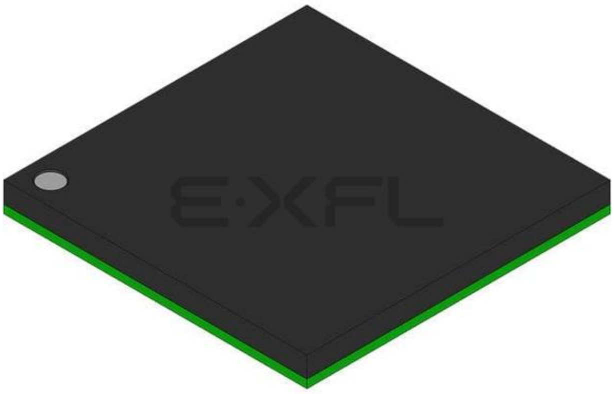

Welcome to E-XFL.COM

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	256KB (256K 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-TFBGA
Supplier Device Package	144-FFBGA (11x11)
Purchase URL	https://www.e-xfl.com/product-detail/atmel/at32uc3a0256-ctut

- **Peripheral Bus A able to run on at divided bus speeds compared to the High Speed Bus**

Figure 4-1 gives an overview of the bus system. All modules connected to the same bus use the same clock, but the clock to each module can be individually shut off by the Power Manager. The figure identifies the number of master and slave interfaces of each module connected to the High Speed Bus, and which DMA controller is connected to which peripheral.

6. Package and Pinout

The device pins are multiplexed with peripheral functions as described in ”Peripheral Multiplexing on I/O lines” on page 31.

Figure 6-1. TQFP100 Pinout

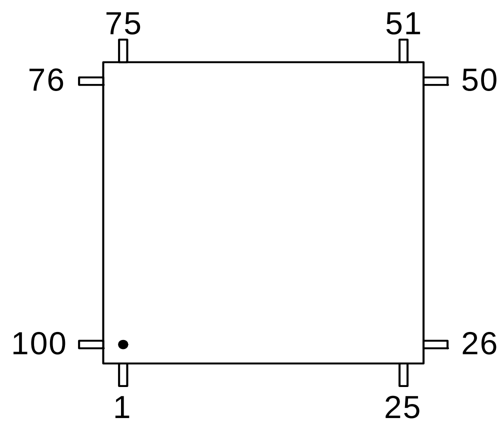

Table 6-1. TQFP100 Package Pinout

1	PB20	**26**	PA05	**51**	PA21	**76**	PB08
2	PB21	**27**	PA06	**52**	PA22	**77**	PB09
3	PB22	**28**	PA07	**53**	PA23	**78**	PB10
4	VDDIO	**29**	PA08	**54**	PA24	**79**	VDDIO
5	GND	**30**	PA09	**55**	PA25	**80**	GND
6	PB23	**31**	PA10	**56**	PA26	**81**	PB11
7	PB24	**32**	N/C	**57**	PA27	**82**	PB12
8	PB25	**33**	PA11	**58**	PA28	**83**	PA29
9	PB26	**34**	VDDCORE	**59**	VDDANA	**84**	PA30
10	PB27	**35**	GND	**60**	ADVREF	**85**	PC02
11	VDDOUT	**36**	PA12	**61**	GNDANA	**86**	PC03
12	VDDIN	**37**	PA13	**62**	VDDPLL	**87**	PB13
13	GND	**38**	VDDCORE	**63**	PC00	**88**	PB14
14	PB28	**39**	PA14	**64**	PC01	**89**	TMS
15	PB29	**40**	PA15	**65**	PB00	**90**	TCK
16	PB30	**41**	PA16	**66**	PB01	**91**	TDO
17	PB31	**42**	PA17	**67**	VDDIO	**92**	TDI
18	RESET_N	**43**	PA18	**68**	VDDIO	**93**	PC04
19	PA00	**44**	PA19	**69**	GND	**94**	PC05
20	PA01	**45**	PA20	**70**	PB02	**95**	PB15
21	GND	**46**	VBUS	**71**	PB03	**96**	PB16
22	VDDCORE	**47**	VDDIO	**72**	PB04	**97**	VDDCORE

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.

