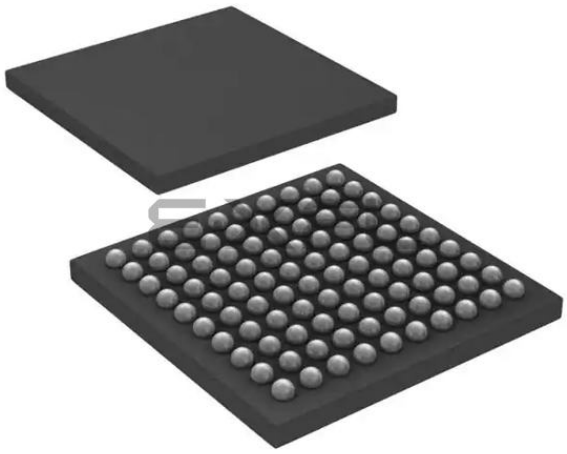


Welcome to E-XFL.COM

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	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	1MB (1M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128K 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 ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	100-VFBGA
Supplier Device Package	100-VFBGA (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsam4s16cb-cfnr

Examples

```
SMUAD    R0, R4, R5  ; Multiplies bottom halfword of R4 with the bottom
                     ; halfword of R5, adds multiplication of top halfword
                     ; of R4 with top halfword of R5, writes to R0
SMUADX   R3, R7, R4  ; Multiplies bottom halfword of R7 with top halfword
                     ; of R4, adds multiplication of top halfword of R7
                     ; with bottom halfword of R4, writes to R3
SMUSD    R3, R6, R2  ; Multiplies bottom halfword of R4 with bottom halfword
                     ; of R6, subtracts multiplication of top halfword of R6
                     ; with top halfword of R3, writes to R3
SMUSDX   R4, R5, R3  ; Multiplies bottom halfword of R5 with top halfword of
                     ; R3, subtracts multiplication of top halfword of R5
                     ; with bottom halfword of R3, writes to R4.
```

12.6.6.10 SMUL and SMULW

Signed Multiply (halfwords) and Signed Multiply (word by halfword)

Syntax

op{*XY*}{*cond*} *Rd*,*Rn*, *Rm*
op{*Y*}{*cond*} *Rd*. *Rn*, *Rm*

For *SMULXY* only:

op is one of:

SMUL{*XY*} Signed Multiply (halfwords).

X and *Y* specify which halfword of the source registers *Rn* and *Rm* is 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 bot tom halfword, bits [15:0], of *Rm* is used.
If *Y* is T, then the top halfword, bits [31:16], of *Rm* is used.

SMULW{Y} Signed Multiply (word by halfword).

Y specifies which halfword of the source register *Rm* is used as the second mul tiply operand.
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.

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

Rd is the destination register.

Rn, Rm are registers holding the first and second operands.

Operation

The SMULBB, SMULTB, SMULBT and SMULTT instructions interprets the values from *Rn* and *Rm* as four signed 16-bit integers. These instructions:

- Multiplies the specified signed halfword, Top or Bottom, values from *Rn* and *Rm*.
- Writes the 32-bit result of the multiplication in *Rd.*

The SMULWT and SMULWB instructions interprets the values from *Rn* as a 32-bit signed integer and *Rm* as two halfword 16-bit signed integers. These instructions:

- Multiplies the first operand and the top, T suffix, or the bottom, B suffix, halfword of the second operand.
- Writes the signed most significant 32 bits of the 48-bit result in the destination register.

Restrictions

12.8 Nested Vectored Interrupt Controller (NVIC)

This section describes the NVIC and the registers it uses. The NVIC supports:

- Up to 35 interrupts
- A programmable priority level of 0–15 for each interrupt. A higher level corresponds to a lower priority, so level 0 is the highest interrupt priority.
- Level detection of interrupt signals
- Dynamic reprioritization of interrupts
- Grouping of priority values into group priority and subpriority fields
- Interrupt tail-chaining
- An external *Non-maskable interrupt* (NMI)

The processor automatically stacks its state on exception entry and unstacks this state on exception exit, with no instruction overhead. This provides low latency exception handling.

12.8.1 Level-sensitive Interrupts

The processor supports level-sensitive interrupts. A level-sensitive interrupt is held asserted until the peripheral deasserts the interrupt signal. Typically, this happens because the ISR accesses the peripheral, causing it to clear the interrupt request.

When the processor enters the ISR, it automatically removes the pending state from the interrupt (see “Hardware and Software Control of Interrupts”). For a level-sensitive interrupt, if the signal is not deasserted before the processor returns from the ISR, the interrupt becomes pending again, and the processor must execute its ISR again. This means that the peripheral can hold the interrupt signal asserted until it no longer requires servicing.

12.8.1.1 Hardware and Software Control of Interrupts

The Cortex-M4 latches all interrupts. A peripheral interrupt becomes pending for one of the following reasons:

- The NVIC detects that the interrupt signal is HIGH and the interrupt is not active
- The NVIC detects a rising edge on the interrupt signal
- A software writes to the corresponding interrupt set-pending register bit, see “Interrupt Set-pending Registers” , or to the NVIC_STIR to make an interrupt pending, see “Software Trigger Interrupt Register” .

A pending interrupt remains pending until one of the following:

- The processor enters the ISR for the interrupt. This changes the state of the interrupt from pending to active. Then:
  - For a level-sensitive interrupt, when the processor returns from the ISR, the NVIC samples the interrupt signal. If the signal is asserted, the state of the interrupt changes to pending, which might cause the processor to immediately re-enter the ISR. Otherwise, the state of the interrupt changes to inactive.
- Software writes to the corresponding interrupt clear-pending register bit.
  For a level-sensitive interrupt, if the interrupt signal is still asserted, the state of the interrupt does not change. Otherwise, the state of the interrupt changes to inactive.

