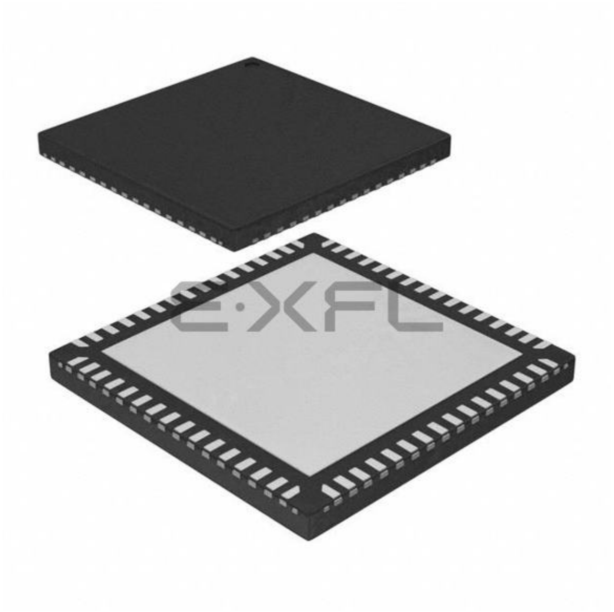
Microchip Technology - AT32UC3C2512C-Z2UT Datasheet



Details	
Product Status	Active
Core Processor	AVR
Core Size	32-Bit Single-Core
Speed	66MHz
Connectivity	CANbus, Ethernet, I²C, IrDA, LINbus, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	45
Program Memory Size	512KB (512K 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/at32uc3c2512c-z2ut

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
		124	PD15	111	VDDIO3	x1/x2	TC0 - A0	USART3 - TXD	EBI - ADDR[11]			
		125	PD16	112	VDDIO3	x1/x2	TC0 - B0	USART3 - RXD	EBI - ADDR[12]			
		126	PD17	113	VDDIO3	x1/x2	TC0 - A1	USART3 - CTS	EBI - ADDR[13]	USART3 - CLK		
		127	PD18	114	VDDIO3	x1/x2	TC0 - B1	USART3 - RTS	EBI - ADDR[14]			
		128	PD19	115	VDDIO3	x1/x2	TC0 - A2		EBI - ADDR[15]			
		129	PD20	116	VDDIO3	x1/x2	TC0 - B2		EBI - ADDR[16]			
57	88	130	PD21	117	VDDIO3	x1/x2	USART3 - TXD	EIC - EXTINT[0]	EBI - ADDR[17]	QDEC1 - QEPI		
	89	131	PD22	118	VDDIO1	x1/x2	USART3 - RXD	TC0 - A2	EBI - ADDR[18]	SCIF - GCLK[0]		
	90	132	PD23	119	VDDIO1	x1/x2	USART3 - CTS	USART3 - CLK	EBI - ADDR[19]	QDEC1 - QEPA		
	91	133	PD24	120	VDDIO1	x1/x2	USART3 - RTS	EIC - EXTINT[8]	EBI - NWE1	QDEC1 - QEPB		
		134	PD25	121	VDDIO1	x1/x2	TC0 - CLK0	USBC - ID	EBI - NWE0		USART4 - CLK	
		135	PD26	122	VDDIO1	x1/x2	TC0 - CLK1	USBC - VBOF	EBI - NRD			
58	92	136	PD27	123	VDDIO1	x1/x2	USART0 - TXD	CANIF - RXLINE[0]	EBI - NCS[1]	TC0 - A0	MACB - RX_ER	
59	93	137	PD28	124	VDDIO1	x1/x2	USART0 - RXD	CANIF - TXLINE[0]	EBI - NCS[2]	TC0 - B0	MACB - RX_DV	
60	94	138	PD29	125	VDDIO1	x1/x2	USART0 - CTS	EIC - EXTINT[6]	USART0 - CLK	TC0 - CLK0	MACB - TX_CLK	
61	95	139	PD30	126	VDDIO1	x1/x2	USART0 - RTS	EIC - EXTINT[3]	EBI - NWAIT	TC0 - A1	MACB - TX_EN	

Note: 1. Refer to "Electrical Characteristics" on page 50 for a description of the electrical properties of the pin types used. See Section 3.3 for a description of the various peripheral signals.

3.2.2 Peripheral Functions

Each GPIO line can be assigned to one of several peripheral functions. The following table describes how the various peripheral functions are selected. The last listed function has priority in case multiple functions are enabled on the same pin.

