Microchip Technology - ATSAM4SD32BB-AN 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	2MB (2M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	160K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.6V
Data Converters	A/D 11x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	64-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsam4sd32bb-an

Table 12-13. Cortex-M4 Instructions (Continued)

Mnemonic	Operands	Description	Flags
LDMDB, LDMEA	Rn{!}, reglist	Load Multiple registers, decrement before	–
LDMFD, LDMIA	Rn{!}, reglist	Load Multiple registers, increment after	–
LDR	Rt, [Rn, #offset]	Load Register with word	–
LDRB, LDRBT	Rt, [Rn, #offset]	Load Register with byte	–
LDRD	Rt, Rt2, [Rn, #offset]	Load Register with two bytes	–
LDREX	Rt, [Rn, #offset]	Load Register Exclusive	–
LDREXB	Rt, [Rn]	Load Register Exclusive with byte	–
LDREXH	Rt, [Rn]	Load Register Exclusive with halfword	–
LDRH, LDRHT	Rt, [Rn, #offset]	Load Register with halfword	–
LDRSB, DRSBT	Rt, [Rn, #offset]	Load Register with signed byte	–
LDRSH, LDRSHT	Rt, [Rn, #offset]	Load Register with signed halfword	–
LDRT	Rt, [Rn, #offset]	Load Register with word	–
LSL, LSLS	Rd, Rm, <Rs|#n>	Logical Shift Left	N,Z,C
LSR, LSRS	Rd, Rm, <Rs|#n>	Logical Shift Right	N,Z,C
MLA	Rd, Rn, Rm, Ra	Multiply with Accumulate, 32-bit result	–
MLS	Rd, Rn, Rm, Ra	Multiply and Subtract, 32-bit result	–
MOV, MOVS	Rd, Op2	Move	N,Z,C
MOVT	Rd, #imm16	Move Top	–
MOVW, MOV	Rd, #imm16	Move 16-bit constant	N,Z,C
MRS	Rd, spec_reg	Move from special register to general register	–
MSR	spec_reg, Rm	Move from general register to special register	N,Z,C,V
MUL, MULS	{Rd,} Rn, Rm	Multiply, 32-bit result	N,Z
MVN, MVNS	Rd, Op2	Move NOT	N,Z,C
NOP	–	No Operation	–
ORN, ORNS	{Rd,} Rn, Op2	Logical OR NOT	N,Z,C
ORR, ORRS	{Rd,} Rn, Op2	Logical OR	N,Z,C
PKHTB, PKHBT	{Rd,} Rn, Rm, Op2	Pack Halfword	–
POP	reglist	Pop registers from stack	–
PUSH	reglist	Push registers onto stack	–
QADD	{Rd,} Rn, Rm	Saturating double and Add	Q
QADD16	{Rd,} Rn, Rm	Saturating Add 16	–
QADD8	{Rd,} Rn, Rm	Saturating Add 8	–
QASX	{Rd,} Rn, Rm	Saturating Add and Subtract with Exchange	–
QDADD	{Rd,} Rn, Rm	Saturating Add	Q
QDSUB	{Rd,} Rn, Rm	Saturating double and Subtract	Q
QSAX	{Rd,} Rn, Rm	Saturating Subtract and Add with Exchange	–
QSUB	{Rd,} Rn, Rm	Saturating Subtract	Q

12.6.5.24 USUB16 and USUB8

Unsigned Subtract 16 and Unsigned Subtract 8

Syntax

```
op{cond}{Rd,} Rn, Rm
```

where

op is any of:

USUB16 Unsigned Subtract 16.

USUB8 Unsigned Subtract 8.

cond is an optional condition code, see "Conditional Execution" .

Rd is the destination register.

Rn is the first operand register.

Rm is the second operand register.

Operation

Use these instructions to subtract 16-bit and 8-bit data before writing the result to the destination register:

The USUB16 instruction:

1. Subtracts each halfword from the second operand register from the corresponding halfword of the first operand register.
2. Writes the unsigned result in the corresponding halfwords of the destination register.

The USUB8 instruction:

1. Subtracts each byte of the second operand register from the corresponding byte of the first operand register.
2. Writes the unsigned byte result in the corresponding byte of the destination register.

Restrictions

Do not use SP and do not use PC.

Condition Flags

These instructions do not change the flags.

Examples

```
USUB16 R1, R0  ; Subtracts halfwords in R0 from corresponding halfword of R1
               ; and writes to corresponding halfword in R1USUB8  R4, R0, R5
               ; Subtracts bytes of R5 from corresponding byte in R0 and
               ; writes to the corresponding byte in R4.
```

X and *Y* specifies which half of the source registers *Rn* and *Rm* are used as the first and second multiply operand.

If *X* is *B*, then the bottom halfword, bits [15:0], of *Rn* is used.
If *X* is *T*, then the top halfword, bits [31:16], of *Rn* is used.

If Y is *B*, then the bottom halfword, bits [15:0], of *Rm* is used.
If *Y* is *T*, then the top halfword, bits [31:16], of *Rm* is used

SMLAW Signed Multiply Accumulate (word by halfword).

