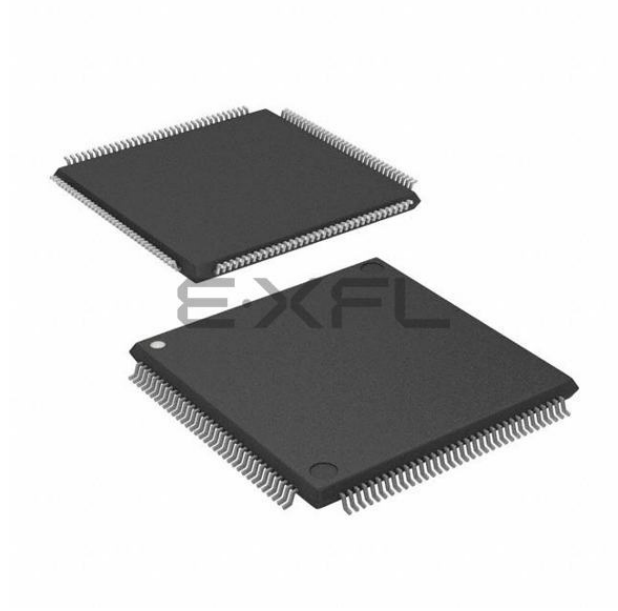


Welcome to E-XFL.COM

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	128KB (128K 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-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at32uc3a3128s-alut

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)