Table 3-2. Peripheral Functions

Function	Description
GPIO Controller Function multiplexing	GPIO and GPIO peripheral selection A to F
Nexus OCD AUX port connections	OCD trace system

depending on the configuration of the OCD AXS register. For details, see the AVR32UC Technical Reference Manual.

Table 3-5. Nexus OCD AUX port connections

Pin	AXS=0	AXS=1	AXS=2
EVTI_N	PA08	PB19	PA10
MDO[5]	PC05	PC31	PB06
MDO[4]	PC04	PC12	PB15
MDO[3]	PA23	PC11	PB14
MDO[2]	PA22	PB23	PA27
MDO[1]	PA19	PB22	PA26
MDO[0]	PA09	PB20	PA19
EVTO_N	PD29	PD29	PD29
MCKO	PD13	PB21	PB26
MSEO[1]	PD30	PD08	PB25
MSEO[0]	PD14	PD07	PB18

3.2.6 Other Functions

The functions listed in Table 3-6 are not mapped to the normal GPIO functions. The aWire DATA pin will only be active after the aWire is enabled. The aWire DATAOUT pin will only be active after the aWire is enabled and the 2_PIN_MODE command has been sent.

Table 3-6. Other Functions

QFN64/ TQFP64 pin	TQFP100 pin	LQFP144 pin	Pad	Oscillator pin
64	98	142	RESET_N	aWire DATA
3	3	3	PA02	aWire DATAOUT

3.3 Signals Description

The following table give details on the signal name classified by peripherals.

Table 3-7. Signal Description List

Signal Name	Function	Type	Active Level	Comments
Power				
VDDIO1 VDDIO2 VDDIO3	I/O Power Supply	Power Input		4.5V to 5.5V or 3.0V to 3.6 V
VDDANA	Analog Power Supply	Power Input		4.5V to 5.5V or 3.0V to 3.6 V

Table 3-7. Signal Description List

Signal Name	Function	Type	Active Level	Comments
SDCK	SDRAM Clock	Output		
SDCKE	SDRAM Clock Enable	Output		
SDWE	SDRAM Write Enable	Output	Low	
External Interrupt Controller - EIC				
EXTINT[8:1]	External Interrupt Pins	Input		
NMI_N = EXTINT[0]	Non-Maskable Interrupt Pin	Input	Low	
General Purpose Input/Output - GPIOA, GPIOB, GPIOC, GPIOD				
PA[29:19] - PA[16:0]	Parallel I/O Controller GPIOA	I/O		
PB[31:0]	Parallel I/O Controller GPIOB	I/O		
PC[31:0]	Parallel I/O Controller GPIOC	I/O		
PD[30:0]	Parallel I/O Controller GPIOD	I/O		
Inter-IC Sound (I2S) Controller - IISC				
IMCK	I2S Master Clock	Output		
ISCK	I2S Serial Clock	I/O		
ISDI	I2S Serial Data In	Input		
ISDO	I2S Serial Data Out	Output		
IWS	I2S Word Select	I/O		
JTAG				
TCK	Test Clock	Input		
TDI	Test Data In	Input		
TDO	Test Data Out	Output		
TMS	Test Mode Select	Input		
Ethernet MAC - MACB				
COL	Collision Detect	Input		
CRS	Carrier Sense and Data Valid	Input		
MDC	Management Data Clock	Output		
MDIO	Management Data Input/Output	I/O		
RXD[3:0]	Receive Data	Input		

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 Negociation	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

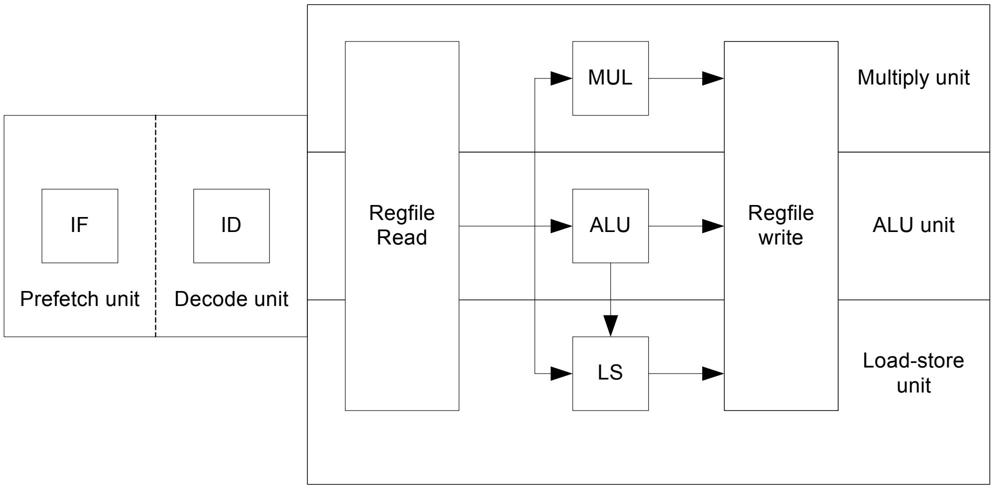
The AVR32UC CPU targets low- and medium-performance applications, and provides an advanced On-Chip Debug (OCD) system, no caches, and a Memory Protection Unit (MPU). A hardware Floating Point Unit (FPU) is also provided through the coprocessor instruction space. Java acceleration hardware is not implemented.

