Microchip Technology - ATSAM4S4CA-AUR Datasheet

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	EBI/EMI, I²C, IrDA, Memory Card, SPI, SSC, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	79
Program Memory Size	256KB (256K 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	100-LQFP
Supplier Device Package	100-LQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsam4s4ca-aur

4.2.5 64-ball WLCSP Pinout

Table 4-5. SAM4SD32/S32/SD16/S16/S8 64-ball WLCSP Pinout

A1	PA31	C1	GND	E1	PA29	G1	PA5
A2	PB7	C2	PA1	E2	TST	G2	PA6
A3	VDDCORE	C3	PA0	E3	NRST	G3	PA9
A4	PB10	C4	PB12	E4	PA28	G4	PA11
A5	VDDIO	C5	ADVREF	E5	PA25	G5	VDDCORE
A6	GND	C6	PB3	E6	PA23	G6	PA14
A7	PB9	C7	PB1	E7	PA18	G7	PA20
A8	PB14	C8	PB0	E8	VDDIN	G8	PA19
B1	PB5	D1	VDDIO	F1	PA27	H1	PA7
B2	JTAGSEL	D2	PA3	F2	VDDCORE	H2	PA8
B3	PB6	D3	PA30	F3	PA4	H3	PA10
B4	PB11	D4	PA2	F4	PB4	H4	PA12
B5	PB13	D5	PA13	F5	PA26	H5	PA24
B6	VDDPLL	D6	PA21	F6	PA16	H6	PA15
B7	PB8	D7	PA17	F7	PA22	H7	VDDIO
B8	GND	D8	PB2	F8	VDDOUT	H8	GND

Table 4-6. SAM4S4/S2 64-ball WLCSP Pinout

A1	PB5	C1	GND	E1	PA3	G1	VDDCORE
A2	PA31	C2	PA0	E2	PA30	G2	PA4
A3	VDDCORE	C3	PB7	E3	PA29	G3	PA9
A4	VDDIO	C4	PB12	E4	PA27	G4	PA11
A5	GND	C5	PA10	E5	PA24	G5	PA25
A6	PB8	C6	PB0	E6	PA18	G6	PA14
A7	PB9	C7	PB2	E7	PA17	G7	VDDIO
A8	ADVREF	C8	PB1	E8	VDDIN	G8	PA19
B1	PA1	D1	VDDIO	F1	TST	H1	PB4
B2	JTAGSEL	D2	PA2	F2	NRST	H2	PA7
B3	PB10	D3	PA28	F3	PA5	H3	PA8
B4	PB11	D4	PB6	F4	PA6	H4	PA12
B5	PB13	D5	PA26	F5	PA13	H5	VDDCORE
B6	VDDPLL	D6	PA23	F6	PA22	H6	PA15
B7	PB14	D7	PA16	F7	PA21	H7	GND
B8	GNDANA	D8	PB3	F8	VDDOUT	H8	PA20

- Prefetches instructions ahead of execution
- Speculatively prefetches from branch target addresses.

12.4.2.4 Software Ordering of Memory Accesses

The order of instructions in the program flow does not always guarantee the order of the corresponding memory transactions. This is because:

- The processor can reorder some memory accesses to improve efficiency, providing this does not affect the behavior of the instruction sequence.
- The processor has multiple bus interfaces
- Memory or devices in the memory map have different wait states
- Some memory accesses are buffered or speculative.

“Memory System Ordering of Memory Accesses” describes the cases where the memory system guarantees the order of memory accesses. Otherwise, if the order of memory accesses is critical, the software must include memory barrier instructions to force that ordering. The processor provides the following memory barrier instructions:

DMB

The *Data Memory Barrier* (DMB) instruction ensures that outstanding memory transactions complete before subsequent memory transactions. See “DMB” .

DSB

The *Data Synchronization Barrier* (DSB) instruction ensures that outstanding memory transactions complete before subsequent instructions execute. See “DSB” .

ISB

The *Instruction Synchronization Barrier* (ISB) ensures that the effect of all completed memory transactions is recognizable by subsequent instructions. See “ISB” .

MPU Programming

Use a DSB followed by an ISB instruction or exception return to ensure that the new MPU configuration is used by subsequent instructions.

12.4.2.5 Bit-banding

A bit-band region maps each word in a *bit-band alias* region to a single bit in the *bit-band region*. The bit-band regions occupy the lowest 1 MB of the SRAM and peripheral memory regions.

The memory map has two 32 MB alias regions that map to two 1 MB bit-band regions:

- Accesses to the 32 MB SRAM alias region map to the 1 MB SRAM bit-band region, as shown in Table 12-6.
- Accesses to the 32 MB peripheral alias region map to the 1 MB peripheral bit-band region, as shown in Table 12-7.

