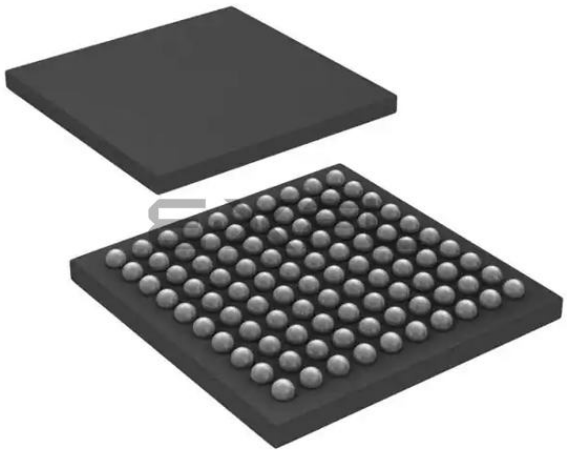


Welcome to E-XFL.COM

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.

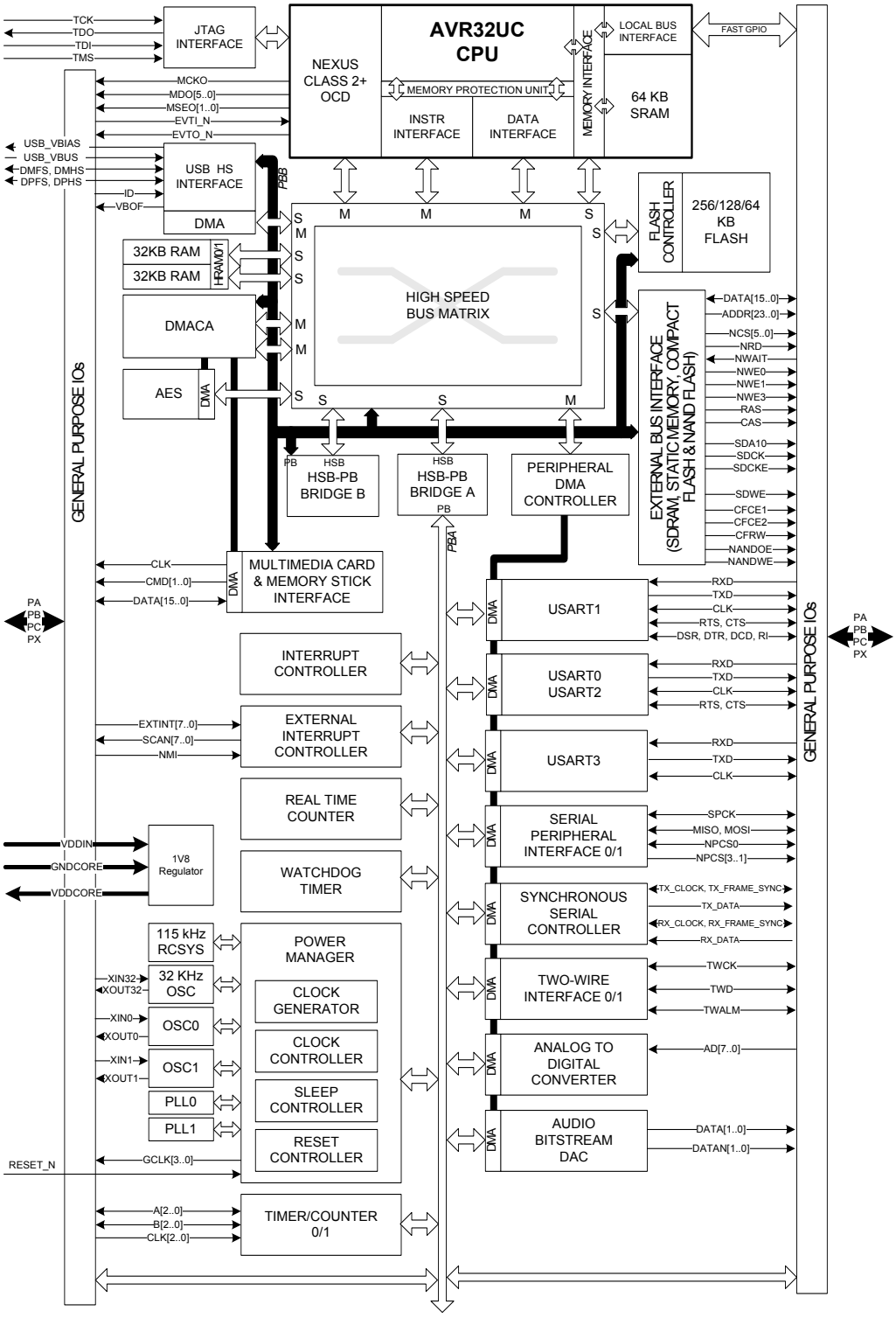
Applications of "Embedded - Microcontrollers"