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), performaing a multiply and accumulate as a single operation with no intermediate rounding, therby 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.

than the oldest instruction. An instruction B is younger than an instruction A if it was sent down the pipeline later than A.

The addresses and priority of simultaneous events are shown in Table 4-4 on page 38. Some of the exceptions are unused in AVR32UC since it has no MMU, coprocessor interface, or floating-point unit.

6. Supply and Startup Considerations

6.1 Supply Considerations

6.1.1 Power Supplies

The AT32UC3C has several types of power supply pins:

- **VDDIO pins (VDDIO1, VDDIO2, VDDIO3): Power I/O lines. Two voltage ranges are available: 5V or 3.3V nominal. The VDDIO pins should be connected together.**
- **VDDANA: Powers the Analog part of the device (Analog I/Os, ADC, ACs, DACs). 2 voltage ranges available: 5V or 3.3V nominal.**
- **VDDIN_5: Input voltage for the 1.8V and 3.3V regulators. Two Voltage ranges are available: 5V or 3.3V nominal.**
- **VDDIN_33:**
  - **USB I/O power supply**
  - **if the device is 3.3V powered: Input voltage, voltage is 3.3V nominal.**
  - **if the device is 5V powered: stabilization for the 3.3V voltage regulator, requires external capacitors**
- **VDDCORE: Stabilization for the 1.8V voltage regulator, requires external capacitors.**
- **GNDCORE: Ground pins for the voltage regulators and the core.**
- **GNDANA: Ground pin for Analog part of the design**
- **GNDPLL: Ground pin for the PLLs**
- **GNDIO pins (GNDIO1, GNDIO2, GNDIO3): Ground pins for the I/O lines. The GNDIO pins should be connected together.**

See ”Electrical Characteristics” on page 50 for power consumption on the various supply pins.

For decoupling recommendations for the different power supplies, please refer to the schematic checklist.

6.1.2 Voltage Regulators

The AT32UC3C embeds two voltage regulators:

- One 1.8V internal regulator that converts from VDDIN_5 to 1.8V. The regulator supplies the output voltage on VDDCORE.
- One 3.3V internal regulator that converts from VDDIN_5 to 3.3V. The regulator supplies the USB pads on VDDIN_33. If the USB is not used or if VDDIN_5 is within the 3V range, the 3.3V regulator can be disabled through the VREG33CTL field of the VREGCTRL SCIF register.

6.1.3 Regulators Connection

The AT32UC3C supports two power supply configurations.

- 5V single supply mode
- 3.3V single supply mode

6.1.3.1 5V Single Supply Mode

In 5V single supply mode, the 1.8V internal regulator is connected to the 5V source (VDDIN_5 pin) and its output feeds VDDCORE.

  - PLL1 stopped
- Clocks
  - External clock on XIN0 as main clock source.
  - CPU, HSB, and PB clocks undivided

Consumption active is the added current consumption when the module clock is turned on and when the module is doing a typical set of operations.

Table 7-5. Typical Current Consumption by Peripheral(2)

Peripheral	**Typ Consumption Active**	**Unit**
ACIFA(1)	3	µA/MHz
ADCIFA(1)	7	
AST	3	
CANIF	25	
DACIFB(1)	3	
EBI	23	
EIC	0.5	
FREQM	0.5	
GPIO	37	
INTC	3	
MDMA	4	
PDCA	24	
PEVC	15	
PWM	40	
QDEC	3	
SAU	3	
SDRAMC	2	
SMC	9	
SPI	5	
TC	8	
TWIM	2	
TWIS	2	
USART	10	
USBC	5	
WDT	2	

Notes:
1. Includes the current consumption on VDDANA.
2. These numbers are valid for the measured condition only and must not be extrapolated to other frequencies.