Table 12-6. SRAM Memory Bit-banding Regions

Address Range	Memory Region	Instruction and Data Accesses
0x20000000–0x200FFFFF	SRAM bit-band region	Direct accesses to this memory range behave as SRAM memory accesses, but this region is also bit-addressable through bit-band alias.
0x22000000–0x23FFFFFF	SRAM bit-band alias	Data accesses to this region are remapped to bit-band region. A write operation is performed as read-modify-write. Instruction accesses are not remapped.

12.6.4.2 LDR and STR, Immediate Offset

Load and Store with immediate offset, pre-indexed immediate offset, or post-indexed immediate offset.

Syntax

```
op{type}{cond} Rt, [Rn {, #offset}]        ; immediate offset
op{type}{cond} Rt, [Rn, #offset]!          ; pre-indexed
op{type}{cond} Rt, [Rn], #offset           ; post-indexed
opD{cond} Rt, Rt2, [Rn {, #offset}]        ; immediate offset, two words
opD{cond} Rt, Rt2, [Rn, #offset]!          ; pre-indexed, two words
opD{cond} Rt, Rt2, [Rn], #offset           ; post-indexed, two words
```

where:

op is one of:

LDR Load Register.

STR Store Register.

type is one of:

B unsigned byte, zero extend to 32 bits on loads.

SB signed byte, sign extend to 32 bits (LDR only).

H unsigned halfword, zero extend to 32 bits on loads.

SH signed halfword, sign extend to 32 bits (LDR only).

- omit, for word.

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

Rt is the register to load or store.

Rn is the register on which the memory address is based.

offset is an offset from *Rn*. If *offset* is omitted, the address is the contents of *Rn*.

Rt2 is the additional register to load or store for two-word operations.

Operation

LDR instructions load one or two registers with a value from memory.

STR instructions store one or two register values to memory.

Load and store instructions with immediate offset can use the following addressing modes:

Offset Addressing

The offset value is added to or subtracted from the address obtained from the register *Rn*. The result is used as the address for the memory access. The register *Rn* is unaltered. The assembly language syntax for this mode is:

```
[Rn, #offset]
```

12.10 System Timer (SysTick)

The processor has a 24-bit system timer, SysTick, that counts down from the reload value to zero, reloads (wraps to) the value in the SYST_RVR on the next clock edge, then counts down on subsequent clocks.

When the processor is halted for debugging, the counter does not decrement.

The SysTick counter runs on the processor clock. If this clock signal is stopped for low power mode, the SysTick counter stops.

Ensure that the software uses aligned word accesses to access the SysTick registers.

The SysTick counter reload and current value are undefined at reset; the correct initialization sequence for the SysTick counter is:

1. Program the reload value.
2. Clear the current value.
3. Program the Control and Status register.

17.5.3 Watchdog Timer Status Register

Name: WDT_SR

Address: 0x400E1458

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
–	–	–	–	–	–	WDERR	WDUNF

- **WDUNF: Watchdog Underflow (cleared on read)**

0: No watchdog underflow occurred since the last read of WDT_SR.

1: At least one watchdog underflow occurred since the last read of WDT_SR.

- **WDERR: Watchdog Error (cleared on read)**

0: No watchdog error occurred since the last read of WDT_SR.

1: At least one watchdog error occurred since the last read of WDT_SR.

18.5.9 System Controller Write Protection Mode Register

Name: SYSC_WPMR

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 0x525443 (“RTC” in ASCII).

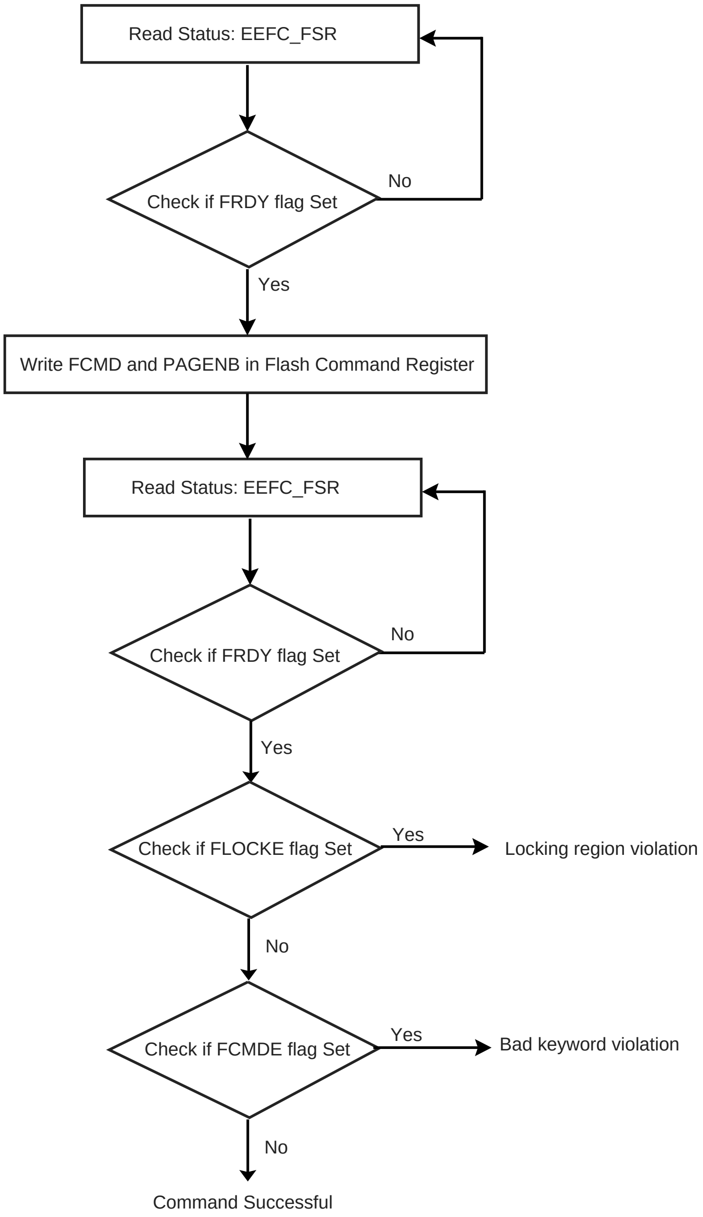
1: Enables the write protection if WPKEY corresponds to 0x525443 (“RTC” in ASCII).