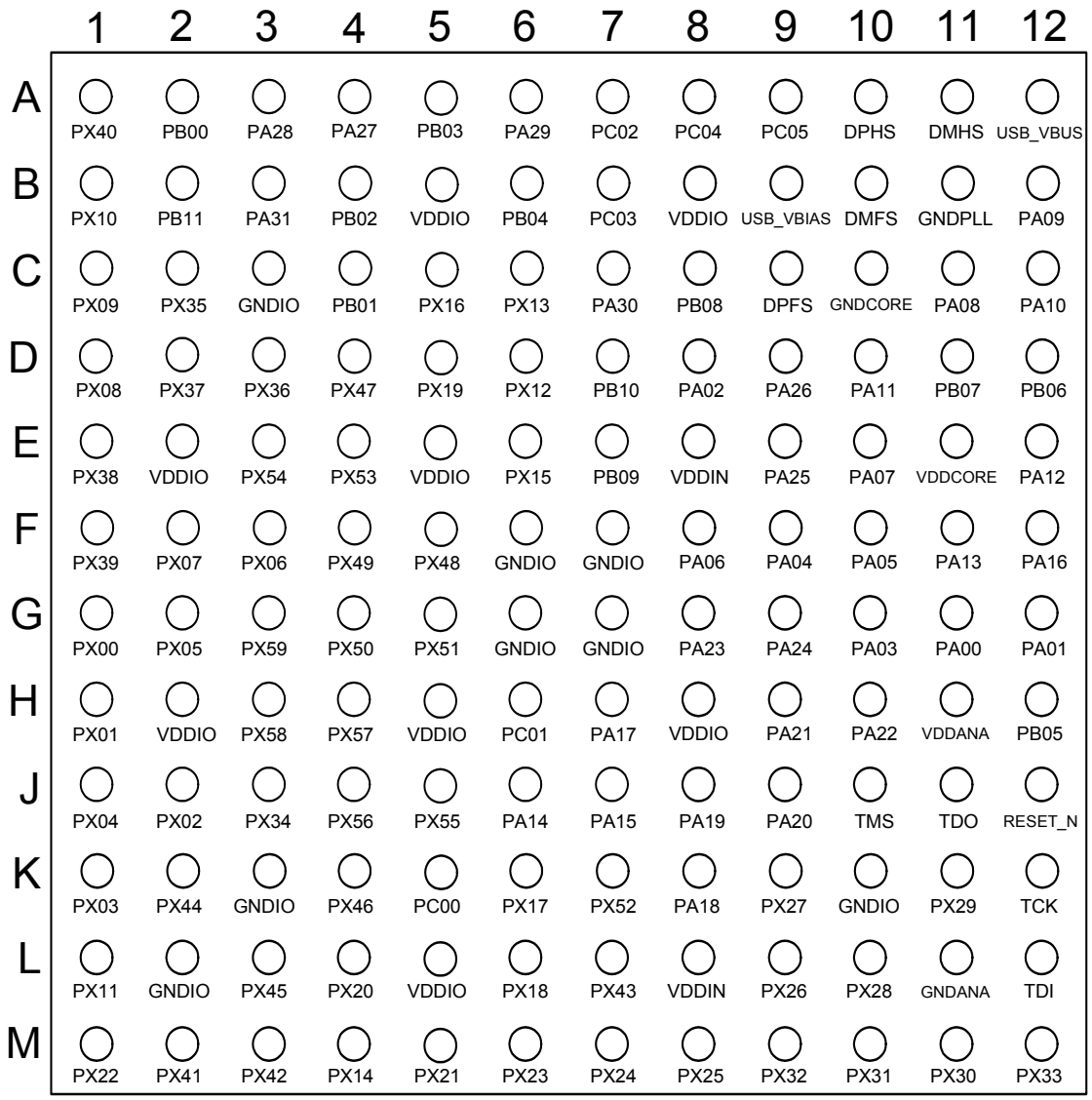


Table 3-1. GPIO Controller Function Multiplexing

BGA 144	QFP 144	BGA 100	PIN	GPIO	Supply	PIN Type (2)	GPIO function			
							A	B	C	D
J4	78		PX56	107	VDDIO	x2	EBI - ADDR[21]	EIC - SCAN[2]	USART2 - TXD	
H4	76		PX57	108	VDDIO	x2	EBI - ADDR[20]	EIC - SCAN[1]	USART3 - RXD	
H3	57		PX58	109	VDDIO	x2	EBI - NCS[0]	EIC - SCAN[0]	USART3 - TXD	
G3	56	F1(1)	PX59	110	VDDIO	x2	EBI - NANDWE		MCI - CMD[1]	

Note: 1. Those balls are physically connected to 2 GPIOs. Software must managed carrefully the GPIO configuration to avoid electrical conflict.

2. Refer to "Electrical Characteristics" on page 40 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 on the same pin.

Table 3-2. Peripheral Functions

Function	Description
GPIO Controller Function multiplexing	GPIO and GPIO peripheral selection A to D
Nexus OCD AUX port connections	OCD trace system
JTAG port connections	JTAG debug port
Oscillators	OSC0, OSC1, OSC32

3.2.3 Oscillator Pinout

The oscillators are not mapped to the normal GPIO functions and their muxings are controlled by registers in the Power Mananger (PM). Please refer to the PM chapter for more information about this.

Table 3-3. Oscillator Pinout

TFBGA144	QFP144	VFBGA100	Pin name	Oscillator pin
A7	18	A5	PC02	XIN0
B7	19	A6	PC03	XOUT0
A8	13	B7	PC04	XIN1
A9	12	A7	PC05	XOUT1
K5	98	K5(1)	PC00	XIN32
H6	99	K6	PC01	XOUT32

Note: 1. This ball is physically connected to 2 GPIOs. Software must managed carrefully the GPIO configuration to avoid electrical conflict

Table 3-6. Signal Description List

Signal Name	Function	Type	Active Level	Comments
DMHS	USB High Speed Data -	Analog		
DPHS	USB High Speed Data +	Analog		
USB_VBIAS	USB VBIAS reference	Analog		Connect to the ground through a 6810 ohms (+/- 1%) resistor in parallel with a 10pf capacitor. If USB hi-speed feature is not required, leave this pin unconnected to save power
USB_VBUS	USB VBUS signal	Output		
VBOF	USB VBUS on/off bus power control port	Output		
ID	ID Pin fo the USB bus	Input		

3.4 I/O Line Considerations

3.4.1 JTAG Pins

TMS and TDI pins have pull-up resistors. TDO pin is an output, driven at up to VDDIO, and has no pull-up resistor.

