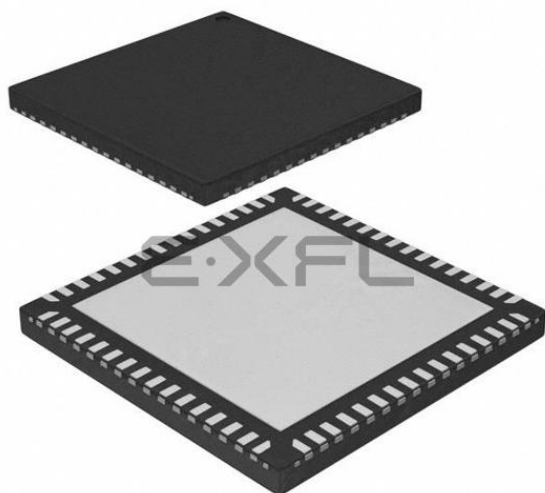




Welcome to [E-XFL.COM](https://www.e-xfl.com)

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	CANbus, Ethernet, I ² C, IrDA, LINbus, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I ² S, POR, PWM, WDT
Number of I/O	45
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	64K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 11x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-VFQFN Exposed Pad
Supplier Device Package	64-QFN (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at32uc3c2256c-z2ut

Table 3-1. GPIO Controller Function Multiplexing

TQFP / QFN 64	TQFP 100	LQFP 144	PIN	G P I O	Supply	Pin Type (1)	GPIO function					
							A	B	C	D	E	F
		14	PB11	43	VDDIO1	x1/x2	USART1 - DSR	SPI0 - MISO	PWM - PWMH[1]			
		15	PB12	44	VDDIO1	x1/x2	USART1 - DCD	SPI0 - SCK	PWM - PWML[2]			
		16	PB13	45	VDDIO1	x1/x2	USART1 - RI	SPI0 - NPCS[0]	PWM - PWMH[2]		MACB - RX_ER	
		17	PB14	46	VDDIO1	x1/x2	USART1 - RTS	SPI0 - NPCS[1]	PWM - PWML[3]		MACB - MDC	
		18	PB15	47	VDDIO1	x1/x2	USART1 - CTS	USART1 - CLK	PWM - PWMH[3]		MACB - MDIO	
		19	PB16	48	VDDIO1	x1/x2	USART1 - RXD	SPI0 - NPCS[2]	PWM - EXT_ FAULTS[0]		CANIF - RXLINE[0]	
		20	PB17	49	VDDIO1	x1/x2	USART1 - TXD	SPI0 - NPCS[3]	PWM - EXT_ FAULTS[1]		CANIF - TXLINE[0]	
		57	PB18	50	VDDIO2	x1/x2	TC0 - CLK2		EIC - EXTINT[4]			
	42	58	PB19	51	VDDIO2	x1/x2	TC0 - A0	SPI1 - MOSI	IISC - ISDO		MACB - CRS	
	43	59	PB20	52	VDDIO2	x1/x2	TC0 - B0	SPI1 - MISO	IISC - ISDI	ACIFA1 - ACAOUT	MACB - COL	
	44	60	PB21	53	VDDIO2	x2/x4	TC0 - CLK1	SPI1 - SCK	IISC - IMCK	ACIFA1 - ACBOUT	MACB - RXD[2]	
	45	61	PB22	54	VDDIO2	x1/x2	TC0 - A1	SPI1 - NPCS[3]	IISC - ISCK	SCIF - GCLK[0]	MACB - RXD[3]	
	46	62	PB23	55	VDDIO2	x1/x2	TC0 - B1	SPI1 - NPCS[2]	IISC - IWS	SCIF - GCLK[1]	MACB - RX_CLK	
		63	PB24	56	VDDIO2	x1/x2	TC0 - CLK0	SPI1 - NPCS[1]				
		64	PB25	57	VDDIO2	x1/x2	TC0 - A2	SPI1 - NPCS[0]	PEVC - PAD_EVT [8]			
		65	PB26	58	VDDIO2	x2/x4	TC0 - B2	SPI1 - SCK	PEVC - PAD_EVT [9]		MACB - TX_EN	
		66	PB27	59	VDDIO2	x1/x2	QDEC0 - QEPA	SPI1 - MISO	PEVC - PAD_EVT [10]	TC1 - CLK0	MACB - TXD[0]	
		67	PB28	60	VDDIO2	x1/x2	QDEC0 - QEPB	SPI1 - MOSI	PEVC - PAD_EVT [11]	TC1 - B0	MACB - TXD[1]	
		68	PB29	61	VDDIO2	x1/x2	QDEC0 - QEPI	SPI0 - NPCS[0]	PEVC - PAD_EVT [12]	TC1 - A0		
31	47	69	PB30	62	VDDIO2	x1						
32	48	70	PB31	63	VDDIO2	x1						
	49	71	PC00	64	VDDIO2	x1/x2	USBC - ID	SPI0 - NPCS[1]	USART2 - CTS	TC1 - B2	CANIF - TXLINE[1]	
	50	72	PC01	65	VDDIO2	x1/x2	USBC - VBOF	SPI0 - NPCS[2]	USART2 - RTS	TC1 - A2	CANIF - RXLINE[1]	