See Section 18.4.8 ”Register Write Protection” for the list of registers that can be write-protected.

- **WPKEY: Write Protection Key**.

Value	Name	Description
0x525443	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Figure 20-7. Command State Chart



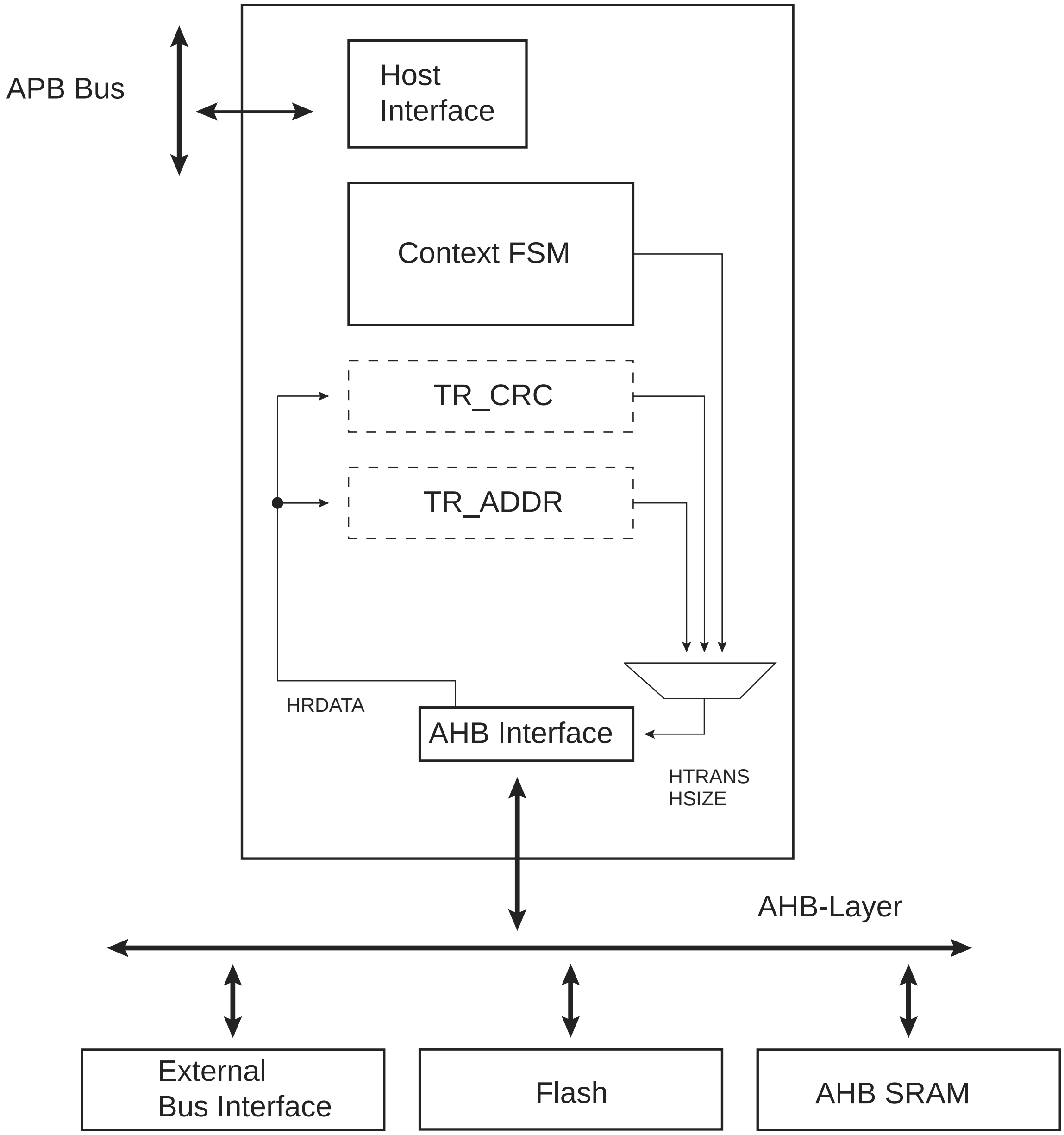
20.4.3.1 Get Flash Descriptor Command

This command provides the system with information on the Flash organization. The system can take full advantage of this information. For instance, a device could be replaced by one with more Flash capacity, and so the software is able to adapt itself to the new configuration.

To get the embedded Flash descriptor, the application writes the GETD command in EEFC_FCR. The first word of the descriptor can be read by the software application in EEFC_FRR as soon as the FRDY flag in EEFC_FSR rises. The next reads of EEFC_FRR provide the following word of the descriptor. If extra read operations to EEFC_FRR are done after the last word of the descriptor has been returned, the EEFC_FRR value is 0 until the next valid command.

23.3 CRCCU Block Diagram

Figure 23-1. Block Diagram



23.7.14 CRCCU Interrupt Mask Register

Name: CRCCU_IMR

Address: 0x40044048

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
–	–	–	–	–	–	–	ERRIMR

- **ERRIMR: CRC Error Interrupt Mask**

0: Interrupt disabled

