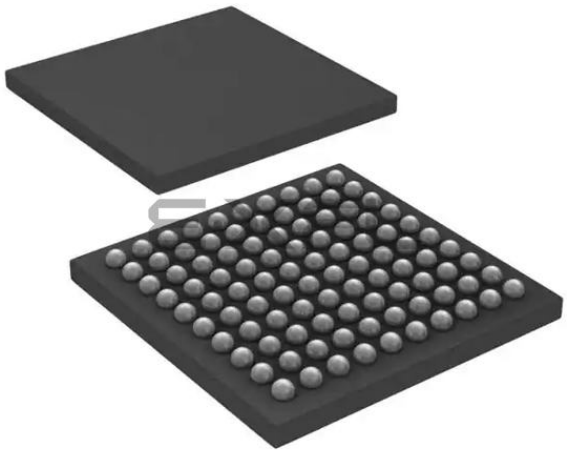


Welcome to E-XFL.COM

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	256KB (256K 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/at32uc3a4256-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.

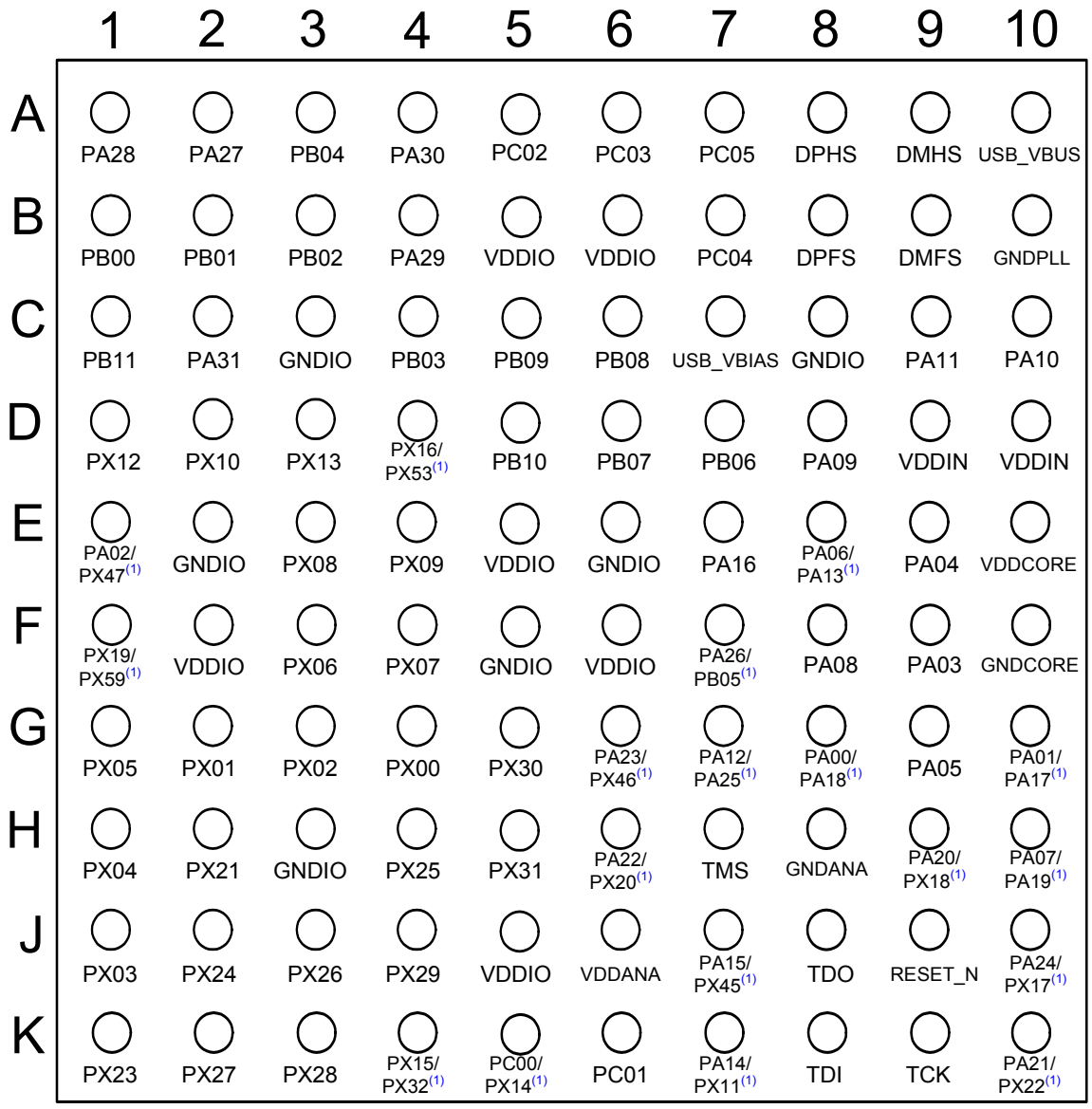
The device embeds a 10-bit ADC and a Digital Audio bistream DAC.