Table 3-1. GPIO Controller Function Multiplexing

TQFP / QFN 64	TQFP 100	LQFP 144	PIN	GPIO	Supply	Pin Type (1)	GPIO function					
							A	B	C	D	E	F
33	51	73	PC02	66	VDDIO2	x1	TWIMS0 - TWD	SPI0 - NPCS[3]	USART2 - RXD	TC1 - CLK1	MACB - MDC	
34	52	74	PC03	67	VDDIO2	x1	TWIMS0 - TWCK	EIC - EXTINT[1]	USART2 - TXD	TC1 - B1	MACB - MDIO	
37	55	77	PC04	68	VDDIO2	x1	TWIMS1 - TWD	EIC - EXTINT[3]	USART2 - TXD	TC0 - B1		
38	56	78	PC05	69	VDDIO2	x1	TWIMS1 - TWCK	EIC - EXTINT[4]	USART2 - RXD	TC0 - A2		
	57	79	PC06	70	VDDIO2	x1	PEVC - PAD_EVT [15]	USART2 - CLK	USART2 - CTS	TC0 - CLK2	TWIMS2 - TWD	TWIMS0 - TWALM
	58	80	PC07	71	VDDIO2	x1	PEVC - PAD_EVT [2]	EBI - NCS[3]	USART2 - RTS	TC0 - B2	TWIMS2 - TWCK	TWIMS1 - TWALM
		81	PC08	72	VDDIO2	x1/x2	PEVC - PAD_EVT [13]	SPI1 - NPCS[1]	EBI - NCS[0]		USART4 - TXD	
		82	PC09	73	VDDIO2	x1/x2	PEVC - PAD_EVT [14]	SPI1 - NPCS[2]	EBI - ADDR[23]		USART4 - RXD	
		83	PC10	74	VDDIO2	x1/x2	PEVC - PAD_EVT [15]	SPI1 - NPCS[3]	EBI - ADDR[22]			
	59	84	PC11	75	VDDIO2	x1/x2	PWM - PWMH[3]	CANIF - RXLINE[1]	EBI - ADDR[21]	TC0 - CLK0		
	60	85	PC12	76	VDDIO2	x1/x2	PWM - PWML[3]	CANIF - TXLINE[1]	EBI - ADDR[20]	USART2 - CLK		
	61	86	PC13	77	VDDIO2	x1/x2	PWM - PWMH[2]	EIC - EXTINT[7]		USART0 - RTS		
	62	87	PC14	78	VDDIO2	x1/x2	PWM - PWML[2]	USART0 - CLK	EBI - SDCKE	USART0 - CTS		
39	63	88	PC15	79	VDDIO2	x1/x2	PWM - PWMH[1]	SPI0 - NPCS[0]	EBI - SDWE	USART0 - RXD	CANIF - RXLINE[1]	
40	64	89	PC16	80	VDDIO2	x1/x2	PWM - PWML[1]	SPI0 - NPCS[1]	EBI - CAS	USART0 - TXD	CANIF - TXLINE[1]	
41	65	90	PC17	81	VDDIO2	x1/x2	PWM - PWMH[0]	SPI0 - NPCS[2]	EBI - RAS	IISC - ISDO		USART3 - TXD
42	66	91	PC18	82	VDDIO2	x1/x2	PWM - PWML[0]	EIC - EXTINT[5]	EBI - SDA10	IISC - ISDI		USART3 - RXD
43	67	92	PC19	83	VDDIO3	x1/x2	PWM - PWML[2]	SCIF - GCLK[0]	EBI - DATA[0]	IISC - IMCK		USART3 - CTS
44	68	93	PC20	84	VDDIO3	x1/x2	PWM - PWMH[2]	SCIF - GCLK[1]	EBI - DATA[1]	IISC - ISCK		USART3 - RTS
45	69	94	PC21	85	VDDIO3	x1/x2	PWM - EXT_FAULTS[0]	CANIF - RXLINE[0]	EBI - DATA[2]	IISC - IWS		
46	70	95	PC22	86	VDDIO3	x1/x2	PWM - EXT_FAULTS[1]	CANIF - TXLINE[0]	EBI - DATA[3]		USART3 - CLK	
	71	96	PC23	87	VDDIO3	x1/x2	QDEC1 - QEPB	CANIF - RXLINE[1]	EBI - DATA[4]	PEVC - PAD_EVT [3]		

Table 3-1. GPIO Controller Function Multiplexing

TQFP / QFN 64	TQFP 100	LQFP 144	PIN	GPIO	Supply	Pin Type (1)	GPIO function					
							A	B	C	D	E	F
	72	97	PC24	88	VDDIO3	x1/x2	QDEC1 - QEPA	CANIF - TXLINE[1]	EBI - DATA[5]	PEVC - PAD_EVT [4]		
		98	PC25	89	VDDIO3	x1/x2		TC1 - CLK2	EBI - DATA[6]	SCIF - GCLK[0]	USART4 - TXD	
		99	PC26	90	VDDIO3	x1/x2	QDEC1 - QEPI	TC1 - B2	EBI - DATA[7]	SCIF - GCLK[1]	USART4 - RXD	
		100	PC27	91	VDDIO3	x1/x2		TC1 - A2	EBI - DATA[8]	EIC - EXTINT[0]	USART4 - CTS	
		101	PC28	92	VDDIO3	x1/x2	SPI0 - NPCS[3]	TC1 - CLK1	EBI - DATA[9]		USART4 - RTS	
		102	PC29	93	VDDIO3	x1/x2	SPI0 - NPCS[1]	TC1 - B1	EBI - DATA[10]			
		105	PC30	94	VDDIO3	x1/x2	SPI0 - NPCS[2]	TC1 - A1	EBI - DATA[11]			
	73	106	PC31	95	VDDIO3	x1/x2	SPI0 - NPCS[3]	TC1 - B0	EBI - DATA[12]	PEVC - PAD_EVT [5]	USART4 - CLK	
47	74	107	PD00	96	VDDIO3	x1/x2	SPI0 - MOSI	TC1 - CLK0	EBI - DATA[13]	QDEC0 - QEPI	USART0 - TXD	
48	75	108	PD01	97	VDDIO3	x1/x2	SPI0 - MISO	TC1 - A0	EBI - DATA[14]	TC0 - CLK1	USART0 - RXD	
49	76	109	PD02	98	VDDIO3	x2/x4	SPI0 - SCK	TC0 - CLK2	EBI - DATA[15]	QDEC0 - QEPA		
50	77	110	PD03	99	VDDIO3	x1/x2	SPI0 - NPCS[0]	TC0 - B2	EBI - ADDR[0]	QDEC0 - QEPB		
		111	PD04	100	VDDIO3	x1/x2	SPI0 - MOSI		EBI - ADDR[1]			
		112	PD05	101	VDDIO3	x1/x2	SPI0 - MISO		EBI - ADDR[2]			
		113	PD06	102	VDDIO3	x2/x4	SPI0 - SCK		EBI - ADDR[3]			
	78	114	PD07	103	VDDIO3	x1/x2	USART1 - DTR	EIC - EXTINT[5]	EBI - ADDR[4]	QDEC0 - QEPI	USART4 - TXD	
	79	115	PD08	104	VDDIO3	x1/x2	USART1 - DSR	EIC - EXTINT[6]	EBI - ADDR[5]	TC1 - CLK2	USART4 - RXD	
	80	116	PD09	105	VDDIO3	x1/x2	USART1 - DCD	CANIF - RXLINE[0]	EBI - ADDR[6]	QDEC0 - QEPA	USART4 - CTS	
	81	117	PD10	106	VDDIO3	x1/x2	USART1 - RI	CANIF - TXLINE[0]	EBI - ADDR[7]	QDEC0 - QEPB	USART4 - RTS	
53	84	120	PD11	107	VDDIO3	x1/x2	USART1 - TXD	USBC - ID	EBI - ADDR[8]	PEVC - PAD_EVT [6]	MACB - TXD[0]	
54	85	121	PD12	108	VDDIO3	x1/x2	USART1 - RXD	USBC - VBOF	EBI - ADDR[9]	PEVC - PAD_EVT [7]	MACB - TXD[1]	
55	86	122	PD13	109	VDDIO3	x2/x4	USART1 - CTS	USART1 - CLK	EBI - SDCK	PEVC - PAD_EVT [8]	MACB - RXD[0]	
56	87	123	PD14	110	VDDIO3	x1/x2	USART1 - RTS	EIC - EXTINT[7]	EBI - ADDR[10]	PEVC - PAD_EVT [9]	MACB - RXD[1]	