Y specifies which half of the source register *Rm* is used as the second multiply operand.

If *Y* is T, then the top halfword, bits [31:16] of *Rm* is used.

If *Y* is B, then the bottom halfword, bits [15:0] of *Rm* is used.

cond is an optional condition code, see “Conditional Execution” .

Rd is the destination register. If *Rd* is omitted, the destination register is *Rn*.

Rn, Rm are registers holding the values to be multiplied.

Ra is a register holding the value to be added or subtracted from.

Operation

The SMALBB, SMLABT, SMLATB, SMLATT instructions:

- Multiplies the specified signed halfword, top or bottom, values from *Rn* and *Rm*.
- Adds the value in *Ra* to the resulting 32-bit product.
- Writes the result of the multiplication and addition in *Rd*.

The non-specified halfwords of the source registers are ignored.

The SMLAWB and SMLAWT instructions:

- Multiply the 32-bit signed values in *Rn* with:
  - The top signed halfword of *Rm*, *T* instruction suffix.
  - The bottom signed halfword of *Rm*, *B* instruction suffix.
- Add the 32-bit signed value in *Ra* to the top 32 bits of the 48-bit product
- Writes the result of the multiplication and addition in *Rd*.

The bottom 16 bits of the 48-bit product are ignored.

If overflow occurs during the addition of the accumulate value, the instruction sets the Q flag in the APSR. No overflow can occur during the multiplication.

Restrictions

In these instructions, do not use SP and do not use PC.

Condition Flags

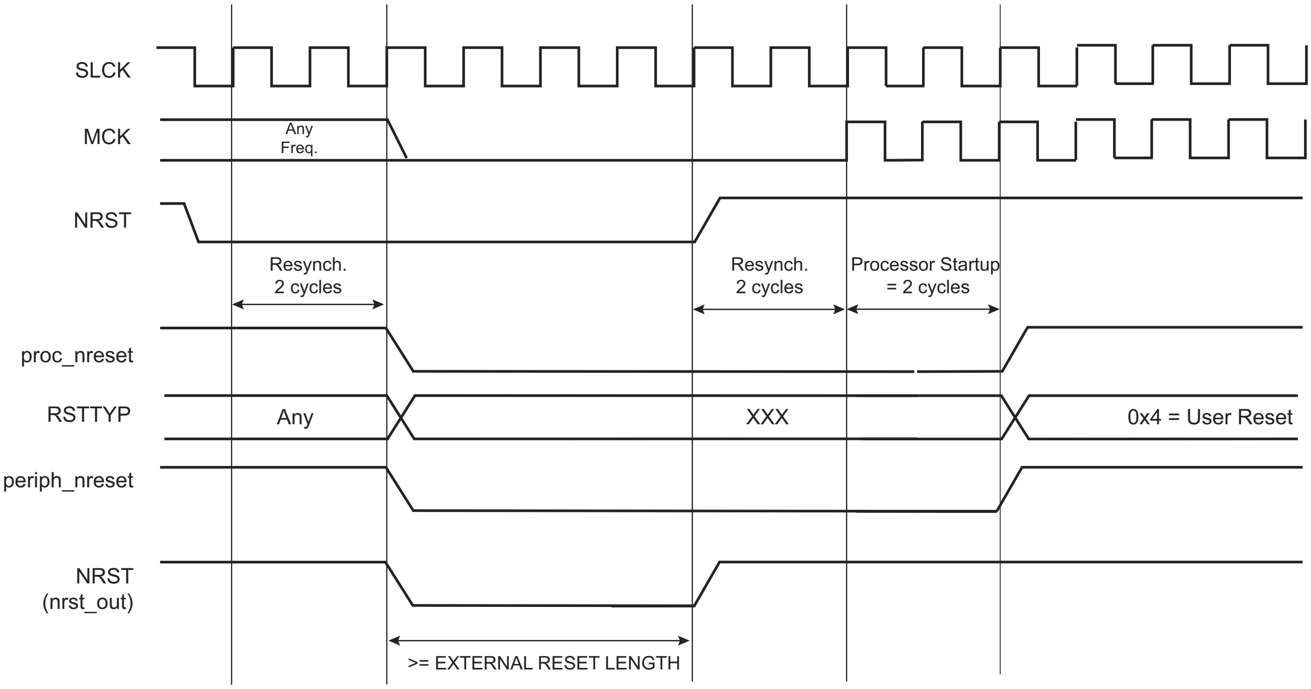
If an overflow is detected, the `Q` flag is set.

Examples

```
SMLABB  R5, R6, R4, R1  ; Multiplies bottom halfwords of R6 and R4, adds
                        ; R1 and writes to R5
SMLATB  R5, R6, R4, R1  ; Multiplies top halfword of R6 with bottom halfword
                        ; of R4, adds R1 and writes to R5
SMLATT  R5, R6, R4, R1  ; Multiplies top halfwords of R6 and R4, adds
                        ; R1 and writes the sum to R5
SMLABT  R5, R6, R4, R1  ; Multiplies bottom halfword of R6 with top halfword
                        ; of R4, adds R1 and writes to R5
SMLABT  R4, R3, R2      ; Multiplies bottom halfword of R4 with top halfword of
                        ; R3, adds R2 and writes to R4
```

The NRST manager guarantees that the NRST line is asserted for External Reset Length slow clock cycles, as programmed in field RSTC_MR.ERSTL. However, if NRST does not rise after External Reset Length because it is driven low externally, the internal reset lines remain asserted until NRST actually rises.

