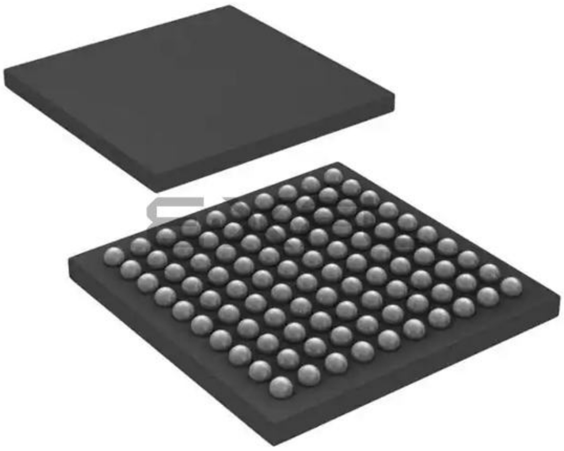
Microchip Technology - AT32UC3A464S-C1UR Datasheet



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/at32uc3a464s-c1ur

1. Description

The AT32UC3A3/A4 is a complete System-On-Chip microcontroller based on the AVR32 UC RISC processor running at frequencies up to 84MHz. AVR32 UC is a high-performance 32-bit RISC microprocessor core, designed for cost-sensitive embedded applications, with particular emphasis on low power consumption, high code density and high performance.

The processor implements a Memory Protection Unit (MPU) and a fast and flexible interrupt controller for supporting modern operating systems and real-time operating systems. Higher computation capabilities are achievable using a rich set of DSP instructions.

The AT32UC3A3/A4 incorporates on-chip Flash and SRAM memories for secure and fast access. 64 KBytes of SRAM are directly coupled to the AVR32 UC for performances optimization. Two blocks of 32 Kbytes SRAM are independently attached to the High Speed Bus Matrix, allowing real ping-pong management.

The Peripheral Direct Memory Access Controller (PDCA) enables data transfers between peripherals and memories without processor involvement. The PDCA drastically reduces processing overhead when transferring continuous and large data streams.

The Power Manager improves design flexibility and security: the on-chip Brown-Out Detector monitors the power supply, the CPU runs from the on-chip RC oscillator or from one of external oscillator sources, a Real-Time Clock and its associated timer keeps track of the time.

The device includes two sets of three identical 16-bit Timer/Counter (TC) channels. Each channel can be independently programmed to perform frequency measurement, event counting, interval measurement, pulse generation, delay timing and pulse width modulation. 16-bit channels are combined to operate as 32-bit channels.

The AT32UC3A3/A4 also features many communication interfaces for communication intensive applications like UART, SPI or TWI. The USART supports different communication modes, like SPI Mode and LIN Mode. Additionally, a flexible Synchronous Serial Controller (SSC) is available. The SSC provides easy access to serial communication protocols and audio standards like I2S.

The AT32UC3A3/A4 includes a powerfull External Bus Interface to interface all standard memory device like SRAM, SDRAM, NAND Flash or parallel interfaces like LCD Module.

The peripheral set includes a High Speed MCI for SDIO/SD/MMC and a hardware encryption module based on AES algorithm.

The device embeds a 10-bit ADC and a Digital Audio bistream DAC.

The Direct Memory Access controller (DMACA) allows high bandwidth data flows between high speed peripherals (USB, External Memories, MMC, SDIO, ...) and through high speed internal features (AES, internal memories).

The High-Speed (480MBit/s) USB 2.0 Device and Host interface supports several USB Classes at the same time thanks to the rich Endpoint configuration. The Embedded Host interface allows device like a USB Flash disk or a USB printer to be directly connected to the processor. This periphal has its own dedicated DMA and is perfect for Mass Storage application.

AT32UC3A3/A4 integrates a class 2+ Nexus 2.0 On-Chip Debug (OCD) System, with non-intrusive real-time trace, full-speed read/write memory access in addition to basic runtime control.