Table 3-2. Peripheral Functions

Function	Description
aWire DATAOUT	aWire output in two-pin mode
JTAG port connections	JTAG debug port
Oscillators	OSC0, OSC32

3.2.3 Oscillator Pinout

The oscillators are not mapped to the normal GPIO functions and their muxings are controlled by registers in the System Control Interface (SCIF). Please refer to the SCIF chapter for more information about this.

Table 3-3. Oscillator pinout

QFN64/ TQFP64 pin	TQFP100 pin	LQFP144 pin	Pad	Oscillator pin
31	47	69	PB30	xin0
	99	143	PB02	xin1
62	96	140	PB00	xin32
32	48	70	PB31	xout0
	100	144	PB03	xout1
63	97	141	PB01	xout32

3.2.4 JTAG port connections

If the JTAG is enabled, the JTAG will take control over a number of pins, irrespectively of the I/O Controller configuration.

Table 3-4. JTAG pinout

QFN64/ TQFP64 pin	TQFP100 pin	LQFP144 pin	Pin name	JTAG pin
2	2	2	PA01	TDI
3	3	3	PA02	TDO
4	4	4	PA03	TMS
1	1	1	PA00	TCK

3.2.5 Nexus OCD AUX port connections

If the OCD trace system is enabled, the trace system will take control over a number of pins, irrespectively of the GPIO configuration. Three different OCD trace pin mappings are possible,

Table 3-7. Signal Description List

Signal Name	Function	Type	Active Level	Comments
RX_CLK	Receive Clock	Input		
RX_DV	Receive Data Valid	Input		
RX_ER	Receive Coding Error	Input		
SPEED	Speed	Output		
TXD[3:0]	Transmit Data	Output		
TX_CLK	Transmit Clock or Reference Clock	Input		
TX_EN	Transmit Enable	Output		
TX_ER	Transmit Coding Error	Output		
WOL	Wake-On-LAN	Output		
Peripheral Event Controller - PEVC				
PAD_EVT[15:0]	Event Input Pins	Input		
Power Manager - PM				
RESET_N	Reset Pin	Input	Low	
Pulse Width Modulator - PWM				
PWMH[3:0] PWML[3:0]	PWM Output Pins	Output		
EXT_FAULT[1:0]	PWM Fault Input Pins	Input		
Quadrature Decoder- QDEC0/QDEC1				
QEPA	QEPA quadrature input	Input		
QEPB	QEPB quadrature input	Input		
QEPI	Index input	Input		
System Controller Interface- SCIF				
XIN0, XIN1, XIN32	Crystal 0, 1, 32K Inputs	Analog		
XOUT0, XOUT1, XOUT32	Crystal 0, 1, 32K Output	Analog		
GCLK0 - GCLK1	Generic Clock Pins	Output		
Serial Peripheral Interface - SPI0, SPI1				
MISO	Master In Slave Out	I/O		
MOSI	Master Out Slave In	I/O		

Table 3-7. Signal Description List

Signal Name	Function	Type	Active Level	Comments
DP	USB Device Port Data +	Analog		
VBUS	USB VBUS Monitor and OTG Negotiation	Analog Input		
ID	ID Pin of the USB Bus	Input		
VBOF	USB VBUS On/off: bus power control port	output		

3.4 I/O Line Considerations

3.4.1 JTAG pins

The JTAG is enabled if TCK is low while the RESET_N pin is released. The TCK, TMS, and TDI pins have pull-up resistors when JTAG is enabled. The TCK pin always have pull-up enabled during reset. The TDO pin is an output, driven at VDDIO1, and has no pull-up resistor. The JTAG pins can be used as GPIO pins and muxed with peripherals when the JTAG is disabled. Please refer to [Section 3.2.4](#) for the JTAG port connections.

3.4.2 RESET_N pin

The RESET_N pin integrates a pull-up resistor to VDDIO1. 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.

The RESET_N pin is also used for the aWire debug protocol. When the pin is used for debugging, it must not be driven by external circuitry.

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 GPIO pins.

3.4.4 GPIO pins

All I/O lines integrate programmable pull-up and pull-down resistors. Most I/O lines integrate drive strength control, see [Table 3-1](#). Programming of this pull-up and pull-down resistor or this drive strength is performed independently for each I/O line through the GPIO Controllers.

After reset, I/O lines default as inputs with pull-up/pull-down resistors disabled. After reset, output drive strength is configured to the lowest value to reduce global EMI of the device.

