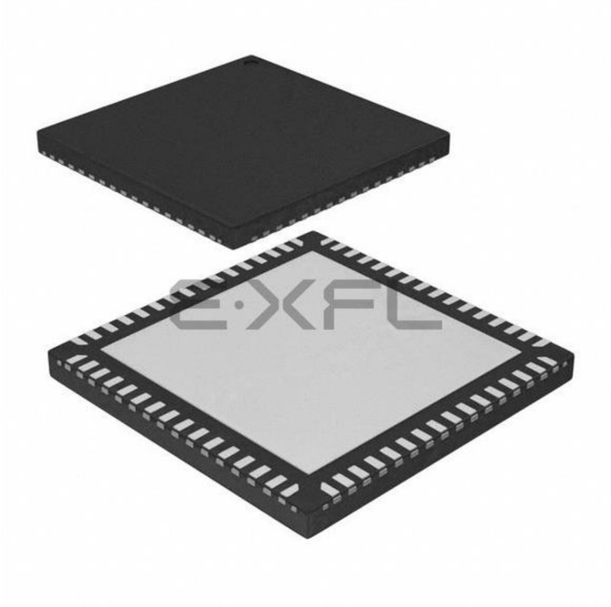
Microchip Technology - ATSAM4S2BA-MU Datasheet



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	ARM® Cortex®-M4
Core Size	32-Bit Single-Core
Speed	120MHz
Connectivity	I²C, IrDA, Memory Card, SPI, SSC, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	47
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	64K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.6V
Data Converters	A/D 16x12b; 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/atsam4s2ba-mu

11.2.1 PIO Controller A Multiplexing

Table 11-2. Multiplexing on PIO Controller A (PIOA)

I/O Line	Peripheral A	Peripheral B	Peripheral C	Peripheral D[1]	Extra Function	System Function	Comments
PA0	PWMH0	TIOA0	A17		WKUP0[2]		
PA1	PWMH1	TIOB0	A18		WKUP1[2]		
PA2	PWMH2	SCK0	DATRG		WKUP2[2]		
PA3	TWD0	NPCS3					
PA4	TWCK0	TCLK0			WKUP3[2]		
PA5	RXD0	NPCS3			WKUP4[2]		
PA6	TXD0	PCK0					
PA7	RTS0	PWMH3				XIN32[3]	
PA8	CTS0	ADTRG			WKUP5[2]	XOUT32[3]	
PA9	URXD0	NPCS1	PWMFI0		WKUP6[2]		
PA10	UTXD0	NPCS2	PWMFI1[1]				
PA11	NPCS0	PWMH0			WKUP7[2]		
PA12	MISO	PWMH1					
PA13	MOSI	PWMH2					
PA14	SPCK	PWMH3			WKUP8[2]		
PA15	TF	TIOA1	PWML3		WKUP14/PIODCEN1[4]		
PA16	TK	TIOB1	PWML2		WKUP15/PIODCEN2[4]		
PA17	TD	PCK1	PWMH3		AD0[5]		
PA18	RD	PCK2	A14	PWMFI2[1]	AD1[5]		
PA19	RK	PWML0	A15		AD2/WKUP9[2]		
PA20	RF	PWML1	A16		AD3/WKUP10[2]		
PA21	RXD1	PCK1			AD8[5]		64-/100-pin versions
PA22	TXD1	NPCS3	NCS2		AD9[5]		64-/100-pin versions
PA23	SCK1	PWMH0	A19		PIODCCLK[6]		64-/100-pin versions
PA24	RTS1	PWMH1	A20		PIODC0		64-/100-pin versions
PA25	CTS1	PWMH2	A23		PIODC1		64-/100-pin versions
PA26	DCD1	TIOA2	MCDA2		PIODC2		64-/100-pin versions
PA27	DTR1	TIOB2	MCDA3		PIODC3		64-/100-pin versions
PA28	DSR1	TCLK1	MCCDA		PIODC4		64-/100-pin versions
PA29	RI1	TCLK2	MCCK		PIODC5		64-/100-pin versions
PA30	PWML2	NPCS2	MCDA0		WKUP11[2]/PIODC6		64-/100-pin versions
PA31	NPCS1	PCK2	MCDA1		PIODC7		64-/100-pin versions

Notes:
1. Only available in SAM4S4x and SAM4S2x.
2. WKUPx can be used if PIO controller defines the I/O line as "input".
3. Refer to Section 6.2 “System I/O Lines”.

12.8.3.3 Interrupt Set-pending Registers

Name: NVIC_ISPRx [x=0..7]

Access: Read/Write

Reset: 0x000000000

31	30	29	28	27	26	25	24
SETPEND							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SETPEND							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SETPEND							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SETPEND							

These registers force interrupts into the pending state, and show which interrupts are pending.

- **SETPEND: Interrupt Set-pending**

Write:

0: No effect.

1: Changes the interrupt state to pending.

Read:

0: Interrupt is not pending.

1: Interrupt is pending.

Notes: 1. Writing a 1 to an ISPR bit corresponding to an interrupt that is pending has no effect.

2. Writing a 1 to an ISPR bit corresponding to a disabled interrupt sets the state of that interrupt to pending.

- **SIZE: Size of the MPU Protection Region**

The minimum permitted value is 3 (b00010).

The SIZE field defines the size of the MPU memory region specified by the MPU_RNR. as follows:

(Region size in bytes) = $2^{(SIZE+1)}$

The smallest permitted region size is 32B, corresponding to a SIZE value of 4. The table below gives an example of SIZE values, with the corresponding region size and value of N in the MPU_RBAR.

SIZE Value	Region Size	Value of N[1]	Note
b00100 (4)	32 B	5	Minimum permitted size
b01001 (9)	1 KB	10	–
b10011 (19)	1 MB	20	–
b11101 (29)	1 GB	30	–
b11111 (31)	4 GB	b01100	Maximum possible size

Note: 1. In the MPU_RBAR; see "MPU Region Base Address Register"

- **ENABLE: Region Enable**

Note: For information about access permission, see "MPU Access Permission Attributes" .