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

Table 9-2. Flash Memory Parameters

Part Number	Flash Size (*FLASH_PW*)	Number of pages (*FLASH_P*)	Page size (*FLASH_W*)	General Purpose Fuse bits (*FLASH_F*)
AT32UC3A0512	512 Kbytes	1024	128 words	32 fuses
AT32UC3A1512	512 Kbytes	1024	128 words	32 fuses
AT32UC3A0256	256 Kbytes	512	128 words	32 fuses
AT32UC3A1256	256 Kbytes	512	128 words	32 fuses
AT32UC3A1128	128 Kbytes	256	128 words	32 fuses
AT32UC3A0128	128 Kbytes	256	128 words	32 fuses

9.3 Bus Matrix Connections

Accesses to unused areas returns an error result to the master requesting such an access.

The bus matrix has the several masters and slaves. Each master has its own bus and its own decoder, thus allowing a different memory mapping per master. The master number in the table below can be used to index the HMATRIX control registers. For example, MCFG0 is associated with the CPU Data master interface.

Table 9-3. High Speed Bus masters

Master 0	CPU Data
Master 1	CPU Instruction
Master 2	CPU SAB
Master 3	PDCA
Master 4	MACB DMA
Master 5	USBB DMA

Each slave has its own arbiter, thus allowing a different arbitration per slave. The slave number in the table below can be used to index the HMATRIX control registers. For example, SCFG3 is associated with the Internal SRAM Slave Interface.

Table 9-4. High Speed Bus slaves

Slave 0	Internal Flash
Slave 1	HSB-PB Bridge 0
Slave 2	HSB-PB Bridge 1
Slave 3	Internal SRAM
Slave 4	USBB DPRAM
Slave 5	EBI

Table 10-1. Peripheral Address Mapping (Continued)

Address	Peripheral Name		Bus
0xFFFF1C00	USART2	Universal Synchronous Asynchronous Receiver Transmitter - USART2	PBA
0xFFFF2000	USART3	Universal Synchronous Asynchronous Receiver Transmitter - USART3	PBA
0xFFFF2400	SPI0	Serial Peripheral Interface - SPI0	PBA
0xFFFF2800	SPI1	Serial Peripheral Interface - SPI1	PBA
0xFFFF2C00	TWI	Two Wire Interface - TWI	PBA
0xFFFF3000	PWM	Pulse Width Modulation Controller - PWM	PBA
0xFFFF3400	SSC	Synchronous Serial Controller - SSC	PBA
0xFFFF3800	TC	Timer/Counter - TC	PBA
0xFFFF3C00	ADC	Analog To Digital Converter - ADC	PBA

10.2 CPU Local Bus Mapping

Some of the registers in the GPIO module are mapped onto the CPU local bus, in addition to being mapped on the Peripheral Bus. These registers can therefore be reached both by accesses on the Peripheral Bus, and by accesses on the local bus.

Mapping these registers on the local bus allows cycle-deterministic toggling of GPIO pins since the CPU and GPIO are the only modules connected to this bus. Also, since the local bus runs at CPU speed, one write or read operation can be performed per clock cycle to the local bus-mapped GPIO registers.

12.2 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 to T_J = 100°C.