12.8.2 NVIC Design Hints and Tips

Ensure that the software uses correctly aligned register accesses. The processor does not support unaligned accesses to NVIC registers. See the individual register descriptions for the supported access sizes.

A interrupt can enter a pending state even if it is disabled. Disabling an interrupt only prevents the processor from taking that interrupt.

14.5.2 Reset Controller Status Register

Name: RSTC_SR

Address: 0x400E1404

Access: Read-only

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

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

15	14	13	12	11	10	9	8
–	–	–	–	–	RSTTYP		

7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	URSTS

- **URSTS: User Reset Status**

A high-to-low transition of the NRST pin sets the URSTS bit. This transition is also detected on the MCK rising edge. If the user reset is disabled (URSTEN = 0 in RSTC_MR) and if the interruption is enabled by the URSTIEN bit in the RSTC_MR, the URSTS bit triggers an interrupt. Reading the RSTC_SR resets the URSTS bit and clears the interrupt.

0: No high-to-low edge on NRST happened since the last read of RSTC_SR.

1: At least one high-to-low transition of NRST has been detected since the last read of RSTC_SR.

- **RSTTYP: Reset Type**

This field reports the cause of the last processor reset. Reading this RSTC_SR does not reset this field.

Value	Name	Description
0	GENERAL_RST	First power-up reset
1	BACKUP_RST	Return from Backup Mode
2	WDT_RST	Watchdog fault occurred
3	SOFT_RST	Processor reset required by the software
4	USER_RST	NRST pin detected low
5	–	Reserved
6	–	Reserved
7	–	Reserved

- **NRSTL: NRST Pin Level**

This bit registers the NRST pin level sampled on each Master Clock (MCK) rising edge.

- **SRCMP: Software Reset Command in Progress**

When set, this bit indicates that a software reset command is in progress and that no further software reset should be performed until the end of the current one. This bit is automatically cleared at the end of the current software reset.

0: No software command is being performed by the Reset Controller. The Reset Controller is ready for a software command.

1: A software reset command is being performed by the Reset Controller. The Reset Controller is busy.

the final product manufacturing by means of measurement equipment embedding such a reference clock. The correction of value must be programmed into the (RTC_MR), and this value is kept as long as the circuitry is powered (backup area). Removing the backup power supply cancels this calibration. This room temperature calibration can be further processed by means of the networking capability of the target application.

To ease the comparison of the inherent crystal accuracy with the reference clock/signal during manufacturing, an internal prescaled 32.768 kHz clock derivative signal can be assigned to drive RTC output. To accommodate the measure, several clock frequencies can be selected among 1 Hz, 32 Hz, 64 Hz, 512 Hz.

The clock calibration correction drives the internal RTC counters but can also be observed in the RTC output when one of the following three frequencies 1 Hz, 32 Hz or 64 Hz is configured. The correction is not visible in the RTC output if 512 Hz frequency is configured.

In any event, this adjustment does not take into account the temperature variation.

The frequency drift (up to -200 ppm) due to temperature variation can be compensated using a reference time if the application can access such a reference. If a reference time cannot be used, a temperature sensor can be placed close to the crystal oscillator in order to get the operating temperature of the crystal oscillator. Once obtained, the temperature may be converted using a lookup table (describing the accuracy/temperature curve of the crystal oscillator used) and RTC_MR configured accordingly. The calibration can be performed on-the-fly. This adjustment method is not based on a measurement of the crystal frequency/drift and therefore can be improved by means of the networking capability of the target application.

If no crystal frequency adjustment has been done during manufacturing, it is still possible to do it. In the case where a reference time of the day can be obtained through LAN/WAN network, it is possible to calculate the drift of the application crystal oscillator by comparing the values read on RTC Time Register (RTC_TIMR) and programming the HIGHPPM and CORRECTION fields on RTC_MR according to the difference measured between the reference time and those of RTC_TIMR.

16.5.8 Waveform Generation

Waveforms can be generated by the RTC in order to take advantage of the RTC inherent prescalers while the RTC is the only powered circuitry (low power mode of operation, backup mode) or in any active modes. Going into backup or low power operating modes does not affect the waveform generation outputs.

The RTC outputs (RTCOUT0 and RTCOUT1) have a source driver selected among seven possibilities.

The first selection choice sticks the associated output at 0 (This is the reset value and it can be used at any time to disable the waveform generation).

Selection choices 1 to 4 respectively select 1 Hz, 32 Hz, 64 Hz and 512 Hz.

32 Hz or 64 Hz can drive, for example, a TN LCD backplane signal while 1 Hz can be used to drive a blinking character like “:” for basic time display (hour, minute) on TN LCDs.

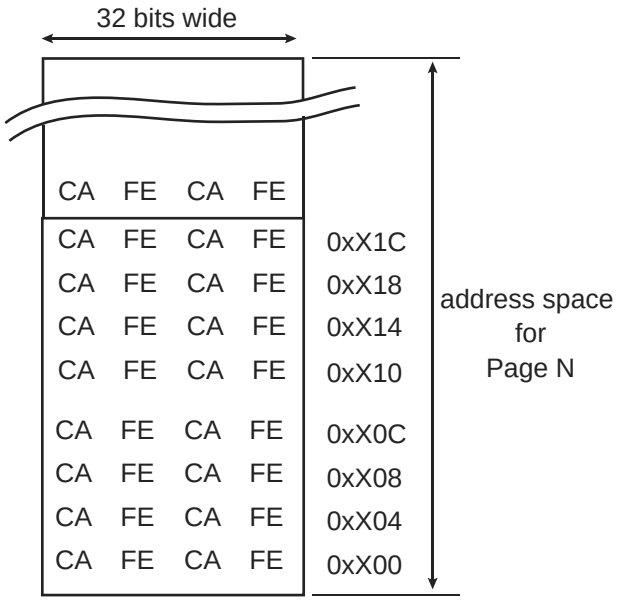
Selection choice 5 provides a toggling signal when the RTC alarm is reached.