24.4 Device Initialization

Initialization follows the steps described below:

1. Stack setup
2. Set up the Embedded Flash Controller
3. External Clock detection (crystal or external clock on XIN)
4. If external crystal or clock with supported frequency, allow USB activation
5. Else, does not allow USB activation and use internal 12 MHz RC oscillator
6. Main oscillator frequency detection if no external clock detected
7. Switch Master Clock on Main Oscillator
8. C variable initialization
9. PLLA setup: PLLA is initialized to generate a 48 MHz clock
10. Disable the Watchdog
11. Initialization of UART0 (115200 bauds, 8, N, 1)
12. Initialization of the USB Device Port (in case USB activation allowed)
13. Wait for one of the following events
    1. Check if USB device enumeration has occurred
    2. Check if characters have been received in UART0
14. Jump to SAM-BA Monitor (see Section 24.5 ”SAM-BA Monitor”)

25.8 Bus Matrix (MATRIX) (MATRIX) User Interface

Table 25-4. Register Mapping

Offset	Register	Name	Access	Reset
0x0000	Master Configuration Register 0	MATRIX_MCFG0	Read/Write	0x00000000
0x0004	Master Configuration Register 1	MATRIX_MCFG1	Read/Write	0x00000000
0x0008	Master Configuration Register 2	MATRIX_MCFG2	Read/Write	0x00000000
0x000C	Master Configuration Register 3	MATRIX_MCFG3	Read/Write	0x00000000
0x0010 - 0x003C	Reserved	–	–	–
0x0040	Slave Configuration Register 0	MATRIX_SCFG0	Read/Write	0x00010010
0x0044	Slave Configuration Register 1	MATRIX_SCFG1	Read/Write	0x00050010
0x0048	Slave Configuration Register 2	MATRIX_SCFG2	Read/Write	0x00000010
0x004C	Slave Configuration Register 3	MATRIX_SCFG3	Read/Write	0x00000010
0x0050	Slave Configuration Register 4	MATRIX_SCFG4	Read/Write	0x00000010
0x0054 - 0x007C	Reserved	–	–	–
0x0080	Priority Register A for Slave 0	MATRIX_PRAS0	Read/Write	0x00000000
0x0084	Reserved	–	–	–
0x0088	Priority Register A for Slave 1	MATRIX_PRAS1	Read/Write	0x00000000
0x008C	Reserved	–	–	–
0x0090	Priority Register A for Slave 2	MATRIX_PRAS2	Read/Write	0x00000000
0x0094	Reserved	–	–	–
0x0098	Priority Register A for Slave 3	MATRIX_PRAS3	Read/Write	0x00000000
0x009C	Reserved	–	–	–
0x00A0	Priority Register A for Slave 4	MATRIX_PRAS4	Read/Write	0x00000000
0x00A4 - 0x0110	Reserved	–	–	–
0x0114	System I/O Configuration register	CCFG_SYSIO	Read/Write	0x00000000
0x0118	Reserved	–	–	–
0x011C	SMC Chip Select NAND Flash Assignment Register	CCFG_SMCNFCS	Read/Write	0x00000000
0x0120 - 0x010C	Reserved	–	–	–
0x1E4	Write Protection Mode Register	MATRIX_WPMR	Read/Write	0x0
0x1E8	Write Protection Status Register	MATRIX_WPSR	Read-only	0x0
0x0110 - 0x01FC	Reserved	–	–	–

Figure 26-20. TDF Optimization Disabled (TDF Mode = 0): TDF wait states between 2 read accesses on different chip selects

