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/M4F
Core Size	32-Bit Dual-Core
Speed	120MHz
Connectivity	EBI/EMI, I²C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, LCD, POR, PWM, WDT
Number of I/O	57
Program Memory Size	512KB (512K 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 6x10b
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/atsam4cms8cc-au

12.6.6.11 UMULL, UMLAL, SMULL, and SMLAL

Signed and Unsigned Long Multiply, with optional Accumulate, using 32-bit operands and producing a 64-bit result.

Syntax

```
op{cond} RdLo, RdHi, Rn, Rm
```

where:

op is one of:

- UMULL Unsigned Long Multiply.
- UMLAL Unsigned Long Multiply, with Accumulate.
- SMULL Signed Long Multiply.
- SMLAL Signed Long Multiply, with Accumulate.

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

RdHi, RdLo are the destination registers. For UMLAL and SMLAL they also hold the accumulating value.

Rn, Rm are registers holding the operands.

Operation

The UMULL instruction interprets the values from *Rn* and *Rm* as unsigned integers. It multiplies these integers and places the least significant 32 bits of the result in *RdLo*, and the most significant 32 bits of the result in *RdHi*.

The UMLAL instruction interprets the values from *Rn* and *Rm* as unsigned integers. It multiplies these integers, adds the 64-bit result to the 64-bit unsigned integer contained in *RdHi* and *RdLo*, and writes the result back to *RdHi* and *RdLo*.

The SMULL instruction interprets the values from *Rn* and *Rm* as two’s complement signed integers. It multiplies these integers and places the least significant 32 bits of the result in *RdLo*, and the most significant 32 bits of the result in *RdHi*.

The SMLAL instruction interprets the values from *Rn* and *Rm* as two’s complement signed integers. It multiplies these integers, adds the 64-bit result to the 64-bit signed integer contained in *RdHi* and *RdLo*, and writes the result back to *RdHi* and *RdLo*.

Restrictions

In these instructions:

- Do not use SP and do not use PC
- *RdHi* and *RdLo* must be different registers.

Condition Flags

These instructions do not affect the condition code flags.

Examples

```
UMULL       R0, R4, R5, R6   ; Unsigned (R4,R0) = R5 x R6
SMLAL       R4, R5, R3, R8   ; Signed (R5,R4) = (R5,R4) + R3 x R8
```

12.6.9.1 BFC and BFI

Bit Field Clear and Bit Field Insert.

Syntax

```
BFC{cond} Rd, #lsb, #width
BFI{cond} Rd, Rn, #lsb, #width
```

where:

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

Rd is the destination register.

Rn is the source register.

lsb is the position of the least significant bit of the bitfield. *lsb* must be in the range 0 to 31.

width is the width of the bitfield and must be in the range 1 to 32-*lsb*.

Operation

BFC clears a bitfield in a register. It clears *width* bits in *Rd*, starting at the low bit position *lsb*. Other bits in *Rd* are unchanged.

BFI copies a bitfield into one register from another register. It replaces *width* bits in *Rd* starting at the low bit position *lsb*, with *width* bits from *Rn* starting at bit[0]. Other bits in *Rd* are unchanged.

Restrictions

Do not use SP and do not use PC.

Condition Flags

These instructions do not affect the flags.

Examples

```
BFC   R4, #8, #12       ; Clear bit 8 to bit 19 (12 bits) of R4 to 0
BFI   R9, R2, #8, #12   ; Replace bit 8 to bit 19 (12 bits) of R9 with
                        ; bit 0 to bit 11 from R2.
```

12.6.11.13 VMOV Immediate

Move floating-point Immediate

Syntax

```
VMOV{cond}.F32 Sd, #imm
```

where:

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

Sd is the branch destination.

imm is a floating-point constant.

Operation

This instruction copies a constant value to a floating-point register.