Figure 14-6. User Reset State



14.4.4 Reset State Priorities

The reset state manager manages the priorities among the different reset sources. The resets are listed in order of priority as follows:

1. General reset
2. Backup reset
3. Watchdog reset
4. Software reset
5. User reset

Particular cases are listed below:

- When in user reset:
  - A watchdog event is impossible because the Watchdog Timer is being reset by the proc_nreset signal.
  - A software reset is impossible, since the processor reset is being activated.
- When in software reset:
  - A watchdog event has priority over the current state.
  - The NRST has no effect.
- When in watchdog reset:
  - The processor reset is active and so a software reset cannot be programmed.
  - A user reset cannot be entered.

15.5.4 Real-time Timer Status Register

Name: RTT_SR

Address: 0x400E143C

Access: Read-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
–	–	–	–	–	–	–	–

7	6	5	4	3	2	1	0
–	–	–	–	–	–	RTTINC	ALMS

- **ALMS: Real-time Alarm Status (cleared on read)**

0: The Real-time Alarm has not occurred since the last read of RTT_SR.

1: The Real-time Alarm occurred since the last read of RTT_SR.

- **RTTINC: Prescaler Roll-over Status (cleared on read)**

0: No prescaler roll-over occurred since the last read of the RTT_SR.

1: Prescaler roll-over occurred since the last read of the RTT_SR.

23.4 Product Dependencies

23.4.1 Power Management

The CRCCU is clocked through the Power Management Controller (PMC), the programmer must first configure the CRCCU in the PMC to enable the CRCCU clock.

23.4.2 Interrupt Source

The CRCCU has an interrupt line connected to the Interrupt Controller. Handling the CRCCU interrupt requires programming the Interrupt Controller before configuring the CRCCU.

27.6.5 Receive Next Pointer Register

Name: PERIPH_RNPR

Access: Read/Write

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

- **RXNPTR: Receive Next Pointer**

RXNPTR contains the next receive buffer address.

When a half-duplex peripheral is connected to the PDC, RXNPTR = TXNPTR.

29.17.13PMC Programmable Clock Register

Name: PMC_PCKx

Address: 0x400E0440

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
–	–	–	–	–	–	–	–

7	6	5	4	3	2	1	0
–	PRES			–	CSS		

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.

- **CSS: Master Clock Source Selection**

Value	Name	Description
0	SLOW_CLK	Slow Clock is selected
1	MAIN_CLK	Main Clock is selected
2	PLLA_CLK	PLLA Clock is selected
3	PLLB_CLK	PLLB Clock is selected
4	MCK	Master Clock is selected

- **PRES: Programmable Clock Prescaler**

Value	Name	Description
0	CLK_1	Selected clock
1	CLK_2	Selected clock divided by 2
2	CLK_4	Selected clock divided by 4
3	CLK_8	Selected clock divided by 8
4	CLK_16	Selected clock divided by 16
5	CLK_32	Selected clock divided by 32
6	CLK_64	Selected clock divided by 64

