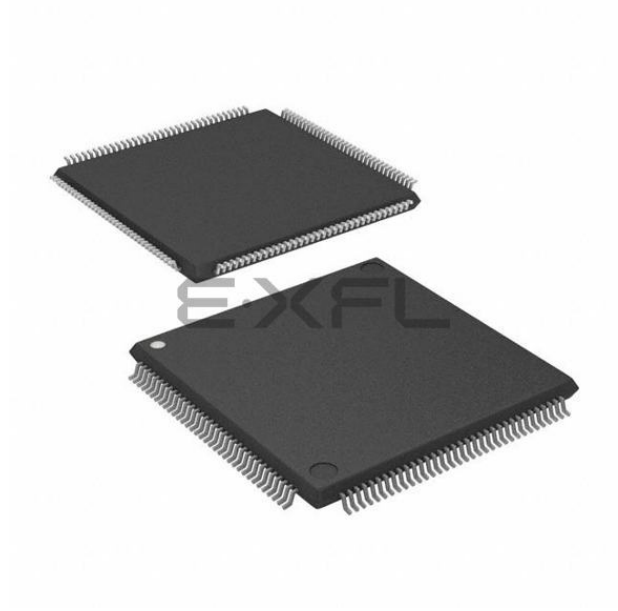


Welcome to E-XFL.COM

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, Ethernet, I²C, SPI, SSC, UART/USART, USB OTG
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	109
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	64K x 8
Voltage - Supply (Vcc/Vdd)	1.65V ~ 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/at32uc3a0512-alur

Table 6-3. BGA144 Package Pinout A1..M8

	1	2	3	4	5	6	7	8
A	VDDIO	PB07	PB05	PB02	PB03	PB01	PC00	PA28
B	PB08	GND	PB06	PB04	VDDIO	PB00	PC01	VDDPLL
C	PB09	PX33	PA29	PC02	PX28	PX26	PX22	PX21
D	PB11	PB13	PB12	PX30	PX29	PX25	PX24	PX20
E	PB10	VDDIO	PX32	PX31	VDDIO	PX27	PX23	VDDANA
F	PA30	PB14	PX34	PB16	TCK	GND	GND	PX16
G	TMS	PC03	PX36	PX35	PX37	GND	GND	PA16
H	TDO	VDDCORE	PX38	PX39	VDDIO	PA01	PA10	VDDCORE
J	TDI	PB17	PB15	PX00	PX01	PA00	PA03	PA04
K	PC05	PC04	PB19	PB20	PX02	PB29	PB30	PA02
L	PB21	GND	PB18	PB24	VDDOUT	PX04	PB31	VDDIN
M	PB22	PB23	PB25	PB26	PX03	PB27	PB28	RESET_N

Table 6-4. BGA144 Package Pinout A9..M12

	9	10	11	12
A	PA26	PA25	PA24	PA23
B	PA27	PA21	GND	PA22
C	ADVREF	GNDANA	PX19	PA19
D	PA18	PA20	DP	DM
E	PX18	PX17	VDDIO	VBUS
F	PA17	PX15	PA15	PA14
G	PA13	PA12	PA11	NC
H	PX11	PA08	VDDCORE	VDDCORE
J	PX14	PA07	PX13	PA09
K	PX08	GND	PA05	PX12
L	PX06	PX10	GND	PA06
M	PX05	PX07	PX09	VDDIO

Note: NC is not connected.

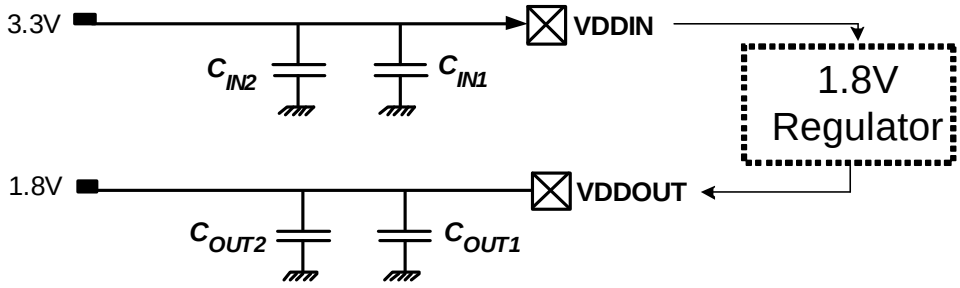
7.2 Voltage Regulator

7.2.1 Single Power Supply

The AT32UC3A embeds a voltage regulator that converts from 3.3V to 1.8V. The regulator takes its input voltage from VDDIN, and supplies the output voltage on VDDOUT. VDDOUT should be externally connected to the 1.8V domains.

Adequate input supply decoupling is mandatory for VDDIN in order to improve startup stability and reduce source voltage drop. Two input decoupling capacitors must be placed close to the chip.