3.4.2 RESET_N Pin

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

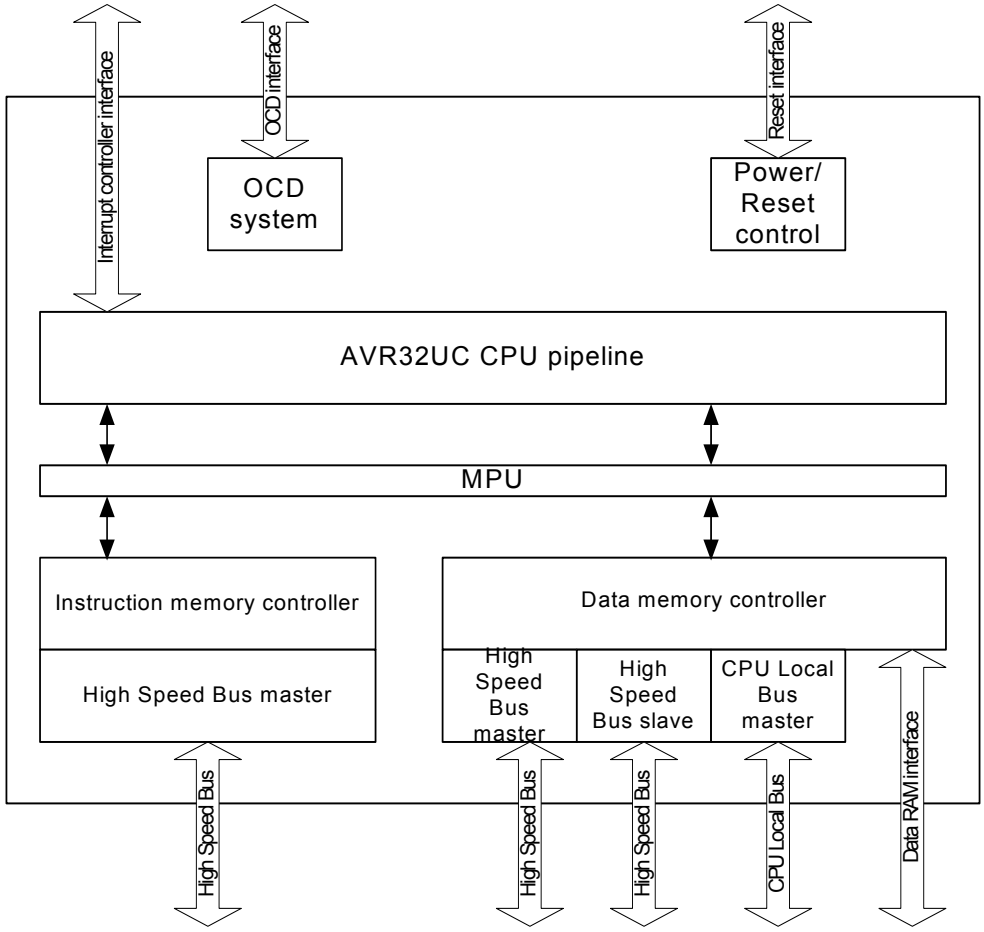
3.4.3 TWI Pins

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

3.4.4 GPIO Pins

All the I/O lines integrate a programmable pull-up resistor. Programming of this pull-up resistor is performed independently for each I/O line through the I/O Controller. After reset, I/O lines default as inputs with pull-up resistors disabled, except when indicated otherwise in the column “Reset State” of the I/O Controller multiplexing tables.

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

Table 4-4. Priority and Handler Addresses for Events

Priority	Handler Address	Name	Event source	Stored Return Address
1	0x8000_0000	Reset	External input	Undefined
2	Provided by OCD system	OCD Stop CPU	OCD system	First non-completed instruction
3	EVBA+0x00	Unrecoverable exception	Internal	PC of offending instruction
4	EVBA+0x04	TLB multiple hit	MPU	
5	EVBA+0x08	Bus error data fetch	Data bus	First non-completed instruction
6	EVBA+0x0C	Bus error instruction fetch	Data bus	First non-completed instruction
7	EVBA+0x10	NMI	External input	First non-completed instruction
8	Autovectored	Interrupt 3 request	External input	First non-completed instruction
9	Autovectored	Interrupt 2 request	External input	First non-completed instruction
10	Autovectored	Interrupt 1 request	External input	First non-completed instruction
11	Autovectored	Interrupt 0 request	External input	First non-completed instruction
12	EVBA+0x14	Instruction Address	CPU	PC of offending instruction
13	EVBA+0x50	ITLB Miss	MPU	
14	EVBA+0x18	ITLB Protection	MPU	PC of offending instruction
15	EVBA+0x1C	Breakpoint	OCD system	First non-completed instruction
16	EVBA+0x20	Illegal Opcode	Instruction	PC of offending instruction
17	EVBA+0x24	Unimplemented instruction	Instruction	PC of offending instruction
18	EVBA+0x28	Privilege violation	Instruction	PC of offending instruction
19	EVBA+0x2C	Floating-point	UNUSED	
20	EVBA+0x30	Coprocessor absent	Instruction	PC of offending instruction
21	EVBA+0x100	Supervisor call	Instruction	PC(Supervisor Call) +2
22	EVBA+0x34	Data Address (Read)	CPU	PC of offending instruction
23	EVBA+0x38	Data Address (Write)	CPU	PC of offending instruction
24	EVBA+0x60	DTLB Miss (Read)	MPU	
25	EVBA+0x70	DTLB Miss (Write)	MPU	
26	EVBA+0x3C	DTLB Protection (Read)	MPU	PC of offending instruction
27	EVBA+0x40	DTLB Protection (Write)	MPU	PC of offending instruction
28	EVBA+0x44	DTLB Modified	UNUSED	

Table 5-1. AT32UC3A3A4 Physical Memory Map

Device	Start Address	Size AT32UC3A3256S AT32UC3A3256 AT32UC3A4256S AT32UC3A4256	Size AT32UC3A3128S AT32UC3A3128 AT32UC3A4128S AT32UC3A4128	Size AT32UC3A364S AT32UC3A364 AT32UC3A464S AT32UC3A464
HRAMC0	0xFF000000	32KByte	32KByte	32KByte
HRAMC1	0xFF008000	32KByte	32KByte	32KByte
HSB-PB Bridge A	0xFFFF0000	64KByte	64KByte	64KByte
HSB-PB Bridge B	0xFFFE0000	64KByte	64KByte	64KByte

5.3 Peripheral Address Map

Table 5-2. Peripheral Address Mapping

Address		Peripheral Name
0xFF100000	DMACA	DMA Controller - DMACA
0xFFFD0000	AES	Advanced Encryption Standard - AES
0xFFFE0000	USB	USB 2.0 Device and Host Interface - USB
0xFFFE1000	HMATRIX	HSB Matrix - HMATRIX
0xFFFE1400	FLASHC	Flash Controller - FLASHC
0xFFFE1C00	SMC	Static Memory Controller - SMC
0xFFFE2000	SDRAMC	SDRAM Controller - SDRAMC
0xFFFE2400	ECCHRS	Error code corrector Hamming and Reed Solomon - ECCHRS
0xFFFE2800	BUSMON	Bus Monitor module - BUSMON
0xFFFE4000	MCI	Mulitmedia Card Interface - MCI
0xFFFE8000	MSI	Memory Stick Interface - MSI
0xFFFF0000	PDCA	Peripheral DMA Controller - PDCA
0xFFFF0800	INTC	Interrupt controller - INTC