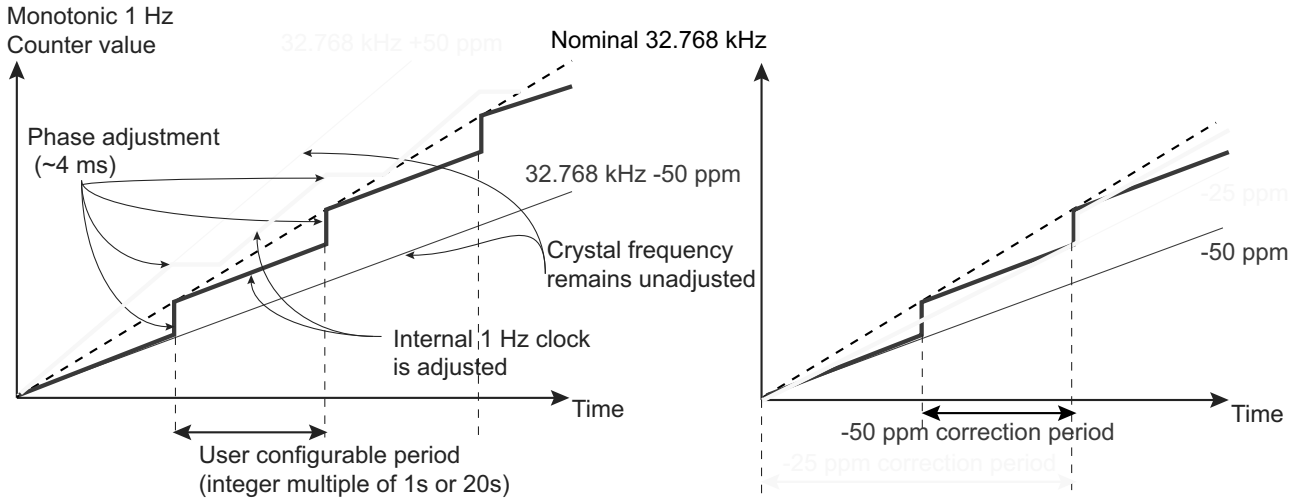
Restrictions

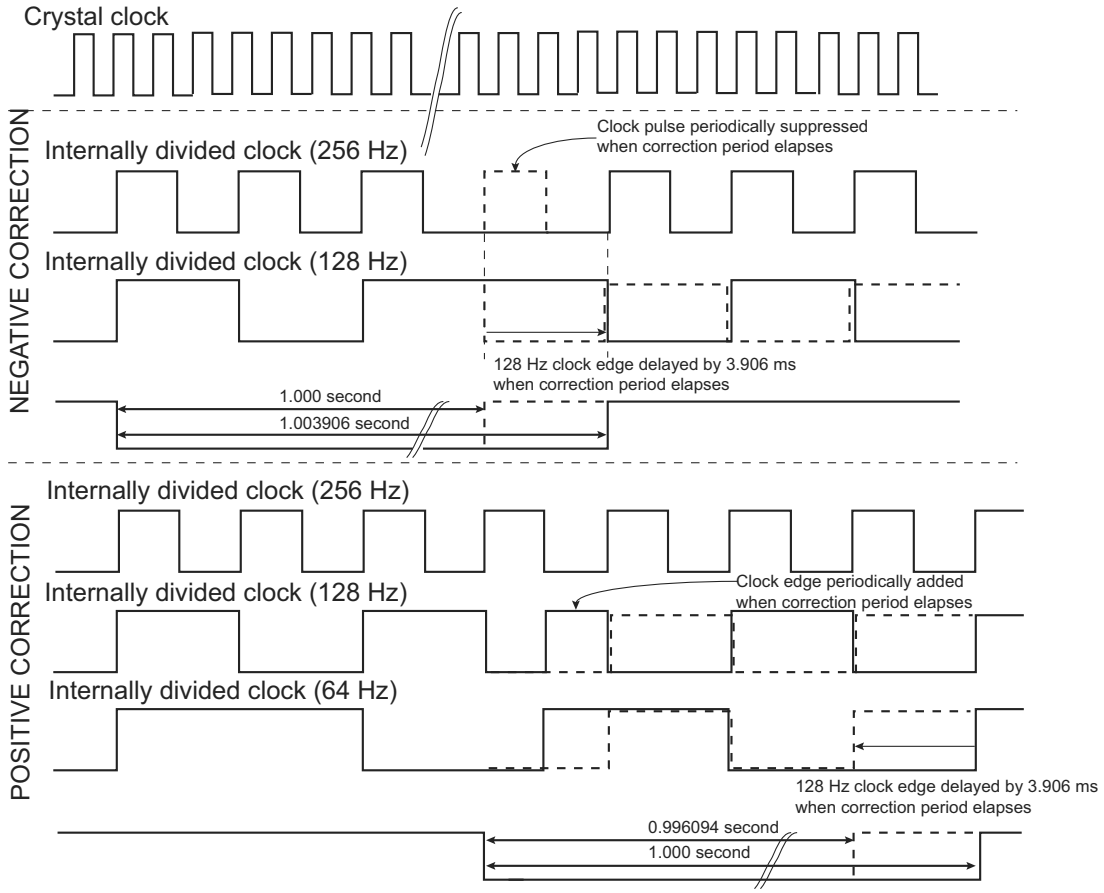
There are no restrictions.