When the I/O line is configured as analog function (ADC I/O, AC inputs, DAC I/O), the pull-up and pull-down resistors are automatically disabled.

single cycle. Load and store instructions have several different formats in order to reduce code size and speed up execution.

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

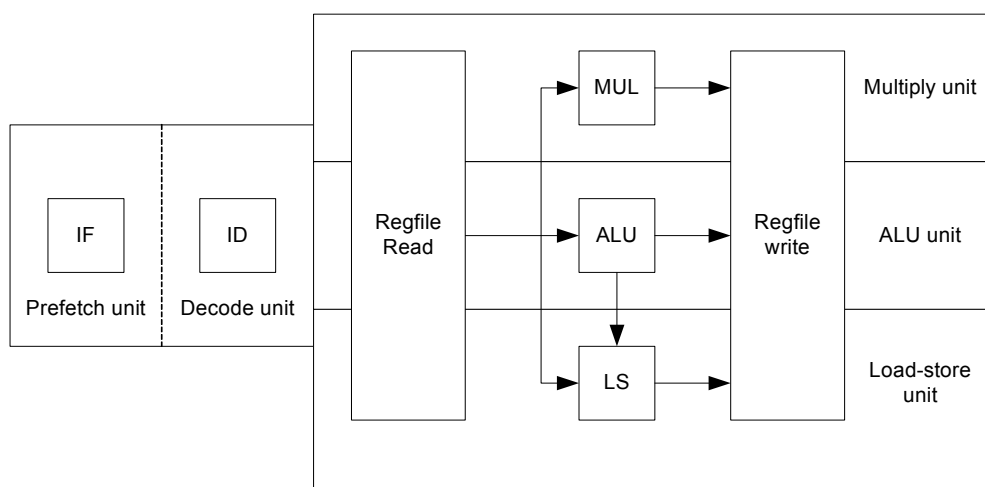
The AVR32UC CPU targets low- and medium-performance applications, and provides an advanced On-Chip Debug (OCD) system, no caches, and a Memory Protection Unit (MPU). A hardware Floating Point Unit (FPU) is also provided through the coprocessor instruction space. Java acceleration hardware is not implemented.

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 I/O controller ports. This local bus has to be enabled by writing a one to 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 CPU Local Bus section in the Memories chapter.

[Figure 4-1 on page 27](#) 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.

4.3.2.1 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.

4.3.2.2 Java Support

AVR32UC does not provide Java hardware acceleration.

4.3.2.3 Floating Point Support

A fused multiply-accumulate Floating Point Unit (FPU), performing a multiply and accumulate as a single operation with no intermediate rounding, thereby increasing precision is provided. The floating point hardware conforms to the requirements of the C standard, which is based on the IEEE 754 floating point standard.

4.3.2.4 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.

Table 4-3. System Registers (Continued)

Reg #	Address	Name	Function
90	360	MPUPSR2	MPU Privilege Select Register region 2
91	364	MPUPSR3	MPU Privilege Select Register region 3
92	368	MPUPSR4	MPU Privilege Select Register region 4
93	372	MPUPSR5	MPU Privilege Select Register region 5
94	376	MPUPSR6	MPU Privilege Select Register region 6
95	380	MPUPSR7	MPU Privilege Select Register region 7
96	384	MPUCRA	Unused in this version of AVR32UC
97	388	MPUCRB	Unused in this version of AVR32UC
98	392	MPUBRA	Unused in this version of AVR32UC
99	396	MPUBRB	Unused in this version of AVR32UC
100	400	MPUAPRA	MPU Access Permission Register A
101	404	MPUAPRB	MPU Access Permission Register B
102	408	MPUCR	MPU Control Register
103	412	SS_STATUS	Secure State Status Register
104	416	SS_ADRF	Secure State Address Flash Register
105	420	SS_ADRR	Secure State Address RAM Register
106	424	SS_ADR0	Secure State Address 0 Register
107	428	SS_ADR1	Secure State Address 1 Register
108	432	SS_SP_SYS	Secure State Stack Pointer System Register
109	436	SS_SP_APP	Secure State Stack Pointer Application Register
110	440	SS_RAR	Secure State Return Address Register
111	444	SS_RSR	Secure State Return Status Register
112-191	448-764	Reserved	Reserved for future use
192-255	768-1020	IMPL	IMPLEMENTATION DEFINED

4.5 Exceptions and Interrupts

In the AVR32 architecture, events are used as a common term for exceptions and interrupts. AVR32UC incorporates a powerful event handling scheme. The different event sources, like Illegal Op-code and interrupt requests, have different priority levels, ensuring a well-defined behavior when multiple events are received simultaneously. Additionally, pending events of a higher priority class may preempt handling of ongoing events of a lower priority class.

When an event occurs, the execution of the instruction stream is halted, and execution is passed to an event handler at an address specified in [Table 4-4 on page 38](#). Most of the handlers are placed sequentially in the code space starting at the address specified by EVBA, with four bytes between each handler. This gives ample space for a jump instruction to be placed there, jumping to the event routine itself. A few critical handlers have larger spacing between them, allowing the entire event routine to be placed directly at the address specified by the EVBA-relative offset generated by hardware. All interrupt sources have autovectored interrupt service routine (ISR) addresses. This allows the interrupt controller to directly specify the ISR address as an address

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 AVR32UC CPU uses unsegmented translation, as described in the AVR32 Architecture Manual. The 32-bit physical address space is mapped as follows:

Table 5-1. AT32UC3C Physical Memory Map

Device	Start Address	AT32UC3 Derivatives							
		C0512C	C1512C C2512C	C0256C	C1256C C2256C	C0128C	C1128C C2128C	C064C	C164C C264C
Embedded SRAM	0x0000_0000	64 KB	64 KB	64 KB	64 KB	32 KB	32 KB	16 KB	16 KB
Embedded Flash	0x8000_0000	512 KB	512 KB	256 KB	256 KB	128 KB	128 KB	64 KB	64 KB
SAU	0x9000_0000	1 KB	1 KB	1 KB	1 KB	1 KB	1 KB	1 KB	1 KB
HSB SRAM	0xA000_0000	4 KB	4 KB	4 KB	4 KB	4 KB	4 KB	4 KB	4 KB
EBI SRAM CS0	0xC000_0000	16 MB	-	16 MB	-	16 MB	-	16 MB	-
EBI SRAM CS2	0xC800_0000	16 MB	-	16 MB	-	16 MB	-	16 MB	-
EBI SRAM CS3	0xCC00_0000	16 MB	-	16 MB	-	16 MB	-	16 MB	-
EBI SRAM /SDRAM CS1	0xD000_0000	128 MB	-	128 MB	-	128 MB	-	128 MB	-
HSB-PB Bridge C	0xFFFD_0000	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB
HSB-PB Bridge B	0xFFFE_0000	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB
HSB-PB Bridge A	0xFFFF_0000	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB

Table 5-2. Flash Memory Parameters

Part Number	Flash Size (FLASH_PW)	Number of pages (FLASH_P)	Page size (FLASH_W)
AT32UC3C0512C AT32UC3C1512C AT32UC3C2512C	512 Kbytes	1024	128 words
AT32UC3C0256C AT32UC3C1256C AT32UC3C2256C	256 Kbytes	512	128 words
AT32UC3C0128C AT32UC3C1128C AT32UC3C2128C	128 Kbytes	256	128 words
AT32UC3C064C AT32UC3C164C AT32UC3C264C	64 Kbytes	128	128 words

5.3 Peripheral Address Map

Table 5-3. Peripheral Address Mapping

Address		Peripheral Name
0xFFFFD0000	PDCA	Peripheral DMA Controller - PDCA
0xFFFFD1000	MDMA	Memory DMA - MDMA
0xFFFFD1400	USART1	Universal Synchronous/Asynchronous Receiver/Transmitter - USART1
0xFFFFD1800	SPI0	Serial Peripheral Interface - SPI0
0xFFFFD1C00	CANIF	Control Area Network interface - CANIF
0xFFFFD2000	TC0	Timer/Counter - TC0
0xFFFFD2400	ADCIFA	ADC controller interface with Touch Screen functionality - ADCIFA
0xFFFFD2800	USART4	Universal Synchronous/Asynchronous Receiver/Transmitter - USART4
0xFFFFD2C00	TWIM2	Two-wire Master Interface - TWIM2
0xFFFFD3000	TWIS2	Two-wire Slave Interface - TWIS2

- Internal 3.3V regulator is off
- $T_A = 25^{\circ}\text{C}$
- I/Os are configured as inputs, with internal pull-up enabled.
- Oscillators
 - OSC0/1 (crystal oscillator) stopped
 - OSC32K (32KHz crystal oscillator) stopped
 - PLL0 running
 - PLL1 stopped
- Clocks
 - External clock on XIN0 as main clock source (10MHz)
 - CPU, HSB, and PBB clocks undivided
 - PBA, PBC clock divided by 4
 - All peripheral clocks running

Table 7-4. Power Consumption for Different Operating Modes

Mode	Conditions	Measured on	Consumption Typ	Unit
Active ⁽¹⁾	CPU running a recursive Fibonacci algorithm	Amp	512	$\mu\text{A}/\text{MHz}$
Idle ⁽¹⁾			258	
Frozen ⁽¹⁾			106	
Standby ⁽¹⁾			48	
Stop			73	μA
DeepStop			43	
Static	OSC32K and AST running		32	
	AST and OSC32K stopped		31	

Note: 1. These numbers are valid for the measured condition only and must not be extrapolated to other frequencies.

7.6.3 Phase Lock Loop (PLL0 and PLL1) Characteristics

Table 7-11. PLL Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{VCO}	Output frequency		80		240	MHz
f_{IN}	Input frequency		4		16	MHz
I_{PLL}	Current consumption	Active mode, $f_{VCO} = 80\text{MHz}$		250		μA
		Active mode, $f_{VCO} = 240\text{MHz}$		600		
$t_{STARTUP}$	Startup time, from enabling the PLL until the PLL is locked	Wide Bandwidth mode disabled		15		μs
		Wide Bandwidth mode enabled		45		

7.6.4 120MHz RC Oscillator (RC120M) Characteristics

Table 7-12. Internal 120MHz RC Oscillator Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{OUT}	Output frequency ⁽¹⁾		88	120	152	MHz
I_{RC120M}	Current consumption			1.85		mA
$t_{STARTUP}$	Startup time			3		μs

Note: 1. These values are based on simulation and characterization of other AVR microcontrollers manufactured in the same process technology. These values are not covered by test limits in production.

7.6.5 System RC Oscillator (RCSYS) Characteristics

Table 7-13. System RC Oscillator Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{OUT}	Output frequency	Calibrated at $T_A = 85^\circ\text{C}$	110	115.2	120	kHz
		$T_A = 25^\circ\text{C}$	105	109	115	
		$T_A = -40^\circ\text{C}$	100	104	108	

7.6.6 8MHz/1MHz RC Oscillator (RC8M) Characteristics

Table 7-14. 8MHz/1MHz RC Oscillator Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{OUT}	Output frequency	SCIF.RCCR8.FREQMODE = 0 ⁽¹⁾	7.6	8	8.4	MHz
		SCIF.RCCR8.FREQMODE = 1 ⁽¹⁾	0.955	1	1.045	
$t_{STARTUP}$	Startup time				20	μs

Notes: 1. Please refer to the SCIF chapter for details.