Selection choice 6 provides a copy of the alarm flag, so the associated output is set high (logical 1) when an alarm occurs and immediately cleared when software clears the alarm interrupt source.

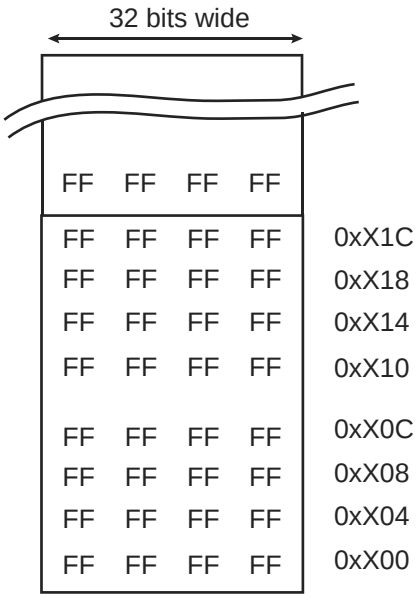
Selection choice 7 provides a 1 Hz periodic high pulse of 15 µs duration that can be used to drive external devices for power consumption reduction or any other purpose.

PIO lines associated to RTC outputs are automatically selecting these waveforms as soon as RTC_MR corresponding fields OUT0 and OUT1 differ from 0.

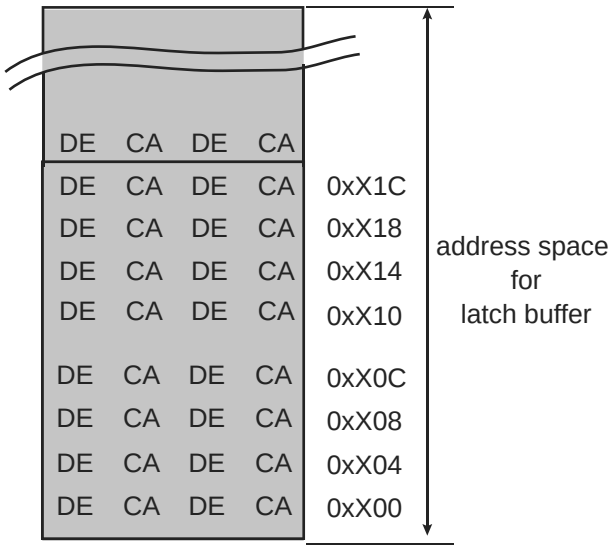
Figure 20-8. Full Page Programming



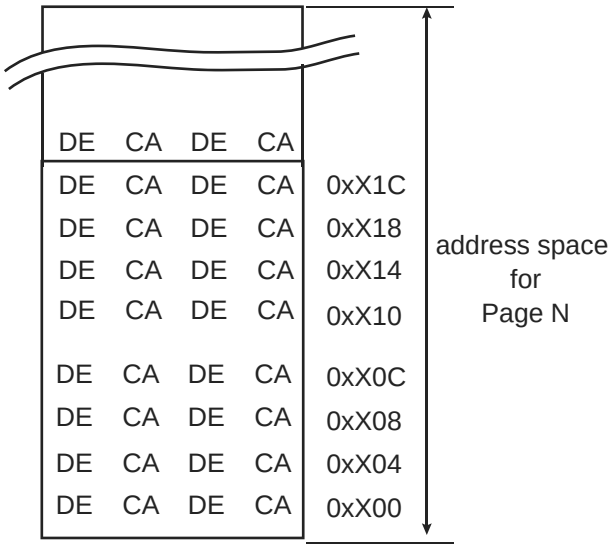
Before programming: Unerased page in Flash array



Step 1: Flash array after page erase



Step 2: Writing a page in the latch buffer



Step 3: Page in Flash array after issuing WP command and FRDY=1

Three round-robin algorithms are implemented:

- Round-Robin arbitration without default master
- Round-Robin arbitration with last access master
- Round-Robin arbitration with fixed default master

25.5.2.1 Round-Robin arbitration without default master

Round-robin arbitration without default master is the main algorithm used by Bus Matrix arbiters. It allows the Bus Matrix to dispatch requests from different masters to the same slave in a pure round-robin manner. At the end of the current access, if no other request is pending, the slave is disconnected from all masters. This configuration incurs one latency cycle for the first access of a burst. Arbitration without default master can be used for masters that perform significant bursts.

25.5.2.2 Round-Robin arbitration with last access master

Round-robin arbitration with last access master is a biased round-robin algorithm used by Bus Matrix arbiters. It allows the Bus Matrix to remove the one latency cycle for the last master that accessed the slave. At the end of the current transfer, if no other master request is pending, the slave remains connected to the last master that performs the access. Other non-privileged masters incur one latency cycle if they want to access the same slave. This technique can be used for masters that mainly perform single accesses.

25.5.2.3 Round-Robin arbitration with fixed default master

Round-robin arbitration with fixed default master is an algorithm used by the Bus Matrix arbiters to remove the one latency cycle for the fixed default master per slave. At the end of the current access, the slave remains connected to its fixed default master. Every request attempted by the fixed default master does not incur latency, whereas other non-privileged masters still incur one latency cycle. This technique can be used for masters that mainly perform single accesses.