29.17.20PMC Fault Output Clear Register

Name: PMC_FOCR

Address: 0x400E0478

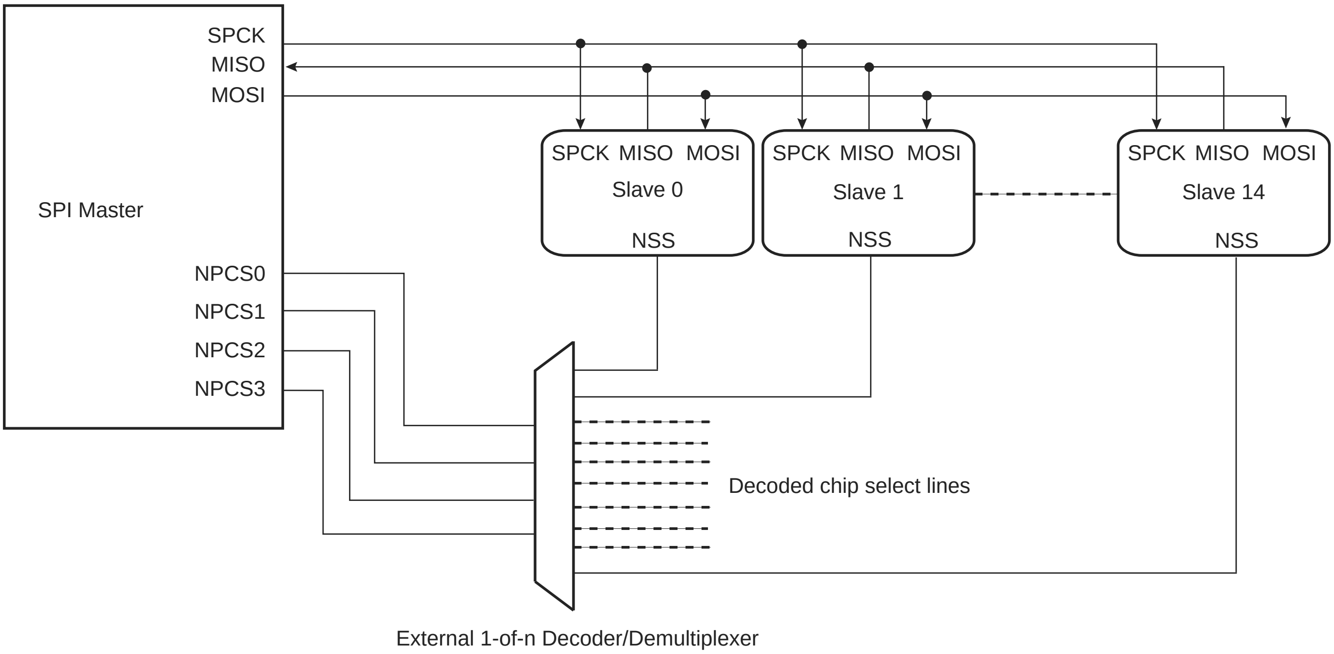
Access: Write-only

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

- **FOCLR: Fault Output Clear**

Clears the clock failure detector fault output.

Figure 33-11. Chip Select Decoding Application Block Diagram: Single Master/Multiple Slave Implementation



33.7.3.8 Peripheral Deselection without PDC

During a transfer of more than one unit of data on a Chip Select without the PDC, the SPI_TDR is loaded by the processor, the TDRE flag rises as soon as the content of the SPI_TDR is transferred into the internal Shift register. When this flag is detected high, the SPI_TDR can be reloaded. If this reload by the processor occurs before the end of the current transfer and if the next transfer is performed on the same chip select as the current transfer, the Chip Select is not de-asserted between the two transfers. But depending on the application software handling the SPI status register flags (by interrupt or polling method) or servicing other interrupts or other tasks, the processor may not reload the SPI_TDR in time to keep the chip select active (low). A null DLYBCT value (delay between consecutive transfers) in the SPI_CSR, gives even less time for the processor to reload the SPI_TDR. With some SPI slave peripherals, if the chip select line must remain active (low) during a full set of transfers, communication errors can occur.

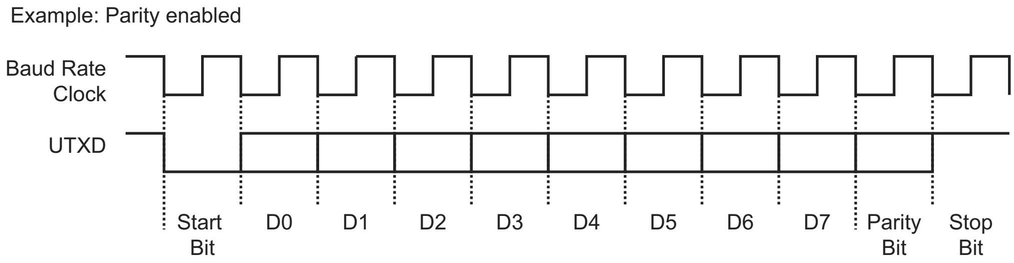
To facilitate interfacing with such devices, the Chip Select registers [CSR0...CSR3] can be programmed with the Chip Select Active After Transfer (CSAAT) bit to 1. This allows the chip select lines to remain in their current state (low = active) until a transfer to another chip select is required. Even if the SPI_TDR is not reloaded, the chip select remains active. To de-assert the chip select line at the end of the transfer, the Last Transfer (LASTXFER) bit in SPI_CR must be set after writing the last data to transmit into SPI_TDR.

33.7.3.9 Peripheral Deselection with PDC

PDC provides faster reloads of the SPI_TDR compared to software. However, depending on the system activity, it is not guaranteed that the SPI_TDR is written with the next data before the end of the current transfer. Consequently, data can be lost by the de-assertion of the NPCS line for SPI slave peripherals requiring the chip select line to remain active between two transfers. The only way to guarantee a safe transfer in this case is the use of the CSAAT and LASTXFER bits.