Table 12-1. DC Characteristics

Symbol	Parameter	Condition	Min.	Typ.	Max	Units
$V_{VDDCORE}$	DC Supply Core		1.65		1.95	V
V_{VDDPLL}	DC Supply PLL		1.65		1.95	V
V_{VDDIO}	DC Supply Peripheral I/Os		3.0		3.6	V
V_{REF}	Analog reference voltage		2.6		3.6	V
V_{IL}	Input Low-level Voltage		-0.3		+0.8	V
V_{IH}	Input High-level Voltage	All GPIOS except for PC00, PC01, PC02, PC03, PC04, PC05.	2.0		5.5V	V
		PC00, PC01, PC02, PC03, PC04, PC05.	2.0		3.6V	V
V_{OL}	Output Low-level Voltage	I_{OL}=-4mA for PA0-PA20, PB0, PB4-PB9, PB11-PB18, PB24-PB26, PB29-PB31, PX0-PX39			0.4	V
		I_{OL}=-8mA for PA21-PA30, PB1-PB3, PB10, PB19-PB23, PB27-PB28, PC0-PC5			0.4	V
V_{OH}	Output High-level Voltage	I_{OH}=4mA for PA0-PA20, PB0, PB4-PB9, PB11-PB18, PB24-PB26, PB29-PB31, PX0-PX39	V_{VDDIO}-0.4			V
		I_{OH}=8mA for PA21-PA30, PB1-PB3, PB10, PB19-PB23, PB27-PB28, PC0-PC5	V_{VDDIO}-0.4			V
I_{OL}	Output Low-level Current	PA0-PA20, PB0, PB4-PB9, PB11-PB18, PB24-PB26, PB29-PB31, PX0-PX39			-4	mA
		PA21-PA30, PB1-PB3, PB10, PB19-PB23, PB27-PB28, PC0-PC5			-8	mA
I_{OH}	Output HIgh-level Current	PA0-PA20, PB0, PB4-PB9, PB11-PB18, PB24-PB26, PB29-PB31, PX0-PX39			4	mA
		PA21-PA30, PB1-PB3, PB10, PB19-PB23, PB27-PB28, PC0-PC5			8	mA
I_{LEAK}	Input Leakage Current	Pullup resistors disabled			1	µA
C_{IN} Input Capacitance		TQFP100 Package		7		pF
		LQFP144 Package		7		pF
R_{PULLUP}	Pull-up Resistance	All GPIO and RESET_N pin.	10K	15K		Ohm

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

Table 12-24. SMC Read Signals with no Hold Settings

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

Table 12-25. SMC Write Signals with Hold Settings

Symbol	Parameter	Min	Units
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	
SMC_{25}	NWE High to NBS0/A0 Change[(1)]	nwe hold length * t_{CPSMC} - 1.9	
SMC_{26}	NWE High to NBS1 Change[(1)]	nwe hold length * t_{CPSMC} - 1.9	
SMC_{29}	NWE High to NBS2/A1 Change[(1)]	nwe hold length * t_{CPSMC} - 1.9	
SMC_{30}	NWE High to NBS3 Change[(1)]	nwe hold length * t_{CPSMC} - 1.9	
SMC_{31}	NWE High to A2 - A25 Change[(1)]	nwe hold length * t_{CPSMC} - 1.7	
SMC_{32}	NWE High to NCS Inactive[(1)]	(nwe hold length - ncs wr hold length)* t_{CPSMC} - 2.9	
SMC_{33}	NWE Pulse Width	nwe pulse length * t_{CPSMC} - 0.9	
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	
SMC_{36}	NCS High to NWE Inactive[(1)]	(ncs wr hold length - nwe hold length)* t_{CPSMC} - 0.6	

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

Figure 12-4. SDRAMC Signals relative to SDCK.