Condition Flags

These instructions do not change the flags.

Figure 17-5. Calibration Circuitry Waveforms





The inaccuracy of a crystal oscillator at typical room temperature (±20 ppm at 20–25 °C) can be compensated if a reference clock/signal is used to measure such inaccuracy. This kind of calibration operation can be set up during 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.

17.6.3 RTC Time Register

Name: RTC_TIMR

Address: 0x400E1468

Access: Read/Write

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

23	22	21	20	19	18	17	16
–	AMPM	HOUR					

15	14	13	12	11	10	9	8
–	MIN						

7	6	5	4	3	2	1	0
–	SEC						

- **SEC: Current Second**

The range that can be set is 0–59 (BCD).

The lowest four bits encode the units. The higher bits encode the tens.

- **MIN: Current Minute**

The range that can be set is 0–59 (BCD).

The lowest four bits encode the units. The higher bits encode the tens.

- **HOUR: Current Hour**

The range that can be set is 1–12 (BCD) in 12-hour mode or 0–23 (BCD) in 24-hour mode.

- **AMPM: Ante Meridiem Post Meridiem Indicator**

This bit is the AM/PM indicator in 12-hour mode.

0: AM.

1: PM.

Figure 22-1. Flash Memory Areas

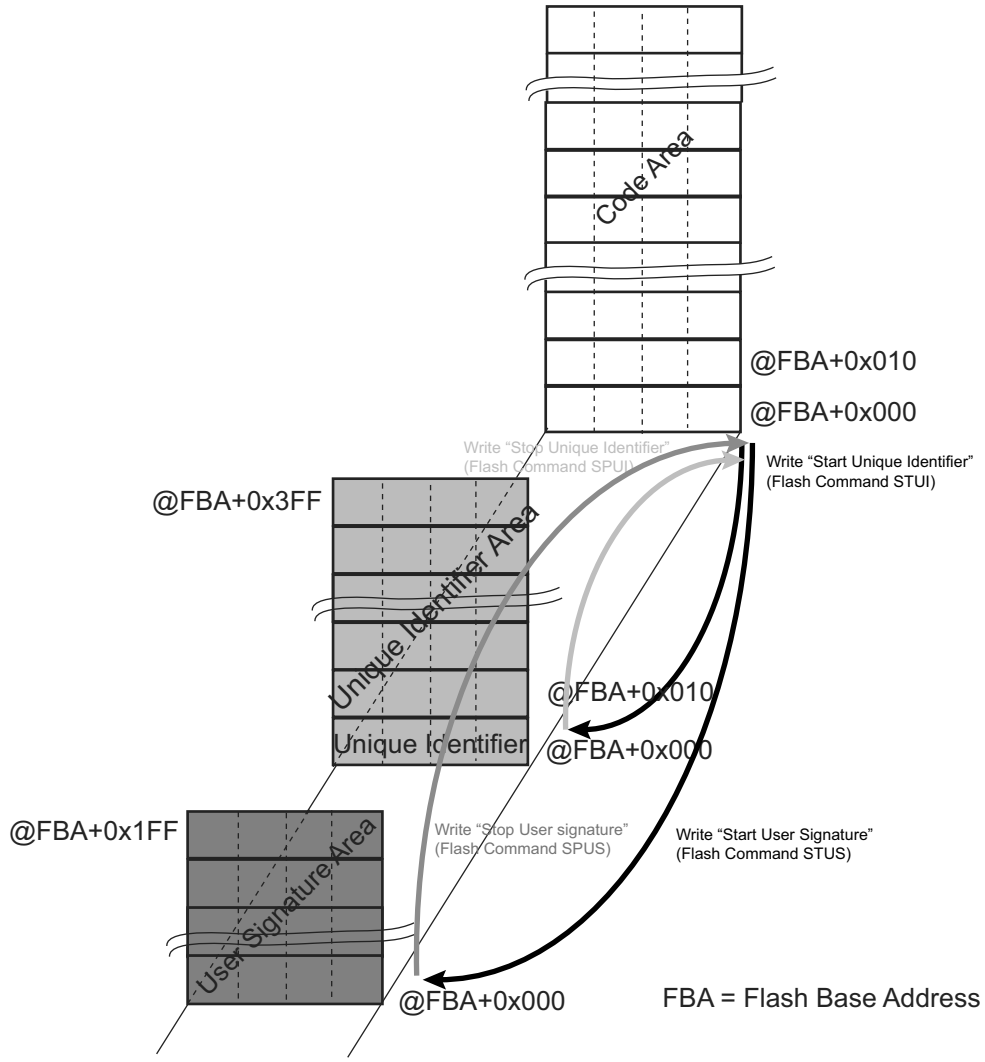


Table 26-5. List of Bus Matrix Slaves

Slave 0	Internal SRAM1
Slave 1	Internal SRAM2