1: Interrupt enabled

27. Peripheral DMA Controller (PDC)

27.1 Description

The Peripheral DMA Controller (PDC) transfers data between on-chip serial peripherals and the target memories. The link between the PDC and a serial peripheral is operated by the AHB to APB bridge.

The user interface of each PDC channel is integrated into the user interface of the peripheral it serves. The user interface of mono-directional channels (receive-only or transmit-only) contains two 32-bit memory pointers and two 16-bit counters, one set (pointer, counter) for the current transfer and one set (pointer, counter) for the next transfer. The bidirectional channel user interface contains four 32-bit memory pointers and four 16-bit counters. Each set (pointer, counter) is used by the current transmit, next transmit, current receive and next receive.

Using the PDC decreases processor overhead by reducing its intervention during the transfer. This lowers significantly the number of clock cycles required for a data transfer, improving microcontroller performance.

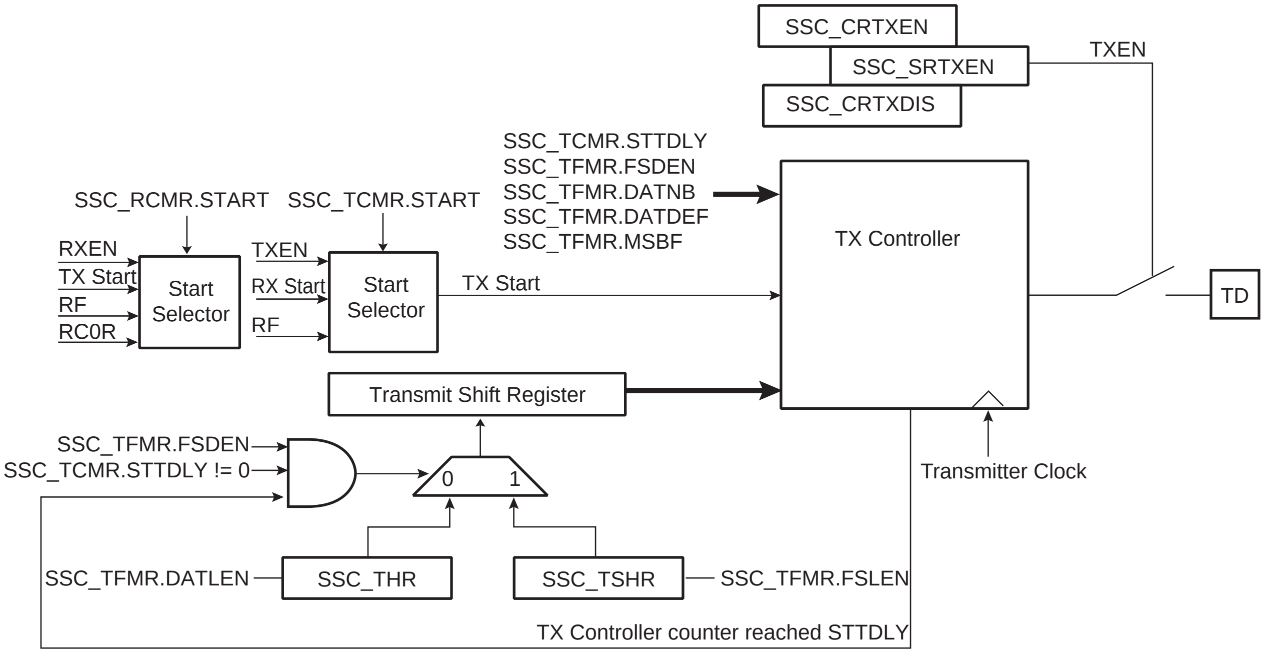
To launch a transfer, the peripheral triggers its associated PDC channels by using transmit and receive signals. When the programmed data is transferred, an end of transfer interrupt is generated by the peripheral itself.