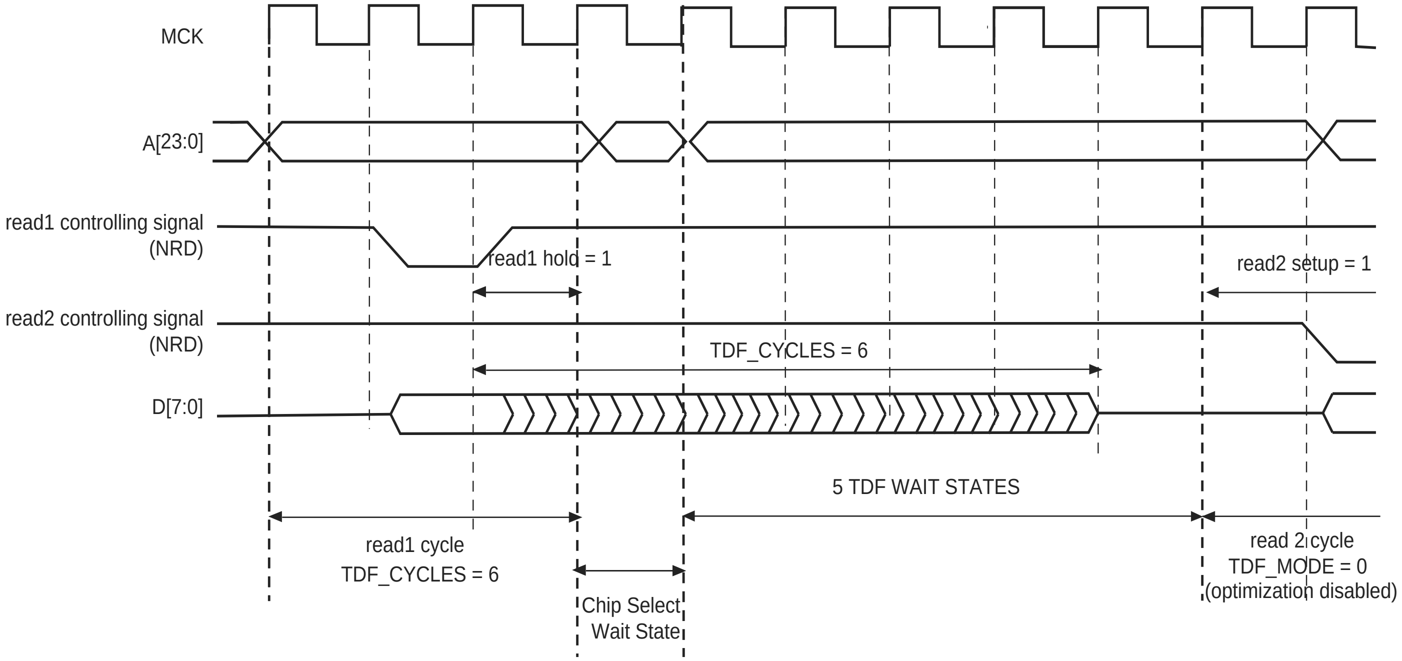
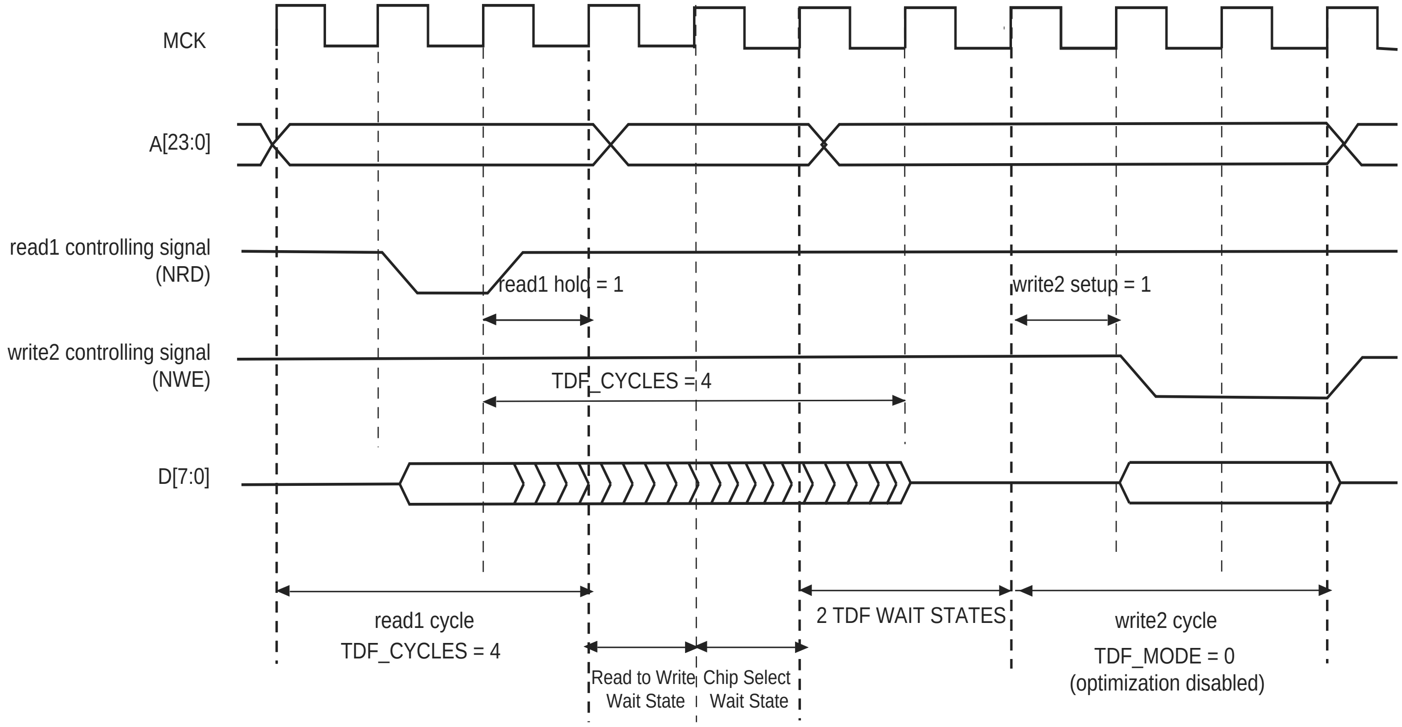


Figure 26-21. TDF Mode = 0: TDF wait states between a read and a write access on different chip selects



27.6.4 Transmit Counter Register

Name: PERIPH_TCR

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
TXCTR							
7	6	5	4	3	2	1	0
TXCTR							

- **TXCTR: Transmit Counter Register**

TXCTR must be set to transmit buffer size.

When a half-duplex peripheral is connected to the PDC, RXCTR = TXCTR.

0: Stops peripheral data transfer to the transmitter.

1–65535: Starts peripheral data transfer if the corresponding channel is active.

28. Clock Generator

28.1 Description

The Clock Generator user interface is embedded within the Power Management Controller and is described in Section 29.17 ”Power Management Controller (PMC) User Interface”. However, the Clock Generator registers are named CKGR_.

28.2 Embedded Characteristics

The Clock Generator is made up of:

- A low-power 32768 Hz slow clock oscillator with Bypass mode
- A low-power RC oscillator
- A 3 to 20 MHz crystal or ceramic resonator-based oscillator, which can be bypassed.
- A factory-programmed fast RC oscillator. Three output frequencies can be selected: 4/8/12 MHz. By default 4 MHz is selected.
- Two 80 to 240 MHz programmable PLL (input from 3 to 32 MHz), capable of providing the clock MCK to the processor and to the peripherals.

It provides the following clocks:

- SLCK, the slow clock, which is the only permanent clock within the system.
- MAINCK is the output of the main clock oscillator selection: either the crystal or ceramic resonator-based oscillator or 4/8/12 MHz fast RC oscillator.
- PLLACK is the output of the divider and 80 to 240 MHz programmable PLL (PLLA).
- PLLBCK is the output of the divider and 80 to 240 MHz programmable PLL (PLLB).

counter starts counting down on the slow clock divided by 8 from the MOSCXTST value. Since the MOSCXTST value is coded with 8 bits, the maximum start-up time is about 62 ms.

When the counter reaches 0, the MOSCXTS bit is set, indicating that the main clock is valid. Setting the MOSCXTS bit in the Interrupt Mask Register (PMC_IMR) can trigger an interrupt to the processor.

28.5.4 Main Clock Oscillator Selection

The user can select the source of the main clock from either the 4/8/12 MHz fast RC oscillator, the 3 to 20 MHz crystal oscillator or the ceramic resonator-based oscillator.