7.8.4 3.3V Brown Out Detector (BOD33) Characteristics

The values in Table 7-23 describe the values of the BOD33.LEVEL field in the SCIF module.

Table 7-23. BOD33.LEVEL Values

BOD33.LEVEL Value	Parameter	Min	Max	Units
17		2.21	2.55	V
22		2.30	2.64	
27		2.39	2.74	
31	threshold at power-up sequence	2.46	2.82	
33		2.50	2.86	
39		2.60	2.98	
44		2.69	3.08	
49		2.78	3.18	
53		2.85	3.27	
60		2.98	3.41	

7.8.5 5V Brown Out Detector (BOD50) Characteristics

The values in Table 7-25 describe the values of the BOD50.LEVEL field in the SCIF module.

Table 7-25. BOD50.LEVEL Values

BOD50.LEVEL Value	Parameter	Min	Max	Units
16		3.20	3.65	V
25		3.42	3.92	
35		3.68	4.22	
44		3.91	4.48	
53		4.15	4.74	
61		4.36	4.97	

7.9 Timing Characteristics

7.9.1 Startup, Reset, and Wake-up Timing

The startup, reset, and wake-up timings are calculated using the following formula:

$$t = t_{CONST} + N_{CPU} \times t_{CPU}$$

Where t_{CONST} and N_{CPU} are found in Table 7-44. t_{CONST} is the delay relative to RCSYS, t_{CPU} is the period of the CPU clock. If another clock source than RCSYS is selected as CPU clock the startup time of the oscillator, $t_{OSCSTART}$, must be added to the wake-up time in the stop, deepstop, and static sleep modes. Please refer to the source for the CPU clock in the "Oscillator Characteristics" on page 57 for more details about oscillator startup times.

Table 7-44. Maximum Reset and Wake-up Timing

Parameter		Measuring	Max t_{CONST} (in µs)	Max N_{CPU}
Startup time from power-up, using regulator		VDDIN_5 rising (10 mV/ms) Time from V_{VDDIN_5}=0 to the first instruction entering the decode stage of CPU. VDDCORE is supplied by the internal regulator.	2600	0
Startup time from reset release		Time from releasing a reset source (except POR, BOD18, and BOD33) to the first instruction entering the decode stage of CPU.	1240	0
Wake-up	Idle	From wake-up event to the first instruction entering the decode stage of the CPU.	0	19
	Frozen		268	209
	Standby		268	209
	Stop		268+ $t_{OSCSTART}$	212
	Deepstop		268+ $t_{OSCSTART}$	212
	Static		268+ $t_{OSCSTART}$	212

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

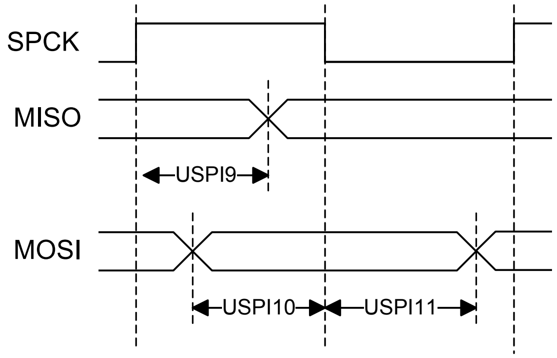


Figure 7-10. USART in SPI Slave Mode NPCS Timing

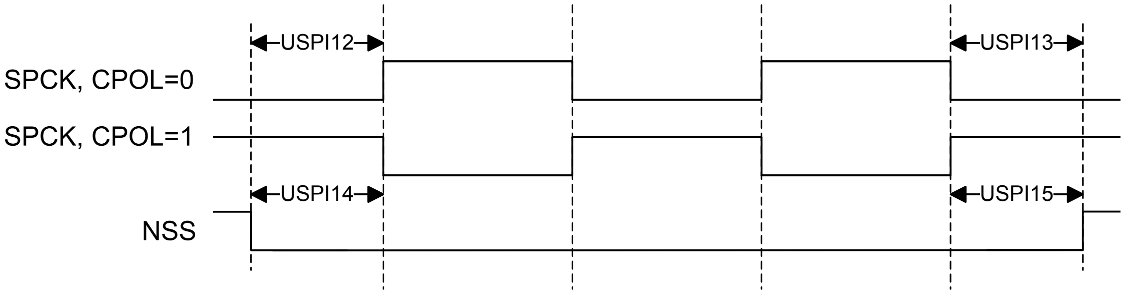


Table 7-47. USART in SPI mode Timing, Slave Mode(1)