When the CSAAT bit is configured to 0, the NPCS does not rise in all cases between two transfers on the same peripheral. During a transfer on a Chip Select, the TDRE flag rises as soon as the content of the SPI_TDR is transferred into the internal shift register. When this flag is detected, the SPI_TDR can be reloaded. If this reload occurs before the end of the current transfer and if the next transfer is performed on the same chip select as the current transfer, the Chip Select is not de-asserted between the two transfers. This can lead to difficulties to interface with some serial peripherals requiring the chip select to be de-asserted after each transfer. To facilitate

Figure 35-9. Character Transmission

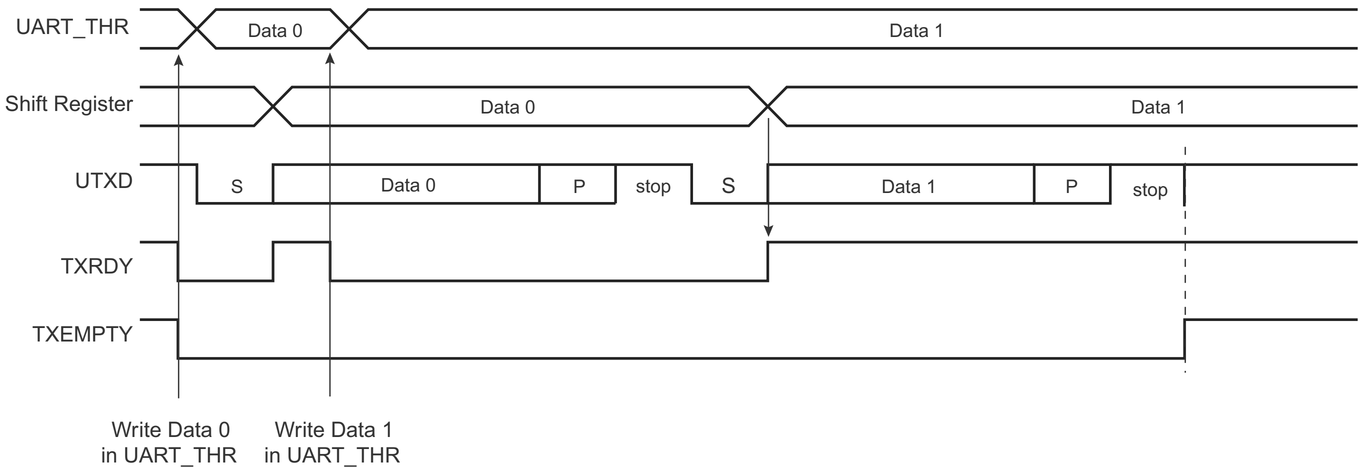


35.5.3.3 Transmitter Control

When the transmitter is enabled, the bit TXRDY (Transmitter Ready) is set in UART_SR. The transmission starts when the programmer writes in the UART_THR, and after the written character is transferred from UART_THR to the internal shift register. The TXRDY bit remains high until a second character is written in UART_THR. As soon as the first character is completed, the last character written in UART_THR is transferred into the internal shift register and TXRDY rises again, showing that the holding register is empty.

When both the internal shift register and UART_THR are empty, i.e., all the characters written in UART_THR have been processed, the TXEMPTY bit rises after the last stop bit has been completed.

Figure 35-10. Transmitter Control



35.5.4 Peripheral DMA Controller (PDC)

Both the receiver and the transmitter of the UART are connected to a PDC.

The PDC channels are programmed via registers that are mapped within the UART user interface from the offset 0x100. The status bits are reported in UART_SR and generate an interrupt.

The RXRDY bit triggers the PDC channel data transfer of the receiver. This results in a read of the data in UART_RHR. The TXRDY bit triggers the PDC channel data transfer of the transmitter. This results in a write of data in UART_THR.

35.5.5 Test Modes

The UART supports three test modes. These modes of operation are programmed by using the CHMODE field in UART_MR.

The Automatic echo mode allows a bit-by-bit retransmission. When a bit is received on the URXD line, it is sent to the UTXD line. The transmitter operates normally, but has no effect on the UTXD line.

- **DSRIC: Data Set Ready Input Change Disable**

- **DCDIC: Data Carrier Detect Input Change Interrupt Disable**

- **CTSIC: Clear to Send Input Change Interrupt Disable**

- **MANE: Manchester Error Interrupt Disable**

36.7.17 USART Transmitter Timeguard Register

Name: US_TTGR

Address: 0x40024028 (0), 0x40028028 (1)

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
–	–	–	–	–	–	–	–

7	6	5	4	3	2	1	0
TG							

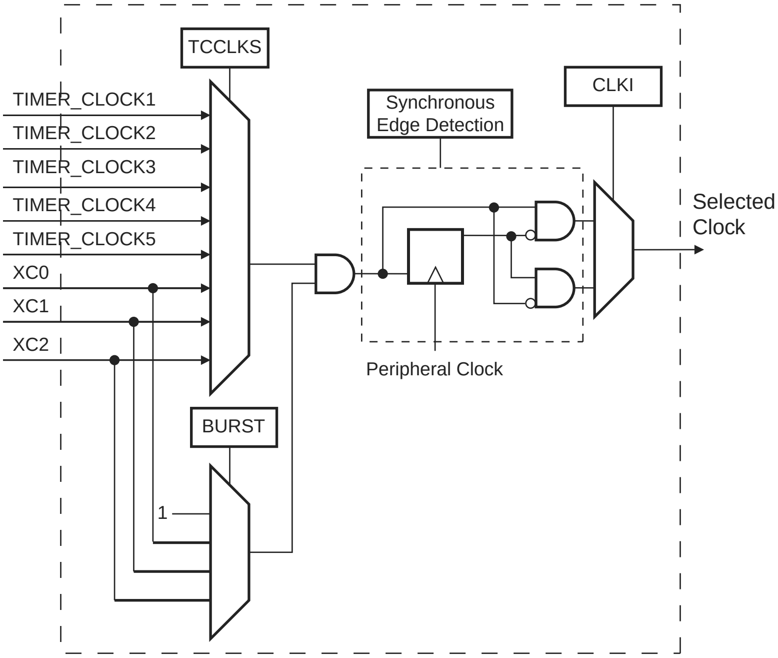
This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

- **TG: Timeguard Value**

0: The transmitter timeguard is disabled.