7.8.6 Analog to Digital Converter (ADC) and sample and hold (S/H) Characteristics

Table 7-27. ADC and S/H characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Units
f_{ADC}	ADC clock frequency	12-bit resolution mode, $V_{VDDANA} = 3V$			1.2	MHz
		10-bit resolution mode, $V_{VDDANA} = 3V$			1.6	
		8-bit resolution mode, $V_{VDDANA} = 3V$			2.2	
		12-bit resolution mode, $V_{VDDANA} = 4.5V$			1.5	
		10-bit resolution mode, $V_{VDDANA} = 4.5V$			2	
		8-bit resolution mode, $V_{VDDANA} = 4.5V$			2.4	
$t_{STARTUP}$	Startup time	ADC cold start-up			1	ms
		ADC hot start-up			24	ADC clock cycles
t_{CONV}	Conversion time (latency)	(ADCIFA.SEQCFGn.SRES)/2 + 2, ADCIFA.CFG.SHD = 1	6		8	ADC clock cycles
		(ADCIFA.SEQCFGn.SRES)/2 + 3, ADCIFA.CFG.SHD = 0	7		9	
	Throughput rate	12-bit resolution, ADC clock = 1.2 MHz, $V_{VDDANA} = 3V$			1.2	MSPS
		10-bit resolution, ADC clock = 1.6 MHz, $V_{VDDANA} = 3V$			1.6	
		12-bit resolution, ADC clock = 1.5 MHz, $V_{VDDANA} = 4.5V$			1.5	
		10-bit resolution, ADC clock = 2 MHz, $V_{VDDANA} = 4.5V$			2	

Table 7-28. ADC Reference Voltage

Symbol	Parameter	Conditions	Min	Typ	Max	Units
$V_{ADCREFO}$	ADCREFO input voltage range	5V Range	1		3.5	V
		3V Range	1		$V_{VDDANA} - 0.7$	
$V_{ADCREF1}$	ADCREF1 input voltage range	5V Range	1		3.5	V
		3V Range	1		$V_{VDDANA} - 0.7$	
$V_{ADCREFP}$	ADCREFP input voltage	5V Range - Voltage reference applied on ADCREFP	1		3.5	V
		3V Range - Voltage reference applied on ADCREFP	1		$V_{VDDANA} - 0.7$	
$V_{ADCREFN}$	ADCREFN input voltage	Voltage reference applied on ADCREFN		GNDANA		V
	Internal 1V reference			1.0		V
	Internal 0.6*VDDANA reference			$0.6 \cdot V_{VDDANA}$		V

Table 7-31. ADC Transfer Characteristics (Continued) 12-bit Resolution Mode⁽¹⁾

Symbol	Parameter	Conditions	Min	Typ	Max	Units
RES	Resolution	Differential mode,			12	Bit
INL	Integral Non-Linearity	$V_{VDDANA} = 5V$,			4	LSB
DNL	Differential Non-Linearity	$V_{ADCREFO} = 3V$,			3	LSB
	Offset error	ADCFIA.SEQCFGn.SRES = 0	-15		15	mV
	Gain error	($F_{adc} = 1.5MHz$)	-25		25	mV

Note: 1. The measures are done without any I/O activity on VDDANA/GNDANA power domain.

Table 7-32. ADC Transfer Characteristics 10-bit Resolution Mode⁽¹⁾

Symbol	Parameter	Conditions	Min	Typ	Max	Units
RES	Resolution	Differential mode,			10	Bit
INL	Integral Non-Linearity	$V_{VDDANA} = 3V$,			1.25	LSB
DNL	Differential Non-Linearity	$V_{ADCREFO} = 1V$,			1	LSB
	Offset error	ADCFIA.SEQCFGn.SRES = 1	-10		10	mV
	Gain error	($F_{adc} = 1.5MHz$)	-20		20	mV
RES	Resolution	Differential mode,			10	Bit
INL	Integral Non-Linearity	$V_{VDDANA} = 5V$,			1.25	LSB
DNL	Differential Non-Linearity	$V_{ADCREFO} = 3V$,			1	LSB
	Offset error	ADCFIA.SEQCFGn.SRES = 1	-15		15	mV
	Gain error	($F_{adc} = 1.5MHz$)	-20		20	mV

Note: 1. The measures are done without any I/O activity on VDDANA/GNDANA power domain.

Table 7-33. ADC Transfer Characteristics 8-bit Resolution Mode⁽¹⁾

Symbol	Parameter	Conditions	Min	Typ	Max	Units
RES	Resolution	Differential mode,			8	Bit
INL	Integral Non-Linearity	$V_{VDDANA} = 3V$,			0.3	LSB
DNL	Differential Non-Linearity	$V_{ADCREFO} = 1V$,			0.25	LSB
	Offset error	ADCFIA.SEQCFGn.SRES = 2	-10		10	mV
	Gain error	($F_{adc} = 1.5MHz$)	-20		20	mV
RES	Resolution	Differential mode,			8	Bit
INL	Integral Non-Linearity	$V_{VDDANA} = 5V$,			0.2	LSB
DNL	Differential Non-Linearity	$V_{ADCREFO} = 3V$,			0.2	LSB
	Offset error	ADCFIA.SEQCFGn.SRES = 2	-20		20	mV
	Gain error	($F_{adc} = 1.5MHz$)	-20		20	mV

Note: 1. The measures are done without any I/O activity on VDDANA/GNDANA power domain.

7.8.8 Analog Comparator Characteristics

Table 7-41. Analog Comparator Characteristics⁽¹⁾

Symbol	Parameter	Conditions	Min	Typ	Max	Units
	Positive input voltage range		0		V_{VDDANA}	V
	Negative input voltage range		0		V_{VDDANA}	V
V_{OFFSET}	Offset	No hysteresis, Low Power mode	-29		29	mV
		No hysteresis, High Speed mode	-16		16	mV
V_{HYST}	Hysteresis	Low hysteresis, Low Power mode	7		44	mV
		Low hysteresis, High Speed mode	5		34	
		High hysteresis, Low Power mode	16		102	mV
		High hysteresis, High Speed mode	12		69	
t_{DELAY}	Propagation delay	Low Power mode			2.9	us
		High Speed mode			0.096	
$t_{STARTUP}$	Start-up time				20	μs

Note: 1. The measures are done without any I/O activity on VDDANA/GNDANA power domain.

Table 7-42. VDDANA scaled reference

Symbol	Parameter	Min	Typ	Max	Units
SCF	ACIFA.SCFi.SCF range	0		32	
V_{VDDANA} scaled			$(64 - SCF) * V_{VDDANA} / 65$		V
	V_{VDDANA} voltage accuracy			3.2	%

7.8.9 USB Transceiver Characteristics

7.8.9.1 Electrical Characteristics

Table 7-43. Electrical Parameters

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
R_{EXT}	Recommended external USB series resistor	In series with each USB pin with $\pm 5\%$		39		Ω

The USB on-chip buffers comply with the Universal Serial Bus (USB) v2.0 standard. All AC parameters related to these buffers can be found within the USB 2.0 electrical specifications.