25.5.3 Fixed Priority Arbitration

The fixed priority arbitration algorithm is used by the Bus Matrix arbiters to dispatch the requests from different masters to the same slave by using the fixed priority defined by the user. If requests from two or more masters are active at the same time, the master with the highest priority is serviced first. If requests from two or more masters with the same priority are active at the same time, the master with the highest number is serviced first.

For each slave, the priority of each master may be defined through the Priority registers for slaves (MATRIX_PRAS and MATRIX_PRBS).

25.6 System I/O Configuration

The System I/O Configuration register (CCFG_SYSIO) configures I/O lines in system I/O mode (such as JTAG, ERASE, USB, etc.) or as general-purpose I/O lines. Enabling or disabling the corresponding I/O lines in peripheral mode or in PIO mode (PIO_PER or PIO_PDR registers) in the PIO controller has no effect. However, the direction (input or output), pull-up, pull-down and other mode control is still managed by the PIO controller.

25.8.5 SMC NAND Flash Chip Select Configuration Register

Name: CCFG_SMCNFCS

Address: 0x400E031C

Type: Read/Write

Reset: 0x0000_0000

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
–	–	–	–	SMC_NFCS3	SMC_NFCS2	SMC_NFCS1	SMC_NFCS0

- **SMC_NFCS0: SMC NAND Flash Chip Select 0 Assignment**

0: NCS0 is not assigned to a NAND Flash (NANDOE and NANWE not used for NCS0)

1: NCS0 is assigned to a NAND Flash (NANDOE and NANWE used for NCS0)

- **SMC_NFCS1: SMC NAND Flash Chip Select 1 Assignment**

0: NCS1 is not assigned to a NAND Flash (NANDOE and NANWE not used for NCS1)

1: NCS1 is assigned to a NAND Flash (NANDOE and NANWE used for NCS1)

- **SMC_NFCS2: SMC NAND Flash Chip Select 2 Assignment**

0: NCS2 is not assigned to a NAND Flash (NANDOE and NANWE not used for NCS2)

1: NCS2 is assigned to a NAND Flash (NANDOE and NANWE used for NCS2)

- **SMC_NFCS3: SMC NAND Flash Chip Select 3 Assignment**

0: NCS3 is not assigned to a NAND Flash (NANDOE and NANWE not used for NCS3)

1: NCS3 is assigned to a NAND Flash (NANDOE and NANWE used for NCS3)

29.17.15PMC Interrupt Disable Register

Name: PMC_IDR

Address: 0x400E0464

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	CFDEV	MOSCRCS	MOSCSELS
15	14	13	12	11	10	9	8
–	–	–	–	–	PCKRDY2	PCKRDY1	PCKRDY0
7	6	5	4	3	2	1	0
–	–	–	–	MCKRDY	LOCKB	LOCKA	MOSCXTS

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

0: No effect.

1: Disables the corresponding interrupt.

- **MOSCXTS: Main Crystal Oscillator Status Interrupt Disable**
- **LOCKA: PLLA Lock Interrupt Disable**
- **LOCKB: PLLB Lock Interrupt Disable**
- **MCKRDY: Master Clock Ready Interrupt Disable**
- **PCKRDYx: Programmable Clock Ready x Interrupt Disable**
- **MOSCSELS: Main Oscillator Selection Status Interrupt Disable**
- **MOSCRCS: Main On-Chip RC Status Interrupt Disable**
- **CFDEV: Clock Failure Detector Event Interrupt Disable**

32.9.3 SSC Receive Clock Mode Register

Name: SSC_RCMR

Address: 0x40004010

Access: Read/Write

31	30	29	28	27	26	25	24
PERIOD							
23	22	21	20	19	18	17	16
STTDLY							
15	14	13	12	11	10	9	8
–	–	–	STOP	START			
7	6	5	4	3	2	1	0
CKG		CKI	CKO			CKS	

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

- **CKS: Receive Clock Selection**

Value	Name	Description
0	MCK	Divided Clock
1	TK	TK Clock signal
2	RK	RK pin

- **CKO: Receive Clock Output Mode Selection**

Value	Name	Description
0	NONE	None, RK pin is an input
1	CONTINUOUS	Continuous Receive Clock, RK pin is an output
2	TRANSFER	Receive Clock only during data transfers, RK pin is an output

- **CKI: Receive Clock Inversion**

0: The data inputs (Data and Frame Sync signals) are sampled on Receive Clock falling edge. The Frame Sync signal output is shifted out on Receive Clock rising edge.

1: The data inputs (Data and Frame Sync signals) are sampled on Receive Clock rising edge. The Frame Sync signal output is shifted out on Receive Clock falling edge.

CKI affects only the Receive Clock and not the output clock signal.

32.9.13 SSC Status Register

Name: SSC_SR

Address: 0x40004040

Access: Read-only

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

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

15	14	13	12	11	10	9	8
–	–	–	–	RXSYN	TXSYN	CP1	CP0

7	6	5	4	3	2	1	0
RXBUFF	ENDRX	OVRUN	RXRDY	TXBUFE	ENDTX	TXEMPTY	TXRDY

- **TXRDY: Transmit Ready**

0: Data has been loaded in SSC_THR and is waiting to be loaded in the transmit shift register (TSR).

1: SSC_THR is empty.

- **TXEMPTY: Transmit Empty**

0: Data remains in SSC_THR or is currently transmitted from TSR.

1: Last data written in SSC_THR has been loaded in TSR and last data loaded in TSR has been transmitted.