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-c1ut

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)

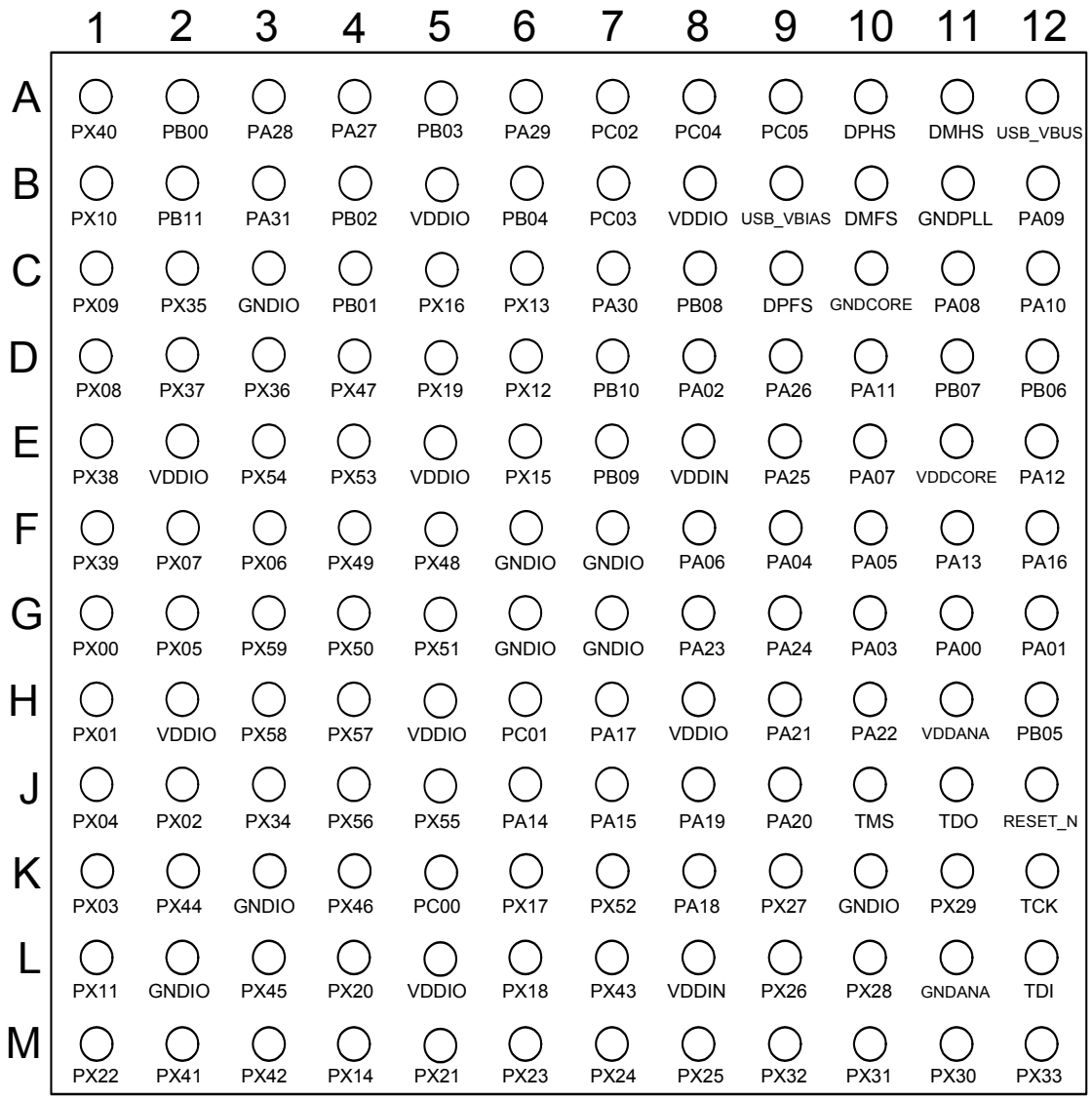


Table 3-1. GPIO Controller Function Multiplexing

BGA 144	QFP 144	BGA 100	PIN	GPIO	Supply	PIN Type (2)	GPIO function			
							A	B	C	D
C7	14	A4	PA30	30	VDDIO	x1	MCI - DATA[1]	USART3 - CLK	DMACA - DMAACK[0]	MSI - DATA[1]
B3	29	C2	PA31	31	VDDIO	x1	MCI - DATA[2]	USART2 - RXD	DMACA - DMARQ[0]	MSI - DATA[2]
A2	30	B1	PB00	32	VDDIO	x1	MCI - DATA[3]	USART2 - TXD	ADC - TRIGGER	MSI - DATA[3]
C4	27	B2	PB01	33	VDDIO	x1	MCI - DATA[4]	ABDAC - DATA[1]	EIC - SCAN[0]	MSI - INS
B4	25	B3	PB02	34	VDDIO	x1	MCI - DATA[5]	ABDAC - DATAN[1]	EIC - SCAN[1]	
A5	24	C4	PB03	35	VDDIO	x1	MCI - DATA[6]	USART2 - CLK	EIC - SCAN[2]	
B6	22	A3	PB04	36	VDDIO	x1	MCI - DATA[7]	USART3 - RXD	EIC - SCAN[3]	
H12	121	F7(1)	PB05	37	VDDIO	x3	USB - ID	TC0 - A0	EIC - SCAN[4]	
D12	134	D7	PB06	38	VDDIO	x1	USB - VBOF	TC0 - B0	EIC - SCAN[5]	
D11	135	D6	PB07	39	VDDIO	x3	SPI1 - SPCK	SSC - TX_CLOCK	EIC - SCAN[6]	
C8	11	C6	PB08	40	VDDIO	x2	SPI1 - MISO	SSC - TX_DATA	EIC - SCAN[7]	
E7	17	C5	PB09	41	VDDIO	x2	SPI1 - NPCS[0]	SSC - RX_DATA	EBI - NCS[4]	
D7	16	D5	PB10	42	VDDIO	x2	SPI1 - MOSI	SSC - RX_FRAME_SYNC	EBI - NCS[5]	
B2	31	C1	PB11	43	VDDIO	x1	USART1 - RXD	SSC - TX_FRAME_SYNC	PM - GCLK[1]	