Slave 2	External Bus Interface
Slave 3	Peripheral Bridge 1
Slave 4	Matrix0
Slave 5	CMCC1

26.2.2.3 Master to Slave Access (Matrix 1)

Table 26-6 gives valid paths for master to slave access on Matrix 1. The paths shown as "-" are forbidden or not wired, e.g. access from the Cortex-M4 S Bus to the Internal ROM.

Table 26-6. Matrix 1 Master to Slave Access

	Masters	0	1	2	3	4	5
Slaves		**Cortex-M4 I/D Bus**	**Cortex-M4 S Bus**	**PDC1**	**Matrix0**	**EBI Matrix 0**	**CMCC1**
0	Internal SRAM1	X	X	X	X	-	-
1	Internal SRAM2	-	X	X	X	-	-
2	External Bus Interface	X	X	X	-	X	X
3	Peripheral Bridge 1	-	X	X	X	-	-
4	Matrix0	X	X	-	-	-	X
5	CMCC1	X	-	-	-	-	-

26.2.2.4 Accesses through Matrix 1

- CM4P1 I/D Bus access to:
  - Flash (through 0x01000000 to 0x01FFFFFF)
  - EBI (through 0x03000000 to 0x06FFFFFF)
  - FLASH and EBI through Cache CMCC1
- CM4P1 S-Bus access to:
  - SRAM1, SRAM2, SRAM0 through Matrix0 (0x20000000),
  - EBI (0x60000000 to 0x63FFFFFF and 0xA0000000 to 0xA3FFFFFF),
  - HBRIDGE1, HBRIDGE0 through Matrix0 (0x40000000)
- PDC1 access to:
  - SRAM1, SRAM2
  - EBI (0x60000000 to 0x63FFFFFF),
  - HBRIDGE1
- Matrix0 access to:
  - SRAM1, SRAM2,
  - HBRIDGE1
- EBI from Matrix 0 access to:
  - EBI (through 0x030000000 to 0x06FFFFFFF, 0x60000000 to 0x63FFFFFF, 0xA0000000 to A3FFFFFF)