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.

Figure 3-3. VFBGA100 Pinout (top view)



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

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

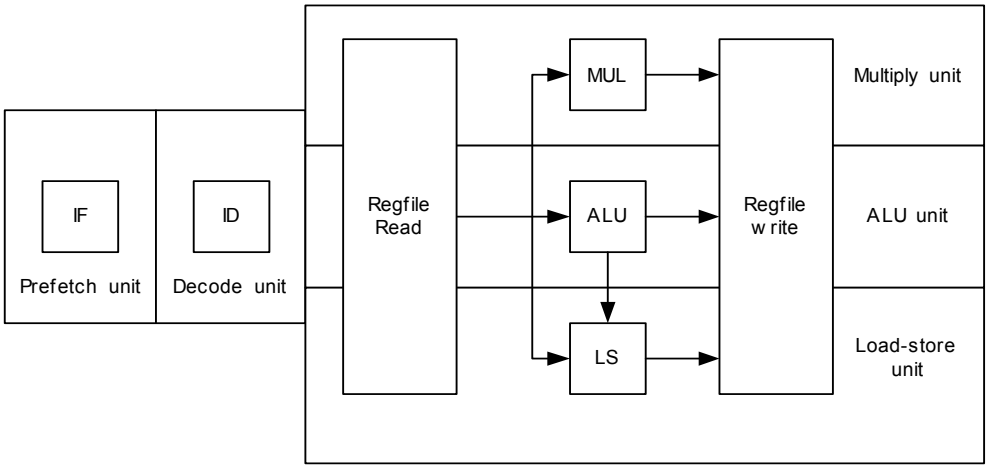
The AVR32UC CPU targets low- and medium-performance applications, and provides an advanced OCD system, no caches, and a Memory Protection Unit (MPU). Java acceleration hardware is not implemented.

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.

Figure 4-2. The AVR32UC Pipeline



4.3.2 AVR32A Microarchitecture Compliance

AVR32UC implements an AVR32A microarchitecture. The AVR32A microarchitecture is targeted at cost-sensitive, lower-end applications like smaller microcontrollers. This microarchitecture does not provide dedicated hardware registers for shadowing of register file registers in interrupt contexts. Additionally, it does not provide hardware registers for the return address registers and return status registers. Instead, all this information is stored on the system stack. This saves chip area at the expense of slower interrupt handling.

Upon interrupt initiation, registers R8-R12 are automatically pushed to the system stack. These registers are pushed regardless of the priority level of the pending interrupt. The return address and status register are also automatically pushed to stack. The interrupt handler can therefore use R8-R12 freely. Upon interrupt completion, the old R8-R12 registers and status register are restored, and execution continues at the return address stored popped from stack.

The stack is also used to store the status register and return address for exceptions and *scall*. Executing the *rete* or *rets* instruction at the completion of an exception or system call will pop this status register and continue execution at the popped return address.

4.3.3 Java Support

AVR32UC does not provide Java hardware acceleration.

4.3.4 Memory Protection

The MPU allows the user to check all memory accesses for privilege violations. If an access is attempted to an illegal memory address, the access is aborted and an exception is taken. The MPU in AVR32UC is specified in the AVR32UC Technical Reference manual.