K5	98	K5(1)	PC00	45	VDDIO	x1				
H6	99	K6	PC01	46	VDDIO	x1				
A7	18	A5	PC02	47	VDDIO	x1				
B7	19	A6	PC03	48	VDDIO	x1				
A8	13	B7	PC04	49	VDDIO	x1				
A9	12	A7	PC05	50	VDDIO	x1				
G1	55	G4	PX00	51	VDDIO	x2	EBI - DATA[10]	USART0 - RXD	USART1 - RI	
H1	59	G2	PX01	52	VDDIO	x2	EBI - DATA[9]	USART0 - TXD	USART1 - DTR	
J2	62	G3	PX02	53	VDDIO	x2	EBI - DATA[8]	USART0 - CTS	PM - GCLK[0]	
K1	63	J1	PX03	54	VDDIO	x2	EBI - DATA[7]	USART0 - RTS		
J1	60	H1	PX04	55	VDDIO	x2	EBI - DATA[6]	USART1 - RXD		
G2	58	G1	PX05	56	VDDIO	x2	EBI - DATA[5]	USART1 - TXD		
F3	53	F3	PX06	57	VDDIO	x2	EBI - DATA[4]	USART1 - CTS		
F2	54	F4	PX07	58	VDDIO	x2	EBI - DATA[3]	USART1 - RTS		
D1	50	E3	PX08	59	VDDIO	x2	EBI - DATA[2]	USART3 - RXD		
C1	49	E4	PX09	60	VDDIO	x2	EBI - DATA[1]	USART3 - TXD		
B1	37	D2	PX10	61	VDDIO	x2	EBI - DATA[0]	USART2 - RXD		
L1	67	K7(1)	PX11	62	VDDIO	x2	EBI - NWE1	USART2 - TXD		
D6	34	D1	PX12	63	VDDIO	x2	EBI - NWE0	USART2 - CTS	MCI - CLK	
C6	33	D3	PX13	64	VDDIO	x2	EBI - NRD	USART2 - RTS	MCI - CLK	
M4	68	K5(1)	PX14	65	VDDIO	x2	EBI - NCS[1]		TC0 - A0	
E6	40	K4(1)	PX15	66	VDDIO	x2	EBI - ADDR[19]	USART3 - RTS	TC0 - B0	
C5	32	D4(1)	PX16	67	VDDIO	x2	EBI - ADDR[18]	USART3 - CTS	TC0 - A1	
K6	83	J10(1)	PX17	68	VDDIO	x2	EBI - ADDR[17]	DMACA - DMARQ[1]	TC0 - B1	