The advantage of the 4/8/12 MHz fast RC oscillator is its fast start-up time. By default, this oscillator is selected to start the system and when entering Wait mode.

The advantage of the 3 to 20 MHz crystal oscillator or ceramic resonator-based oscillator is the high level of accuracy provided.

The selection of the oscillator is made by writing the MOSCSEL bit in CKGR_MOR. The switch of the main clock source is glitch-free, so there is no need to run out of SLCK, PLLACK in order to change the selection. The MOSCSELS bit of PMC_SR indicates when the switch sequence is done.

Setting the MOSCSELS bit in PMC_IMR can trigger an interrupt to the processor.

Enabling the fast RC oscillator (MOSCRCEN = 1) and changing the fast RC frequency (MOSCCRF) at the same time is not allowed.

The fast RC must be enabled first and its frequency changed in a second step.

28.5.5 Bypassing the Main Crystal Oscillator

Prior to bypassing the 3 to 20 MHz crystal oscillator, the external clock frequency provided on the XIN pin must be stable and within the values specified in the XIN Clock characteristics in the section “Electrical Characteristics”.

The sequence is as follows:

1. Make sure an external clock is connected on XIN.
2. Enable the bypass by writing a 1 to CKGR_MOR.MOSCXTBY.
3. Disable the 3 to 20 MHz oscillator by writing a 0 to bit CKGR_MOR.MOSCXTEN.

28.5.6 Switching Main Clock between the Main RC Oscillator and Fast Crystal Oscillator

Both sources must be enabled during the switchover operation. Only after completion can the unused oscillator be disabled. If switching to fast crystal oscillator, the clock presence must first be checked according to what is described in Section 28.5.7 ”Software Sequence to Detect the Presence of Fast Crystal” because the source may not be reliable (crystal failure or bypass on a non-existent clock).

28.5.7 Software Sequence to Detect the Presence of Fast Crystal

The frequency meter carried on CKGR_MCFR is operating on the selected main clock and not on the fast crystal clock nor on the fast RC oscillator clock.

Therefore, to check for the presence of the fast crystal clock, it is necessary to have the main clock (MAINCK) driven by the fast crystal clock (MOSCSEL = 1).

The following software sequence order must be followed:

1. MCK must select the slow clock (CSS = 0 in the Master Clock Register (PMC_MCKR)).
2. Wait for the MCKRDY flag in PMC_SR to be 1.
3. The fast crystal must be enabled by programming 1 in the MOSCXTEN field in the CKGR_MOR with the MOSCXTST field being programmed to the appropriate value (see the “Electrical Characteristics” section).

35. Universal Asynchronous Receiver Transmitter (UART)

35.1 Description

The Universal Asynchronous Receiver Transmitter (UART) features a two-pin UART that can be used for communication and trace purposes and offers an ideal medium for in-situ programming solutions.

Moreover, the association with a peripheral DMA controller (PDC) permits packet handling for these tasks with processor time reduced to a minimum.

35.2 Embedded Characteristics

- Two-pin UART
  - Independent Receiver and Transmitter with a Common Programmable Baud Rate Generator
  - Even, Odd, Mark or Space Parity Generation
  - Parity, Framing and Overrun Error Detection
  - Automatic Echo, Local Loopback and Remote Loopback Channel Modes
  - Interrupt Generation
  - Support for Two PDC Channels with Connection to Receiver and Transmitter

35.3 Block Diagram

Figure 35-1. UART Block Diagram

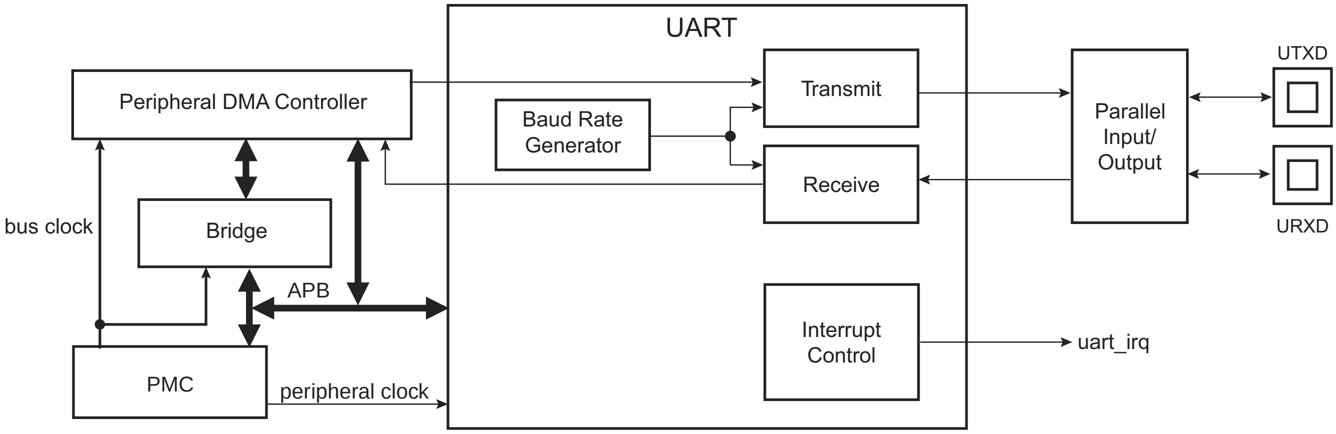


Table 35-1. UART Pin Description

Pin Name	Description	Type
URXD	UART Receive Data	Input
UTXD	UART Transmit Data	Output

36.7.14 USART Transmit Holding Register

Name: US_THR

Address: 0x4002401C (0), 0x4002801C (1)

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–

15	14	13	12	11	10	9	8
TXSYNH	–	–	–	–	–	–	TXCHR

7	6	5	4	3	2	1	0
TXCHR							

- **TXCHR: Character to be Transmitted**

Next character to be transmitted after the current character if TXRDY is not set.

- **TXSYNH: Sync Field to be Transmitted**

0: The next character sent is encoded as a data. Start frame delimiter is DATA SYNC.

1: The next character sent is encoded as a command. Start frame delimiter is COMMAND SYNC.