27.2 Embedded Characteristics

- Performs Transfers to/from APB Communication Serial Peripherals
- Supports Half-duplex and Full-duplex Peripherals

When both the SSC_THR and the transmit shift register are empty, the status flag TXEMPTY is set in the SSC_SR. When the Transmit Holding register is transferred in the transmit shift register, the status flag TXRDY is set in the SSC_SR and additional data can be loaded in the holding register.

Figure 32-11. Transmitter Block Diagram



32.8.3 Receiver Operations

A received frame is triggered by a start event and can be followed by synchronization data before data transmission.

The start event is configured setting the Receive Clock Mode Register (SSC_RCMR). See Section 32.8.4 “Start” on page 652.

The frame synchronization is configured setting the Receive Frame Mode Register (SSC_RFMR). See Section 32.8.5 “Frame Sync” on page 654.

The receiver uses a shift register clocked by the receiver clock signal and the start mode selected in the SSC_RCMR. The data is transferred from the shift register depending on the data format selected.

When the receiver shift register is full, the SSC transfers this data in the holding register, the status flag RXRDY is set in the SSC_SR and the data can be read in the receiver holding register. If another transfer occurs before read of the Receive Holding Register (SSC_RHR), the status flag OVERUN is set in the SSC_SR and the receiver shift register is transferred in the SSC_RHR.

35.6.3 UART Interrupt Enable Register

Name: UART_IER

Address: 0x400E0608 (0), 0x400E0808 (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
–	–	–	RXBUFF	TXBUFE	–	TXEMPTY	–

7	6	5	4	3	2	1	0
PARE	FRAME	OVRE	ENDTX	ENDRX	–	TXRDY	RXRDY

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

0: No effect.

1: Enables the corresponding interrupt.

- **RXRDY: Enable RXRDY Interrupt**

- **TXRDY: Enable TXRDY Interrupt**

- **ENDRX: Enable End of Receive Transfer Interrupt**

- **ENDTX: Enable End of Transmit Interrupt**

- **OVRE: Enable Overrun Error Interrupt**

- **FRAME: Enable Framing Error Interrupt**

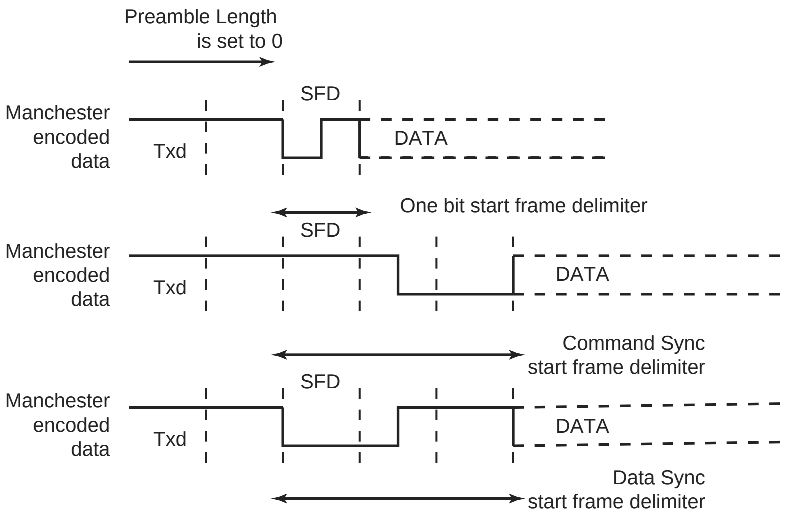
- **PARE: Enable Parity Error Interrupt**