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
L6	84	H9(1)	PX18	69	VDDIO	x2	EBI - ADDR[16]	DMACA - DMAACK[1]	TC0 - A2	
D5	35	F1(1)	PX19	70	VDDIO	x2	EBI - ADDR[15]	EIC - SCAN[0]	TC0 - B2	
L4	73	H6(1)	PX20	71	VDDIO	x2	EBI - ADDR[14]	EIC - SCAN[1]	TC0 - CLK0	
M5	80	H2	PX21	72	VDDIO	x2	EBI - ADDR[13]	EIC - SCAN[2]	TC0 - CLK1	
M1	72	K10(1)	PX22	73	VDDIO	x2	EBI - ADDR[12]	EIC - SCAN[3]	TC0 - CLK2	
M6	85	K1	PX23	74	VDDIO	x2	EBI - ADDR[11]	EIC - SCAN[4]	SSC - TX_CLOCK	
M7	86	J2	PX24	75	VDDIO	x2	EBI - ADDR[10]	EIC - SCAN[5]	SSC - TX_DATA	
M8	92	H4	PX25	76	VDDIO	x2	EBI - ADDR[9]	EIC - SCAN[6]	SSC - RX_DATA	
L9	90	J3	PX26	77	VDDIO	x2	EBI - ADDR[8]	EIC - SCAN[7]	SSC - RX_FRAME_SYNC	
K9	89	K2	PX27	78	VDDIO	x2	EBI - ADDR[7]	SPI0 - MISO	SSC - TX_FRAME_SYNC	
L10	91	K3	PX28	79	VDDIO	x2	EBI - ADDR[6]	SPI0 - MOSI	SSC - RX_CLOCK	
K11	94	J4	PX29	80	VDDIO	x2	EBI - ADDR[5]	SPI0 - SPCK		
M11	96	G5	PX30	81	VDDIO	x2	EBI - ADDR[4]	SPI0 - NPCS[0]		
M10	97	H5	PX31	82	VDDIO	x2	EBI - ADDR[3]	SPI0 - NPCS[1]		
M9	93	K4(1)	PX32	83	VDDIO	x2	EBI - ADDR[2]	SPI0 - NPCS[2]		
M12	95		PX33	84	VDDIO	x2	EBI - ADDR[1]	SPI0 - NPCS[3]		
J3	61		PX34	85	VDDIO	x2	EBI - ADDR[0]	SPI1 - MISO	PM - GCLK[0]	
C2	38		PX35	86	VDDIO	x2	EBI - DATA[15]	SPI1 - MOSI	PM - GCLK[1]	
D3	44		PX36	87	VDDIO	x2	EBI - DATA[14]	SPI1 - SPCK	PM - GCLK[2]	
D2	45		PX37	88	VDDIO	x2	EBI - DATA[13]	SPI1 - NPCS[0]	PM - GCLK[3]	
E1	51		PX38	89	VDDIO	x2	EBI - DATA[12]	SPI1 - NPCS[1]	USART1 - DCD	
F1	52		PX39	90	VDDIO	x2	EBI - DATA[11]	SPI1 - NPCS[2]	USART1 - DSR	
A1	36		PX40	91	VDDIO	x2		MCI - CLK		
M2	71		PX41	92	VDDIO	x2	EBI - CAS			
M3	69		PX42	93	VDDIO	x2	EBI - RAS			
L7	88		PX43	94	VDDIO	x2	EBI - SDA10	USART1 - RI		
K2	66		PX44	95	VDDIO	x2	EBI - SDWE	USART1 - DTR		
L3	70	J7(1)	PX45	96	VDDIO	x3	EBI - SDCK			
K4	74	G6(1)	PX46	97	VDDIO	x2	EBI - SDCKE			
D4	39	E1(1)	PX47	98	VDDIO	x2	EBI - NANDOE	ADC - TRIGGER	MCI - DATA[11]	
F5	41		PX48	99	VDDIO	x2	EBI - ADDR[23]	USB - VBOF	MCI - DATA[10]	
F4	43		PX49	100	VDDIO	x2	EBI - CFRNW	USB - ID	MCI - DATA[9]	
G4	75		PX50	101	VDDIO	x2	EBI - CFCE2	TC1 - B2	MCI - DATA[8]	
G5	77		PX51	102	VDDIO	x2	EBI - CFCE1	DMACA - DMAACK[0]	MCI - DATA[15]	
K7	87		PX52	103	VDDIO	x2	EBI - NCS[3]	DMACA - DMARQ[0]	MCI - DATA[14]	
E4	42	D4(1)	PX53	104	VDDIO	x2	EBI - NCS[2]		MCI - DATA[13]	
E3	46		PX54	105	VDDIO	x2	EBI - NWAIT	USART3 - TXD	MCI - DATA[12]	
J5	79		PX55	106	VDDIO	x2	EBI - ADDR[22]	EIC - SCAN[3]	USART2 - RXD	

3.3 Signal Descriptions

The following table gives details on signal name classified by peripheral.