37.6.11.1 WAVSEL = 00

When WAVSEL = 00, the value of TC_CV is incremented from 0 to 2^{16}-1. Once 2^{16}-1 has been reached, the value of TC_CV is reset. Incrementation of TC_CV starts again and the cycle continues. See Figure 37-7.

An external event trigger or a software trigger can reset the value of TC_CV. It is important to note that the trigger may occur at any time. See Figure 37-8.

RC Compare cannot be programmed to generate a trigger in this configuration. At the same time, RC Compare can stop the counter clock (CPCSTOP = 1 in TC_CMR) and/or disable the counter clock (CPCDIS = 1 in TC_CMR).

Figure 37-7. WAVSEL = 00 without Trigger

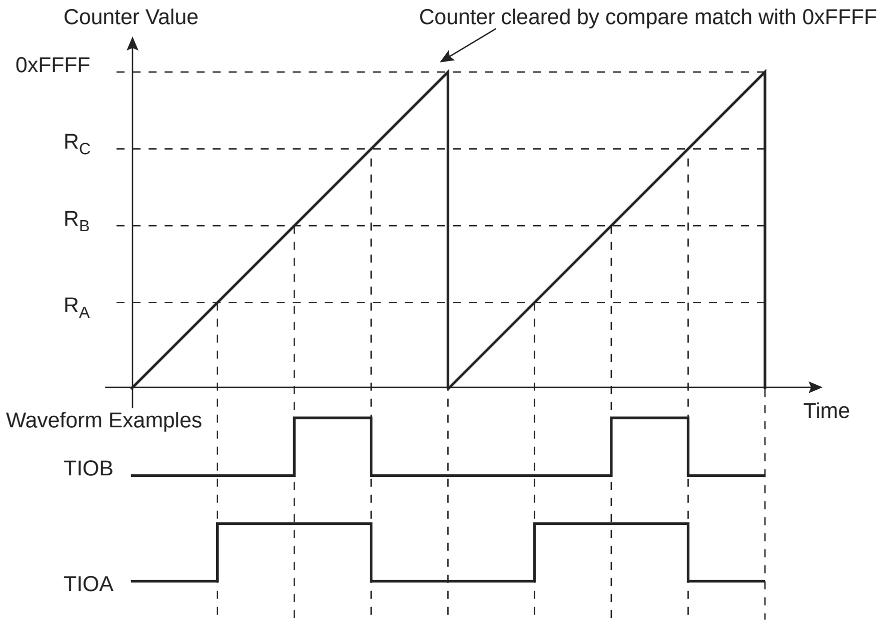
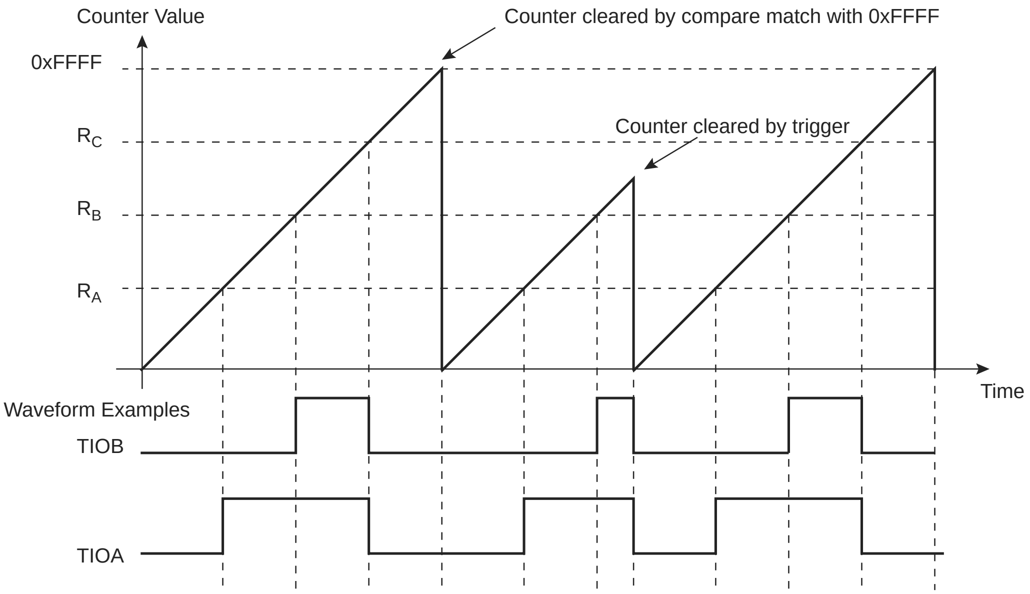


Figure 37-8. WAVSEL = 00 with Trigger



Counter Value
Counter cleared by compare match with 0xFFFF
0xFFFF
R_C
Counter cleared by trigger
R_B
R_A
Time
Waveform Examples
TIOB
TIOA