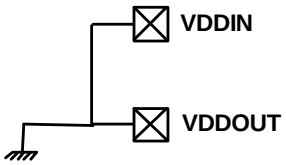
Adequate output supply decoupling is mandatory for VDDOUT to reduce ripple and avoid oscillations. The best way to achieve this is to use two capacitors in parallel between VDDOUT and GND as close to the chip as possible



Refer to Section 12.3 on page 42 for decoupling capacitors values and regulator characteristics

7.2.2 Dual Power Supply

In case of dual power supply, VDDIN and VDDOUT should be connected to ground to prevent from leakage current.



8. I/O Line Considerations

8.1 JTAG pins

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

8.2 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.

8.3 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 PIO pins.

8.4 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 GPIO Controllers. After reset, I/O lines default as inputs with pull-up resistors disabled, except when indicated otherwise in the column “Reset State” of the GPIO Controller multiplexing tables.

9. Memories

9.1 Embedded Memories

- **Internal High-Speed Flash**
  - **512 KBytes (AT32UC3A0512, AT32UC3A1512)**
  - **256 KBytes (AT32UC3A0256, AT32UC3A1256)**
  - **128 KBytes (AT32UC3A1128, AT32UC3A2128)**
    - **0 Wait State Access at up to 33 MHz in Worst Case Conditions**
    - **1 Wait State Access at up to 66 MHz 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**
    - **4 ms Page Programming Time, 8 ms Chip Erase Time**
    - **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, Single-cycle access at full speed**
  - **64 KBytes (AT32UC3A0512, AT32UC3A0256, AT32UC3A1512, AT32UC3A1256)**
  - **32KBytes (AT32UC3A1128)**

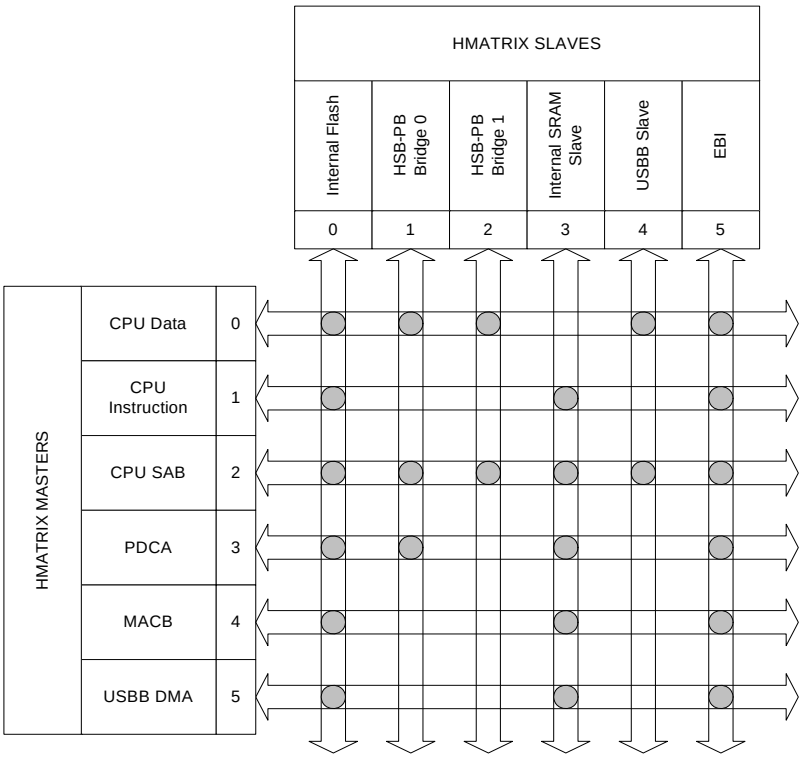
9.2 Physical Memory Map

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 AVR32 Architecture Manual. The 32-bit physical address space is mapped as follows:

Table 9-1. AT32UC3A Physical Memory Map

Device	Start Address	Size					
		AT32UC3A0512	AT32UC3A1512	AT32UC3A0256	AT32UC3A1256	AT32UC3A0128	AT32UC3A1128
Embedded SRAM	0x0000_0000	64 Kbyte	64 Kbyte	64 Kbyte	64 Kbyte	32 Kbyte	32 Kbyte
Embedded Flash	0x8000_0000	512 Kbyte	512 Kbyte	256 Kbyte	256 Kbyte	128 Kbyte	128 Kbyte
EBI SRAM CS0	0xC000_0000	16 Mbyte	-	16 Mbyte	-	16 Mbyte	-
EBI SRAM CS2	0xC800_0000	16 Mbyte	-	16 Mbyte	-	16 Mbyte	-
EBI SRAM CS3	0xCC00_0000	16 Mbyte	-	16 Mbyte	-	16 Mbyte	-
EBI SRAM CS1 /SDRAM CS0	0xD000_0000	128 Mbyte	-	128 Mbyte	-	128 Mbyte	-
USB Configuration	0xE000_0000	64 Kbyte	64 Kbyte	64 Kbyte	64 Kbyte	64 Kbyte	64 Kbyte
HSB-PB Bridge A	0xFFFE_0000	64 Kbyte	64 Kbyte	64 Kbyte	64 Kbyte	64 Kbyte	64 Kbyte
HSB-PB Bridge B	0xFFFF_0000	64 Kbyte	64 Kbyte	64 kByte	64 kByte	64 Kbyte	64 Kbyte