7.5.4 RESET_N Characteristics

Table 7-11. RESET_N Waveform Parameters

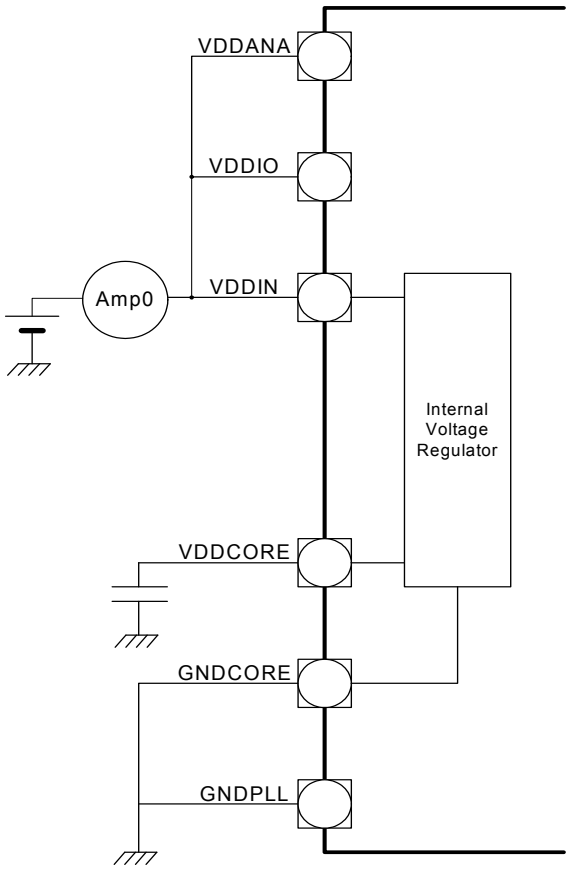
Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
t_{RESET}	RESET_N minimum pulse width		10			ns

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 7-26. 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 7-34. 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.