38.14.14HSMCI Interrupt Disable Register

Name: HSMCI_IDR

Address: 0x40000048

Access: Write-only

31	30	29	28	27	26	25	24
UNRE	OVRE	ACKRCVE	ACKRCV	XFRDONE	FIFOEMPTY	–	–

23	22	21	20	19	18	17	16
CSTOE	DTOE	DCRCE	RTOE	RENDE	RCRCE	RDIRE	RINDE

15	14	13	12	11	10	9	8
TXBUFE	RXBUFF	CSRCV	SDIOWAIT	–	–	–	SDIOIRQA

7	6	5	4	3	2	1	0
ENDTX	ENDRX	NOTBUSY	DTIP	BLKE	TXRDY	RXRDY	CMDRDY

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

- **CMDRDY: Command Ready Interrupt Disable**

- **RXRDY: Receiver Ready Interrupt Disable**

- **TXRDY: Transmit Ready Interrupt Disable**

- **BLKE: Data Block Ended Interrupt Disable**

- **DTIP: Data Transfer in Progress Interrupt Disable**

- **NOTBUSY: Data Not Busy Interrupt Disable**

- **ENDRX: End of Receive Buffer Interrupt Disable**

- **ENDTX: End of Transmit Buffer Interrupt Disable**

- **SDIOIRQA: SDIO Interrupt for Slot A Interrupt Disable**

- **SDIOWAIT: SDIO Read Wait Operation Status Interrupt Disable**

- **CSRCV: Completion Signal received interrupt Disable**

- **RXBUFF: Receive Buffer Full Interrupt Disable**

- **TXBUFE: Transmit Buffer Empty Interrupt Disable**

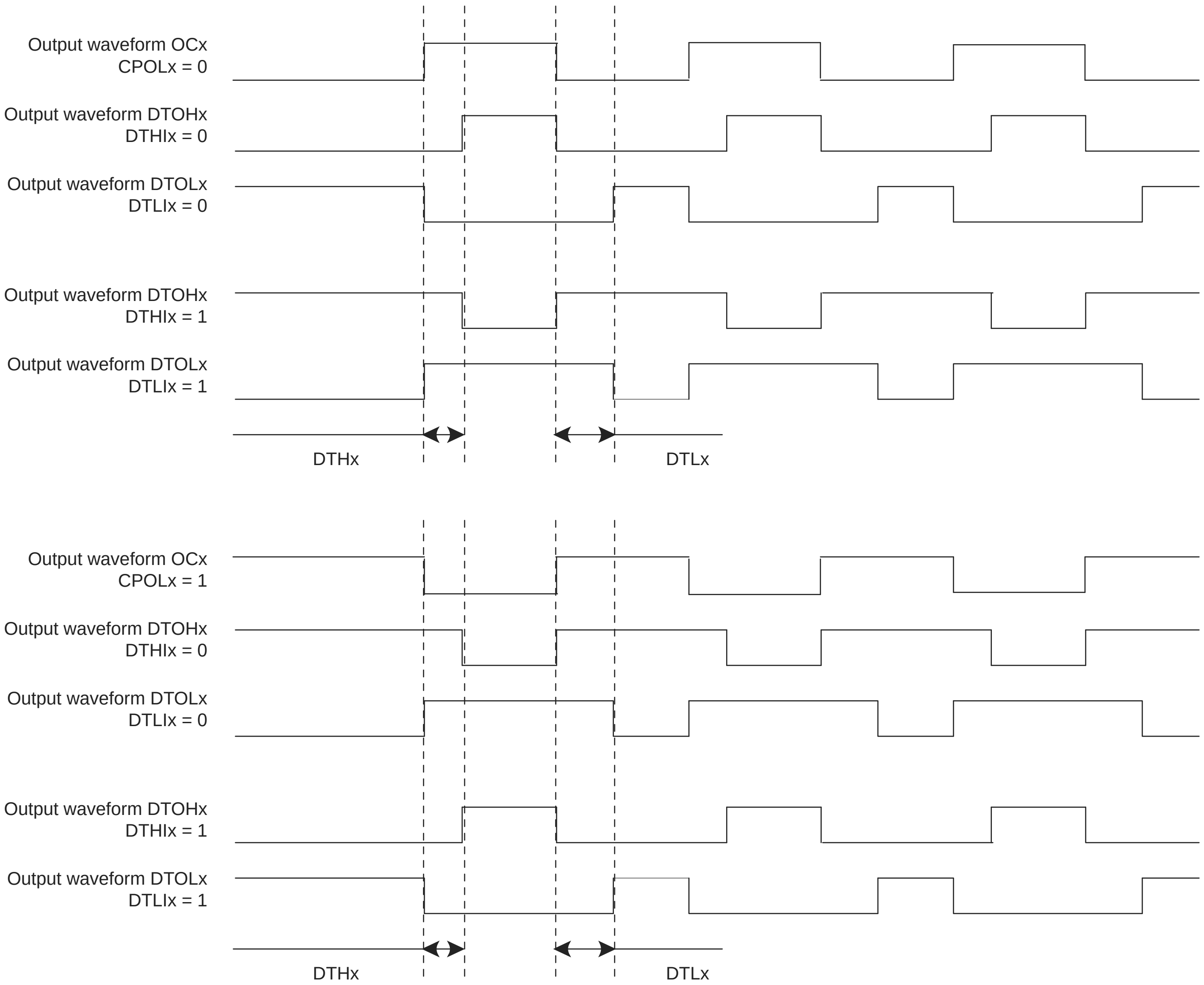
- **RINDE: Response Index Error Interrupt Disable**

- **RDIRE: Response Direction Error Interrupt Disable**

- **RCRCE: Response CRC Error Interrupt Disable**

- **RENDE: Response End Bit Error Interrupt Disable**

Figure 39-7. Complementary Output Waveforms



39.6.2.5 Output Override

The two complementary outputs DTOHx and DTOLx of the dead-time generator can be forced to a value defined by the software.

40.4 Product Dependencies

For further details on the USB Device hardware implementation, see the specific Product Properties document.

The USB physical transceiver is integrated into the product. The bidirectional differential signals DDP and DDM are available from the product boundary.

One I/O line may be used by the application to check that VBUS is still available from the host. Self-powered devices may use this entry to be notified that the host has been powered off. In this case, the pull-up on DDP must be disabled in order to prevent feeding current to the host. The application should disconnect the transceiver, then remove the pull-up.

40.4.1 I/O Lines

The USB pins are shared with PIO lines. By default, the USB function is activated, and pins DDP and DDM are used for USB. To configure DDP or DDM as PIOs, the user needs to configure the system I/O configuration register (CCFG_SYSIO) in the MATRIX.

40.4.2 Power Management

The USB device peripheral requires a 48 MHz clock. This clock must be generated by a PLL driven by a clock source with an accuracy of ± 0.25% (note that the fast RC oscillator cannot be used).

Thus, the USB device receives two clocks from the Power Management Controller (PMC): the master clock, MCK, used to drive the peripheral user interface, and the UDPCK, used to interface with the bus USB signals (recovered 12 MHz domain).

WARNING: The UDP peripheral clock in the Power Management Controller (PMC) must be enabled before any read/write operations to the UDP registers including the UDP_TXVC register.