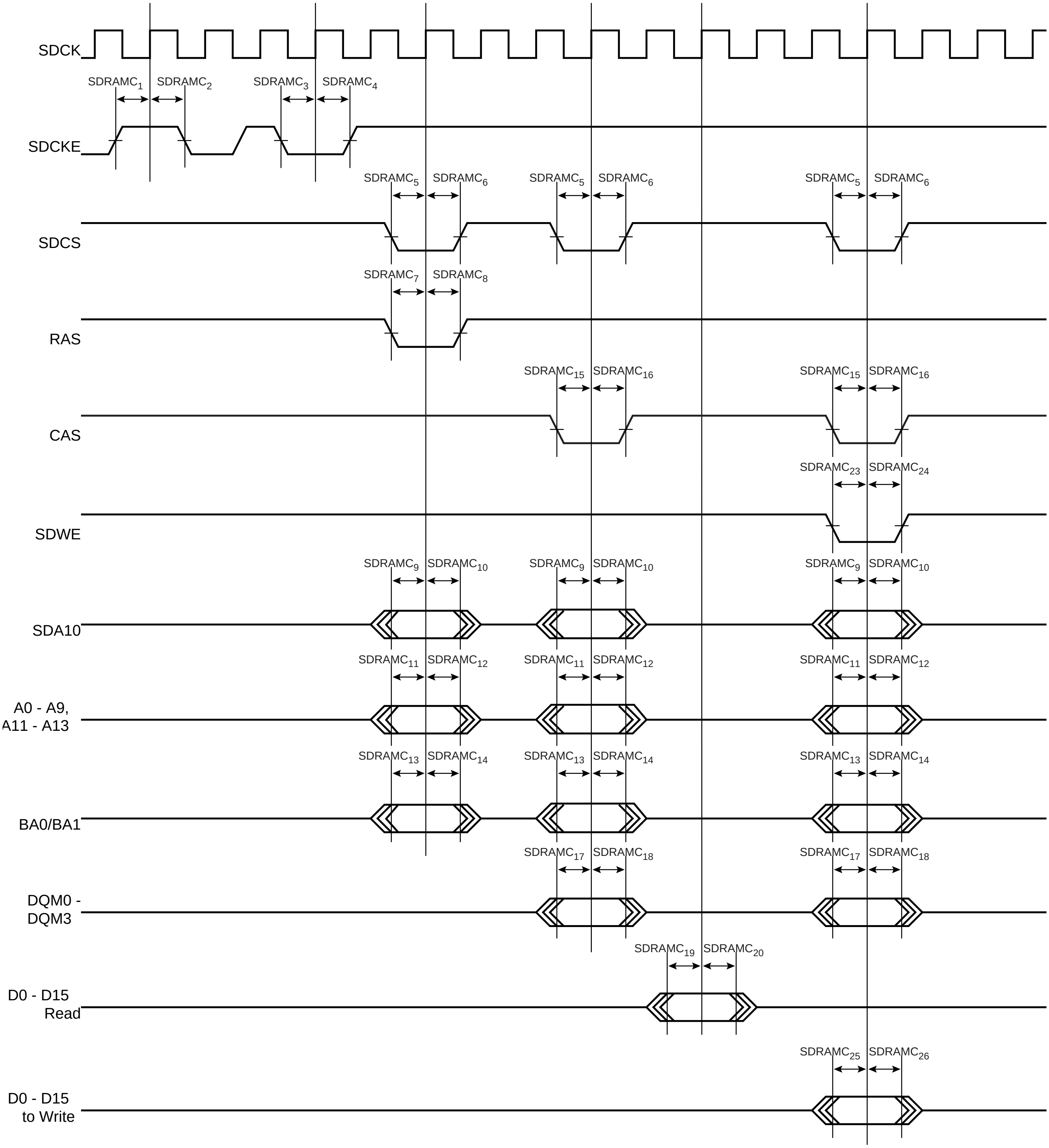

Figure 12-7. SPI Master mode with (CPOL=0 and NCPHA=1) or (CPOL=1 and NCPHA=0)

