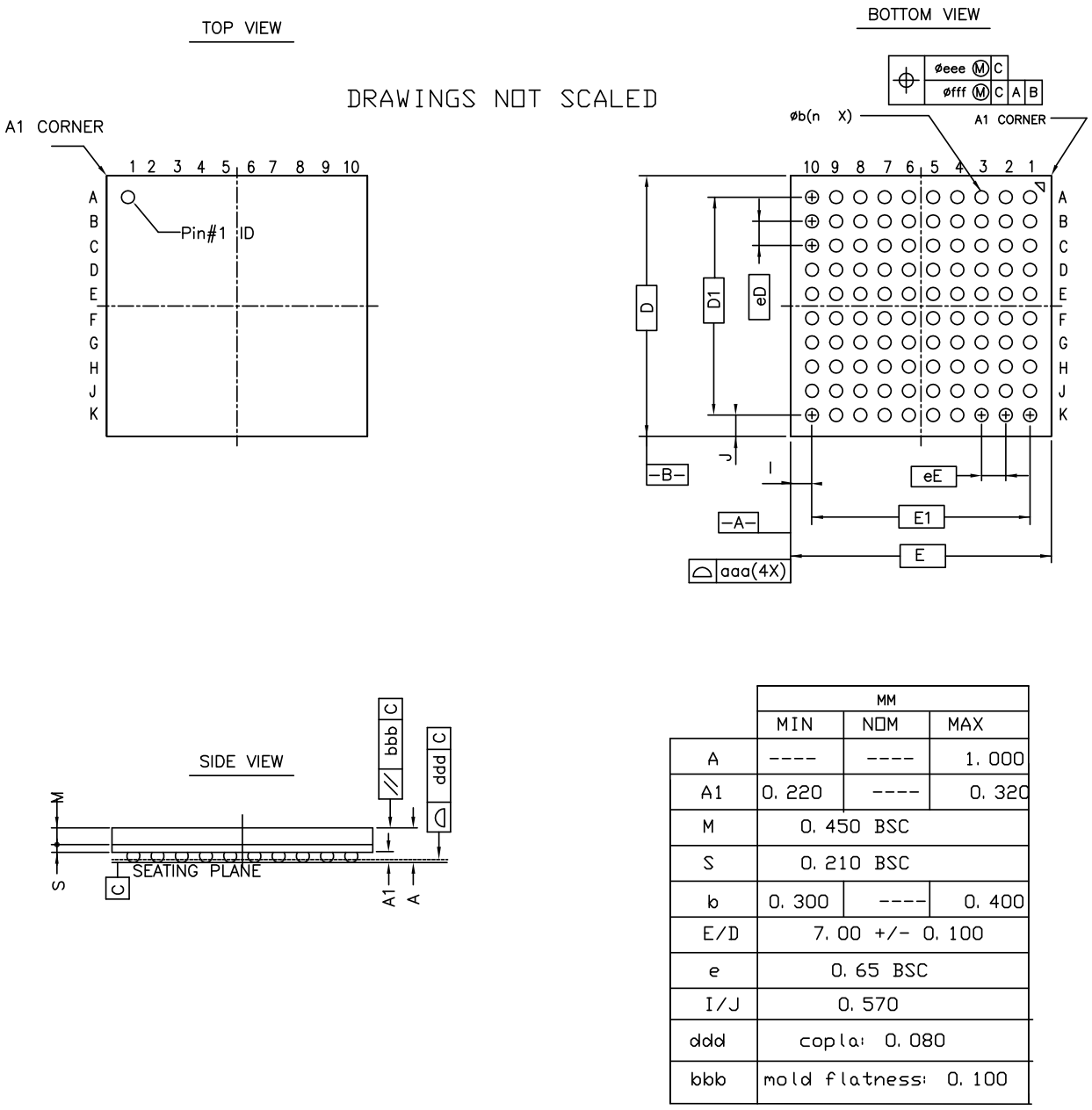


Figure 8-3. VFBGA-100 package drawing



	MM		
	MIN	NOM	MAX
A	----	----	1.000
A1	0.220	----	0.320
M	0.450 BSC		
S	0.210 BSC		
b	0.300	----	0.400
E/D	7.00 +/- 0.100		
e	0.65 BSC		
I/J	0.570		
ddd	copla: 0.080		
bbb	mold flatness: 0.100		

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.