Table 3-6. Signal Description List

Signal Name	Function	Type	Active Level	Comments
Power				
VDDIO	I/O Power Supply	Power		3.0 to 3.6V
VDDANA	Analog Power Supply	Power		3.0 to 3.6V
VDDIN	Voltage Regulator Input Supply	Power		3.0 to 3.6V
VDDCORE	Voltage Regulator Output for Digital Supply	Power Output		1.65 to 1.95 V
GNDANA	Analog Ground	Ground		
GNDIO	I/O Ground	Ground		
GNDCORE	Digital Ground	Ground		
GNDPLL	PLL Ground	Ground		
Clocks, Oscillators, and PLL's				
XIN0, XIN1, XIN32	Crystal 0, 1, 32 Input	Analog		
XOUT0, XOUT1, XOUT32	Crystal 0, 1, 32 Output	Analog		
JTAG				
TCK	Test Clock	Input		
TDI	Test Data In	Input		
TDO	Test Data Out	Output		
TMS	Test Mode Select	Input		
Auxiliary Port - AUX				
MCKO	Trace Data Output Clock	Output		
MDO[5:0]	Trace Data Output	Output		
MSEO[1:0]	Trace Frame Control	Output		
EVTI_N	Event In	Input	Low	
EVTO_N	Event Out	Output	Low	
Power Manager - PM				
GCLK[3:0]	Generic Clock Pins	Output		

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		

3.4 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.

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

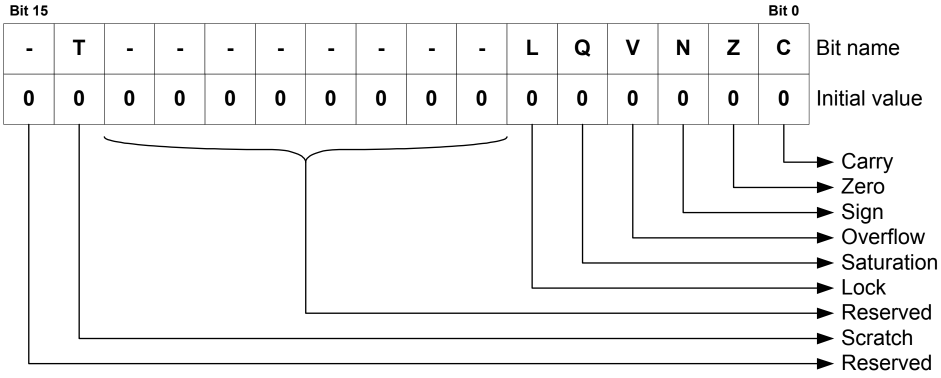
4.3.7 CPU and Architecture Revision

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.

Figure 4-5. The Status Register Low Halfword



4.4.3 Processor States

4.4.3.1 Normal RISC State

The AVR32 processor supports several different execution contexts as shown in Table 4-2 on page 27.

Table 4-2. Overview of Execution Modes, their Priorities and Privilege Levels.

Priority	Mode	Security	Description
1	Non Maskable Interrupt	Privileged	Non Maskable high priority interrupt mode
2	Exception	Privileged	Execute exceptions
3	Interrupt 3	Privileged	General purpose interrupt mode
4	Interrupt 2	Privileged	General purpose interrupt mode
5	Interrupt 1	Privileged	General purpose interrupt mode
6	Interrupt 0	Privileged	General purpose interrupt mode
N/A	Supervisor	Privileged	Runs supervisor calls
N/A	Application	Unprivileged	Normal program execution mode

Mode changes can be made under software control, or can be caused by external interrupts or exception processing. A mode can be interrupted by a higher priority mode, but never by one with lower priority. Nested exceptions can be supported with a minimal software overhead.

When running an operating system on the AVR32, user processes will typically execute in the application mode. The programs executed in this mode are restricted from executing certain instructions. Furthermore, most system registers together with the upper halfword of the status register cannot be accessed. Protected memory areas are also not available. All other operating modes are privileged and are collectively called System Modes. They have full access to all privileged and unprivileged resources. After a reset, the processor will be in supervisor mode.

4.4.3.2 Debug State

The AVR32 can be set in a debug state, which allows implementation of software monitor routines that can read out and alter system information for use during application development. This implies that all system and application registers, including the status registers and program counters, are accessible in debug state. The privileged instructions are also available.