Figure 9-1. HMatrix Master / Slave Connections



  - Optional Manchester Encoding
- RS485 with driver control signal
- ISO7816, T = 0 or T = 1 Protocols for interfacing with smart cards
  - NACK handling, error counter with repetition and iteration limit
- IrDA modulation and demodulation
  - Communication at up to 115.2 Kbps
- Test Modes
  - Remote Loopback, Local Loopback, Automatic Echo
- SPI Mode
  - Master or Slave
  - Serial Clock Programmable Phase and Polarity
  - SPI Serial Clock (SCK) Frequency up to Internal Clock Frequency PBA/4
- Supports Connection of Two Peripheral DMA Controller Channels (PDC)
  - Offers Buffer Transfer without Processor Intervention

10.11.8 Serial Synchronous Controller

- Provides serial synchronous communication links used in audio and telecom applications (with CODECs in Master or Slave Modes, I2S, TDM Buses, Magnetic Card Reader, etc.)
- Contains an independent receiver and transmitter and a common clock divider
- Offers a configurable frame sync and data length
- Receiver and transmitter can be programmed to start automatically or on detection of different event on the frame sync signal
- Receiver and transmitter include a data signal, a clock signal and a frame synchronization signal

10.11.9 Timer Counter

- Three 16-bit Timer Counter Channels
- Wide range of functions including:
  - Frequency Measurement
  - Event Counting
  - Interval Measurement
  - Pulse Generation
  - Delay Timing
  - Pulse Width Modulation
  - Up/down Capabilities
- Each channel is user-configurable and contains:
  - Three external clock inputs
  - Five internal clock inputs
  - Two multi-purpose input/output signals
- Two global registers that act on all three TC Channels

10.11.10 Pulse Width Modulation Controller

- 7 channels, one 20-bit counter per channel
- Common clock generator, providing Thirteen Different Clocks
  - A Modulo n counter providing eleven clocks
  - Two independent Linear Dividers working on modulo n counter outputs
- Independent channel programming
  - Independent Enable Disable Commands
  - Independent Clock
  - Independent Period and Duty Cycle, with Double Bufferization
  - Programmable selection of the output waveform polarity
  - Programmable center or left aligned output waveform

11. Boot Sequence

This chapter summarizes the boot sequence of the AT32UC3A. The behaviour after power-up is controlled by the Power Manager. For specific details, refer to Section 13. ”Power Manager (PM)” on page 53.

11.1 Starting of clocks

After power-up, the device will be held in a reset state by the Power-On Reset circuitry, until the power has stabilized throughout the device. Once the power has stabilized, the device will use the internal RC Oscillator as clock source.

On system start-up, the PLLs are disabled. All clocks to all modules are running. No clocks have a divided frequency, all parts of the system recieves a clock with the same frequency as the internal RC Oscillator.

11.2 Fetching of initial instructions

After reset has been released, the AVR32 UC CPU starts fetching instructions from the reset address, which is 0x8000_0000. This address points to the first address in the internal Flash.

The code read from the internal Flash is free to configure the system to use for example the PLLs, to divide the frequency of the clock routed to some of the peripherals, and to gate the clocks to unused peripherals.

These figures represent the power consumption measured on the power supplies.