4.3.5 Unaligned Reference Handling

AVR32UC does not support unaligned accesses, except for doubleword accesses. AVR32UC is able to perform word-aligned *st.d* and *ld.d*. Any other unaligned memory access will cause an address exception. Doubleword-sized accesses with word-aligned pointers will automatically be performed as two word-sized accesses.

4.4 Programming Model

4.4.1 Register File Configuration

The AVR32UC register file is shown below.

Figure 4-3. The AVR32UC Register File

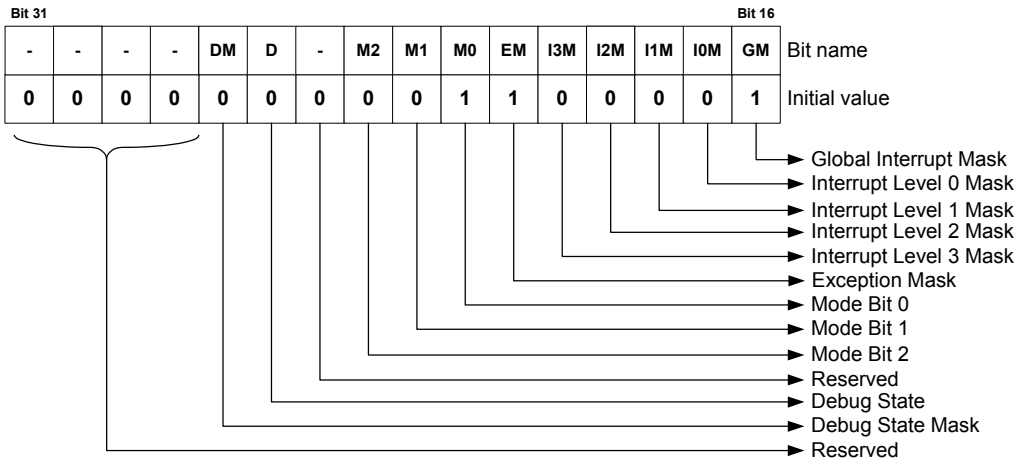
Application	Supervisor	INT0	INT1	INT2	INT3	Exception	NMI	Secure
Bit 31 Bit 0	Bit 31 Bit 0	Bit 31 Bit 0	Bit 31 Bit 0	Bit 31 Bit 0	Bit 31 Bit 0	Bit 31 Bit 0	Bit 31 Bit 0	Bit 31 Bit 0
PC	PC	PC	PC	PC	PC	PC	PC	PC
LR	LR	LR	LR	LR	LR	LR	LR	LR
SP_APP	SP_SYS	SP_SYS	SP_SYS	SP_SYS	SP_SYS	SP_SYS	SP_SYS	SP_SEC
R12	R12	R12	R12	R12	R12	R12	R12	R12
R11	R11	R11	R11	R11	R11	R11	R11	R11
R10	R10	R10	R10	R10	R10	R10	R10	R10
R9	R9	R9	R9	R9	R9	R9	R9	R9
R8	R8	R8	R8	R8	R8	R8	R8	R8
R7	R7	R7	R7	R7	R7	R7	R7	R7
R6	R6	R6	R6	R6	R6	R6	R6	R6
R5	R5	R5	R5	R5	R5	R5	R5	R5
R4	R4	R4	R4	R4	R4	R4	R4	R4
R3	R3	R3	R3	R3	R3	R3	R3	R3
R2	R2	R2	R2	R2	R2	R2	R2	R2
R1	R1	R1	R1	R1	R1	R1	R1	R1
R0	R0	R0	R0	R0	R0	R0	R0	R0
SR	SR	SR	SR	SR	SR	SR	SR	SR

SS_STATUS
SS_ADRF
SS_ADRR
SS_ADR0
SS_ADR1
SS_SP_SYS
SS_SP_APP
SS_RAR
SS_RSR

4.4.2 Status Register Configuration

The Status Register (SR) is split into two halfwords, one upper and one lower, see Figure 4-4 on page 26 and Figure 4-5 on page 27. The lower word contains the C, Z, N, V, and Q condition code flags and the R, T, and L bits, while the upper halfword contains information about the mode and state the processor executes in. Refer to the *AVR32 Architecture Manual* for details.