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 5-1. AT32UC3A3A4 Physical Memory Map

Device	Start Address	Size	Size	Size
		AT32UC3A3256S AT32UC3A3256 AT32UC3A4256S AT32UC3A4256	AT32UC3A3128S AT32UC3A3128 AT32UC3A4128S AT32UC3A4128	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

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. Electrical Characteristics

7.1 Absolute Maximum Ratings*

Operating Temperature	-40°C to +85°C
Storage Temperature	-60°C to +150°C
Voltage on Input Pin with respect to Ground	-0.3V to 3.6V
Maximum Operating Voltage (VDDCORE)	1.95V
Maximum Operating Voltage (VDDIO)	3.6V
Total DC Output Current on all I/O Pin for TQFP144 package	370 mA
for TFBGA144 package	370 mA

*NOTICE: Stresses beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

7.5 Analog characteristics

7.5.1 ADC

Table 7-5. Electrical Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
V_{VDDANA}	Analog Power Supply		3.0		3.6	V

Table 7-6. Decoupling Requirements

Symbol	Parameter	Conditions	Typ.	Technology	Unit
C_{VDDANA}	Power Supply Capacitor		100	NPO	nF

7.5.2 BOD

Table 7-7. 1.8V BOD Level Values

Symbol	Parameter Value	Conditions	Min.	Typ.	Max.	Unit
BODLEVEL	00 1111b			1.79		V
	01 0111b			1.70		V
	01 1111b			1.61		V
	10 0111b			1.52		V

Table 7-7 describes the values of the BODLEVEL field in the flash FGPFR register.

Table 7-8. 3.3V BOD Level Values

Symbol	Parameter Value	Conditions	Min.	Typ.	Max.	Unit
BOD33LEVEL	Reset value			2.71		V
	1011			2.27		V
	1010			2.37		V
	1001			2.46		V
	1000			2.56		V
	0111			2.66		V
	0110			2.76		V
	0101			2.86		V
	0100			2.96		V
	0011			3.06		V
	0010			3.15		V
	0001			3.25		V
	0000			3.35		V

Table 7-8 describes the values of the BOD33.LEVEL field in the PM module

7.6.1 Power Consumtion for Different Sleep Modes

Table 7-12. Power Consumption for Different Sleep Modes

<table>
<tr><th>Mode</th><th colspan="2">Conditions[1]</th><th>Typ.</th><th>Unit</th></tr>
<tr><td>Active</td><td colspan="2">- CPU running a recursive Fibonacci Algorithm from flash and clocked from PLL0 at f MHz.
- Flash High Speed mode disable (f < 66 MHz)
- Voltage regulator is on.
- XIN0: external clock. Xin1 Stopped. XIN32 stopped.
- All peripheral clocks activated with a division by 8.
- GPIOs are inactive with internal pull-up, JTAG unconnected with external pullup and Input pins are connected to GND</td><td>0.626xf(MHz)+2.257</td><td>mA/MHz</td></tr>
<tr><td></td><td colspan="2">Same conditions with Flash High Speed mode enable (66< f < 84 MHz)</td><td>0.670xf(MHz)+2.257</td><td>mA/MHz</td></tr>
<tr><td></td><td colspan="2">Same conditions with Flash High Speed mode disable at 60 MHz</td><td>40</td><td>mA</td></tr>
<tr><td>Idle</td><td colspan="2">See Active mode conditions</td><td>0.349xf(MHz)+0.968</td><td>mA/MHz</td></tr>
<tr><td></td><td colspan="2">Same conditions at 60 MHz</td><td>21.8</td><td>mA</td></tr>
<tr><td>Frozen</td><td colspan="2">See Active mode conditions</td><td>0.098xf(MHz)+1.012</td><td>mA/MHz</td></tr>
<tr><td></td><td colspan="2">Same conditions at 60 MHz</td><td>6.6</td><td>mA</td></tr>
<tr><td>Standby</td><td colspan="2">See Active mode conditions</td><td>0.066xf(MHz)+1.010</td><td>mA/MHz</td></tr>
<tr><td></td><td colspan="2">Same conditions at 60 MHz</td><td>4.6</td><td>mA</td></tr>
<tr><td>Stop</td><td colspan="2">- CPU running in sleep mode
- XIN0, Xin1 and XIN32 are stopped.
- All peripheral clocks are desactived.
- GPIOs are inactive with internal pull-up, JTAG unconnected with external pullup and Input pins are connected to GND.</td><td>96</td><td>µA</td></tr>
<tr><td>Deepstop</td><td colspan="2">See Stop mode conditions</td><td>54</td><td>µA</td></tr>
<tr><td>Static</td><td>T_A = 25 °C
CPU is in static mode
GPIOs on internal pull-up
All peripheral clocks de-activated
DM and DP pins connected to ground
XIN0, Xin1 and XIN32 are stopped</td><td>on Amp0</td><td>31</td><td>µA</td></tr>
</table>

Notes: 1. Core frequency is generated from XIN0 using the PLL.