Table 12-9. Power Consumption for Different Modes

Mode	Conditions		Typ.	Unit
Active	Typ : Ta =25 °C CPU running from flash (1). VDDIN=3.3 V. VDDCORE =1.8V. CPU clocked from PLL0 at f MHz Voltage regulator is on. XIN0 : external clock. (1) XIN1 stopped. XIN32 stopped PLL0 running All peripheral clocks activated. GPIOs on internal pull-up. JTAG unconnected with ext pull-up.	f = 12 MHz	9	mA
		f = 24 MHz	15	mA
		f = 36MHz	20	mA
		f = 50 MHz	28	mA
		f = 66 MHz	36.3	mA
Idle	Typ : Ta = 25 °C CPU running from flash (1). VDDIN=3.3 V. VDDCORE =1.8V. CPU clocked from PLL0 at f MHz Voltage regulator is on. XIN0 : external clock. XIN1 stopped. XIN32 stopped PLL0 running All peripheral clocks activated. GPIOs on internal pull-up. JTAG unconnected with ext pull-up.	f = 12 MHz	5	mA
		f = 24 MHz	10	mA
		f = 36MHz	14	mA
		f = 50 MHz	19	mA
		f = 66 MHz	25.5	mA
Frozen	Typ : Ta = 25 °C CPU running from flash (1). CPU clocked from PLL0 at f MHz Voltage regulator is on. XIN0 : external clock. XIN1 stopped. XIN32 stopped PLL0 running All peripheral clocks activated. GPIOs on internal pull-up. JTAG unconnected with ext pull-up.	f = 12 MHz	3	mA
		f = 24 MHz	6	mA
		f = 36MHz	9	mA
		f = 50 MHz	13	mA
		f = 66 MHz	16.8	mA
Standby	Typ : Ta = 25 °C CPU running from flash (1). CPU clocked from PLL0 at f MHz Voltage regulator is on. XIN0 : external clock. XIN1 stopped. XIN32 stopped PLL0 running All peripheral clocks activated. GPIOs on internal pull-up. JTAG unconnected with ext pull-up.	f = 12 MHz	1	mA
		f = 24 MHz	2	mA
		f = 36MHz	3	mA
		f = 50 MHz	4	mA
		f = 66 MHz	4.8	mA

- V_{DDCORE} = 1.8V
- Ambient Temperature = 25°C

12.6.1 CPU/HSB Clock Characteristics

Table 12-11. Core Clock Waveform Parameters

Symbol	Parameter	Conditions	Min	Max	Units
$1/(t_{CPCPU})$	CPU Clock Frequency			66	MHz
t_{CPCPU}	CPU Clock Period		15,15		ns

12.6.2 PBA Clock Characteristics

Table 12-12. PBA Clock Waveform Parameters

Symbol	Parameter	Conditions	Min	Max	Units
$1/(t_{CPPBA})$	PBA Clock Frequency			66	MHz
t_{CPPBA}	PBA Clock Period		15,15		ns

12.6.3 PBB Clock Characteristics

Table 12-13. PBB Clock Waveform Parameters

Symbol	Parameter	Conditions	Min	Max	Units
$1/(t_{CPPBB})$	PBB Clock Frequency			66	MHz
t_{CPPBB}	PBB Clock Period		15,15		ns

12.7 Crystal Oscillator Characteristis

The following characteristics are applicable to the operating temperature range: T_A = -40°C to 85°C and worst case of power supply, unless otherwise specified.

12.7.1 32 KHz Oscillator Characteristics

Table 12-14. 32 KHz Oscillator Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$1/(t_{CP32KHz})$	Crystal Oscillator Frequency				32 768	Hz
C_L	Equivalent Load Capacitance		6		12.5	pF
t_{ST}	Startup Time	C_L = 6pF(1) C_L = 12.5pF(1)			600 1200	ms
I_{OSC}	Current Consumption	Active mode			1.8	µA
		Standby mode			0.1	µA

Note: 1. C_L is the equivalent load capacitance.

Table 12-21. Transfer Characteristics in 10-bit mode

Parameter	Conditions	Min	Typ	Max	Units
Resolution			10		Bit
Absolute Accuracy	f=5MHz			3	LSB
Integral Non-linearity	f=5MHz		1.5	2	LSB
Differential Non-linearity	f=5MHz		1	2	LSB
	f=2.5MHz		0.6	1	LSB
Offset Error	f=5MHz	-2		2	LSB
Gain Error	f=5MHz	-2		2	LSB

12.9 EBI Timings

These timings are given for worst case process, T = 85·C, VDDCORE = 1.65V, VDDIO = 3V and 40 pF load capacitance.

Table 12-22. SMC Clock Signal.

Symbol	Parameter	Max[1]	Units
$1/(t_{CPSMC})$	SMC Controller Clock Frequency	$1/(t_{cpcpu})$	MHz