Figure 4-4. The Status Register High Halfword



5. Memories

5.1 Embedded Memories

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

5.2 Physical Memory Map

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

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

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

Table 5-1. AT32UC3A3A4 Physical Memory Map

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

Table 5-3. Local Bus Mapped GPIO Registers

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

Table 7-2. Normal I/O Pin Characteristics

Symbol	Parameter	Conditions	drive x2	drive x2	drive x3	Unit
f_{MAX}	Output frequency	10pf	40	66	100	MHz
		30pf	18.2	35.7	61.6	MHz
		60pf	7.5	18.5	36.3	MHz
t_{RISE}	Rise time	10pf	2.7	1.4	0.9	ns
		30pf	6.9	3.5	1.9	ns
		60pf	13.4	6.7	3.5	ns
t_{FALL}	Fall time	10pf	3.2	1.7	0.9	ns
		30pf	8.6	4.3	2.26	ns
		60pf	16.5	8.3	4.3	ns

7.4 Regulator characteristics

Table 7-3. Electrical Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
V_{VDDIN}	Supply voltage (input)		3.0	3.3	3.6	V
$V_{VDDCORE}$	Supply voltage (output)		1.75	1.85	1.95	V
I_{OUT}	Maximum DC output current	V_{VDDIN} = 3.3V			100	mA

Table 7-4. Decoupling Requirements

Symbol	Parameter	Conditions	Typ.	Technology	Unit
C_{IN1}	Input Regulator Capacitor 1		1	NPO	nF
C_{IN2}	Input Regulator Capacitor 2		4.7	X7R	µF
C_{OUT1}	Output Regulator Capacitor 1		470	NPO	pF
C_{OUT2}	Output Regulator Capacitor 2		2.2	X7R	µF