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
SDA10	SDRAM Address 10 Line	Output		
SDCK	SDRAM Clock	Output		
SDCKE	SDRAM Clock Enable	Output		
SDWE	SDRAM Write Enable	Output	Low	
MultiMedia Card Interface - MCI				
CLK	Multimedia Card Clock	Output		
CMD[1:0]	Multimedia Card Command	I/O		
DATA[15:0]	Multimedia Card Data	I/O		
Memory Stick Interface - MSI				
SCLK	Memory Stick Clock	Output		
BS	Memory Stick Command	I/O		
DATA[3:0]	Multimedia Card Data	I/O		
Serial Peripheral Interface - SPI0, SPI1				
MISO	Master In Slave Out	I/O		
MOSI	Master Out Slave In	I/O		
NPCS[3:0]	SPI Peripheral Chip Select	I/O	Low	
SPCK	Clock	Output		
Synchronous Serial Controller - SSC				
RX_CLOCK	SSC Receive Clock	I/O		
RX_DATA	SSC Receive Data	Input		
RX_FRAME_SYNC	SSC Receive Frame Sync	I/O		
TX_CLOCK	SSC Transmit Clock	I/O		
TX_DATA	SSC Transmit Data	Output		
TX_FRAME_SYNC	SSC Transmit Frame Sync	I/O		
Timer/Counter - TC0, TC1				
A0	Channel 0 Line A	I/O		
A1	Channel 1 Line A	I/O		
A2	Channel 2 Line A	I/O		

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		

The register file is organized as sixteen 32-bit registers and includes the Program Counter, the Link Register, and the Stack Pointer. In addition, register R12 is designed to hold return values from function calls and is used implicitly by some instructions.

4.3 The AVR32UC CPU

The AVR32UC CPU targets low- and medium-performance applications, and provides an advanced OCD system, no caches, and a Memory Protection Unit (MPU). Java acceleration hardware is not implemented.

AVR32UC provides three memory interfaces, one High Speed Bus master for instruction fetch, one High Speed Bus master for data access, and one High Speed Bus slave interface allowing other bus masters to access data RAMs internal to the CPU. Keeping data RAMs internal to the CPU allows fast access to the RAMs, reduces latency, and guarantees deterministic timing. Also, power consumption is reduced by not needing a full High Speed Bus access for memory accesses. A dedicated data RAM interface is provided for communicating with the internal data RAMs.