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

Table 12-23. SMC Read Signals with Hold Settings

Symbol	Parameter	Min	Units
NRD Controlled (READ_MODE = 1)			
SMC_1	Data Setup before NRD High	12	ns
SMC_2	Data Hold after NRD High	0	
SMC_3	NRD High to NBS0/A0 Change[1]	nrd hold length * t_{CPSMC} - 1.3	
SMC_4	NRD High to NBS1 Change[1]	nrd hold length * t_{CPSMC} - 1.3	
SMC_5	NRD High to NBS2/A1 Change[1]	nrd hold length * t_{CPSMC} - 1.3	
SMC_6	NRD High to NBS3 Change[1]	nrd hold length * t_{CPSMC} - 1.3	
SMC_7	NRD High to A2 - A25 Change[1]	nrd hold length * t_{CPSMC} - 1.3	
SMC_8	NRD High to NCS Inactive[1]	(nrd hold length - ncs rd hold length) * t_{CPSMC} - 2.3	
SMC_9	NRD Pulse Width	nrd pulse length * t_{CPSMC} - 1.4	
NRD Controlled (READ_MODE = 0)			
SMC_{10}	Data Setup before NCS High	11.5	ns
SMC_{11}	Data Hold after NCS High	0	
SMC_{12}	NCS High to NBS0/A0 Change[1]	ncs rd hold length * t_{CPSMC} - 2.3	
SMC_{13}	NCS High to NBS0/A0 Change[1]	ncs rd hold length * t_{CPSMC} - 2.3	
SMC_{14}	NCS High to NBS2/A1 Change[1]	ncs rd hold length * t_{CPSMC} - 2.3	
SMC_{15}	NCS High to NBS3 Change[1]	ncs rd hold length * t_{CPSMC} - 2.3	
SMC_{16}	NCS High to A2 - A25 Change[1]	ncs rd hold length * t_{CPSMC} - 4	
SMC_{17}	NCS High to NRD Inactive[1]	ncs rd hold length - nrd hold length)* t_{CPSMC} - 1.3	
SMC_{18}	NCS Pulse Width	ncs rd pulse length * t_{CPSMC} - 3.6	

Note: 1. hold length = total cycle duration - setup duration - pulse duration. “hold length” is for “ncs rd hold length” or “nrd hold length”.

Table 12-26. SMC Write Signals with No Hold Settings (NWE Controlled only).

Symbol	Parameter	Min	Units
SMC_{37}	NWE Rising to A2-A25 Valid	5.4	ns
SMC_{38}	NWE Rising to NBS0/A0 Valid	5	
SMC_{39}	NWE Rising to NBS1 Change	5	
SMC_{40}	NWE Rising to A1/NBS2 Change	5	
SMC_{41}	NWE Rising to NBS3 Change	5	
SMC_{42}	NWE Rising to NCS Rising	5.1	
SMC_{43}	Data Out Valid before NWE Rising	(nwe pulse length - 1) * t_{CPSMC} - 1.2	
SMC_{44}	Data Out Valid after NWE Rising	5	
SMC_{45}	NWE Pulse Width	nwe pulse length * t_{CPSMC} - 0.9	

Figure 12-2. SMC Signals for NCS Controlled Accesses.