1–255: The transmitter timeguard is enabled and TG is Timeguard Delay / Bit Period.

Figure 37-3. Clock Selection



37.6.4 Clock Control

The clock of each counter can be controlled in two different ways: it can be enabled/disabled and started/stopped. See Figure 37-4.

- The clock can be enabled or disabled by the user with the CLKEN and the CLKDIS commands in the TC Channel Control Register (TC_CCR). In Capture mode it can be disabled by an RB load event if LDBDIS is set to 1 in the TC_CMR. In Waveform mode, it can be disabled by an RC Compare event if CPCDIS is set to 1 in TC_CMR. When disabled, the start or the stop actions have no effect: only a CLKEN command in the TC_CCR can re-enable the clock. When the clock is enabled, the CLKSTA bit is set in the TC_SR.
- The clock can also be started or stopped: a trigger (software, synchro, external or compare) always starts the clock. The clock can be stopped by an RB load event in Capture mode (LDBSTOP = 1 in TC_CMR) or an RC compare event in Waveform mode (CPCSTOP = 1 in TC_CMR). The start and the stop commands are effective only if the clock is enabled.

39.7.9 PWM Sync Channels Mode Register

Name: PWM_SCM

Address: 0x40020020

Access: Read/Write

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

23	22	21	20	19	18	17	16
PTRCS			PTRM	–	–	UPDM	

15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–

7	6	5	4	3	2	1	0
–	–	–	–	SYNC3	SYNC2	SYNC1	SYNC0

This register can only be written if bits WPSWS2 and WPHWS2 are cleared in the PWM Write Protection Status Register.

- **SYNCx: Synchronous Channel x**

0: Channel x is not a synchronous channel.

1: Channel x is a synchronous channel.

- **UPDM: Synchronous Channels Update Mode**

Value	Name	Description
0	MODE0	Manual write of double buffer registers and manual update of synchronous channels(1)
1	MODE1	Manual write of double buffer registers and automatic update of synchronous channels(2)
2	MODE2	Automatic write of duty-cycle update registers by the Peripheral DMA Controller and automatic update of synchronous channels(2)

Notes: 1. The update occurs at the beginning of the next PWM period, when the UPDULOCK bit in PWM Sync Channels Update Control Register is set.

2. The update occurs when the Update Period is elapsed.

- **PTRM: Peripheral DMA Controller Transfer Request Mode**

UPDM	PTRM	WRDY Flag and Peripheral DMA Controller Transfer Request
0	x	The WRDY flag in PWM Interrupt Status Register 2 and the PDC transfer request are never set to ‘1’.
1	x	The WRDY flag in PWM Interrupt Status Register 2 is set to ‘1’ as soon as the update period is elapsed, the Peripheral DMA Controller transfer request is never set to ‘1’.
2	0	The WRDY flag in PWM Interrupt Status Register 2 and the PDC transfer request are set to ‘1’ as soon as the update period is elapsed.
	1	The WRDY flag in PWM Interrupt Status Register 2 and the PDC transfer request are set to ‘1’ as soon as the selected comparison matches.

- **PTRCS: Peripheral DMA Controller Transfer Request Comparison Selection**

Selection of the comparison used to set the flag WRDY and the corresponding Peripheral DMA Controller transfer request.

through the UDP_FDRx. Once a transfer is done, the device firmware must release Bank 0 to the USB peripheral device by clearing RX_DATA_BK0.

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

- **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.

- **STALLSENT: Stall Sent**

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

This ends a STALL handshake.

Read:

0: Host has not acknowledged a stall

1: Host has acknowledged the stall

Write:

0: Resets the STALLSENT flag, clears the interrupt

1: No effect

This is mandatory for the device firmware to clear this flag. Otherwise the interrupt remains.

Refer to chapters 8.4.5 and 9.4.5 of the *Universal Serial Bus Specification, Rev. 2.0* for more information on the STALL handshake.

- **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.

43.7.12 DACC Write Protection Mode Register

Name: DACC_WPMR

Address: 0x4003C0E4

Access: Read/Write

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

- **WPEN: Write Protection Enable**

0: Disables the write protection if WPKEY corresponds to 0x444143 (“DAC” in ASCII).

1: Enables the write protection if WPKEY corresponds to 0x444143 (“DAC” in ASCII).

See Section 43.6.7 ”Register Write Protection” for the list of registers that can be write-protected.