40.4.3 Interrupt

The USB device interface has an interrupt line connected to the Interrupt Controller.

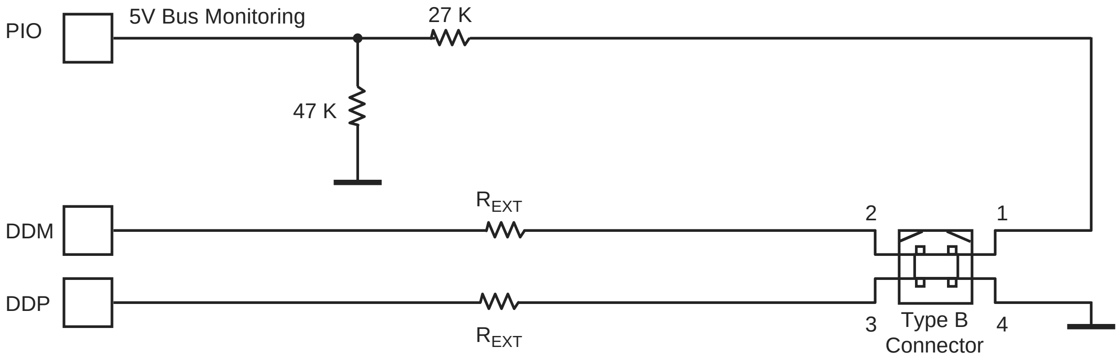
Handling the USB device interrupt requires programming the Interrupt Controller before configuring the UDP.

Table 40-3. Peripheral IDs

Instance	ID
UDP	34

40.5 Typical Connection

Figure 40-2. Board Schematic to Interface Device Peripheral



- **RXSETUP: Received Setup**

This flag generates an interrupt while it is set to one.

Read:

0: No setup packet available.

1: A setup data packet has been sent by the host and is available in the FIFO.

Write:

0: Device firmware notifies the USB peripheral device that it has read the setup data in the FIFO.

1: No effect.

This flag is used to notify the USB device firmware that a valid Setup data packet has been sent by the host and successfully received by the USB device. The USB device firmware may transfer Setup data from the FIFO by reading the UDP_FDRx to the microcontroller memory. Once a transfer has been done, RXSETUP must be cleared by the device firmware.

Ensuing Data OUT transaction is not accepted while RXSETUP is set.

- **ISOERROR: A CRC error has been detected in an isochronous transfer**

This flag generates an interrupt while it is set to one.

Read:

0: No error in the previous isochronous transfer.

1: CRC error has been detected, data available in the FIFO are corrupted.

Write:

0: Resets the ISOERROR flag, clears the interrupt.

1: No effect.

- **TXPKTRDY: Transmit Packet Ready**

This flag is cleared by the USB device.

This flag is set by the USB device firmware.

Read:

0: There is no data to send.

1: The data is waiting to be sent upon reception of token IN.

Write:

0: Can be used in the procedure to cancel transmission data. (See Section 40.6.2.5 “Transmit Data Cancellation” on page 1041)

1: A new data payload has been written in the FIFO by the firmware and is ready to be sent.

This flag is used to generate a Data IN transaction (device to host). Device firmware checks that it can write a data payload in the FIFO, checking that TXPKTRDY is cleared. Transfer to the FIFO is done by writing in the UDP_FDRx. Once the data payload has been transferred to the FIFO, the firmware notifies the USB device setting TXPKTRDY to one. USB bus transactions can start. TXCOMP is set once the data payload has been received by the host.

After setting or clearing this bit, a wait time of 3 UDPCK clock cycles and 3 peripheral clock cycles is required before accessing DPR.

Value	Name	Description
6	BULK_IN	Bulk IN
3	INT_OUT	Interrupt OUT
7	INT_IN	Interrupt IN

- **DTGLE: Data Toggle (Read-only)**

0: Identifies DATA0 packet

1: Identifies DATA1 packet

Refer to Chapter 8 of the *Universal Serial Bus Specification, Rev. 2.0* for more information on DATA0, DATA1 packet definitions.

- **EPEDS: Endpoint Enable Disable**

Read:

0: Endpoint disabled

1: Endpoint enabled

Write:

0: Disables endpoint

1: Enables endpoint

Control endpoints are always enabled. Reading or writing this field has no effect on control endpoints.

Note: After reset, all endpoints are configured as control endpoints (zero).

- **RXBYTECNT[10:0]: Number of Bytes Available in the FIFO (Read-only)**

When the host sends a data packet to the device, the USB device stores the data in the FIFO and notifies the microcontroller. The microcontroller can load the data from the FIFO by reading RXBYTECENT bytes in the UDP_FDRx.

Figure 44-7. SAM4S16/S8 Current Consumption in Sleep Mode (AMP1) vs Master Clock Ranges (refer to Table 44-14)