A local bus interface is provided for connecting the CPU to device-specific high-speed systems, such as floating-point units and fast GPIO ports. This local bus has to be enabled by writing the LOCEN bit in the CPUCR system register. The local bus is able to transfer data between the CPU and the local bus slave in a single clock cycle. The local bus has a dedicated memory range allocated to it, and data transfers are performed using regular load and store instructions. Details on which devices that are mapped into the local bus space is given in the Memories chapter of this data sheet.

Figure 4-1 on page 23 displays the contents of AVR32UC.

The following table shows the instructions with support for unaligned addresses. All other instructions require aligned addresses.

Table 4-1. Instructions with Unaligned Reference Support

Instruction	Supported alignment
ld.d	Word
st.d	Word

4.3.6 Unimplemented Instructions

The following instructions are unimplemented in AVR32UC, and will cause an Unimplemented Instruction Exception if executed:

- All SIMD instructions
- All coprocessor instructions if no coprocessors are present
- retj, incjosp, popjc, pushjc
- tlbr, tlbs, tlbw
- cache

4.3.7 CPU and Architecture Revision

Three major revisions of the AVR32UC CPU currently exist.

The Architecture Revision field in the CONFIG0 system register identifies which architecture revision is implemented in a specific device.

AVR32UC CPU revision 3 is fully backward-compatible with revisions 1 and 2, ie. code compiled for revision 1 or 2 is binary-compatible with revision 3 CPUs.

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

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.

4.5.5 Entry Points for Events

Several different event handler entry points exists. In AVR32UC, the reset address is 0x8000_0000. 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.

AVR32UC uses the ITLB and DTLB protection exceptions to signal a MPU protection violation. ITLB and DTLB miss exceptions are used to signal that an access address did not map to any of the entries in the MPU. TLB multiple hit exception indicates that an access address did map to multiple TLB entries, signalling an error.

All external interrupt requests have entry points located at an offset relative to EVBA. This autovector offset is specified by an external 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, or optionally in a privileged memory protection region if an MPU is present.

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

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

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

Table 7-32. SMC Read Signals with no Hold Settings

Symbol	Parameter	Min.	Unit
NRD Controlled (READ_MODE = 1)			
SMC_{19}	Data Setup before NRD High	13.7	ns
SMC_{20}	Data Hold after NRD High	1	ns
NRD Controlled (READ_MODE = 0)			
SMC_{21}	Data Setup before NCS High	13.3	ns
SMC_{22}	Data Hold after NCS High	0	ns

Table 7-33. SMC Write Signals with Hold Settings

Symbol	Parameter	Min.	Unit
NRD Controlled (READ_MODE = 1)			
SMC_{23}	Data Out Valid before NWE High	(nwe pulse length - 1) * t_{CPSMC} - 0.9	ns
SMC_{24}	Data Out Valid after NWE High(1)	nwe hold length * t_{CPSMC} - 6	ns
SMC_{25}	NWE High to NBS0/A0 Change(1)	nwe hold length * t_{CPSMC} - 1.9	ns
SMC_{26}	NWE High to NBS1 Change(1)	nwe hold length * t_{CPSMC} - 1.9	ns
SMC_{29}	NWE High to A1 Change(1)	nwe hold length * t_{CPSMC} - 1.9	ns
SMC_{31}	NWE High to A2 - A23 Change(1)	nwe hold length * t_{CPSMC} - 1.7	ns
SMC_{32}	NWE High to NCS Inactive(1)	(nwe hold length - ncs wr hold length)* t_{CPSMC} - 2.9	ns
SMC_{33}	NWE Pulse Width	nwe pulse length * t_{CPSMC} - 0.9	ns
NRD Controlled (READ_MODE = 0)			
SMC_{34}	Data Out Valid before NCS High	(ncs wr pulse length - 1)* t_{CPSMC} - 4.6	ns
SMC_{35}	Data Out Valid after NCS High(1)	ncs wr hold length * t_{CPSMC} - 5.8	ns
SMC_{36}	NCS High to NWE Inactive(1)	(ncs wr hold length - nwe hold length)* t_{CPSMC} - 0.6	ns