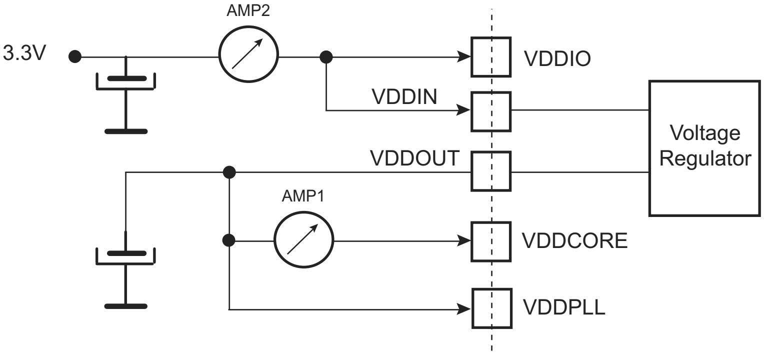
- **WPKEY: Write Protection Key**

Value	Name	Description
0x444143	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

44.4.2 Sleep and Wait Mode Current Consumption

The Wait mode and Sleep mode configuration and measurements are defined below.

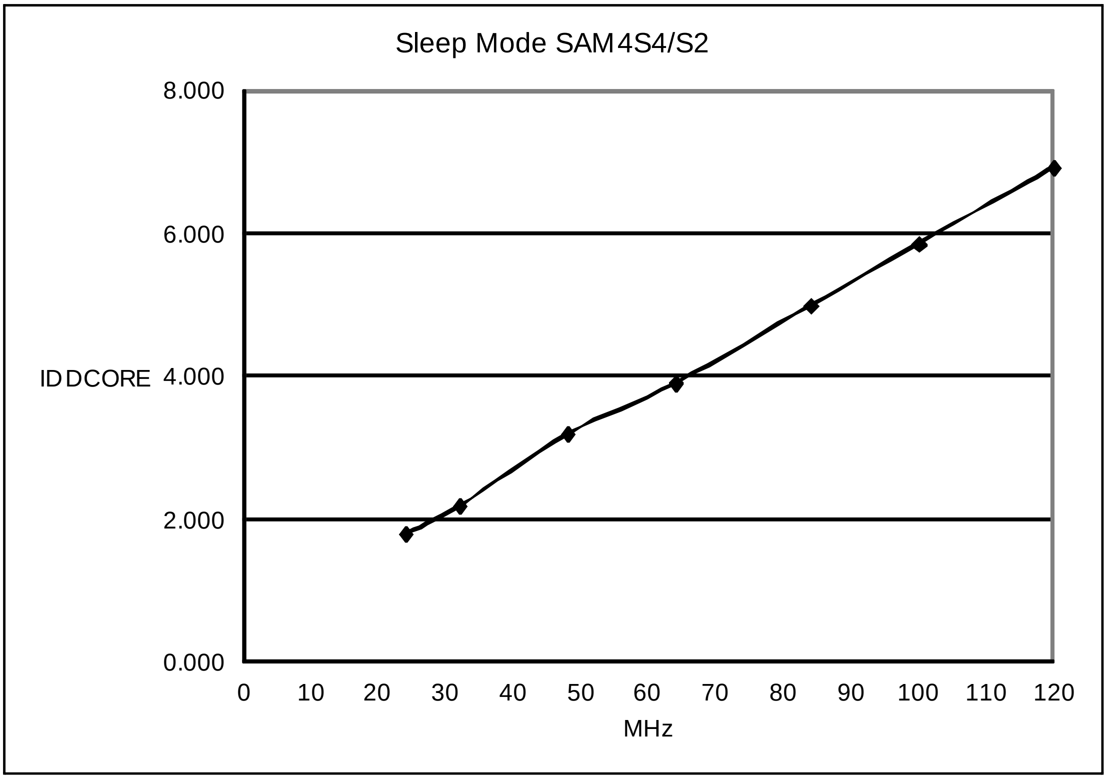
Figure 44-5. Measurement Setup for Sleep Mode



44.4.2.1 Sleep Mode

- Core clock off
- VDDIO = VDDIN = 3.3V
- Master clock (MCK) running at various frequencies with PLLA or the fast RC oscillator
- Fast start-up through pins WKUP0–15
- Current measurement as shown in Figure 44-5
- All peripheral clocks deactivated
- T_A = 25°C

Figure 44-6. SAM4S4/2 Current Consumption in Sleep Mode (AMP1) vs Master Clock Ranges (refer to Table 44-12)



Note that in SPI Master Mode the SAM4S does not sample the data (MISO) on the opposite edge where data clocks out (MOSI) but the same edge is used. This is shown in Figure 44-22 and Figure 44-23.

44.12.4 HSMCI Timings

The High-speed MultiMedia Card Interface (HSMCI) supports the MultiMedia Card (MMC) Specification V4.3, the SD Memory Card Specification V2.0, the SDIO V2.0 specification and CE-ATA V1.1.