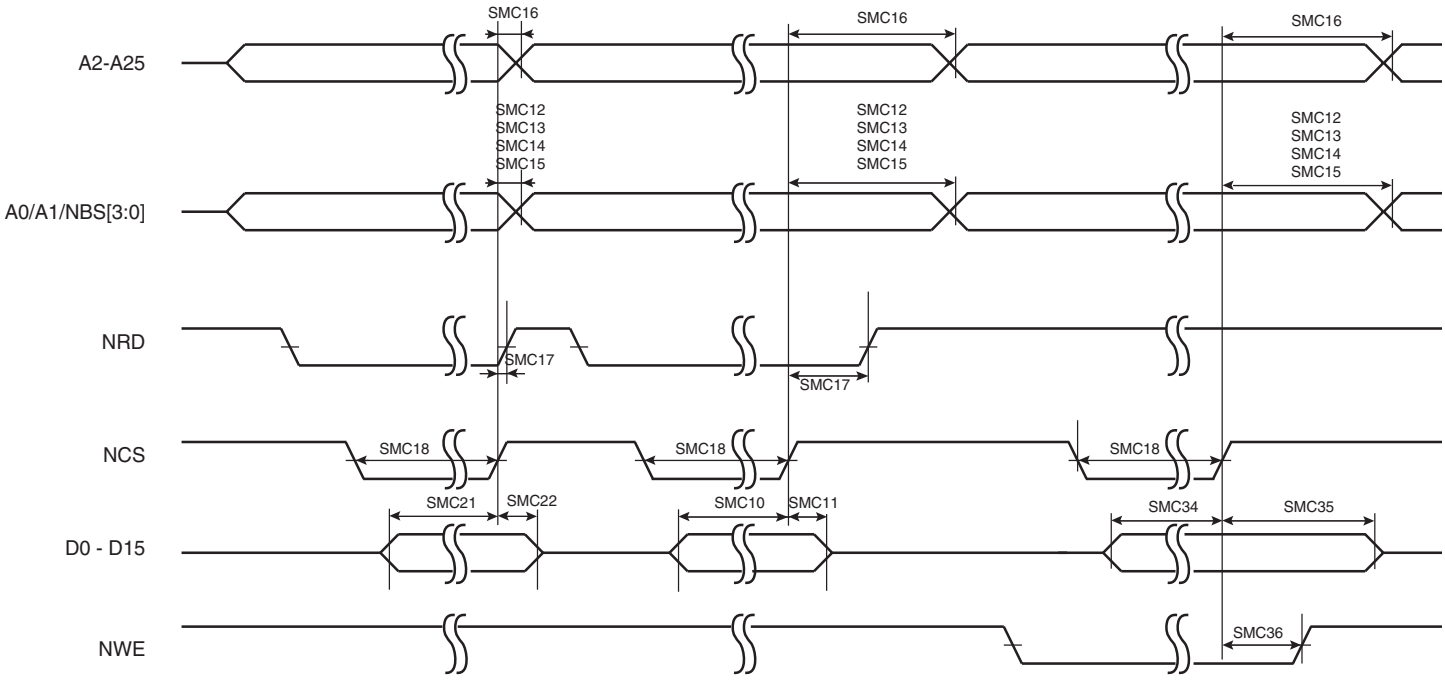
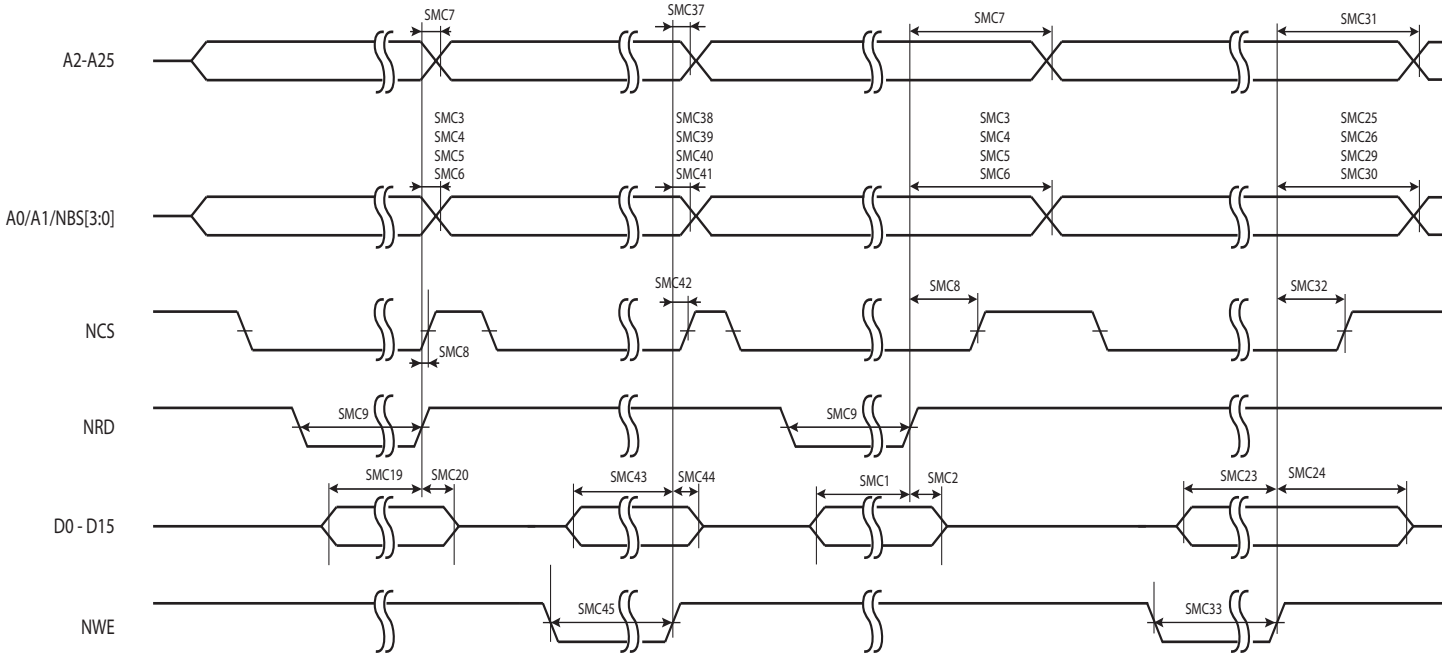


Figure 7-8. SMC Signals for NRD and NRW Controlled Accesses.



7.11.2 SDRAM Signals

These timings are given for 10 pF load on SDCK and 40 pF on other signals.

Table 7-35. SDRAM Clock Signal.