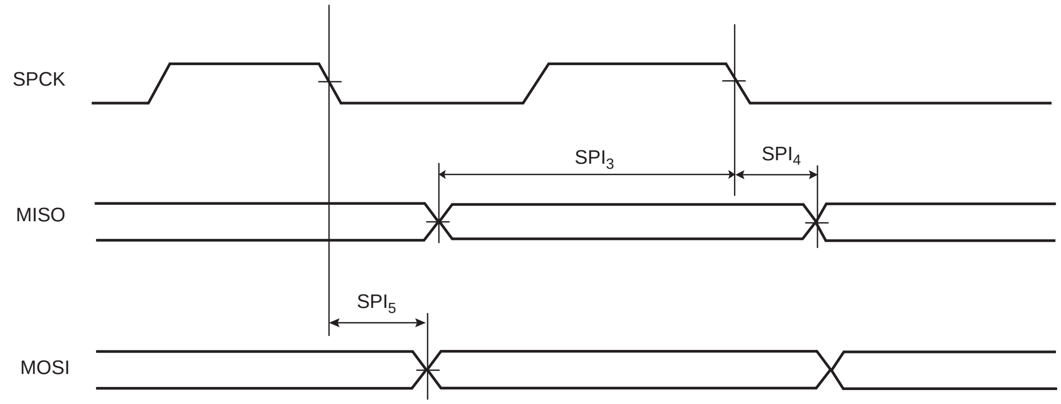

Figure 12-8. SPI Slave mode with (CPOL=0 and NCPHA=1) or (CPOL=1 and NCPHA=0)

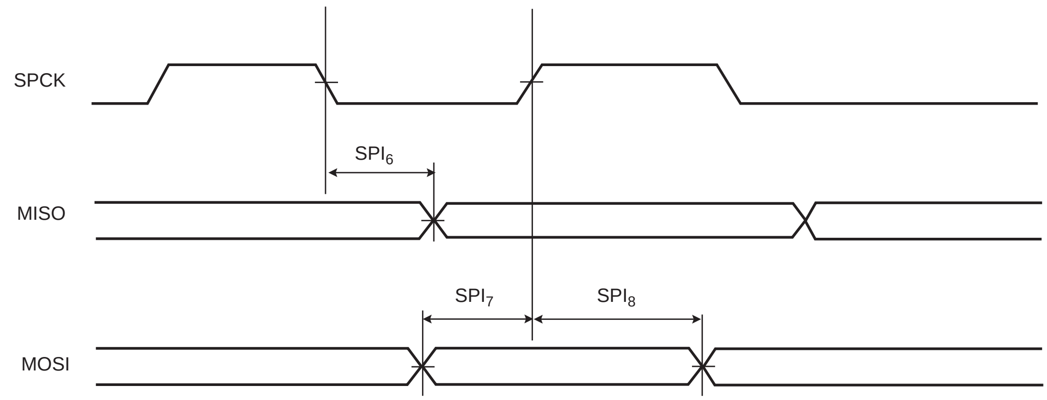

Figure 12-9. SPI Slave mode with (CPOL = NCPHA = 0) or (CPOL= NCPHA= 1)

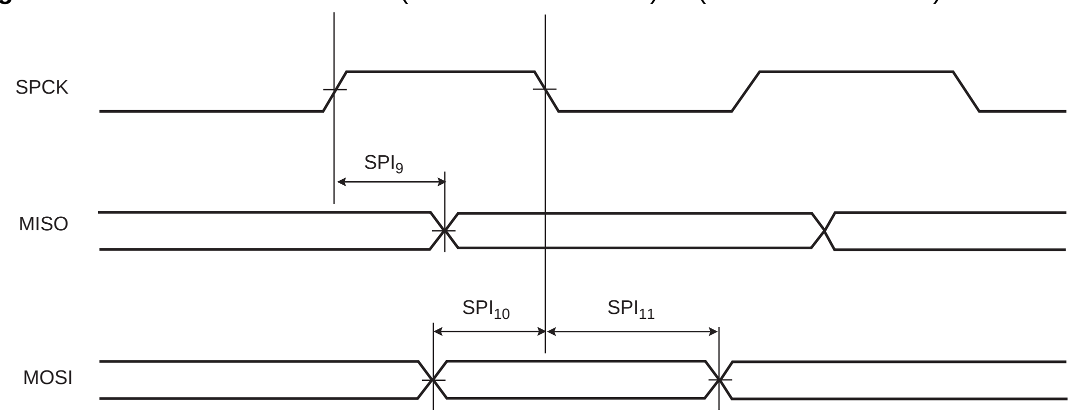

Table 12-35. Programming Time

Temperature Operating Range Part	Page Programming Time (ms)	Chip Erase Time (ms)
Industrial	4	4
Automotive	16	16

13. Mechanical Characteristics

13.1 Thermal Considerations

13.1.1 Thermal Data

Table 13-1 summarizes the thermal resistance data depending on the package.