27.3 I/O Lines Description

Table 27-1. I/O Line Description

Name	Description	Type	Active Level
NCS[3:0]	Static Memory Controller Chip Select Lines	Output	Low
NRD	Read Signal	Output	Low
NWR0/NWE	Write 0/Write Enable Signal	Output	Low
NWR1/NBS1	Write 1/Byte 1 Select Signal	Output	Low
A0/NBS0	Address Bit 0/Byte 0 Select Signal	Output	Low
A[23:1]	Address Bus	Output	–
D[15:0]	Data Bus	I/O	–
NWAIT	External Wait Signal	Input	Low
NANDCS	NAND Flash Chip Select Line	Output	Low
NANDOE	NAND Flash Output Enable	Output	Low
NANDWE	NAND Flash Write Enable	Output	Low
NANDALE	NAND Flash Address Latch Enable	Output	–
NANDCLE	NAND Flash Command Latch Enable	Output	–

27.4 Product Dependencies

27.4.1 I/O Lines

The pins used for interfacing the SMC are multiplexed with the PIO lines. The programmer must first program the PIO controller to assign the SMC pins to their peripheral function. If I/O Lines of the SMC are not used by the application, they can be used for other purposes by the PIO Controller.

27.4.2 Power Management

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

27.5 Multiplexed Signals

Table 27-2. Static Memory Controller (SMC) Multiplexed Signals

Multiplexed Signals		Related Function
NWR0	NWE	Byte-write or Byte-select access. See Section 27.7.2.1 ”Byte Write Access” and Section 27.7.2.2 ”Byte Select Access”
A0	NBS0	8-bit or 16-bit data bus. See Section 27.7.1 ”Data Bus Width”
NWR1	NBS1	Byte-write or Byte-select access. See Section 27.7.2.1 ”Byte Write Access” and Section 27.7.2.2 ”Byte Select Access”
A22	NANDCLE	NAND Flash Command Latch Enable
A21	NANDALE	NAND Flash Address Latch Enable

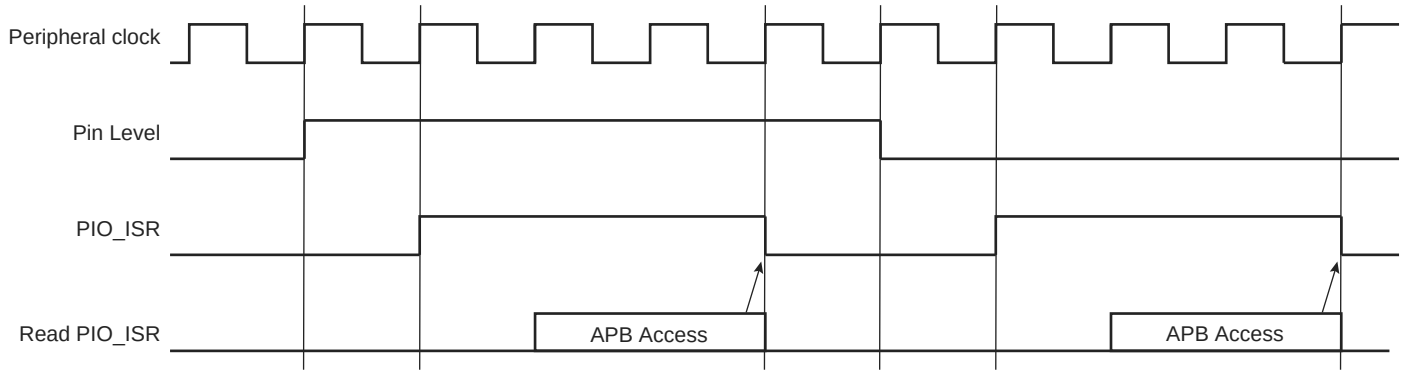
- Falling edge on PIO line 6
- Rising edge on PIO line 7
- Any edge on the other lines

Table 32-2 provides the required configuration for this example.