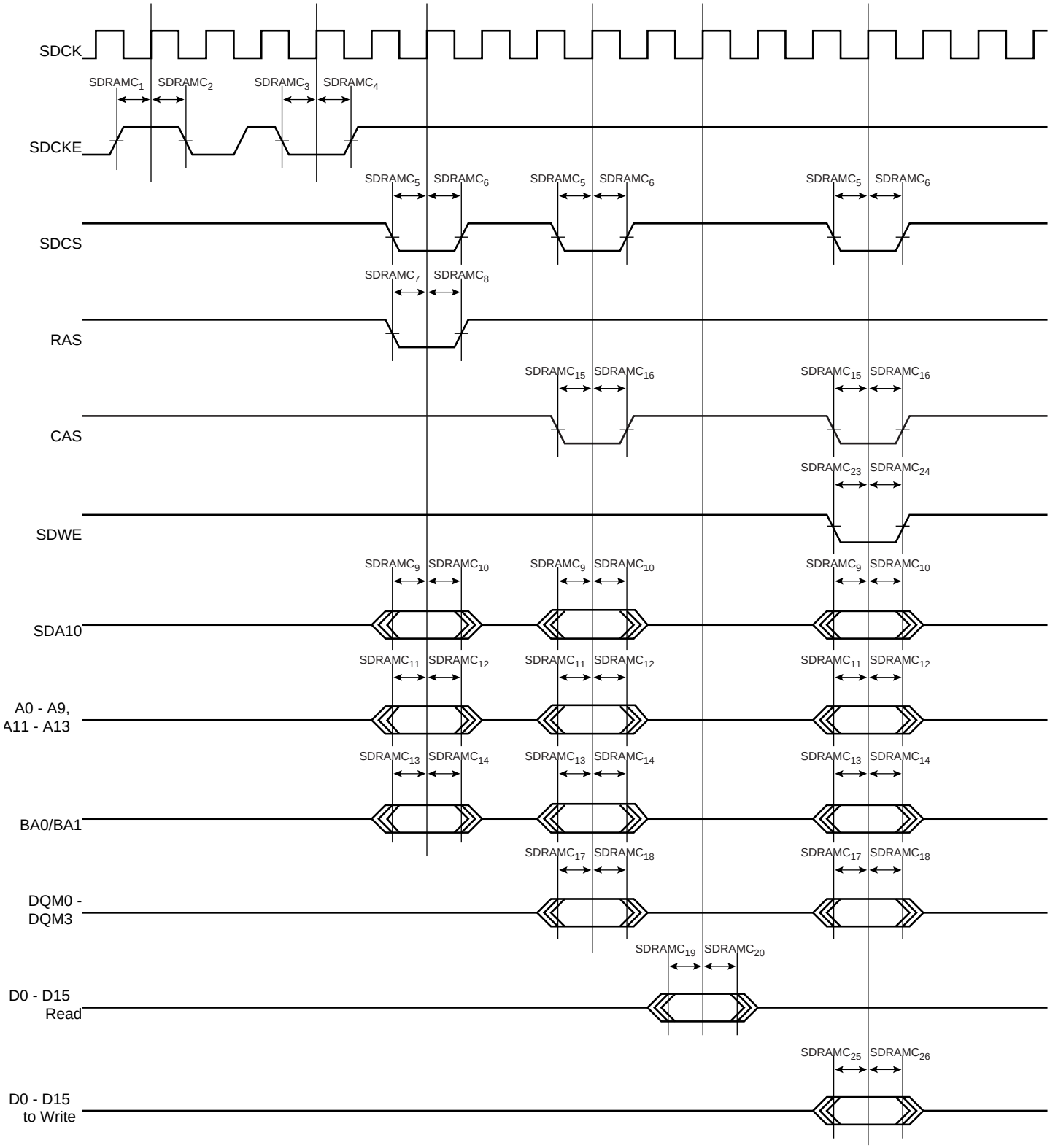
Symbol	Parameter	Conditions	Min.	Max.(1)	Unit
$1/(t_{CPSDCK})$	SDRAM Controller Clock Frequency			$1/(t_{cpcpu})$	MHz