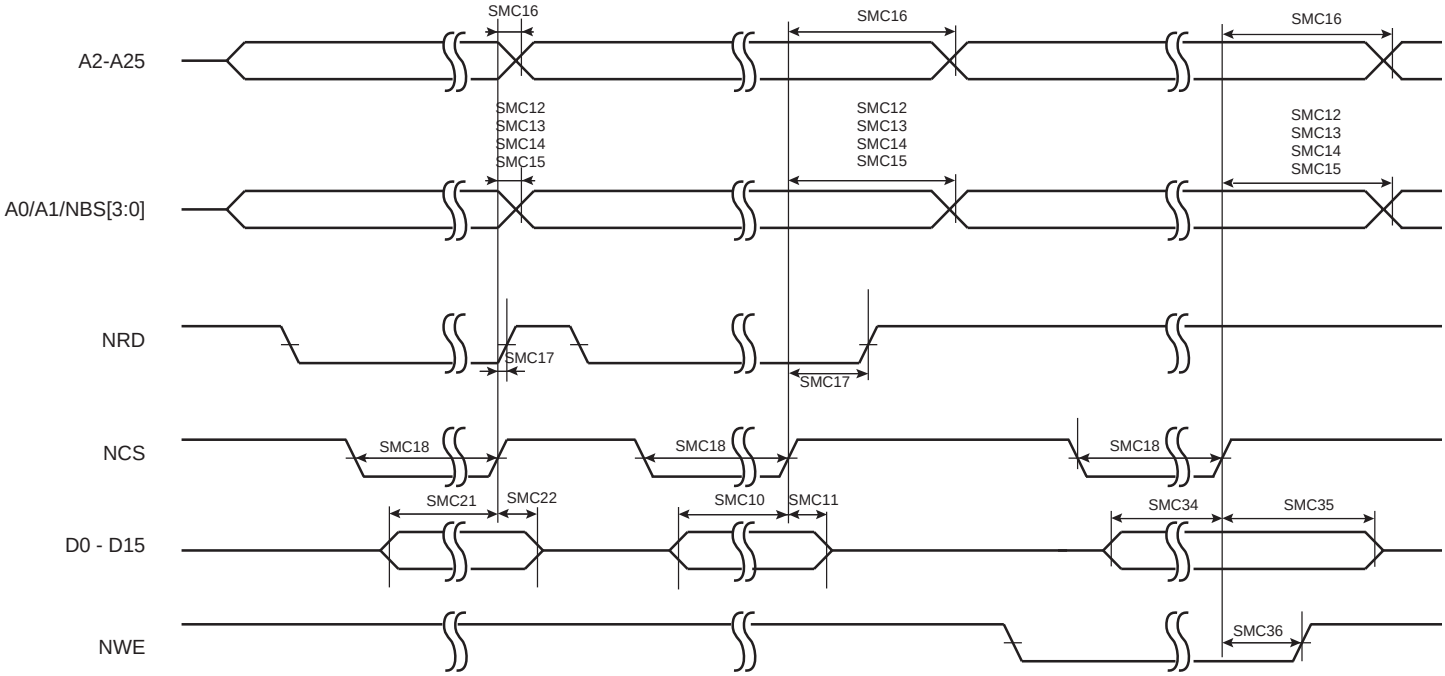
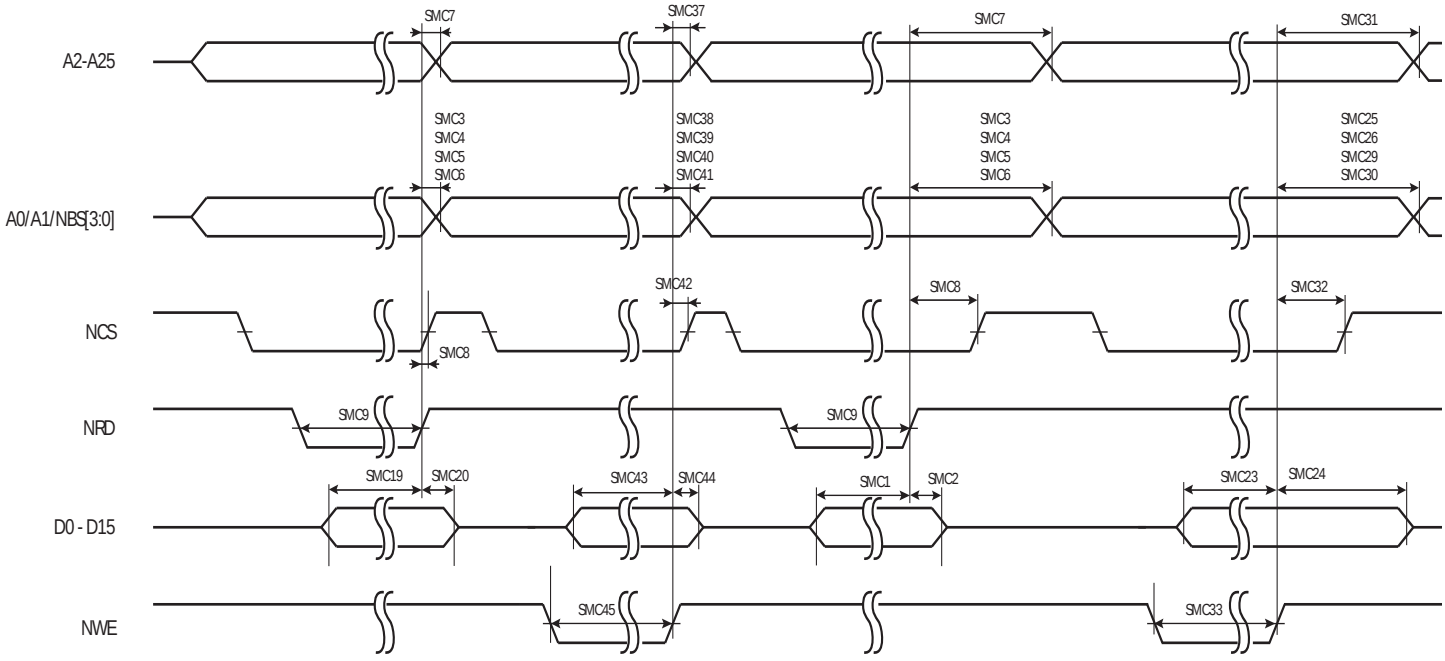


Figure 12-3. SMC Signals for NRD and NRW Controlled Accesses.



12.9.1 SDRAM Signals

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

Table 12-27. SDRAM Clock Signal.

Symbol	Parameter	Max[1]	Units
$1/(t_{CPSDCK})$	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 12-28. SDRAM Clock Signal.

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

15.2 Rev. J

15.2.1 PWM

1. **PWM channel interrupt enabling triggers an interrupt**
   When enabling a PWM channel that is configured with center aligned period (CALG=1), an interrupt is signalled.

   Fix/Workaround
   When using center aligned mode, enable the channel and read the status before channel interrupt is enabled.