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

7.9.4.2 Slave mode

Figure 7-13. SPI Slave Mode With (CPOL= 0 and NCPHA= 1) or (CPOL= 1 and NCPHA= 0)

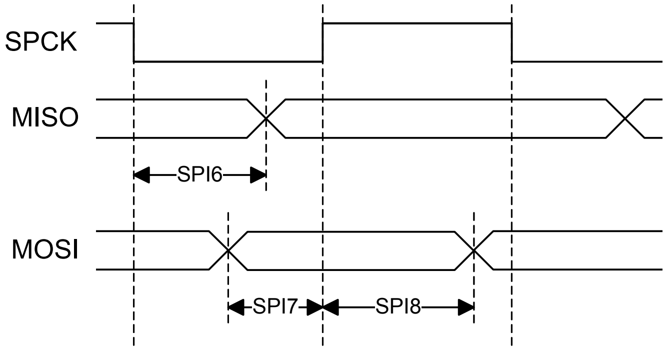


Figure 7-14. SPI Slave Mode With (CPOL= NCPHA= 0) or (CPOL= NCPHA= 1)

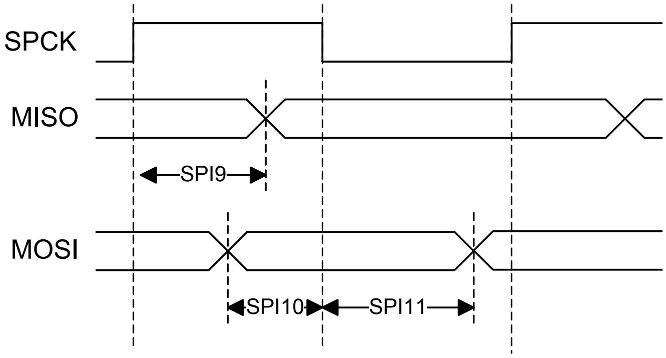
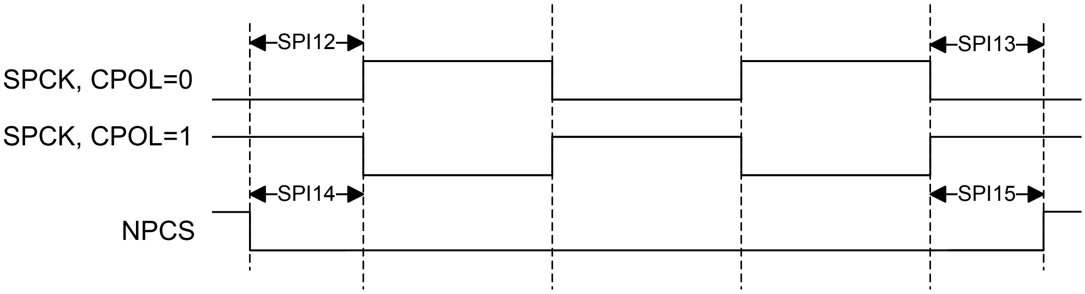