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"

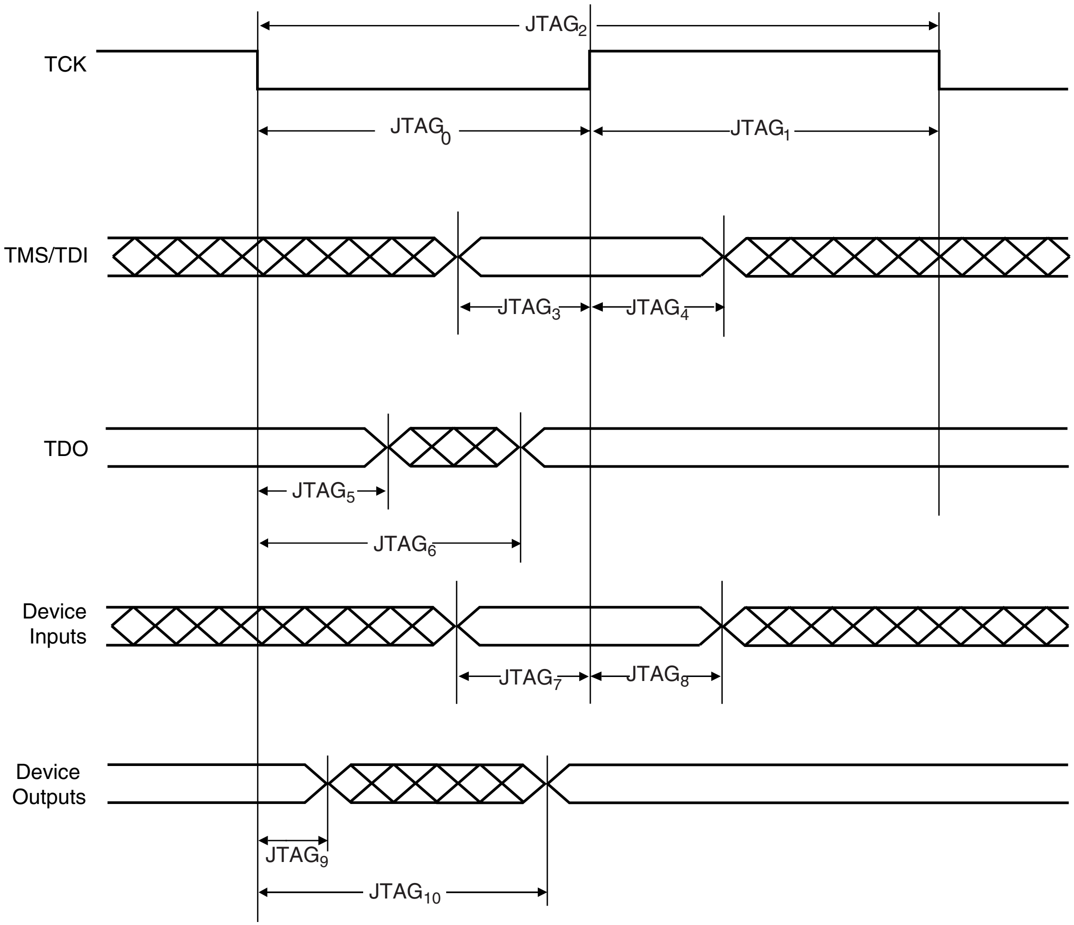
Table 7-34. SMC Write Signals with No Hold Settings (NWE Controlled only)

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

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 7-10. JTAG Interface Signals