7.9.3 USART in SPI Mode Timing

7.9.3.1 Master mode

Figure 7-6. USART in SPI Master Mode With (CPOL= CPHA= 0) or (CPOL= CPHA= 1)

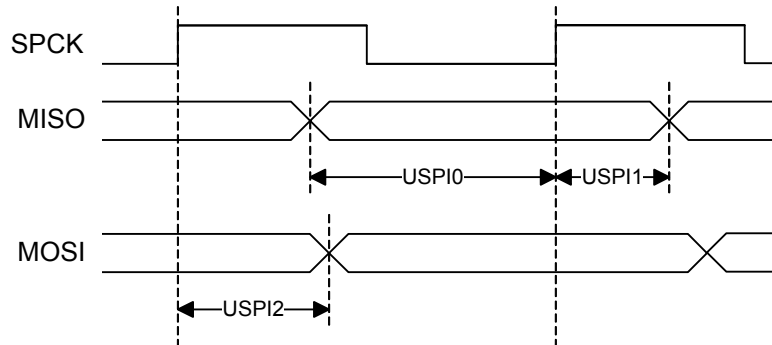


Figure 7-7. USART in SPI Master Mode With (CPOL= 0 and CPHA= 1) or (CPOL= 1 and CPHA= 0)

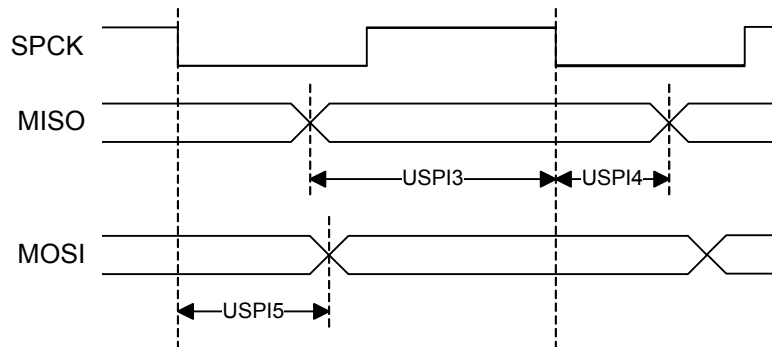


Table 7-46. USART in SPI Mode Timing, Master Mode⁽¹⁾

Symbol	Parameter	Conditions	Min	Max	Units
USPI0	MISO setup time before SPCK rises	external capacitor = 40pF	26+ t _{SAMPLE} ⁽²⁾		ns
USPI1	MISO hold time after SPCK rises		0		ns
USPI2	SPCK rising to MOSI delay			11	ns
USPI3	MISO setup time before SPCK falls		26+ t _{SAMPLE} ⁽²⁾		ns
USPI4	MISO hold time after SPCK falls		0		ns
USPI5	SPCK falling to MOSI delay			11.5	ns

Note: 1. These values are based on simulation and characterization of other AVR microcontrollers manufactured in the same process technology. These values are not covered by test limits in production.

2. Where: $t_{SAMPLE} = t_{SPCK} - \left(\left\lfloor \frac{t_{SPCK}}{2 \times t_{CLKUSART}} \right\rfloor \times \frac{1}{2} \right) \times t_{CLKUSART}$

Figure 7-9. USART in SPI Slave Mode With (CPOL= CPHA= 0) or (CPOL= CPHA= 1)

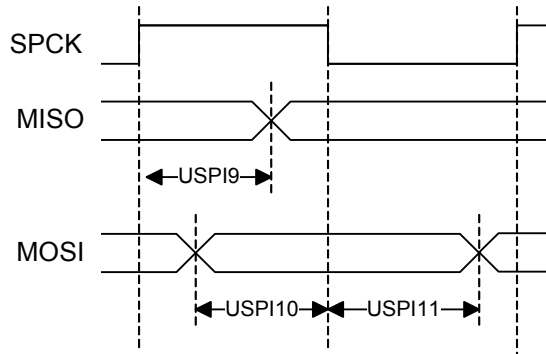


Figure 7-10. USART in SPI Slave Mode NPCS Timing

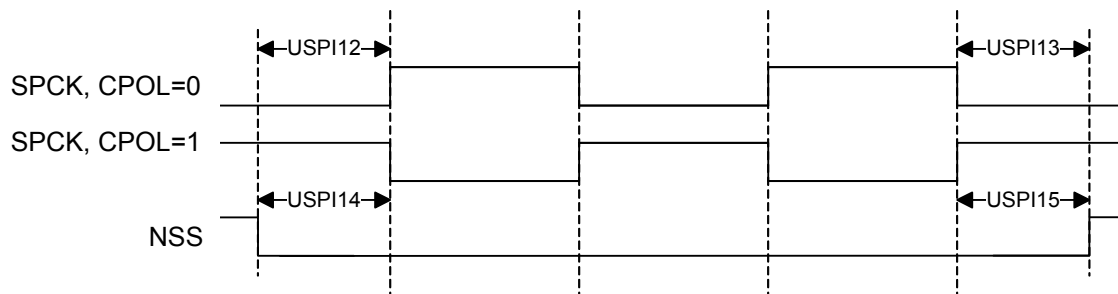


Table 7-47. USART in SPI mode Timing, Slave Mode⁽¹⁾

Symbol	Parameter	Conditions	Min	Max	Units
USPI6	SPCK falling to MISO delay	external capacitor = 40pF		27	ns
USPI7	MOSI setup time before SPCK rises		$t_{SAMPLE}^{(2)} + t_{CLK_USART}$		ns
USPI8	MOSI hold time after SPCK rises		0		ns
USPI9	SPCK rising to MISO delay			28	ns
USPI10	MOSI setup time before SPCK falls		$t_{SAMPLE}^{(2)} + t_{CLK_USART}$		ns
USPI11	MOSI hold time after SPCK falls		0		ns
USPI12	NSS setup time before SPCK rises		33		ns
USPI13	NSS hold time after SPCK falls		0		ns
USPI14	NSS setup time before SPCK falls		33		ns
USPI15	NSS hold time after SPCK rises		0		ns

Note: 1. These values are based on simulation and characterization of other AVR microcontrollers manufactured in the same process technology. These values are not covered by test limits in production.