Figure 7-15. SPI Slave Mode NPCS Timing



7.9.6 JTAG Timing

Figure 7-16. JTAG Interface Signals

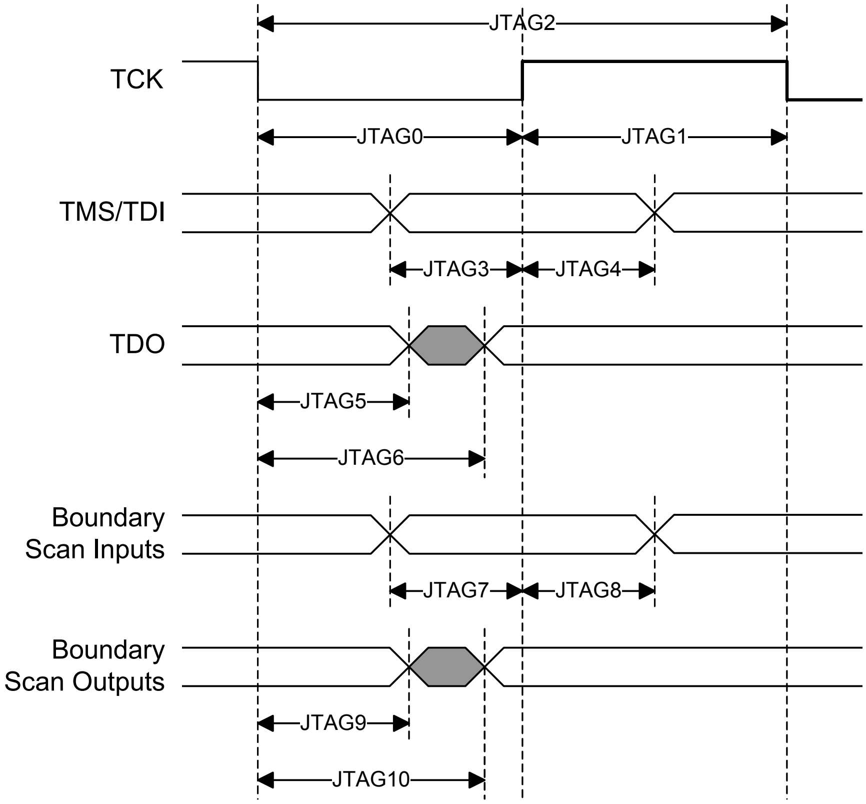


Table 7-51. JTAG Timings[1]

Symbol	Parameter	Conditions	Min	Max	Units
JTAG0	TCK Low Half-period	external capacitor = 40pF	21.5		ns
JTAG1	TCK High Half-period		8.5		ns
JTAG2	TCK Period		29		ns
JTAG3	TDI, TMS Setup before TCK High		6.5		ns
JTAG4	TDI, TMS Hold after TCK High		0		ns
JTAG5	TDO Hold Time		12.5		ns
JTAG6	TCK Low to TDO Valid			21.5	ns
JTAG7	Boundary Scan Inputs Setup Time		0		ns
JTAG8	Boundary Scan Inputs Hold Time		4.5		ns
JTAG9	Boundary Scan Outputs Hold Time		11		ns
JTAG10	TCK to Boundary Scan Outputs Valid			18	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.

Table 7-54. SMC Read Signals with no Hold Settings(1)

Symbol	Parameter	Conditions	Min	Units
NRD Controlled (READ_MODE = 1)				
SMC_{19}	Data setup before NRD high	V_{VDD} = 3.0V, drive strength of the pads set to the lowest, external capacitor = 40pF	32.5	ns
SMC_{20}	Data hold after NRD high		0	
NRD Controlled (READ_MODE = 0)				
SMC_{21}	Data setup before NCS high	V_{VDD} = 3.0V, drive strength of the pads set to the lowest, external capacitor = 40pF	28.5	ns
SMC_{22}	Data hold after NCS high		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.

Table 7-55. SMC Write Signals with Hold Settings(1)

Symbol	Parameter	Conditions	Min	Units
NRD Controlled (READ_MODE = 1)				
SMC_{23}	Data Out valid before NWE high	V_{VDD} = 3.0V, drive strength of the pads set to the lowest, external capacitor = 40pF	(nwe pulse length - 1) * t_{CPSMC} - 1.4	ns
SMC_{24}	Data Out valid after NWE high(2)		nwe pulse length * t_{CPSMC} - 4.7	
SMC_{25}	NWE high to NBS0/A0 change(2)		nwe pulse length * t_{CPSMC} - 2.7	
SMC_{29}	NWE high to NBS2/A1 change(2)		nwe pulse length * t_{CPSMC} - 0.7	
SMC_{31}	NWE high to A2 - A25 change(2)		nwe pulse length * t_{CPSMC} - 6.8	
SMC_{32}	NWE high to NCS inactive(2)		(nwe hold pulse - ncs wr hold length) * t_{CPSMC} - 2.5	
SMC_{33}	NWE pulse width		nwe pulse length * t_{CPSMC} - 0.2	
NRD Controlled (READ_MODE = 0)				
SMC_{34}	Data Out valid before NCS high	V_{VDD} = 3.0V, drive strength of the pads set to the lowest, external capacitor = 40pF	(ncs wr pulse length - 1) * t_{CPSMC} - 2.2	ns
SMC_{35}	Data Out valid after NCS high(2)		ncs wr hold length * t_{CPSMC} - 5.1	
SMC_{36}	NCS high to NWE inactive(2)		(ncs wr hold length - nwe hold length) * t_{CPSMC} - 2	

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

Figure 7-19. SDRAMC Signals relative to SDCK.