Figure 7-3. MCU Cold Start-Up

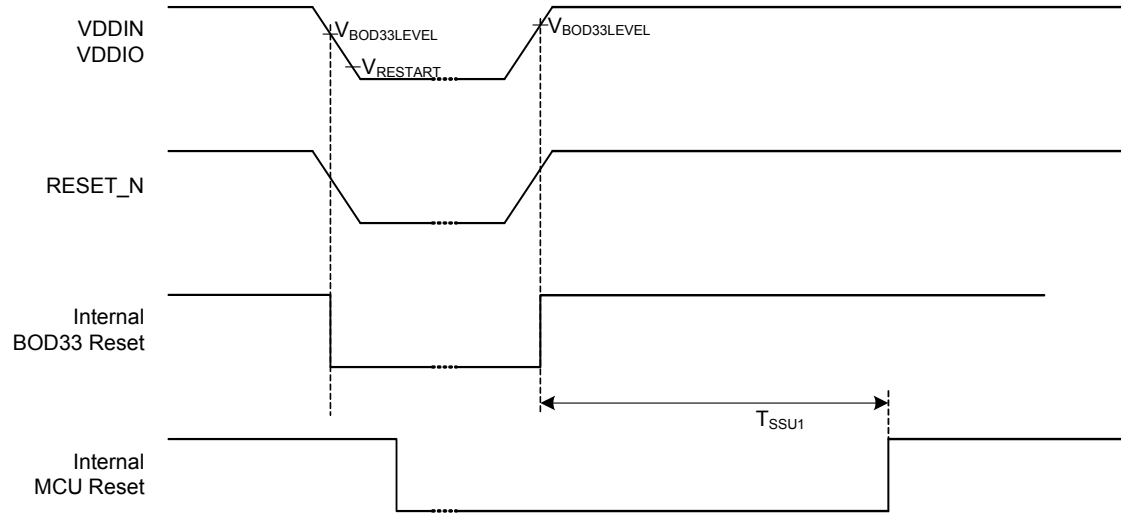


Figure 7-4. MCU Cold Start-Up RESET_N Externally Driven

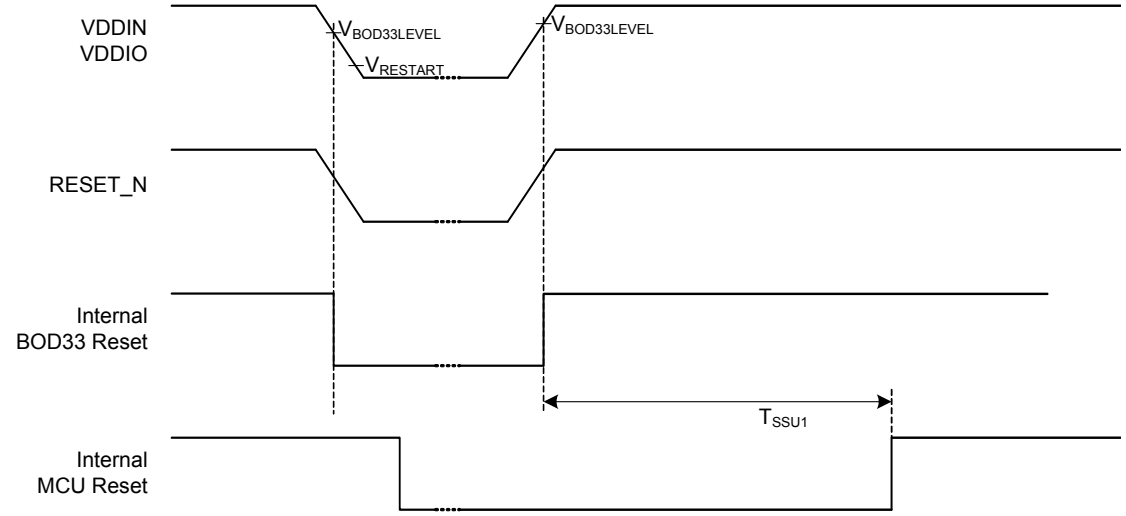
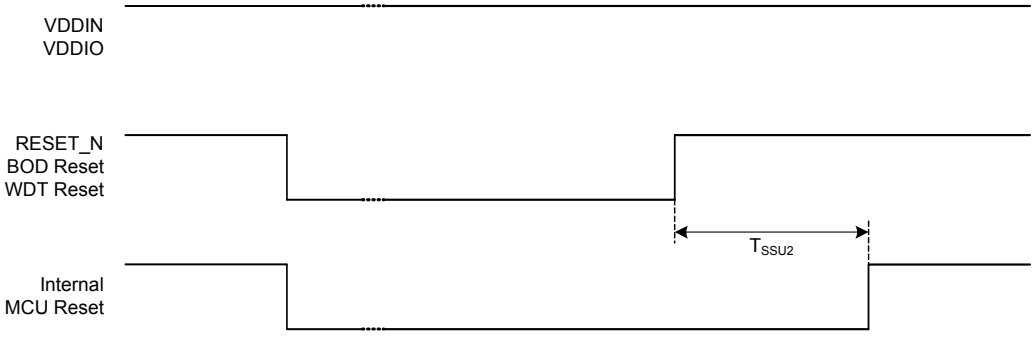


Figure 7-5. MCU Hot Start-Up

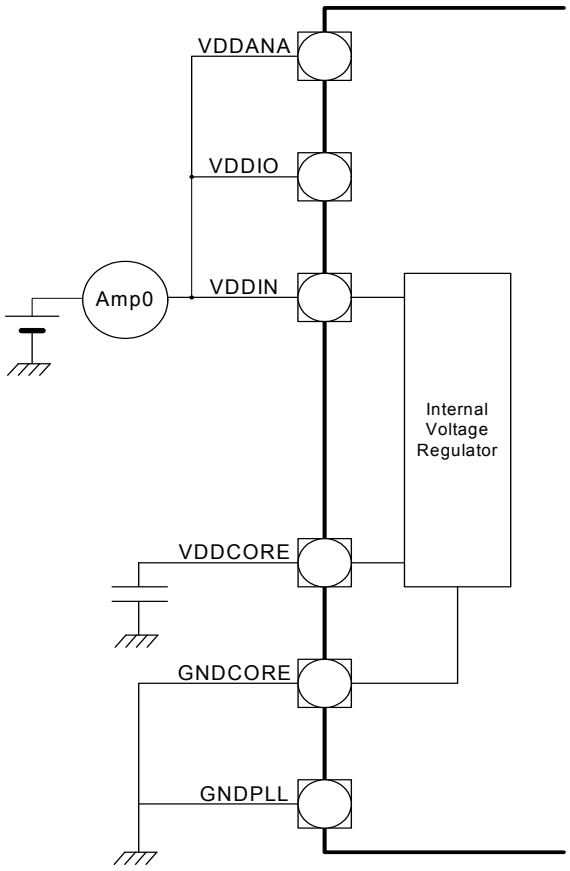


7.6 Power Consumption

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 7-36. 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

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. 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.