- **ENDTX: End of Transmission**

0: The register SSC_TCR has not reached 0 since the last write in SSC_TCR or SSC_TNCR.

1: The register SSC_TCR has reached 0 since the last write in SSC_TCR or SSC_TNCR.

- **TXBUFE: Transmit Buffer Empty**

0: SSC_TCR or SSC_TNCR have a value other than 0.

1: Both SSC_TCR and SSC_TNCR have a value of 0.

- **RXRDY: Receive Ready**

0: SSC_RHR is empty.

1: Data has been received and loaded in SSC_RHR.

- **OVRUN: Receive Overrun**

0: No data has been loaded in SSC_RHR while previous data has not been read since the last read of the Status Register.

1: Data has been loaded in SSC_RHR while previous data has not yet been read since the last read of the Status Register.

- **ENDRX: End of Reception**

0: Data is written on the Receive Counter Register or Receive Next Counter Register.

1: End of PDC transfer when Receive Counter Register has arrived at zero.

After a master write transfer, the SCL is stretched (tied low) as long as no new data is written in the TWI_THR or until a STOP command is performed.

See Figure 34-5, Figure 34-6, and Figure 34-7.

Figure 34-5. Master Write with One Data Byte

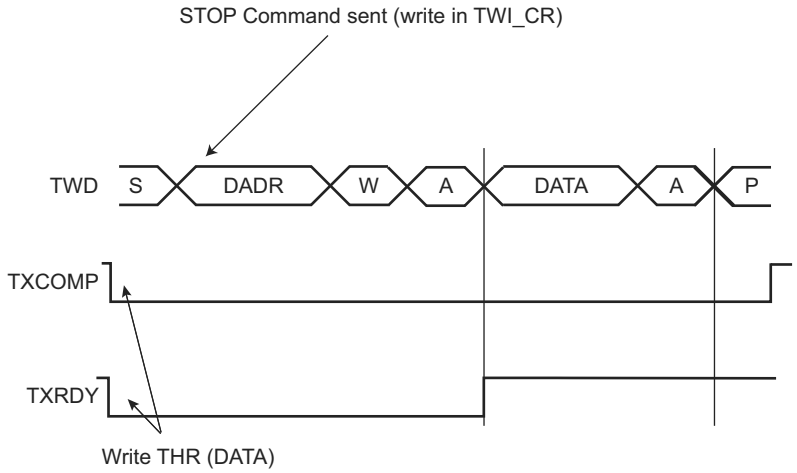
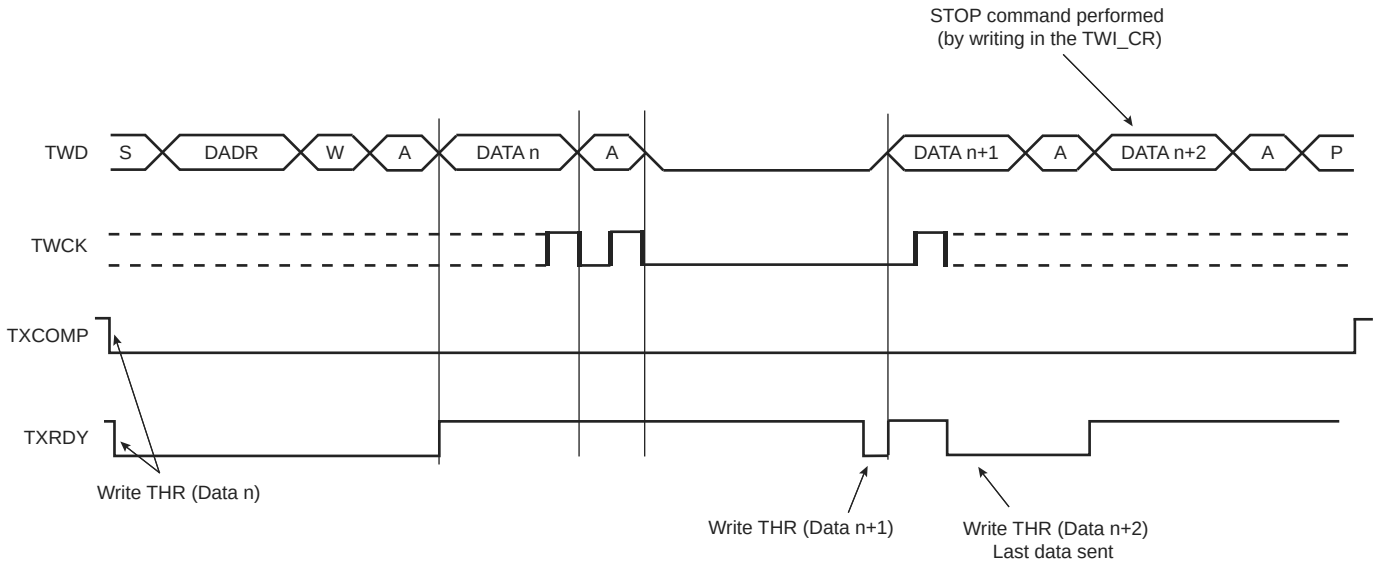


Figure 34-6. Master Write with Multiple Data Bytes



35.6.2 UART Mode Register

Name: UART_MR

Address: 0x400E0604 (0), 0x400E0804 (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
CHMODE		–	–	PAR			–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

- **PAR: Parity Type**

Value	Name	Description
0	EVEN	Even Parity
1	ODD	Odd Parity
2	SPACE	Space: parity forced to 0
3	MARK	Mark: parity forced to 1
4	NO	No parity

- **CHMODE: Channel Mode**

Value	Name	Description
0	NORMAL	Normal mode
1	AUTOMATIC	Automatic echo
2	LOCAL_LOOPBACK	Local loopback
3	REMOTE_LOOPBACK	Remote loopback

36.7.8 USART Interrupt Disable Register (SPI_MODE)

Name: US_IDR (SPI_MODE)

Address: 0x4002400C (0), 0x4002800C (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	UNRE	TXEMPTY	–
7	6	5	4	3	2	1	0
–	–	OVRE	ENDTX	ENDRX	–	TXRDY	RXRDY

This configuration is relevant only if USART_MODE = 0xE or 0xF in the USART Mode Register.