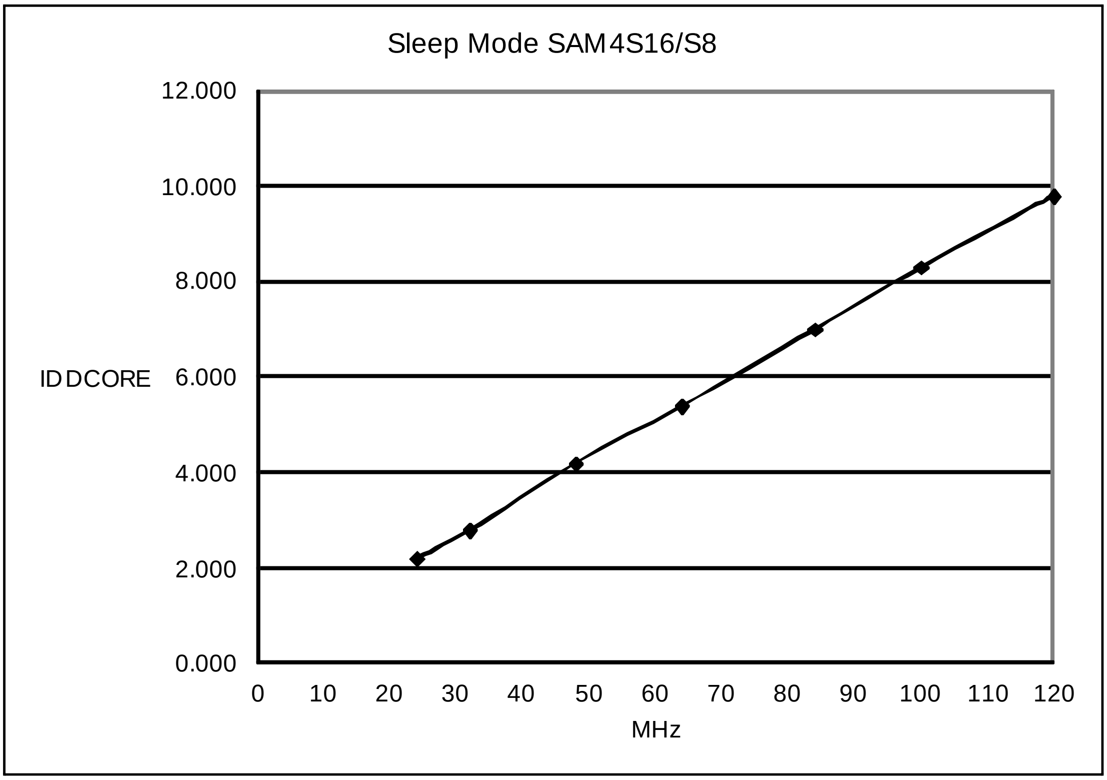


Table 44-14. SAM4S16/S8 Typical Sleep Mode Current Consumption vs Master Clock (MCK) Variation with PLLA

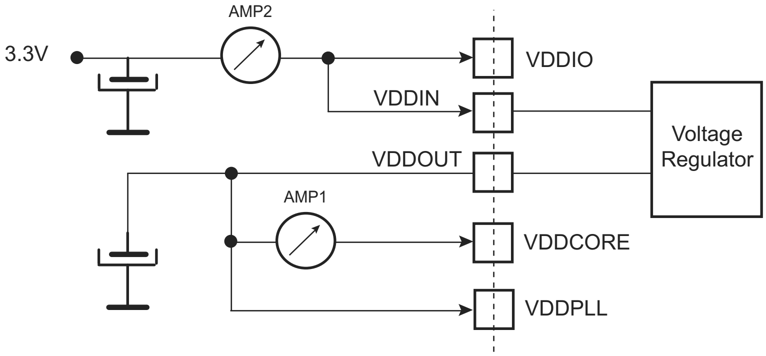
	Typical Value @ 25°C		
Core Clock/MCK (MHz)	**VDDCORE Consumption (AMP1)**	**Total Consumption (AMP2)**	**Unit**
120	8.1	9.6	mA
100	7.1	8.1	
84	6.0	6.8	
64	4.7	5.2	
48	3.5	3.9	
32	2.4	2.6	
24	1.8	2.0	

44.4.3 Active Mode Power Consumption

The active mode configuration and measurements are defined as follows:

- VDDIO = VDDIN = 3.3V
- VDDCORE = 1.2V (internal voltage regulator used)
- T_A = 25°C
- Application running from Flash memory with128-bit access mode
- All peripheral clocks are deactivated.
- Master Clock (MCK) running at various frequencies with PLLA or the fast RC oscillator.
- Current measurement on AMP1 (VDDCORE) and total current on AMP2

Figure 44-10. Active Mode Measurement Setup



The following tables give Active mode current consumption in typical conditions.

- VDDCORE at 1.2V
- T_A = 25°C

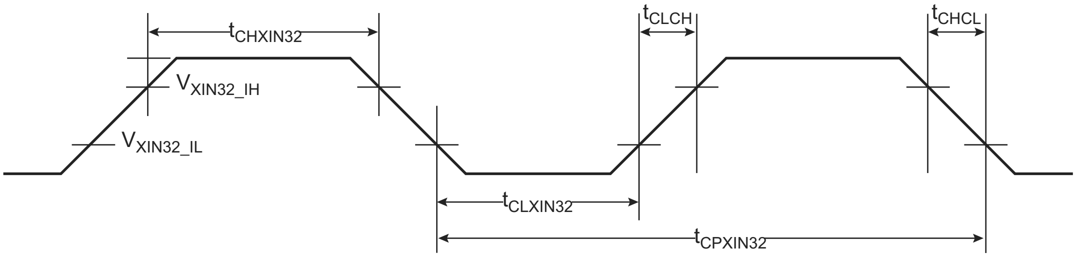
44.5.5 32.768 kHz XIN32 Clock Input Characteristics in Bypass Mode

Table 44-30. XIN32 Clock Electrical Characteristics (In Bypass Mode)

Symbol	Parameter	Conditions	Min	Max	Unit
$1/(t_{CPXIN32})$	XIN32 Clock Frequency	[1]		44	kHz
$t_{CPXIN32}$	XIN32 Clock Period	[1]	22		µs
$t_{CHXIN32}$	XIN32 Clock High Half-period	[1]	11		µs
$t_{CLXIN32}$	XIN32 Clock Low Half-period	[1]	11		µs
t_{CLCH}	Rise Time		400		ns
t_{CHCL}	Fall Time		400		ns
C_i	XIN32 Input Capacitance			6	pF
R_{IN}	XIN32 Pull-down Resistor		3	5	MΩ
V_{XIN32_IL}	V_{XIN32} Input Low-level Voltage		-0.3	$0.3 \times V_{DDIO}$	V
V_{XIN32_IH}	V_{XIN32} Input High-level Voltage		$0.7 \times V_{DDIO}$	$V_{DDIO} + 0.3$	V

Note: 1. These characteristics apply only when the 32.768 kHz crystal oscillator is in Bypass mode (i.e., when OSCBYPASS = 1 in SUPC_MR and XTALSEL = 1 in SUPC_CR).

Figure 44-14. XIN32 Clock Timing


$t_{CHXIN32}$
t_{CLCH}
t_{CHCL}
V_{XIN32_IH}
V_{XIN32_IL}
$t_{CLXIN32}$
$t_{CPXIN32}$