- **TXEMPTY: Enable TXEMPTY Interrupt**

- **TXBUFE: Enable Buffer Empty Interrupt**

- **RXBUFF: Enable Buffer Full Interrupt**

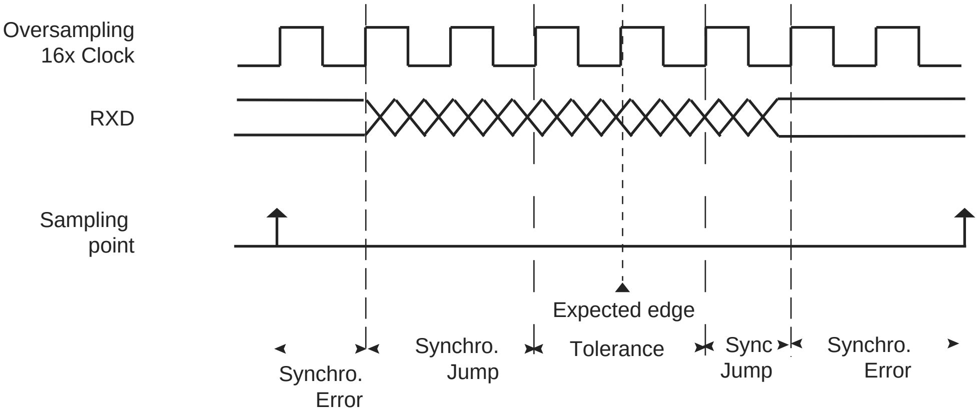
Figure 36-9. Start Frame Delimiter



Drift Compensation

Drift compensation is available only in 16X oversampling mode. An hardware recovery system allows a larger clock drift. To enable the hardware system, the bit in the USART_MAN register must be set. If the RXD edge is one 16X clock cycle from the expected edge, this is considered as normal jitter and no corrective actions is taken. If the RXD event is between 4 and 2 clock cycles before the expected edge, then the current period is shortened by one clock cycle. If the RXD event is between 2 and 3 clock cycles after the expected edge, then the current period is lengthened by one clock cycle. These intervals are considered to be drift and so corrective actions are automatically taken.

Figure 36-10. Bit Resynchronization



36.6.3.3 Asynchronous Receiver

If the USART is programmed in Asynchronous operating mode (SYNC = 0), the receiver oversamples the RXD input line. The oversampling is either 16 or 8 times the baud rate clock, depending on the OVER bit in the US_MR.

The receiver samples the RXD line. If the line is sampled during one half of a bit time to 0, a start bit is detected and data, parity and stop bits are successively sampled on the bit rate clock.

If the oversampling is 16 (OVER = 0), a start is detected at the eighth sample to 0. Data bits, parity bit and stop bit are assumed to have a duration corresponding to 16 oversampling clock cycles. If the oversampling is 8 (OVER = 1), a start bit is detected at the fourth sample to 0. Data bits, parity bit and stop bit are assumed to have a duration corresponding to 8 oversampling clock cycles.

37.7.20 TC Write Protection Mode Register

Name: TC_WPMR

Address: 0x400100E4 (0), 0x400140E4 (1)

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 0x54494D (“TIM” in ASCII).

1: Enables the write protection if WPKEY corresponds to 0x54494D (“TIM” in ASCII).

The Timer Counter clock of the first channel must be enabled to access this register.

See Section 37.6.17 “Register Write Protection” for a list of registers that can be write-protected and Timer Counter clock conditions.

- **WPKEY: Write Protection Key**

Value	Name	Description
0x54494D	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

38.14.1 HSMCI Control Register

Name: HSMCI_CR

Address: 0x40000000

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
SWRST	–	–	–	PWSDIS	PWSEN	MCIDIS	MCIEN

- **MCIEN: Multi-Media Interface Enable**

0: No effect.

1: Enables the Multi-Media Interface if MCDIS is 0.

- **MCIDIS: Multi-Media Interface Disable**

0: No effect.

1: Disables the Multi-Media Interface.

- **PWSEN: Power Save Mode Enable**

0: No effect.

1: Enables the Power Saving Mode if PWSDIS is 0.

Warning: Before enabling this mode, the user must set a value different from 0 in the PWSDIV field of the HSMCI_MR.

- **PWSDIS: Power Save Mode Disable**

0: No effect.

1: Disables the Power Saving Mode.

- **SWRST: Software Reset**

0: No effect.

1: Resets the HSMCI. A software triggered hardware reset of the HSMCI is performed.

- **RTOE: Response Time-out Error Interrupt Enable**

- **DCRCE: Data CRC Error Interrupt Enable**

- **DTOE: Data Time-out Error Interrupt Enable**

- **CSTOE: Completion Signal Timeout Error Interrupt Enable**

- **FIFOEMPTY: FIFO empty Interrupt enable**

- **XFRDONE: Transfer Done Interrupt enable**

- **ACKRCV: Boot Acknowledge Interrupt Enable**

- **ACKRCVE: Boot Acknowledge Error Interrupt Enable**

- **OVRE: Overrun Interrupt Enable**

- **UNRE: Underrun Interrupt Enable**

39.7.24 PWM Fault Status Register

Name: PWM_FSR

Address: 0x40020060

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
FS							

7	6	5	4	3	2	1	0
FIV							

- **FIV: Fault Input Value**

For each bit y of FIV, where y is the fault input number:

0: The current sampled value of the fault input y is 0 (after filtering if enabled).