2. **PWM counter restarts at 0x0001**
   The PWM counter restarts at 0x0001 and not 0x0000 as specified. Because of this the first PWM period has one more clock cycle.
   Fix/Workaround
   - The first period is 0x0000, 0x0001, ..., period
   - Consecutive periods are 0x0001, 0x0002, ..., period

3. **PWM update period to a 0 value does not work**
   It is impossible to update a period equal to 0 by the using the PWM update register (PWM_CUPD).

   Fix/Workaround
   Do not update the PWM_CUPD register with a value equal to 0.

15.2.2 ADC

1. **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.

15.2.3 SPI

1. **SPI Slave / PDCA transfer: no TX UNDERRUN flag**
   There is no TX UNDERRUN flag available, therefore in SPI slave mode, there is no way to be informed of a character lost in transmission.

   Fix/Workaround
   For PDCA transfer: none.

2. **SPI FDIV option does not work**
   Selecting clock signal using FDIV = 1 does not work as specified.

   Fix/Workaround
   Do not set FDIV = 1.

3. **SPI Bad Serial Clock Generation on 2nd chip_select when SCBR = 1, CPOL=1 and NCPHA=0**
   When multiple CS are in use, if one of the baudrate equals to 1 and one of the others doesn't equal to 1, and CPOL=1 and CPHA=0, then an aditional pulse will be generated on SCK.
   Fix/workaround

(data read or code fetch) the second half (last 256 kB) of the flash may fail. This may lead to an exception or to other errors derived from this corrupted read access.

Fix/Workaround

Flashc WP, EP, EA, WUP, EUP commands: these commands must be issued from RAM or through the EBI. After these commands, read twice one flash page initialized to 00h in each half part of the flash.

15.4.6 PDCA

1. **Wrong PDCA behavior when using two PDCA channels with the same PID.**
Workaround/fix
The same PID should not be assigned to more than one channel.

15.4.7 TWI

1. **The TWI RXRDY flag in SR register is not reset when a software reset is performed.**
Fix/Workaround
After a Software Reset, the register TWI RHR must be read.

15.4.8 SDRAMC

1. **Code execution from external SDRAM does not work**
Code execution from SDRAM does not work.

Fix/Workaround
Do not run code from SDRAM.

15.4.9 GPIO

1. **Some GPIO VIH (input high voltage) are 3.6V max instead of 5V tolerant**
Only 11 GPIOs remain 5V tolerant (VIHmax=5V):PB01, PB02, PB03, PB10, PB19, PB20, PB21, PB22, PB23, PB27, PB28**.**
Workaround/fix
None.

15.4.10 USART

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

15.4.11 Processor and Architecture

1. **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.

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

3. **Exceptions when system stack is protected by MPU**

15.5 Rev. E

15.5.1 SPI

1. **SPI FDIV option does not work**
   Selecting clock signal using FDIV = 1 does not work as specified.

   Fix/Workaround
   Do not set FDIV = 1.

2. **SPI Slave / PDCA transfer: no TX UNDERRUN flag**
   There is no TX UNDERRUN flag available, therefore in SPI slave mode, there is no way to be informed of a character lost in transmission.

   Fix/Workaround
   For PDCA transfer: none.

3. **SPI Bad serial clock generation on 2nd chip select when SCBR=1, CPOL=1 and CNCPHA=0**
   When multiple CS are in use, if one of the baudrate equals to 1 and one of the others doesn't equal to 1, and CPOL=1 and CPHA=0, then an additional pulse will be generated on SCK.

   Fix/Workaround
   When multiple CS are in use, if one of the baudrate equals to 1, the other must also equal 1 if CPOL=1 and CPHA=0.

4. **SPI Glitch on RXREADY flag in slave mode when enabling the SPI or during the first transfer**
   In slave mode, the SPI can generate a false RXREADY signal during enabling of the SPI or during the first transfer.

   Fix/Workaround
   1. Set slave mode, set required CPOL/CPHA.
   2. Enable SPI.
   3. Set the polarity CPOL of the line in the opposite value of the required one.
   4. Set the polarity CPOL to the required one.
   5. Read the RXHOLDING register.

   Transfers can now befin and RXREADY will now behave as expected.

5. **SPI CSNAAT bit 2 in register CSR0...CSR3 is not available.**
   Fix/Workaround

   Do not use this bit.

6. **SPI disable does not work in SLAVE mode.**
   Fix/Workaround
   Read the last received data, then perform a Software Reset.

7. **SPI Bad Serial Clock Generation on 2nd chip_select when SCBR = 1, CPOL=1 and NCPHA=0**
   When multiple CS are in use, if one of the baudrate equals to 1 and one of the others doesn't equal to 1, and CPOL=1 and CPHA=0, then an aditional pulse will be generated on SCK.

Fix/workaround

When multiple CS are in use, if one of the baudrate equals 1, the other must also equal 1 if CPOL=1 and CPHA=0.