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

0: No effect

1: Disables the corresponding interrupt.

- **RXRDY: RXRDY Interrupt Disable**
- **TXRDY: TXRDY Interrupt Disable**
- **ENDRX: End of Receive Buffer Transfer Interrupt Disable**
- **ENDTX: End of Transmit Buffer Interrupt Disable**
- **OVRE: Overrun Error Interrupt Disable**
- **TXEMPTY: TXEMPTY Interrupt Disable**
- **UNRE: SPI Underrun Error Interrupt Disable**
- **TXBUFE: Transmit Buffer Empty Interrupt Disable**
- **RXBUFF: Receive Buffer Full Interrupt Disable**

37.7.3 TC Channel Mode Register: Waveform Mode

Name: TC_CMRx [x=0..2] (WAVEFORM_MODE)

Address: 0x40010004 (0)[0], 0x40010044 (0)[1], 0x40010084 (0)[2], 0x40014004 (1)[0], 0x40014044 (1)[1], 0x40014084 (1)[2]

Access: Read/Write

31	30	29	28	27	26	25	24
BSWTRG		BEEVT		BCPC		BCPB	

23	22	21	20	19	18	17	16
ASWTRG		AEEVT		ACPC		ACPA	

15	14	13	12	11	10	9	8
WAVE	WAVSEL		ENETRG	EEVT		EEVTEDG	

7	6	5	4	3	2	1	0
CPCDIS	CPCSTOP	BURST		CLKI	TCCLKS		

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

- **TCCLKS: Clock Selection**

Value	Name	Description
0	TIMER_CLOCK1	Clock selected: internal MCK/2 clock signal (from PMC)
1	TIMER_CLOCK2	Clock selected: internal MCK/8 clock signal (from PMC)
2	TIMER_CLOCK3	Clock selected: internal MCK/32 clock signal (from PMC)
3	TIMER_CLOCK4	Clock selected: internal MCK/128 clock signal (from PMC)
4	TIMER_CLOCK5	Clock selected: internal SLCK clock signal (from PMC)
5	XC0	Clock selected: XC0
6	XC1	Clock selected: XC1
7	XC2	Clock selected: XC2

- **CLKI: Clock Invert**

0: Counter is incremented on rising edge of the clock.

1: Counter is incremented on falling edge of the clock.

- **BURST: Burst Signal Selection**

Value	Name	Description
0	NONE	The clock is not gated by an external signal.
1	XC0	XC0 is ANDed with the selected clock.
2	XC1	XC1 is ANDed with the selected clock.
3	XC2	XC2 is ANDed with the selected clock.

- **CPCSTOP: Counter Clock Stopped with RC Compare**

0: Counter clock is not stopped when counter reaches RC.

1: Counter clock is stopped when counter reaches RC.

38.14.11HSMCI Transmit Data Register

Name: HSMCI_TDR

Address: 0x40000034

Access: Write-only

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

- **DATA: Data to Write**

40.6.2.4 Stall Handshake

A stall handshake can be used in one of two distinct occasions. (For more information on the stall handshake, refer to Chapter 8 of the *Universal Serial Bus Specification, Rev 2.0.*)

- A functional stall is used when the halt feature associated with the endpoint is set. (Refer to Chapter 9 of the *Universal Serial Bus Specification, Rev 2.0,* for more information on the halt feature.)
- To abort the current request, a protocol stall is used, but uniquely with control transfer.

The following procedure generates a stall packet:

1. The microcontroller sets the FORCESTALL flag in the UDP_CSRx endpoint's register.
2. The host receives the stall packet.
3. The microcontroller is notified that the device has sent the stall by polling the STALLSENT to be set. An endpoint interrupt is pending while STALLSENT is set. The microcontroller must clear STALLSENT to clear the interrupt.

When a setup transaction is received after a stall handshake, STALLSENT must be cleared in order to prevent interrupts due to STALLSENT being set.

Figure 40-12. Stall Handshake (Data IN Transfer)