1: The current sampled value of the fault input y is 1 (after filtering if enabled).

- **FS: Fault Status**

For each bit y of FS, where y is the fault input number:

0: The fault y is not currently active.

1: The fault y is currently active.

41.7.6 ACC Interrupt Status Register

Name: ACC_ISR

Address: 0x40040030

Access: Read-only

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

- **CE: Comparison Edge (cleared on read)**

0: No edge occurred (defined by EDGETYP) on analog comparator output since the last read of ACC_ISR.

1: A selected edge (defined by EDGETYP) on analog comparator output occurred since the last read of ACC_ISR.

- **SCO: Synchronized Comparator Output**

Returns an image of the analog comparator output after being pre-processed (refer to Figure 41-1 on page 1069).

If INV = 0

SCO = 0 if inn > inp

SCO = 1 if inp > inn

If INV = 1

SCO = 1 if inn > inp

SCO = 0 if inp > inn

- **MASK: Flag Mask**

0: The CE flag and SCO value are valid.

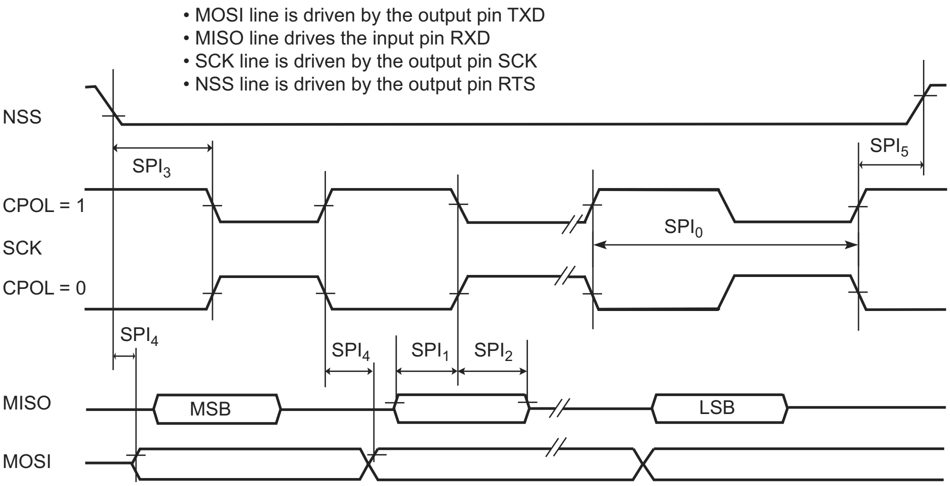
1: The CE flag and SCO value are invalid.

44.12.7 USART in SPI Mode Timings

Timings are given in the following domain:

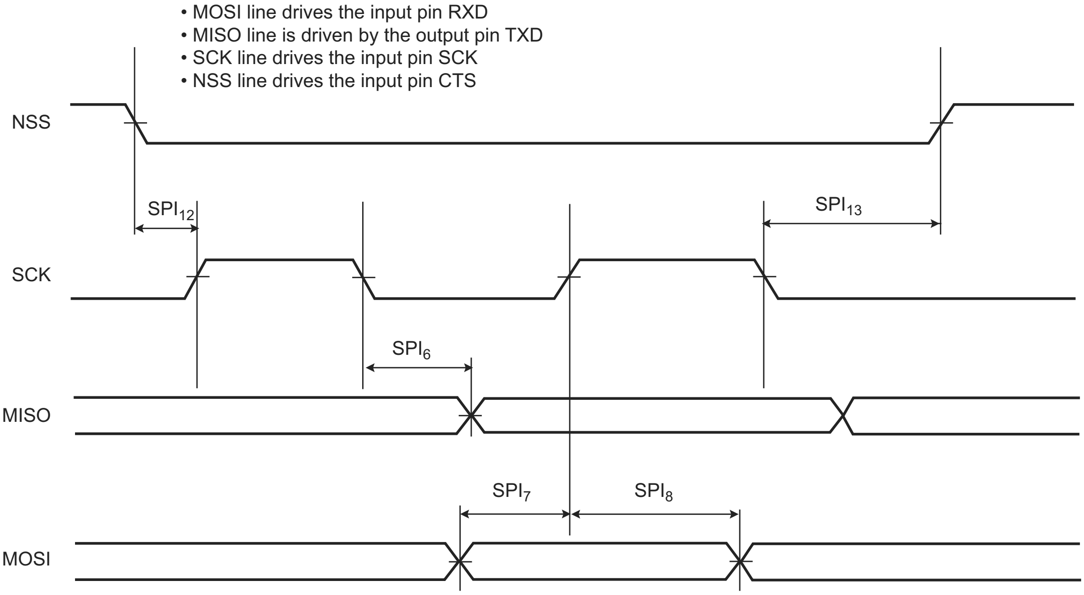
- 1.8V domain: V_{DDIO} from 1.65 to 1.95 V, maximum external capacitor = 20 pF
- 3.3V domain: V_{DDIO} from 2.85 to 3.6 V, maximum external capacitor = 40 pF