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.

10.1.8 Power Manager

OSC32 not functionnal in Crystal Modes (OSC32CTRL.MODE=1 or OSC32CTRL.MODE=2)
OSC32 clock output is not active even if the oscillation signal is present on XIN32/XOUT32 pins.
OSC32RDY bit may still set even if the CLK32 is not active.
External clock mode (OSC32CTRL.MODE=0) is not affected.

Fix/Workaround
None.

Clock sources will not be stopped in STATIC sleep mode if the difference between CPU and PBx division factor is too high
If the division factor between the CPU/HSB and PBx frequencies is more than 4 when going to a sleep mode where the system RC oscillator is turned off, then high speed clock sources will not be turned off. This will result in a significantly higher power consumption during the sleep mode.
Fix/Workaround
Before going to sleep modes where the system RC oscillator is stopped, make sure that the factor between the CPU/HSB and PBx frequencies is less than or equal to 4.

10.1.9 PDCA

PCONTROL.CHxRES is non-functional
PCONTROL.CHxRES is non-functional. Counters are reset at power-on, and cannot be reset by software.
Fix/Workaround
Software needs to keep history of performance counters.

Transfer error will stall a transmit peripheral handshake interface
If a transfer error is encountered on a channel transmitting to a peripheral, the peripheral handshake of the active channel will stall and the PDCA will not do any more transfers on the affected peripheral handshake interface.
Fix/Workaround
Disable and then enable the peripheral after the transfer error.

10.1.10 AES

URAD (Unspecified Register Access Detection Status) does not detect read accesses to the write-only KEYW[5..8]R registers
Fix/Workaround
None.

10.1.11 HMATRIX

In the PRAS and PRBS registers, the MxPR fields are only two bits
In the PRAS and PRBS registers, the MxPR fields are only two bits wide, instead of four bits. The unused bits are undefined when reading the registers.
Fix/Workaround
Mask undefined bits when reading PRAS and PRBS.

10.1.12 TWIM

TWIM SR.IDLE goes high immediately when NAK is received
When a NAK is received and there is a non-zero number of bytes to be transmitted, SR.IDLE goes high immediately and does not wait for the STOP condition to be sent. This does not cause any problem just by itself, but can cause a problem if software waits for SR.IDLE to go high and then immediately disables the TWIM by writing a one to CR.MDIS. Disabling the TWIM causes the TWCK and TWD pins to go high immediately, so the STOP condition will not be transmitted correctly.
Fix/Workaround
If possible, do not disable the TWIM. If it is absolutely necessary to disable the TWIM, there must be a software delay of at least two TWCK periods between the detection of SR.IDLE==1 and the disabling of the TWIM.

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

SMBALERT bit may be set after reset
The SMBus Alert (SMBALERT) bit in the Status Register (SR) might be erroneously set after system reset.
Fix/Workaround
After system reset, clear the SR.SMBALERT bit before commencing any TWI transfer.

10.1.13 TWIS

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.

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.

10.1.14 SSC

Frame Synchro and Frame Synchro Data are delayed by one clock cycle
The frame synchro and the frame synchro data are delayed from 1 SSC_CLOCK when:
- Clock is CKDIV
- The START is selected on either a frame synchro edge or a level
- Frame synchro data is enabled
- Transmit clock is gated on output (through CKO field)

Fix/Workaround
Transmit or receive CLOCK must not be gated (by the mean of CKO field) when START condition is performed on a generated frame synchro.