7.13 SPI Characteristics

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

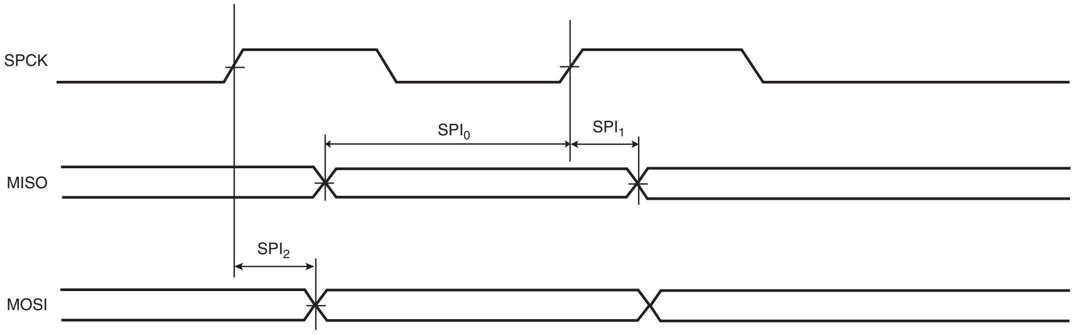


Table 7-38. SPI Timings

Symbol	Parameter	Conditions [1]	Min.	Max.	Unit
SPI_0	MISO Setup time before SPCK rises (master)	3.3V domain	22 + $(t_{CPMCK})/2$ [2]		ns
SPI_1	MISO Hold time after SPCK rises (master)	3.3V domain	0		ns
SPI_2	SPCK rising to MOSI Delay (master)	3.3V domain		7	ns
SPI_3	MISO Setup time before SPCK falls (master)	3.3V domain	22 + $(t_{CPMCK})/2$ [3]		ns
SPI_4	MISO Hold time after SPCK falls (master)	3.3V domain	0		ns
SPI_5	SPCK falling to MOSI Delay master)	3.3V domain		7	ns
SPI_6	SPCK falling to MISO Delay (slave)	3.3V domain		26.5	ns
SPI_7	MOSI Setup time before SPCK rises (slave)	3.3V domain	0		ns
SPI_8	MOSI Hold time after SPCK rises (slave)	3.3V domain	1.5		ns
SPI_9	SPCK rising to MISO Delay (slave)	3.3V domain		27	ns
SPI_{10}	MOSI Setup time before SPCK falls (slave)	3.3V domain	0		ns
SPI_{11}	MOSI Hold time after SPCK falls (slave)	3.3V domain	1		ns

1. 3.3V domain: V_{VDDIO} from 3.0V to 3.6V, maximum external capacitor = 40 pF
2. t_{CPMCK}: Master Clock period in ns.
3. t_{CPMCK}: Master Clock period in ns.

7.14 MCI

The High Speed MultiMedia Card Interface (MCI) supports the MultiMedia Card (MMC) Specification V4.2, the SD Memory Card Specification V2.0, the SDIO V1.1 specification and CE-ATA V1.1.

8. Mechanical Characteristics

8.1 Thermal Considerations

8.1.1 Thermal Data

Table 8-1 summarizes the thermal resistance data depending on the package.

Table 8-1. Thermal Resistance Data

Symbol	Parameter	Condition	Package	Typ	Unit
θ_{JA}	Junction-to-ambient thermal resistance	Still Air	TQFP144	40.3	°C/W
θ_{JC}	Junction-to-case thermal resistance		TQFP144	9.5	
θ_{JA}	Junction-to-ambient thermal resistance	Still Air	TFBGA144	28.5	°C/W
θ_{JC}	Junction-to-case thermal resistance		TFBGA144	6.9	
θ_{JA}	Junction-to-ambient thermal resistance	Still Air	VFBGA100	31.1	°C/W
θ_{JC}	Junction-to-case thermal resistance		VFBGA100	6.9	

8.1.2 Junction Temperature

The average chip-junction temperature, T_J, in °C can be obtained from the following:

1. $T_J = T_A + (P_D \times \theta_{JA})$
2. $T_J = T_A + (P_D \times (\theta_{HEATSINK} + \theta_{JC}))$

where:

- θ_{JA} = package thermal resistance, Junction-to-ambient (°C/W), provided in Table 8-1 on page 68.
- θ_{JC} = package thermal resistance, Junction-to-case thermal resistance (°C/W), provided in Table 8-1 on page 68.
- $\theta_{HEAT\ SINK}$ = cooling device thermal resistance (°C/W), provided in the device datasheet.
- P_D = device power consumption (W) estimated from data provided in the section ”Regulator characteristics” on page 43.
- T_A = ambient temperature (°C).

From the first equation, the user can derive the estimated lifetime of the chip and decide if a cooling device is necessary or not. If a cooling device is to be fitted on the chip, the second equation should be used to compute the resulting average chip-junction temperature T_J in °C.

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).

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.

10.2.9 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.

10.2.10 AES

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

10.2.11 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.

10.2.12 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.