Table 13-1. Thermal Resistance Data

Symbol	Parameter	Condition	Package	Typ	Unit
θ_{JA}	Junction-to-ambient thermal resistance	Still Air	TQFP100	43.4	·C/W
θ_{JC}	Junction-to-case thermal resistance		TQFP100	5.5	
θ_{JA}	Junction-to-ambient thermal resistance	Still Air	LQFP144	39.8	·C/W
θ_{JC}	Junction-to-case thermal resistance		LQFP144	8.9	

13.1.2 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 13-1 on page 64.
- θ_{JC} = package thermal resistance, Junction-to-case thermal resistance (°C/W), provided in Table 13-1 on page 64.
- $\theta_{HEAT\ SINK}$ = cooling device thermal resistance (°C/W), provided in the device datasheet.
- P_D = device power consumption (W) estimated from data provided in the section "Power Consumption" on page 44.
- 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.

14. Ordering Information

Table 14-1. Ordering Information

Device	Ordering Code	Package	Conditioning	Temperature Operating Range
AT32UC3A0512	AT32UC3A0512-ALUT	144 LQFP	Tray	Industrial (-40·C to 85·C)
	AT32UC3A0512-ALUR	144 LQFP	Reel	Industrial (-40·C to 85·C)
	AT32UC3A0512-ALTR	144 LQFP	Reel	Automotive (-40·C to 85·C)
	AT32UC3A0512-ALTT	144 LQFP	Tray	Automotive (-40·C to 85·C)
	AT32UC3A0512-ALTES	144 LQFP	Tray	Automotive (-40·C to 85·C) samples
	AT32UC3A0512-CTUT	144 FFBGA	Tray	Industrial (-40·C to 85·C)
	AT32UC3A0512-CTUR	144 FFBGA	Reel	Industrial (-40·C to 85·C)
AT32UC3A0256	AT32UC3A0256-ALUT	144 LQFP	Tray	Industrial (-40·C to 85·C)
	AT32UC3A0256-ALUR	144 LQFP	Reel	Industrial (-40·C to 85·C)
	AT32UC3A0256-CTUT	144 FFBGA	Tray	Industrial (-40·C to 85·C)
	AT32UC3A0256-CTUR	144 FFBGA	Reel	Industrial (-40·C to 85·C)
AT32UC3A0128	AT32UC3A0128-ALUT	144 LQFP	Tray	Industrial (-40·C to 85·C)
	AT32UC3A0128-ALUR	144 LQFP	Reel	Industrial (-40·C to 85·C)
	AT32UC3A0128-CTUT	144 FFBGA	Tray	Industrial (-40·C to 85·C)
	AT32UC3A0128-CTUR	144 FFBGA	Reel	Industrial (-40·C to 85·C)
AT32UC3A1512	AT32UC3A1512-AUT	100 TQFP	Tray	Industrial (-40·C to 85·C)
	AT32UC3A1512-AUR	100 TQFP	Reel	Industrial (-40·C to 85·C)
AT32UC3A1256	AT32UC3A1256-AUT	100 TQFP	Tray	Industrial (-40·C to 85·C)
	AT32UC3A1256-AUR	100 TQFP	Reel	Industrial (-40·C to 85·C)
AT32UC3A1128	AT32UC3A1128-AUT	100 TQFP	Tray	Industrial (-40·C to 85·C)
	AT32UC3A1128-AUR	100 TQFP	Reel	Industrial (-40·C to 85·C)

14.1 Automotive Quality Grade

The AT32UC3A have been developed and manufactured according to the most stringent requirements of the international standard ISO-TS-16949. This data sheet will contain limit values extracted from the results of extensive characterization (Temperature and Voltage). The quality and reliability of the AT32UC3A is verified during regular product qualification as per AEC-Q100 grade 3.