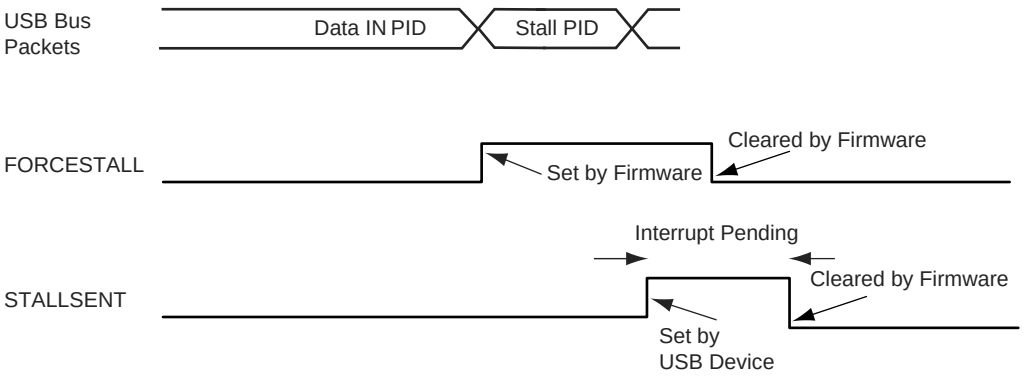
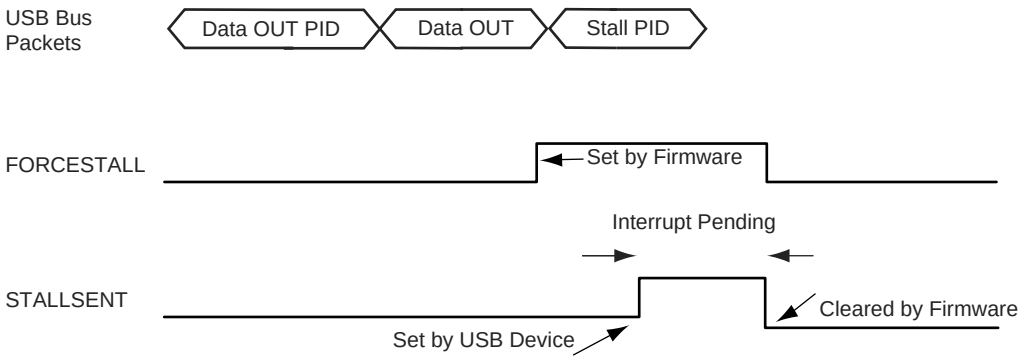


Figure 40-13. Stall Handshake (Data OUT Transfer)



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.

Table 44-3. DC Characteristics (Continued)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$R_{PULLDOWN}$	Pull-down Resistor	PA0–PA31, PB0–PB14, PC0–PC31, NRST	70	100	130	kΩ
R_{ODT}	On-die Series Termination Resistor	PA4–PA31, PB0–PB9, PB12–PB14, PC0–PC31	–	36	–	Ω
		PA0–PA3	–	18	–	
I_{CC}	Flash Active Current on VDDCORE	Random 144-bit Read @ 25°C: Maximum read frequency at VDDCORE = 1.2V, VDDIO = 3.3V	–	16	25	mA
		Random 72-bit Read @ 25°C: Maximum read frequency at VDDCORE = 1.2V, VDDIO = 3.3V	–	10	18	
		Program(3) onto VDDCORE = 1.2V, VDDIO = 3.3V @ 25°C	–	3	5	
I_{CC33}	Flash Active Current on VDDIO	Random 144-bit read: Maximum read frequency at VDDCORE = 1.2V, VDDIO = 3.3V @ 25°C	–	3	16	mA
		Random 72-bit read: Maximum read frequency at VDDCORE = 1.2V, VDDIO = 3.3V @ 25°C	–	3	5	
		Program(3) onto VDDCORE = 1.2V, VDDIO = 3.3V @ 25°C	–	10	15	

Note: 1. PA[4–11], PA[15–25], PB[0–9], PB[12–14], PC[0–31]
2. Refer to Section 5.2.2 “VDDIO Versus VDDIN”
3. The Flash programming characteristics are applicable at operating temperature range: T_A = -40 to 85 °C.

44.4.3.3 SAM4SD32/SD16/SA16 Active Power Consumption

Table 44-23. SAM4SD32/SA16/SD16 Active Power Consumption with VDDCORE @ 1.2V running from Flash Memory (IDDCORE- AMP1) or SRAM

	CoreMark					
	Cache Enable (CE)		Cache Disable (CD)			
Core Clock (MHz)	128-bit Flash access(1)	64-bit Flash access(1)	128-bit Flash access(1)	64-bit Flash access(1)	SRAM	Unit
120	20.7	20.7	24.8	18.1	19.9	mA
100	17.5	17.4	21.6	16.5	16.8	
84	14.7	14.7	18.3	13.9	14.2	
64	11.3	11.3	14.4	11.5	10.9	
48	8.5	8.5	11.3	9.4	8.3	
32	5.7	5.7	8.0	7.1	5.6	
24	4.3	4.3	6.3	5.9	4.2	
12	2.5	2.5	3.5	3.3	1.9	
8	1.7	1.7	2.4	2.3	1.7	
4	0.9	0.9	1.2	1.2	0.7	
2	0.5	0.5	0.7	0.7	0.5	
1	0.4	0.4	0.5	0.5	0.4	
0.5	0.3	0.3	0.3	0.3	0.3	

Note: 1. Flash Wait State (FWS) in EEFC_FMR adjusted versus core frequency

Figure 44-12. SAM4SD32/SD16/SA16 Current Consumption in Active Mode (AMP1) versus Master Clock Ranges