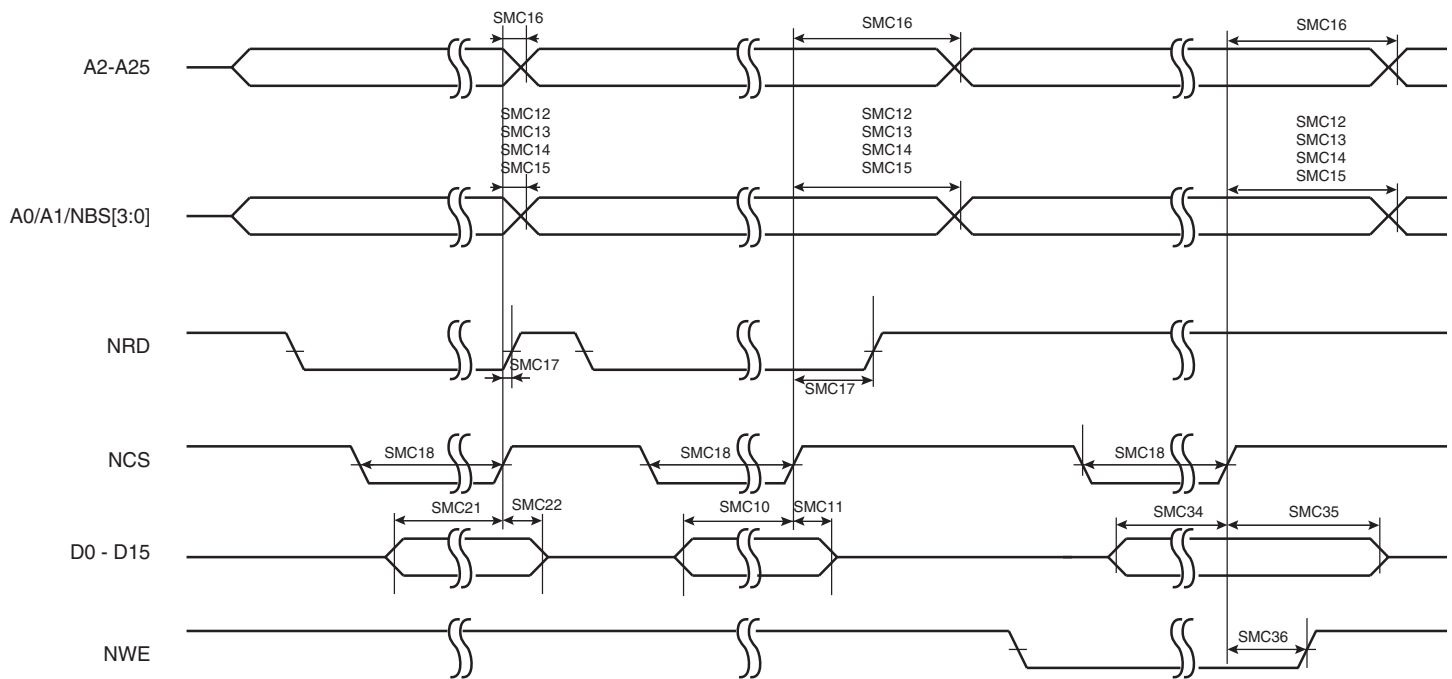
2. Where: $t_{SAMPLE} = t_{SPCK} - \left(\left\lfloor \frac{t_{SPCK}}{2 \times t_{CLK_USART}} \right\rfloor + \frac{1}{2} \right) \times t_{CLK_USART}$

Table 7-56. SMC Write Signals with No Hold Settings (NWE Controlled only)⁽¹⁾

Symbol	Parameter	Conditions	Min	Units
SMC ₃₇	NWE rising to A2-A25 valid	$V_{DD} = 3.0V$, drive strength of the pads set to the lowest, external capacitor = 40pF	8.7	ns
SMC ₃₈	NWE rising to NBS0/A0 valid		7.6	
SMC ₄₀	NWE rising to A1/NBS2 change		8.7	
SMC ₄₂	NWE rising to NCS rising		8.4	
SMC ₄₃	Data Out valid before NWE rising		$(nwe \text{ pulse length} - 1) * tc_{PSMC} - 1.2$	
SMC ₄₄	Data Out valid after NWE rising		8.4	
SMC ₄₅	NWE pulse width		$nwe \text{ pulse length} * tc_{PSMC} - 0$	

Note: 1. These values are based on simulation and characterization of other AVR microcontrollers manufactured in the same process technology. These values are not covered by test limits in production.

Figure 7-17. SMC Signals for NCS Controlled Accesses



10.2.12 WDT

1 Clearing the Watchdog Timer (WDT) counter in second half of timeout period will issue a Watchdog reset

If the WDT counter is cleared in the second half of the timeout period, the WDT will immediately issue a Watchdog reset.

Fix/Workaround

Use twice as long timeout period as needed and clear the WDT counter within the first half of the timeout period. If the WDT counter is cleared after the first half of the timeout period, you will get a Watchdog reset immediately. If the WDT counter is not cleared at all, the time before the reset will be twice as long as needed.

2 WDT Control Register does not have synchronization feedback

When writing to the Timeout Prescale Select (PSEL), Time Ban Prescale Select (TBAN), Enable (EN), or WDT Mode (MODE) fields of the WDT Control Register (CTRL), a synchronizer is started to propagate the values to the WDT clock domain. This synchronization takes a finite amount of time, but only the status of the synchronization of the EN bit is reflected back to the user. Writing to the synchronized fields during synchronization can lead to undefined behavior.

Fix/Workaround

-When writing to the affected fields, the user must ensure a wait corresponding to 2 clock cycles of both the WDT peripheral bus clock and the selected WDT clock source.

-When doing writes that changes the EN bit, the EN bit can be read back until it reflects the written value.