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

Table 7-36. SDRAM Clock Signal

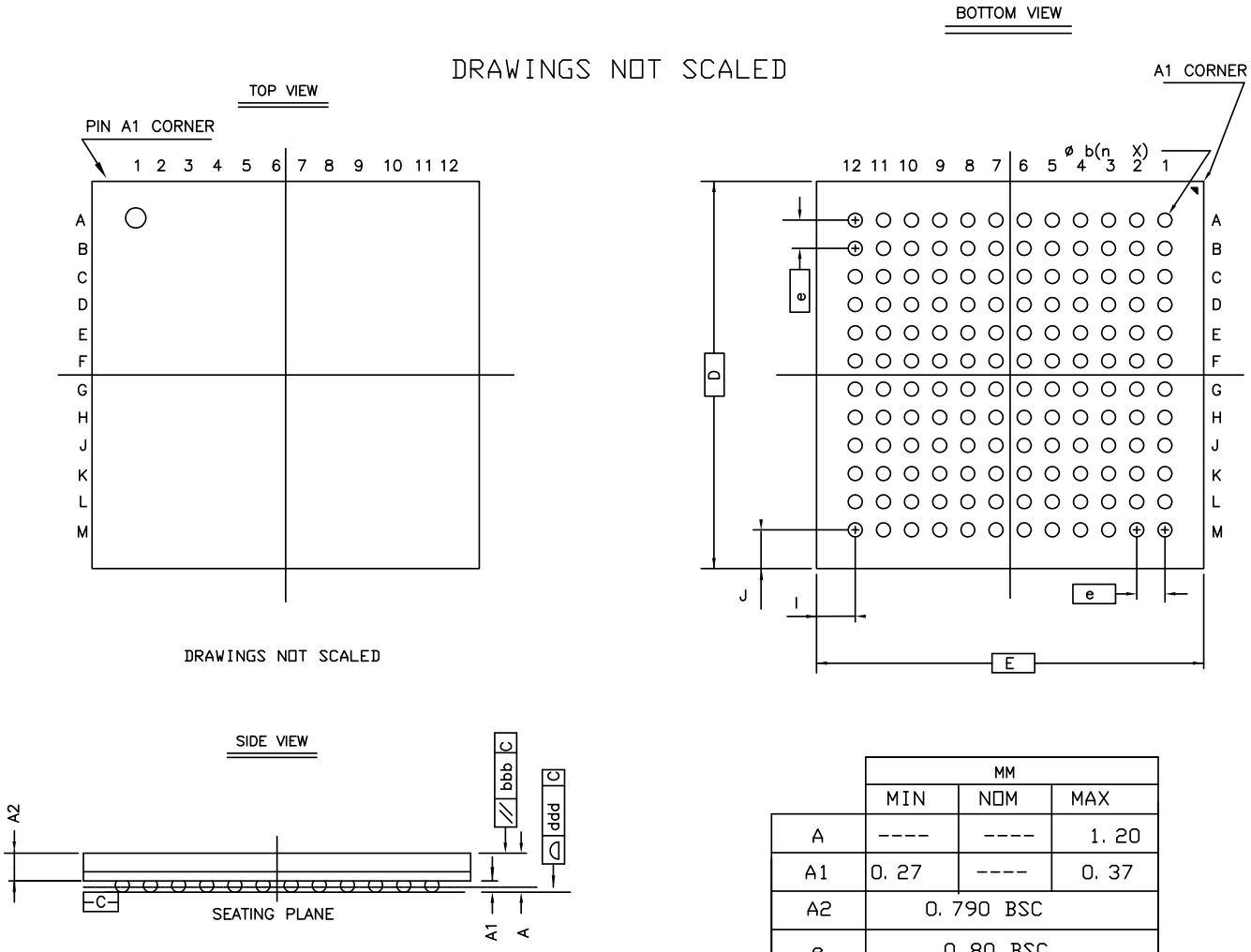
Symbol	Parameter	Conditions	Min.	Max.	Unit
$SDRAMC_1$	SDCKE High before SDCK Rising Edge		7.4		ns
$SDRAMC_2$	SDCKE Low after SDCK Rising Edge		3.2		ns
$SDRAMC_3$	SDCKE Low before SDCK Rising Edge		7		ns
$SDRAMC_4$	SDCKE High after SDCK Rising Edge		2.9		ns
$SDRAMC_5$	SDCS Low before SDCK Rising Edge		7.5		ns
$SDRAMC_6$	SDCS High after SDCK Rising Edge		1.6		ns
$SDRAMC_7$	RAS Low before SDCK Rising Edge		7.2		ns
$SDRAMC_8$	RAS High after SDCK Rising Edge		2.3		ns
$SDRAMC_9$	SDA10 Change before SDCK Rising Edge		7.6		ns
$SDRAMC_{10}$	SDA10 Change after SDCK Rising Edge		1.9		ns
$SDRAMC_{11}$	Address Change before SDCK Rising Edge		6.2		ns
$SDRAMC_{12}$	Address Change after SDCK Rising Edge		2.2		ns

Figure 7-9. SDRAMC Signals relative to SDCK.



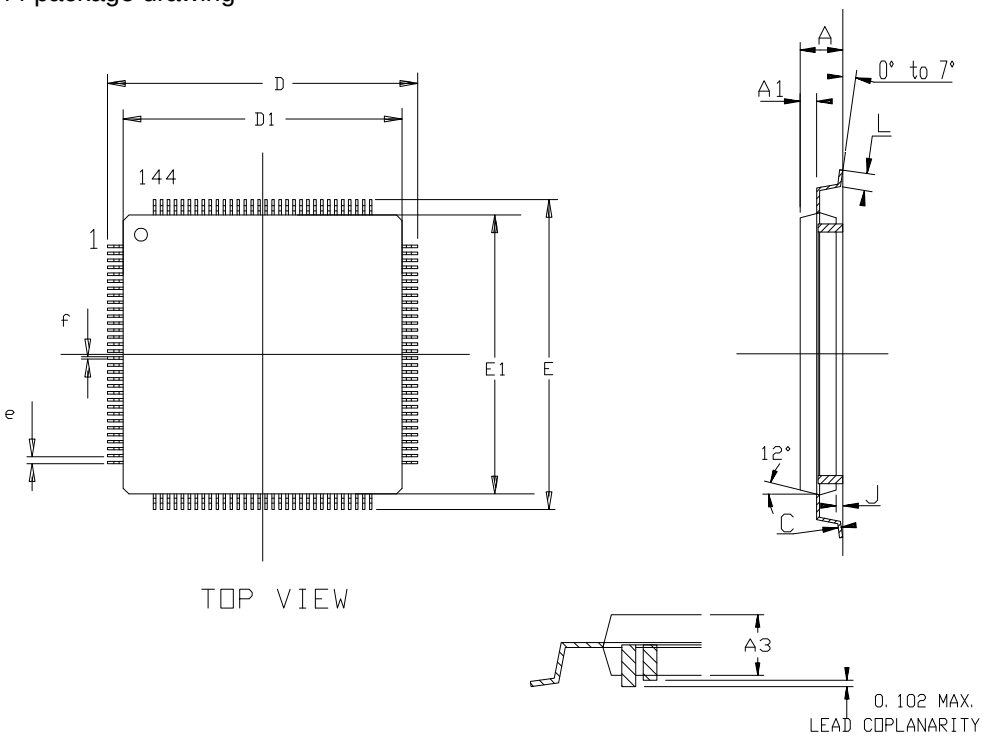
8.2 Package Drawings

Figure 8-1. TFBGA 144 package drawing



	MM		
	MIN	NOM	MAX
A	----	----	1.20
A1	0.27	----	0.37
A2	0.790 BSC		
e	0.80 BSC		
b	0.38	----	0.48
E/D	11.00 BSC		
I/J	1.100 REF		
ddd	coplar 0.120		
bbb	coplar 0.200		