10.1.15 FLASHC

Corrupted read in flash may happen after fuses write or erase operations (FLASHC LP, UP, WGPB, EGPB, SSB, PGPFB, EAGPF commands)
After a flash fuse write or erase operation (FLASHC LP, UP, WGPB, EGPB, SSB, PGPFB, EAGPF commands), reading (data read or code fetch) in flash may fail. This may lead to an exception or to other errors derived from this corrupted read access.
Fix/Workaround
Before the flash fuse write or erase operation, enable the flash high speed mode (FLASHC HSEN command). The flash fuse write or erase operations (FLASHC LP, UP, WGPB, EGPB, SSB, PGPFB, EAGPF commands) must be issued from RAM or through the EBI. After these commands, read 3 times one flash page initialized to 00h. Disable the flash high speed mode (FLASHC HSDIS command). It is then possible to safely read or code fetch the flash.

10.2 Rev. E

10.2.1 General

Devices cannot operate with CPU frequency higher than 66MHz in 1WS and 36MHz in 0WS
Fix/Workaround
None

Increased Power Consumption in VDDIO in sleep modes
If the OSC0 is enabled in crystal mode when entering a sleep mode where the OSC0 is disabled, this will lead to an increased power consumption in VDDIO.
Fix/Workaround
Disable the OSC0 through the System Control Interface (SCIF) before going to any sleep mode where the OSC0 is disabled, or pull down or up XIN0 and XOUT0 with 1 Mohm resistor.

Power consumption in static mode The power consumption in static mode can be up to 330µA on some parts (typical at 25°C)

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.

RETE instruction does not clear SREG[L] from interrupts
The RETE instruction clears SREG[L] as expected from exceptions.
Fix/Workaround
When using the STCOND instruction, clear SREG[L] in the stacked value of SR before returning from interrupts with RETE.

RETS behaves incorrectly when MPU is enabled
RETS behaves incorrectly when MPU is enabled and MPU is configured so that system stack is not readable in unprivileged mode.
Fix/Workaround
Make system stack readable in unprivileged mode, or return from supervisor mode using rete instead of rets. This requires:
1. Changing the mode bits from 001 to 110 before issuing the instruction. Updating the mode bits to the desired value must be done using a single mtsr instruction so it is done atomically. Even if this step is generally described as not safe in the UC technical reference manual, it is safe in this very specific case.
2. Execute the RETE instruction.

In the PRAS and PRBS registers, the MxPR fields are only two bits
In the PRAS and PRBS registers, the MxPR fields are only two bits wide, instead of four bits. The unused bits are undefined when reading the registers.
Fix/Workaround
Mask undefined bits when reading PRAS and PRBS.

Multiply instructions do not work on RevD
All the multiply instructions do not work.
Fix/Workaround
Do not use the multiply instructions.

10.3.3 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.

10.3.4 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).
Fix/Workaround
For higher polling time, the software must freeze the pipe for the desired period in order to prevent any "extra" token.

11. Datasheet Revision History

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

11.1 Rev. H– 10/12

1. Updated max frequency
2. Added Flash Read High Speed Mode description in FLASHC chapter
3. Updated Electrical Characteristics accordingly to new max frequency
4. Fixed wrong description of PLLOPT[0] in PM chapter
5. Updated Errata section according to new maximum frequency
6. Added USB hi-speed PLL electrical characteristics
7. Added OSC32 Errata in Power Management sections for Rev D,E and H

11.2 Rev. G– 11/11

1. Add recommandation for MCI connection with more than 1 slot

11.3 Rev. F – 08/11

1. Final version

11.4 Rev. E – 06/11

1. Updated Errata for E and D
2. Updated FLASHC chapter with HSEN and HSDIS commands

11.5 Rev. D – 04/11

1. Updated Errata for revision H and E
2. Updated Reset Sequence
3. Updated Peripherals' current consumption and others minor electrical charateristics
4. Updated Peripherals chapters

11.6 Rev. C – 03/10

1. Updated the datasheet with new revision H features.