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.

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

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

10.1.5 ADC

Sleep Mode activation needs additional A to D conversion
If the ADC sleep mode is activated when the ADC is idle the ADC will not enter sleep mode before after the next AD conversion.
Fix/Workaround
Activate the sleep mode in the mode register and then perform an AD conversion.

10.1.6 USART

ISO7816 info register US_NER cannot be read
The NER register always returns zero.
Fix/Workaround
None.

The LIN ID is not transmitted in mode PDCM='0'
Fix/Workaround
Using USART in mode LIN master with the PDCM bit = '0', the LINID written at the first address of the transmit buffer is not used. The LINID must be written in the LINIR register, after the configuration and start of the PDCA transfer. Writing the LINID in the LINIR register will start the transfer whenever the PDCA transfer is ready.

The LINID interrupt is only available for the header reception and not available for the header transmission
Fix/Workaround
None.

USART LIN mode is not functional with the PDCA if PDCM bit in LINMR register is set to 1
If a PDCA transfer is initiated in USART LIN mode with PDCM bit set to 1, the transfer never starts.
Fix/Workaround
Only use PDCM=0 configuration with the PDCA transfer.

10.1.7 SPI

SPI disable does not work in SLAVE mode
SPI disable does not work in SLAVE mode.
Fix/Workaround
Read the last received data, then perform a software reset by writing a one to the Software Reset bit in the Control Register (CR.SWRST).

SPI bad serial clock generation on 2nd chip_select when SCBR=1, CPOL=1, and NCPHA=0
When multiple chip selects (CS) are in use, if one of the baudrates equal 1 while one (CSRn.SCBR=1) of the others do not equal 1, and CSRn.CPOL=1 and CSRn.NCPHA=0, then an additional pulse will be generated on SCK.
Fix/Workaround
When multiple CS are in use, if one of the baudrates equals 1, the others must also equal 1 if CSRn.CPOL=1 and CSRn.NCPHA=0.

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.

Fix/Workaround
SOFT_RESET on RX+ Config US_MR + Config_US_CR.

10.2.7 SPI

SPI disable does not work in SLAVE mode
SPI disable does not work in SLAVE mode.
Fix/Workaround
Read the last received data, then perform a software reset by writing a one to the Software Reset bit in the Control Register (CR.SWRST).

SPI bad serial clock generation on 2nd chip_select when SCBR=1, CPOL=1, and NCPHA=0
When multiple chip selects (CS) are in use, if one of the baudrates equal 1 while one (CSRn.SCBR=1) of the others do not equal 1, and CSRn.CPOL=1 and CSRn.NCPHA=0, then an additional pulse will be generated on SCK.
Fix/Workaround
When multiple CS are in use, if one of the baudrates equals 1, the others must also equal 1 if CSRn.CPOL=1 and CSRn.NCPHA=0.

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.

10.2.8 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
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.

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: (+1)(408) 441-0311
Fax: (+1)(408) 487-2600
www.atmel.com

Atmel Asia Limited
Unit 1-5 & 16, 19/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
HONG KONG
Tel: (+852) 2245-6100
Fax: (+852) 2722-1369

Atmel Munich GmbH
Business Campus
Parkring 4
D-85748 Garching b. Munich
GERMANY
Tel: (+49) 89-31970-0
Fax: (+49) 89-3194621

Atmel Japan
16F, Shin Osaki Kangyo Bldg.
1-6-4 Osaka Shinagawa-ku
Tokyo 104-0032
JAPAN
Tel: (+81) 3-6417-0300
Fax: (+81) 3-6417-0370