15.5.2 PWM

1. **PWM counter restarts at 0x0001**

   The PWM counter restarts at 0x0001 and not 0x0000 as specified. Because of this the first PWM period has one more clock cycle.

   Fix/Workaround

   - The first period is 0x0000, 0x0001, ..., period

   - Consecutive periods are 0x0001, 0x0002, ..., period

2. **PWM channel interrupt enabling triggers an interrupt**

   When enabling a PWM channel that is configured with center aligned period (CALG=1), an interrupt is signalled.

   Fix/Workaround

   When using center aligned mode, enable the channel and read the status before channel interrupt is enabled.

3. **PWM update period to a 0 value does not work**

   It is impossible to update a period equal to 0 by the using the PWM update register (PWM_CUPD).

   Fix/Workaround

   Do not update the PWM_CUPD register with a value equal to 0.

4. **PWM channel status may be wrong if disabled before a period has elapsed**

   Before a PWM period has elapsed, the read channel status may be wrong. The CHIDx-bit for a PWM channel in the PWM Enable Register will read '1' for one full PWM period even if the channel was disabled before the period elapsed. It will then read '0' as expected.

   Fix/Workaround

   Reading the PWM channel status of a disabled channel is only correct after a PWM period has elapsed.

15.5.3 SSC

1. **SSC does not trigger RF when data is low**

   The SSC cannot transmit or receive data when CKS = CKDIV and CKO = none, in TCMR or RCMR respectively.

   Fix/Workaround

   Set CKO to a value that is not "none" and bypass the output of the TK/RK pin with the PIO.

2. **SSC Data is not sent unless clock is set as output**

   The SSC cannot transmit or receive data when CKS = CKDIV and CKO = none, in TCMR or RCMR respectively.

   Fix/Workaround

   Set CKO to a value that is not "none" and bypass the output of the TK/RK pin with the PIO.

15.5.4 USB

1. **USB No end of host reset signaled upon disconnection**
   In host mode, in case of an unexpected device disconnection whereas a usb reset is being sent by the usb controller, the UHCON.RESET bit may not been cleared by the hardware at the end of the reset.

   Fix/Workaround
   A software workaround consists in testing (by polling or interrupt) the disconnection (UHINT.DDISCI == 1) while waiting for the end of reset (UHCON.RESET == 0) to avoid being stuck.

2. **USBFSM and UHADDR1/2/3 registers are not available.**
   Do not use USBFSM register.

   Fix/Workaround
   Do not use USBFSM register and use HCON[6:0] field instead for all the pipes.

15.5.5 Processor and Architecture

1. **Incorrect Processor ID**
   The processor ID reads 0x01 and not 0x02 as it should.

   Fix/Workaround
   None.

2. **Bus error should be masked in Debug mode**
   If a bus error occurs during debug mode, the processor will not respond to debug commands through the DINST register.

   Fix/Workaround
   A reset of the device will make the CPU respond to debug commands again.

3. **Read Modify Write (RMW) instructions on data outside the internal RAM does not work.**
   Read Modify Write (RMW) instructions on data outside the internal RAM does not work.

   Fix/Workaround
   Do not perform RMW instructions on data outside the internal RAM.

4. **CRC calculation of a locked device will calculate CRC for 512 kB of flash memory, even though the part has less flash.**
   Fix/Workaround
   The flash address space is wrapping, so it is possible to use the CRC value by calculating CRC of the flash content concatenated with itself N times. Where N is 512 kB/flash size.

5. **Need two NOPs instruction after instructions masking interrupts**
   The instructions following in the pipeline the instruction masking the interrupt through SR may behave abnormally.

   Fix/Workaround
   Place two NOPs instructions after each SSRF or MTSR instruction setting IxM or GM in SR.

2. **The RTC CLKEN bit (bit number 16) of CTRL register is not available.**
   Fix/Workaround
   Do not use the CLKEN bit of the RTC on Rev E**.**

15.5.14 OCD

1. **Stalled memory access instruction writeback fails if followed by a HW breakpoint.**
   Consider the following assembly code sequence:
   A
   B
   If a hardware breakpoint is placed on instruction B, and instruction A is a memory access instruction, register file updates from instruction A can be discarded.
   Fix/Workaround
   Do not place hardware breakpoints, use software breakpoints instead.
   Alternatively, place a hardware breakpoint on the instruction before the memory access instruction and then single step over the memory access instruction.

15.5.15 PDCA

1. **Wrong PDCA behavior when using two PDCA channels with the same PID.**
   Workaround/fix
   The same PID should not be assigned to more than one channel.

15.5.16 TWI

1. **The TWI RXRDY flag in SR register is not reset when a software reset is performed.**
   Fix/Workaround
   After a Software Reset, the register TWI RHR must be read.