Note: 1. hold length = total cycle duration - setup duration - pulse duration. “hold length” is for “ncs wr hold length” or “nwe hold length"

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

Symbol	Parameter	Min.	Unit
SMC_{37}	NWE Rising to A2-A25 Valid	5.4	ns
SMC_{38}	NWE Rising to NBS0/A0 Valid	5	ns
SMC_{39}	NWE Rising to NBS1 Change	5	ns
SMC_{40}	NWE Rising to A1/NBS2 Change	5	ns
SMC_{41}	NWE Rising to NBS3 Change	5	ns
SMC_{42}	NWE Rising to NCS Rising	5.1	ns

Figure 7-9. SDRAMC Signals relative to SDCK.

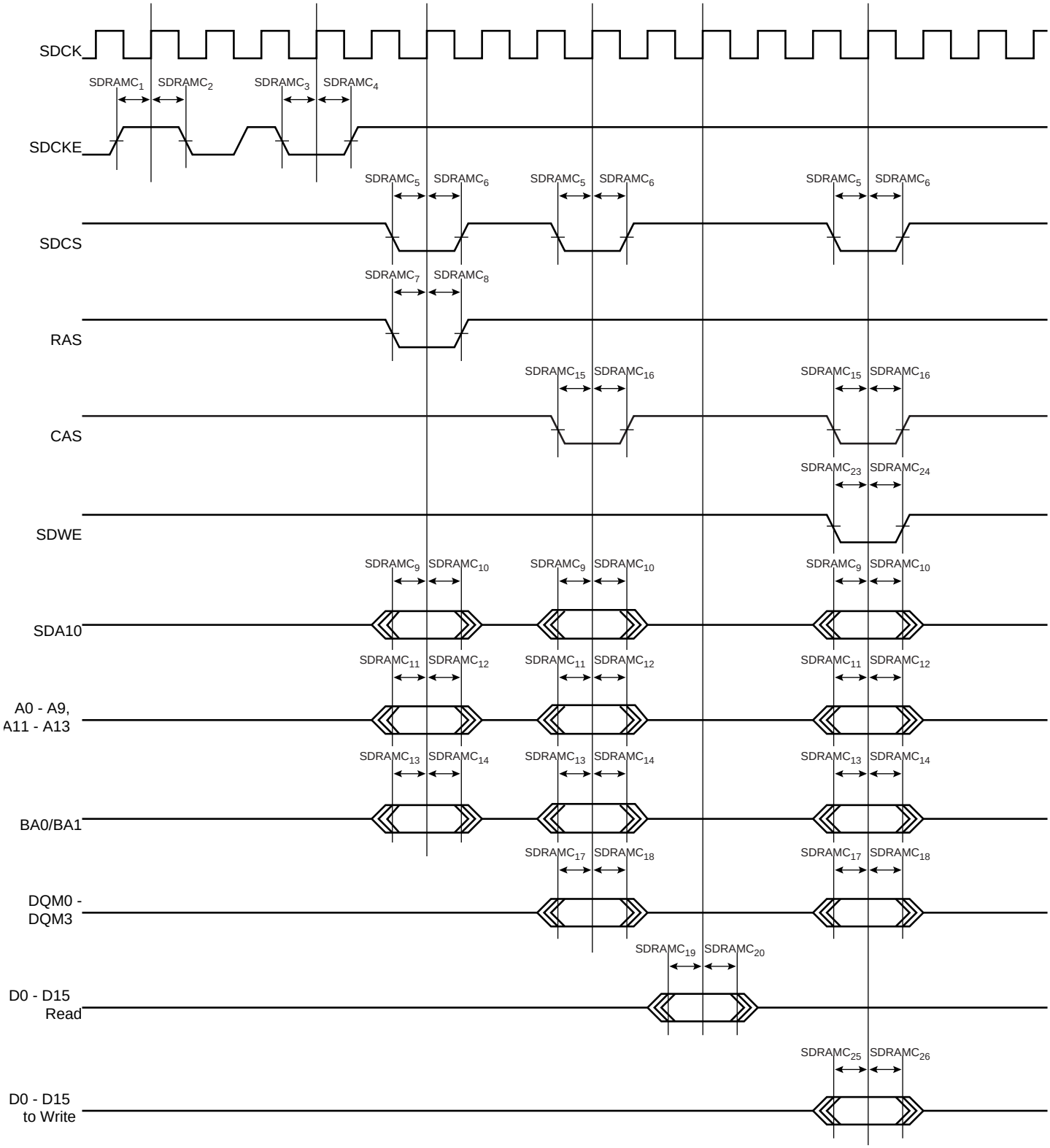
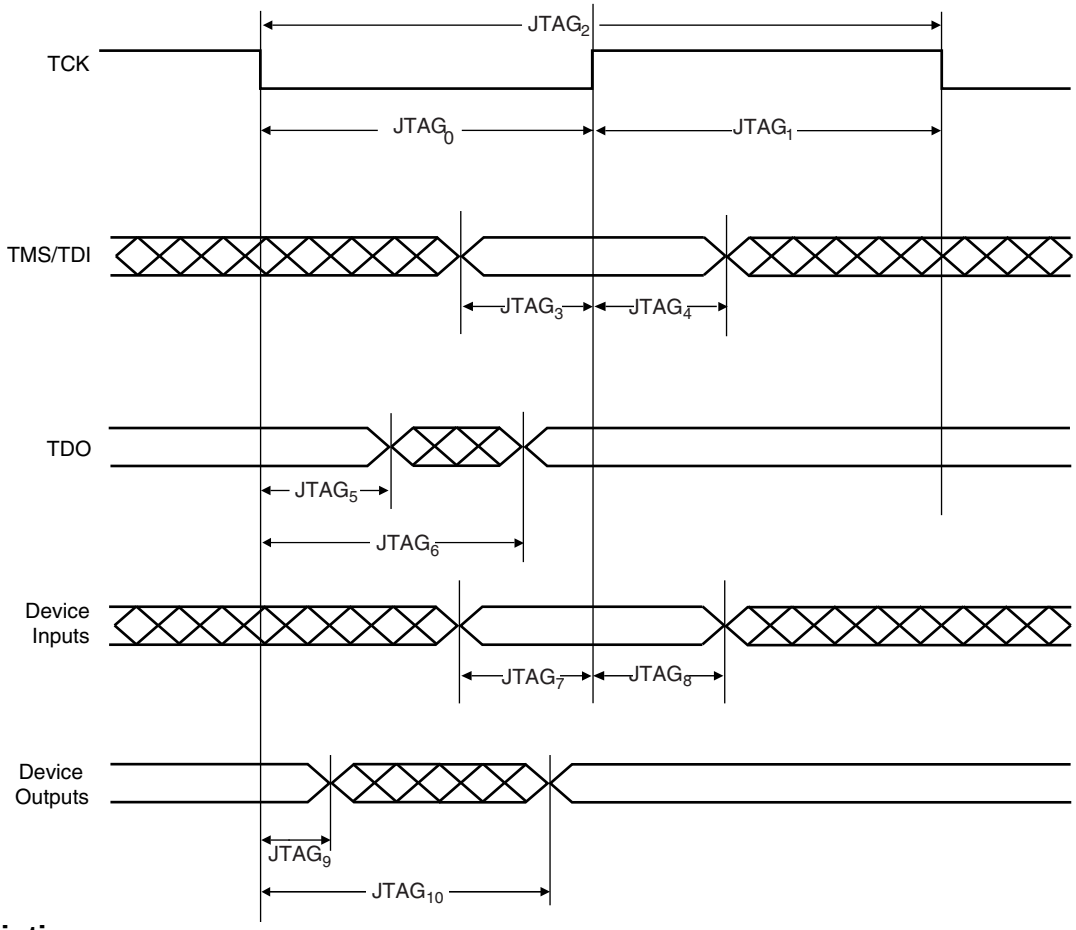
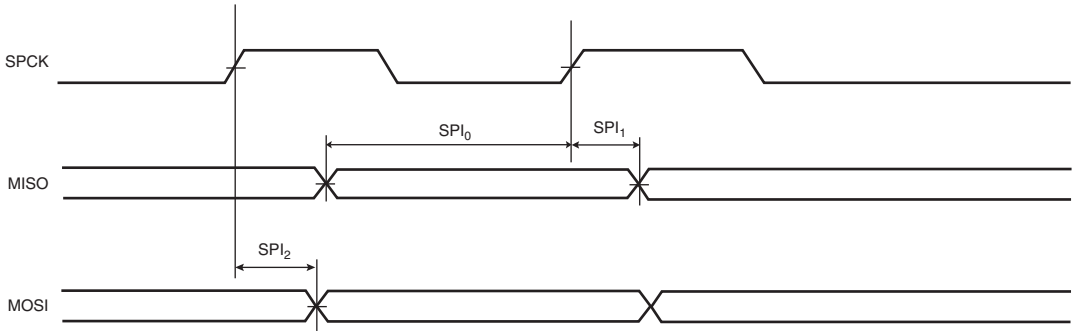