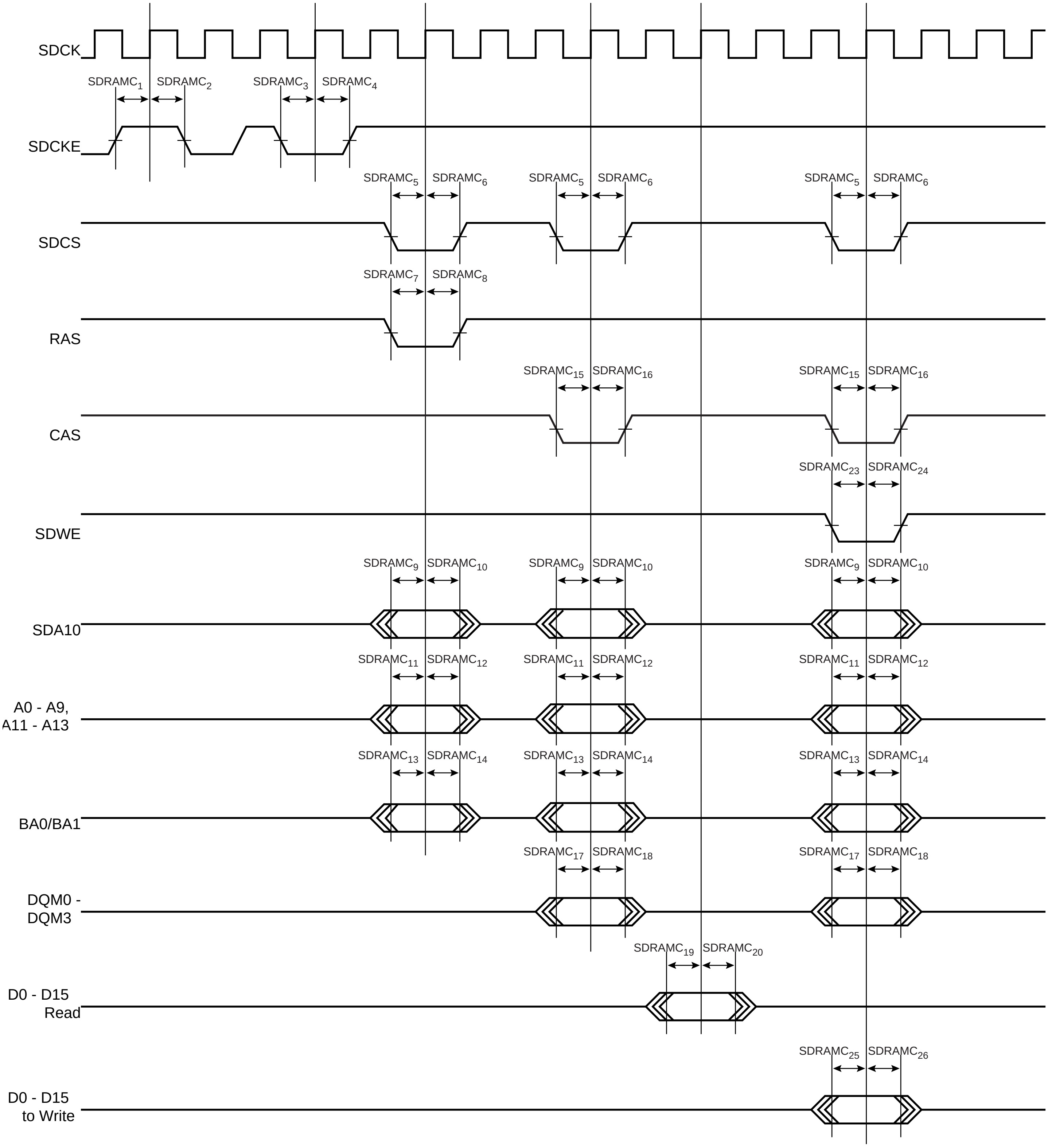
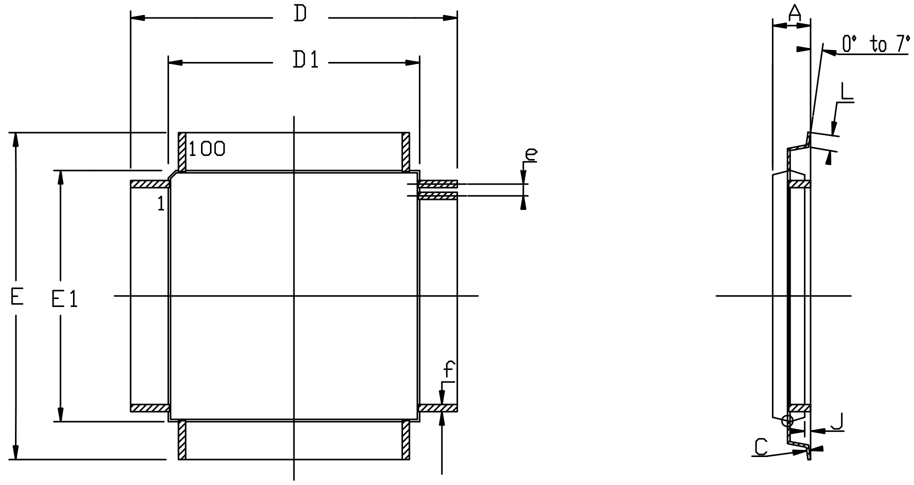