As indicated in the ordering information paragraph, the product is available in only one temperature grade T: -40°C / + 85°C.

Workaround/fix
The same PID should not be assigned to more than one channel.

15.3.7 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.

15.3.8 USART

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

15.3.9 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.

15.3.10 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.

15.3.11 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**
   RETS behaves incorrectly when MPU is enabled and MPU is configured so that system stack is not readable in unprivileged mode.
   Fix/Woraround
   Workaround 1: Make system stack readable in unprivileged mode,
   or
   Workaround 2: Return from supervisor mode using rete instead of rets. This requires :
   1. Changing the mode bits from 001b to 110b before issuing the instruction. Updating the mode bits to the desired value must be done using a single mtsr instruction so it is done atomically. Even if this step is described in general as not safe in the UC technical reference guide, it is safe in this very

4. **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.

5. **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.

6. **SPI Disable does not work in Slave mode**
Fix/workaround
Read the last received data then perform a Software reset.

15.4.4 Power Manager

1. **Wrong reset causes when BOD is activated**
Setting the BOD enable fuse will cause the Reset Cause Register to list BOD reset as the reset source even though the part was reset by another source.

Fix/Workaround
Do not set the BOD enable fuse, but activate the BOD as soon as your program starts.

2. **If the BOD level is higher than VDDCORE, the part is constantly under reset**
If the BOD level is set to a value higher than VDDCORE and enabled by fuses, the part will be in constant reset.

Fix/Workaround
Apply an external voltage on VDDCORE that is higher than the BOD level and is lower than VDDCORE max and disable the BOD.

15.4.5 FLASHC

1. **On AT32UC3A0512 and AT32UC3A1512, corrupted read in flash after FLASHC WP, EP, EA, WUP, EUP commands may happen**
- After a FLASHC Write Page (WP) or Erase Page (EP) command applied to a page in a given half of the flash (first or last 256 kB of flash), reading (data read or code fetch) the other half of the flash may fail. This may lead to an exception or to other errors derived from this corrupted read access.
- After a FLASHC Erase All (EA) command, reading (data read or code fetch) the flash may fail. This may lead to an exception or to other errors derived from this corrupted read access.
- After a FLASHC Write User Page (WUP) or Erase User Page (EUP) command, reading

(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**

12. **CPU cannot operate on a divided slow clock (internal RC oscillator)**
    Fix/Workaround
    Do not run the CPU on a divided slow clock.

13. **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.

14. **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.

15. **Exceptions when system stack is protected by MPU**
    RETS behaves incorrectly when MPU is enabled and MPU is configured so that system stack is not readable in unprivileged mode.
    Fix/Woraround
    Workaround 1: Make system stack readable in unprivileged mode,
    or
    Workaround 2: Return from supervisor mode using rete instead of rets. This requires :
    1. Changing the mode bits from 001b to 110b before issuing the instruction. Updating the mode bits to the desired value must be done using a single mtsr instruction so it is done atomically. Even if this step is described in general as not safe in the UC technical reference guide, it is safe in this very specific case.

2. Execute the RETE instruction.

15.5.6 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.

2. **SDRAM SDCKE rise at the same time as SDCK while exiting self-refresh mode**
    SDCKE rise at the same time as SDCK while exiting self-refresh mode.

    Fix/Workaround
    None.

15.5.7 USART

1. **USART Manchester Encoder Not Working**
    Manchester encoding/decoding is not working.

    Fix/Workaround
    Do not use manchester encoding.

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.