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

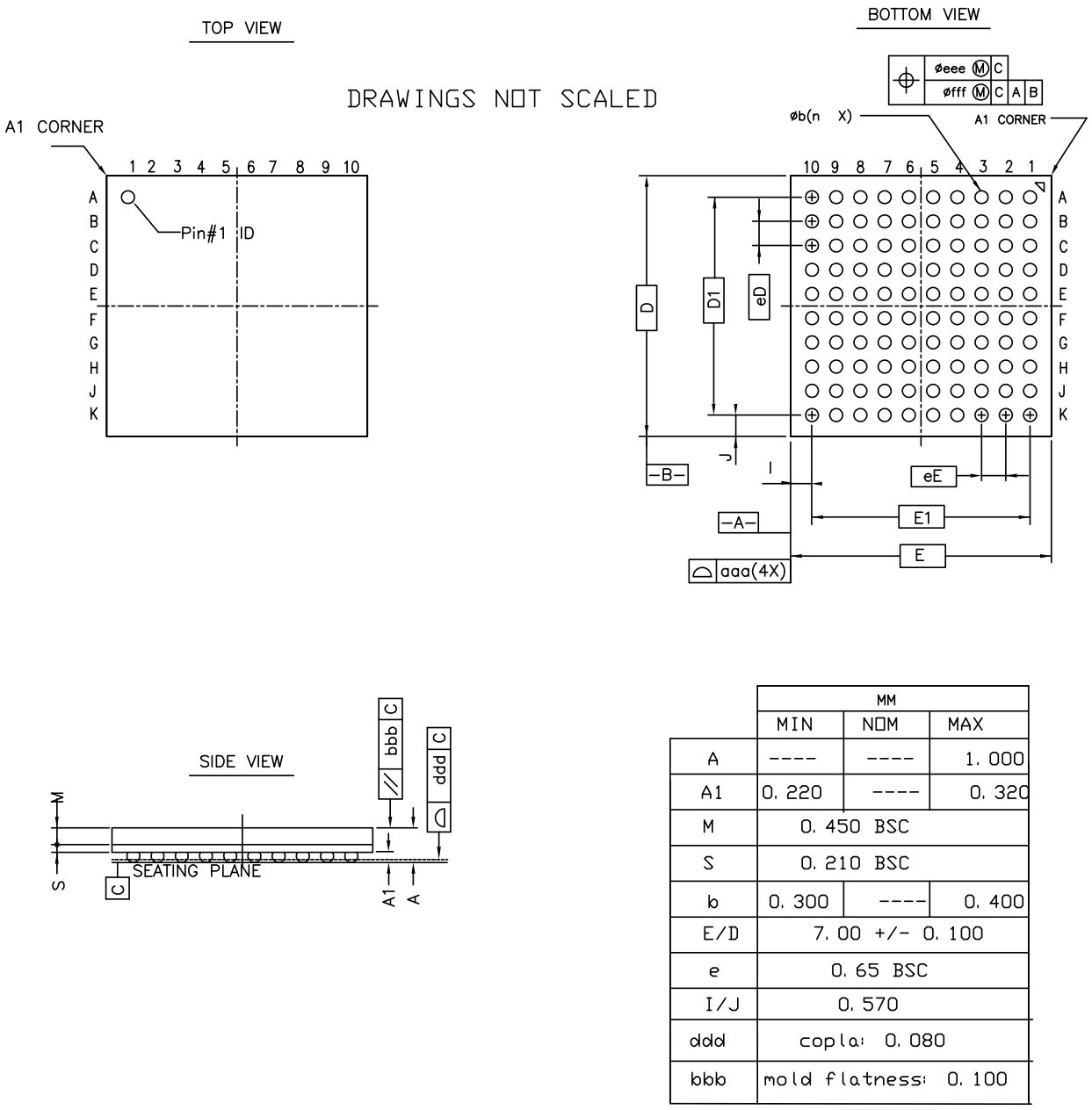
Table 8-3. Package Characteristics

Moisture Sensitivity Level	MSL3

Table 8-4. Package Reference

JEDEC Drawing Reference	MS-026
JESD97 Classification	E3

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		

10. Errata

10.1 Rev. H

10.1.1 General

Devices with Date Code lower than 1233 cannot operate with CPU frequency higher than 66MHz in 1WS and 36MHz in 0WS in the whole temperature range
Fix/Workaround
None

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.

10.1.2 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.

10.1.3 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.

10.1.4 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

10.2.16 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.

10.3 Rev. D

10.3.1 General

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

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.

10.3.2 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.

11.7 Rev. B – 08/09

1. Updated the datasheet with new device AT32UC3A4.

11.8 Rev. A – 03/09

1. Initial revision.