Figure 7-10. JTAG Interface Signals



7.13 SPI Characteristics

Figure 7-11. SPI Master mode with (CPOL= NCPHA= 0) or (CPOL= NCPHA= 1)



7.15 Flash Memory Characteristics

The following table gives the device maximum operating frequency depending on the field FWS of the Flash FSR register. This field defines the number of wait states required to access the Flash Memory. Flash operating frequency equals the CPU/HSB frequency.

Table 7-39. Flash Operating Frequency

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
F_{FOP}	Flash Operating Frequency	FWS = 0 High Speed Read Mode Disable -40°C < Ambient Temperature < 85°C			36	MHz
		FWS = 1 High Speed Read Mode Disable -40°C < Ambient Temperature < 85°C			66	MHz
		FWS = 0 High Speed Read Mode Enable -40°C < Ambient Temperature < 70°C			42	MHz
		FWS = 1 High Speed Read Mode Enable -40°C < Ambient Temperature < 70°C			84	MHz

Table 7-40. Parts Programming Time

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
T_{FPP}	Page Programming Time			5		ms
T_{FFP}	Fuse Programming Time			0.5		ms
T_{FCE}	Chip erase Time			8		ms

Table 7-41. Flash Parameters

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
N_{FARRAY}	Flash Array Write/Erase cycle				100K	cycle
N_{FFUSE}	General Purpose Fuses write cycle				1000	cycle
T_{FDR}	Flash Data Retention Time			15		year

8.3 Soldering Profile

Table 8-5 gives the recommended soldering profile from J-STD-20.

Table 8-5. Soldering Profile

Profile Feature	Green Package
Average Ramp-up Rate (217°C to Peak)	3°C/Second max
Preheat Temperature 175°C ±25°C	150-200°C
Time Maintained Above 217°C	60-150 seconds
Time within 5°C of Actual Peak Temperature	30 seconds
Peak Temperature Range	260 (+0/-5°C)
Ramp-down Rate	6°C/Second max.
Time 25°C to Peak Temperature	8 minutes max

Note: It is recommended to apply a soldering temperature higher than 250°C.

A maximum of three reflow passes is allowed per component.

Fix/Workaround
None.

10.3.14 MCI

The busy signal of the responses R1b is not taken in account for CMD12 STOP_TRANSFER
It is not possible to know the busy status of the card during the response (R1b) for the commands CMD12.
Fix/Workaround
The card busy line should be polled through the GPIO Input Value register (IVR) for commands CMD12.

10.3.15 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.

10.3.16 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.

11.7 Rev. B – 08/09

1. Updated the datasheet with new device AT32UC3A4.

11.8 Rev. A – 03/09

1. Initial revision.