Figure 44-37. USART SPI Master Mode



- MOSI line is driven by the output pin TXD
- MISO line drives the input pin RXD
- SCK line is driven by the output pin SCK
- NSS line is driven by the output pin RTS

Figure 44-38. USART SPI Slave Mode: (Mode 1 or 2)



- MOSI line drives the input pin RXD
- MISO line is driven by the output pin TXD
- SCK line drives the input pin SCK
- NSS line drives the input pin CTS

Table 49-3. SAM4S Datasheet Rev. 11100I Revision History (Continued)

Doc. Date	Changes
03-Apr-15	Section 34.8, “Two-wire Interface (TWI) User Interface”: - removed line “Reset:” from above all register tables; redundant with Table 34-7 ”Register Mapping”. - replaced instance of acronym “TWI2” with “TWI” in section title Figure 34-4 Master Mode Typical Application Block Diagram: renamed lines from host with SDA and SCL; placed TWD and TWCK in host block Updated Section 34.7.3.4, “Master Transmitter Mode”, Section 34.7.3.5 “Master Receiver Mode”, Figure 34-23 Slave Mode Typical Application Block Diagram:, Figure 34-24 Read Access Ordered by a Master, Figure 34-25 Write Access Ordered by a Master, Figure 34-26 Master Performs a General Call, Figure 34-28 Clock Synchronization in Write Mode, Figure 34-31 Read Write Flowchart in Slave Mode, Section 34.8.1, “TWI Control Register”, Section 34.8.5, “TWI Clock Waveform Generator Register” and Section 34.8.6, “TWI Status Register”
	Section 35., “Universal Asynchronous Receiver Transmitter (UART)” Figure 35-1 UART Block Diagram: changed ‘MCK’ to ‘peripheral clock’; added ‘bus clock’ to diagram. In diagrams and equations: changed ‘MCK’ to ‘peripheral clock’ Figure 35-2 Baud Rate Generator: removed ‘(PMC)’ from figure. Section 35.5.1, “Baud Rate Generator”: deleted equation for Baud Rate calculation. Now in Section 35.6.9, “UART Baud Rate Generator Register”. Section 35.5.2.4, “Receiver Overrun”: replaced “(or the Peripheral Data Controller or DMA Controller)” with “(or the PDC)” Section 35.5.4, “Peripheral DMA Controller (PDC)”: replaced “peripheral data controller” with “PDC” Section 35.6.6, “UART Status Register”: replaced “Peripheral Data Controller” with “PDC” in ENDRX and ENDTX bit descriptions Section 35.6.9, “UART Baud Rate Generator Register”: added equation to CD bit description.
	Section 36., “Universal Synchronous Asynchronous Receiver Transceiver (USART)” Replaced all references to ‘MCK’ with ’peripheral clock’ Removed section “Application Block Diagram” Modified Figure 36-1, “USART Block Diagram”, Figure 36-3, “Fractional Baud Rate Generator” and Figure 36-36, “Example of RTS Drive with Timeguard” Updated Section 36.2, “Embedded Characteristics” Modified Table 36-15 “Register Mapping” Updated Table 36-7 “Possible Values for the Fi/Di Ratio” Modified Section 36.5.3, “Interrupt Sources” and Section 36.6.1, “Baud Rate Generator” Removed all references to bit RXIDLEV Replaced bit ITERATION with bit ITER (US_CSR register) Section 36.6.1.3, “Baud Rate in Synchronous Mode or SPI Mode”: replaced “CLK0” with “CLKO” Modified Section 36.6.3.8 “Parity”, Section 36.6.3.3 “Asynchronous Receiver” Section 36.6.3.8 “Parity”, ”Transmit Character Repetition” and ”Disable Successive Receive NACK” Modified Section 36.6.10, “Register Write Protection”, Section 36.7.1, “USART Control Register”, Section 36.7.3, “USART Mode Register” , , Section 36.7.5, “USART Interrupt Enable Register”, Section 36.7.6, “USART Interrupt Enable Register (SPI_MODE)”, Section 36.7.7, “USART Interrupt Disable Register”, Section 36.7.8, “USART Interrupt Disable Register (SPI_MODE)”, Section 36.7.9, “USART Interrupt Mask Register”, Section 36.7.10, “USART Interrupt Mask Register (SPI_MODE)”, Section 36.7.11, “USART Channel Status Register”, Section 36.7.12, “USART Channel Status Register (SPI_MODE)”, Section 36.7.15, “USART Baud Rate Generator Register”, Section 36.7.16, “USART Receiver Time-out Register”, Section 36.7.17, “USART Transmitter Timeguard Register” and Section 36.7.21, “USART Manchester Configuration Register”