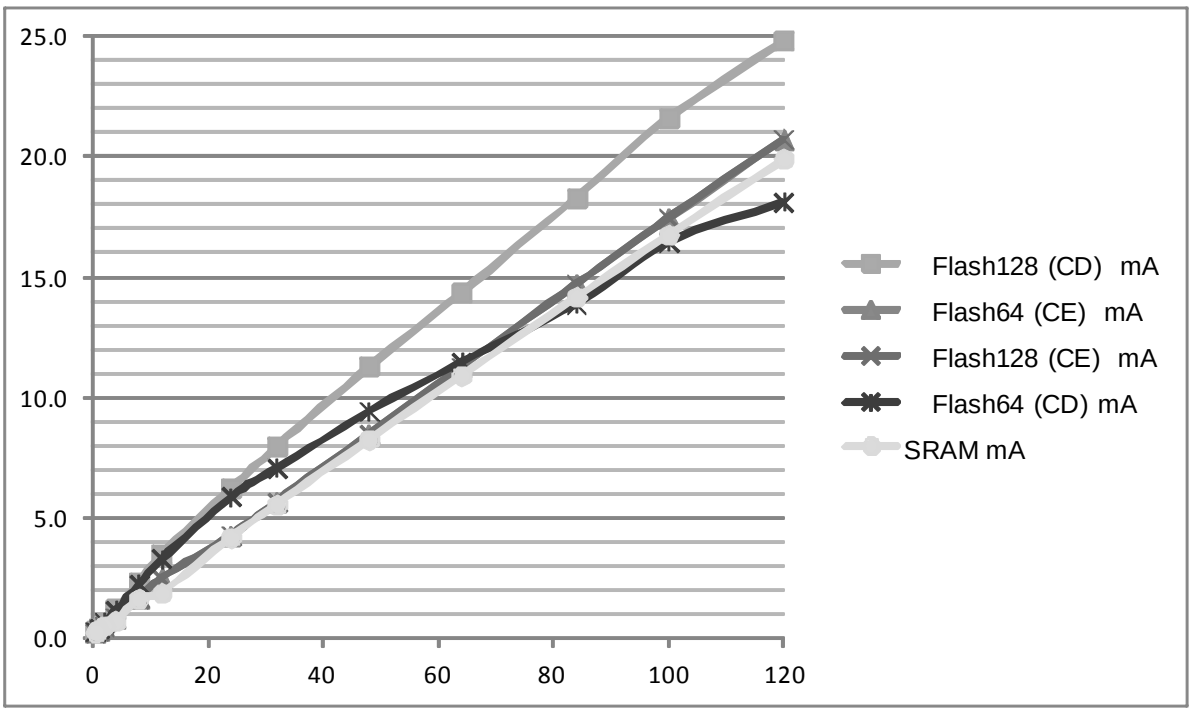


Table 49-3. SAM4S Datasheet Rev. 11100I Revision History (Continued)

Doc. Date	Changes
03-Apr-15	**Section 18., “Supply Controller (SUPC)”** Updated Figure 18-1 “Supply Controller Block Diagram”. Section 18.4.2, “Slow Clock Generator”: modified last paragraph with information on entering Bypass mode. Modfied Section 18.4.7.1, “Wake-up Inputs”, Section 18.5.5, “Supply Controller Mode Register”, Section 18.5.7, “Supply Controller Wake-up Inputs Register” and Section 18.5.8, “Supply Controller Status Register” Section 18.4.7.2, “Low-power Tamper Detection and Anti-Tampering”: corrected ‘...LPDBCCLR bit must be set in SUPC_MR’ to ‘...LPDBCCLR bit must be set in SUPC_WUMR’Updated Figure 18-4 “Wake-up Sources” Section 18.5.2, “Supply Controller (SUPC) User Interface”: corrected reset value for SUPC_MR.
	Section 19., “General Purpose Backup Registers (GPBR)” Modified Section 19.1, “Description” and Section 19.2, “Embedded Characteristics” Table 19-1 “Register Mapping”: added reset value 0x00000000 for all registers SYS_GPBRx Section 19.3.1, “General Purpose Backup Register x”: inserted sentence “These registers are reset at first power-up and on each loss of VVDIO”
	Section 20., “Enhanced Embedded Flash Controller (EEFC)”. Updated Section 20.2, “Embedded Characteristics” Added Figure 20-1 Flash Memory Areas Section 20.3.2, “Interrupt Sources”: changed last sentence Section 20.4.1, “Embedded Flash Organization”: corrected instance of command name “Get descriptor” to “Get Flash Descriptor” Figure 20-7 Command State Chart: replaced two instances of “MC_FSR” with “EEFC_FSR” Modified Section 20.4.3.2, “Write Commands”, Section 20.4.3.3, “Erase Commands”, Section 20.4.3.8, “Unique Identifier Area” and Section 20.4.3.9, “User Signature Area” Section 20.5, “Enhanced Embedded Flash Controller (EEFC) User Interface”: deleted address offsets from individual register descriptions (offsets are provided in Section 20-6, “Register Mapping”) Modified Section 20.5.1, “EEFC Flash Mode Register”, Section 20.5.2, “EEFC Flash Command Register” and Section 20.5.3, “EEFC Flash Status Register”: added new field ‘ZERO’ (register bits 26:25) Updated Section 20-6, “Register Mapping”
	Section 21., “Fast Flash Programming Interface (FFPI)” Section 21-1, “16-bit Parallel Programming Interface”: renamed figure
	Section 22., “Cortex-M Cache Controller (CMCC)” Section 22.4.3, “Cache Performance Monitoring”: corrected “MODE field of the CMCC_CFG register” to “MODE field of the CMCC_MCFG register” Table 22-1 "Register Mapping": removed CMCC_CTRL reset value for this write-only register; replaced “Cache ...” with “Cache Controller ...” in “Register” column contents Updated Section 22.5.1, “Cache Controller Type Register” Updated bit descriptions in Section 22.5.3, “Cache Controller Control Register”, Section 22.5.4, “Cache Controller Status Register”, Section 22.5.5, “Cache Controller Maintenance Register 0”, Section 22.5.8, “Cache Controller Monitor Enable Register” and Section 22.5.9, “Cache Controller Monitor Control Register”. Section 22.5.7, “Cache Controller Monitor Configuration Register”: changed access from Write-only to Read/Write. Section 22.5.8, “Cache Controller Monitor Enable Register”: removed reset value from several write-only registers (reset values are found in Table 22-1 "Register Mapping") and changed access from Write-only to Read/Write.