44.12.5 SSC Timings

SSC timings are given for the following domains:

- 1.8V domain: V_{DDIO} from 1.65 to 1.95 V, maximum external capacitor = 20 pF
- 3.3V domain: V_{DDIO} from 2.85 to 3.6 V, maximum external capacitor = 30 pF

Figure 44-26. SSC Transmitter, TK and TF as Output

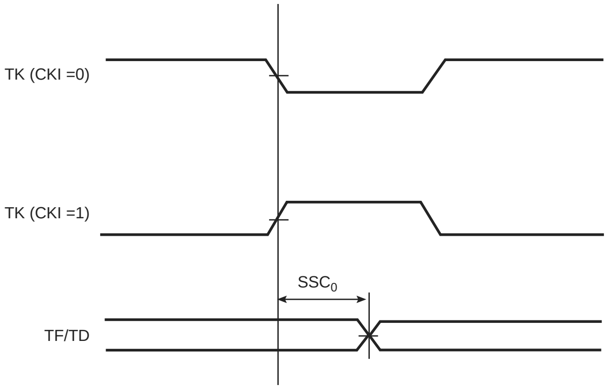


Figure 44-27. SSC Transmitter, TK as Input and TF as Output

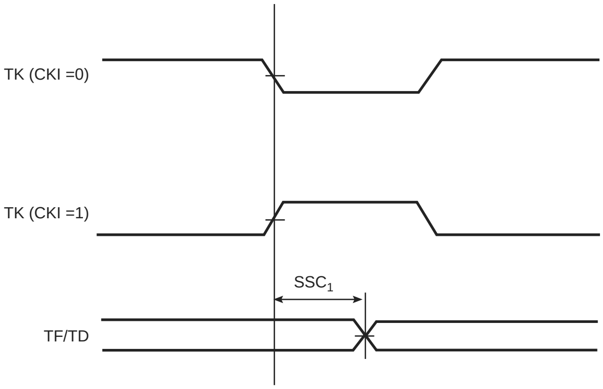


Table 49-6. SAM4S Datasheet Rev. 11100F 29-Jan-14 Revision History (Continued)

Doc. Date	Changes
	Section 43. “Digital-to-Analog Converter Controller (DACC)” Section 43.7.7 “DACC Interrupt Enable Register”, Section 43.7.8 “DACC Interrupt Disable Register” and Section 43.7.9 “DACC Interrupt Mask Register”: modified bit descriptions. Rework of all “refresh” related paragraphs, Section 43.7.3 “DACC Channel Enable Register” and Section 43.6.7 “DACC Timings”. Modified description for ”REFRESH: Automatic Refresh Period” field in Section 43.7.2 “DACC Mode Register”. Re-worked Section 43.6.8 “Register Write Protection” and associated registers and bit/field descriptions in Section 43.7.12 “DACC Write Protection Mode Register” and Section 43.7.13 “DACC Write Protection Status Register”.
	Section 44. “Electrical Characteristics” Added Section 44.2 “Recommended Operating Conditions”. Section 44.4 “Power Consumption”: Added power consumption values for SAM4S4/SAM4S2. Updated Section 44.4.1 “Backup Mode Current Consumption”. Removed Supply Ripple Voltage parameter from Table 44-30, “3 to 20 MHz Crystal Oscillator Characteristics” Table 44-32 ”XIN Clock Electrical Characteristics (In Bypass Mode)”: Added $C_{PARASTANDBY}$ AND $R_{PARASTANDBY}$ parameters. Updated and re-worked Section 44.8 “12-bit ADC Characteristics”: Updated Section 44.9 “12-bit DAC Characteristics”. Removed Max Voltage Ripple parameter from Table 44-55, “Analog Power Supply Characteristics”. Added Refresh Time to Table 44-56, “Channel Conversion Time and DAC Clock”. In Section 44.12 “AC Characteristics” modified ● Table 44-64, “SPI Timings”. ● Table 44-65, “SSC Timings” ● Table 44-66, “SMC Read Signals - NRD Controlled (READ_MODE = 1)” ● Table 44-68, “SMC Write Signals - NWE Controlled (WRITE_MODE = 1)” ● Table 44-69, “SMC Write Signals - NCS Controlled (WRITE_MODE = 0)” ● Table 44-70, “USART SPI Timings” Table 44-71 ”Two-wire Serial Bus Requirements”: Added parameter t_{BUF} Section 44.12.9 “Embedded Flash Characteristics”: modified Table 44-72, “Embedded Flash Wait State at 105°C”. Table 44-73, “AC Flash Characteristics”: Full Chip Erase: Added values for 256 Kbytes and 128 Kbytes. Added new parameter Page Program Time.
	Section 45. “Mechanical Characteristics” Table 45-20 ”64-ball WLCSP Package Dimensions (in mm)” Added body size for SAM4S4 for WLCSP64 package. Figure 45-8 ”48-lead LQFP Package Drawing” and corresponding characteristics added. Figure 45-9 ”48-lead QFN Package Drawing” and corresponding characteristics added.
	Section 48. “Errata” Added Section 48.3 “Errata SAM4S4/S2 Rev. A Parts”.
	Section 47. “Ordering Information” Added information on carrier type availability. Updated Table 47-1 ”Ordering Codes for SAM4S Devices”. Added new ordering codes for SAM4S4 and SAM4S2 devices.