Figure 8-3. TQFP-100 package drawing



COMMON DIMENSIONS IN MM

SYMBOL	Min	Max	NOTES
A	----	1.20	
A1	0.95	1.05	
C	0.09	0.20	
D	16.00 BSC		
D1	14.00 BSC		
E	16.00 BSC		
E1	14.00 BSC		
J	0.05	0.15	
L	0.45	0.75	
e	0.50 BSC		
f	0.17	0.27	

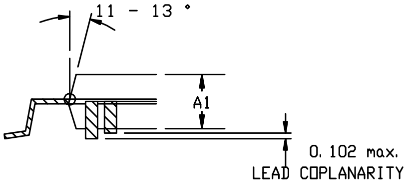


Table 8-8. Device and Package Maximum Weight

500	mg

Table 8-9. Package Characteristics

Moisture Sensitivity Level	Jdec J-STD0-20D - MSL 3

Table 8-10. Package Reference

JEDEC Drawing Reference	MS-026
JESD97 Classification	E3

10.2 rev D

10.2.1 ADCIFA

1 **ADCREFP/ADCREFN can not be selected as an external ADC reference by setting the ADCIFA.CFG.EXREF bit to one**

Fix/Workaround

A voltage reference can be applied on ADCREFP/ADCREFN pins if the ADCIFA.CFG.EXREF bit is set to zero, the ADCIFA.CFG.RS bit is set to zero and the voltage reference applied on ADCREFP/ADCREFN pins is higher than the internal 1V reference.

10.2.2 AST

1 **AST wake signal is released one AST clock cycle after the BUSY bit is cleared**

After writing to the Status Clear Register (SCR) the wake signal is released one AST clock cycle after the BUSY bit in the Status Register (SR.BUSY) is cleared. If entering sleep mode directly after the BUSY bit is cleared the part will wake up immediately.

Fix/Workaround

Read the Wake Enable Register (WER) and write this value back to the same register. Wait for BUSY to clear before entering sleep mode.

10.2.3 aWire

1 **aWire MEMORY_SPEED_REQUEST command does not return correct CV**

The aWire MEMORY_SPEED_REQUEST command does not return a CV corresponding to the formula in the aWire Debug Interface chapter.

Fix/Workaround

Issue a dummy read to address 0x100000000 before issuing the MEMORY_SPEED_REQUEST command and use this formula instead:

$$f_{sab} = \frac{7f_{aw}}{CV - 3}$$

10.2.4 GPIO

1 **Clearing Interrupt flags can mask other interrupts**

When clearing interrupt flags in a GPIO port, interrupts on other pins of that port, happening in the same clock cycle will not be registered.

Fix/Workaround

Read the PVR register of the port before and after clearing the interrupt to see if any pin change has happened while clearing the interrupt. If any change occurred in the PVR between the reads, they must be treated as an interrupt.

10.2.5 Power Manager

1 **Clock Failure Detector (CFD) can be issued while turning off the CFD**

While turning off the CFD, the CFD bit in the Status Register (SR) can be set. This will change the main clock source to RCSYS.

Fix/Workaround

Solution 1: Enable CFD interrupt. If CFD interrupt is issues after turning off the CFD, switch back to original main clock source.

Solution 2: Only turn off the CFD while running the main clock on RCSYS.

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) fieldss of the WDT Control Register (CTRL), a synchronizer is started to propagate the values to the WDT clcok 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.