Table 32-2. Configuration for Example Interrupt Generation

Configuration	Description
Interrupt Mode	All the interrupt sources are enabled by writing 32'hFFFF_FFFF in PIO_IER. Then the additional interrupt mode is enabled for lines 0 to 7 by writing 32'h0000_00FF in PIO_AIMER.
Edge or Level Detection	Lines 3, 4 and 5 are configured in level detection by writing 32'h0000_0038 in PIO_LSR. The other lines are configured in edge detection by default, if they have not been previously configured. Otherwise, lines 0, 1, 2, 6 and 7 must be configured in edge detection by writing 32'h0000_00C7 in PIO_ESR.
Falling/Rising Edge or Low/High-Level Detection	Lines 0, 2, 4, 5 and 7 are configured in rising edge or high-level detection by writing 32'h0000_00B5 in PIO_REHLSR. The other lines are configured in falling edge or low-level detection by default if they have not been previously configured. Otherwise, lines 1, 3 and 6 must be configured in falling edge/low-level detection by writing 32'h0000_004A in PIO_FELLSR.

Figure 32-7. Input Change Interrupt Timings When No Additional Interrupt Modes



32.5.11 Programmable I/O Drive

It is possible to configure the I/O drive for pads PA0 to PA31. Refer to Section 46. "Electrical Characteristics".

32.5.12 Programmable Schmitt Trigger

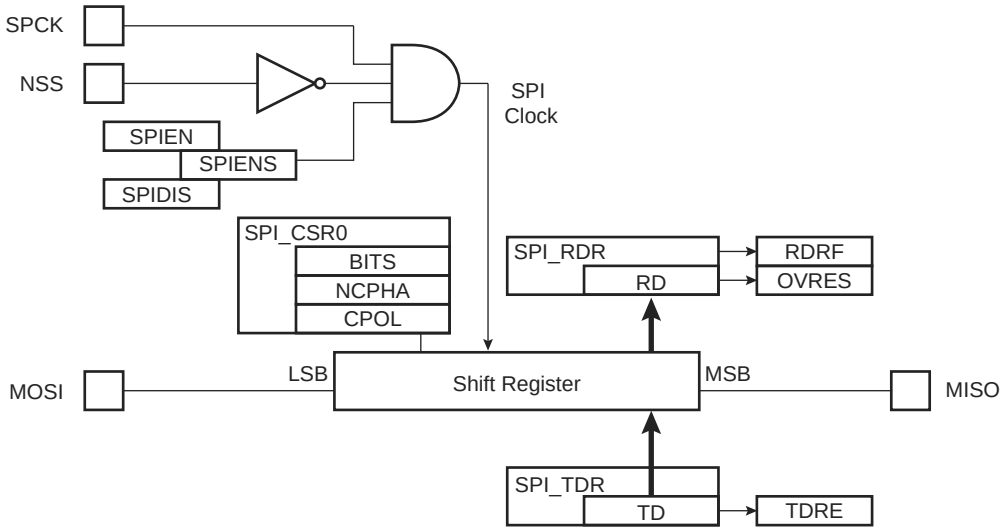
It is possible to configure each input for the Schmitt trigger. By default the Schmitt trigger is active. Disabling the Schmitt trigger is requested when using the QTouch® Library.

32.5.13 I/O Lines Programming Example

The programming example shown in Table 32-3 is used to obtain the following configuration:

- 4-bit output port on I/O lines 0 to 3 (should be written in a single write operation), open-drain, with pull-up resistor
- Four output signals on I/O lines 4 to 7 (to drive LEDs for example), driven high and low, no pull-up resistor, no pull-down resistor
- Four input signals on I/O lines 8 to 11 (to read push-button states for example), with pull-up resistors, glitch filters and input change interrupts

Figure 33-13. Slave Mode Functional Block Diagram



33.7.5 Register Write Protection

To prevent any single software error from corrupting SPI behavior, certain registers in the address space can be write-protected by setting the WPEN bit in the SPI Write Protection Mode Register (SPI_WPMR).

If a write access to a write-protected register is detected, the WPVS flag in the SPI Write Protection Status Register (SPI_WPSR) is set and the WPVSRC field indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading SPI_WPSR.

The following registers can be write-protected:

- SPI Mode Register
- SPI Chip Select Register

- **TXBUFE: TX Buffer Empty (cleared by writing SPI_TCR or SPI_TNCR)**

0: SPI_TCR(1) or SPI_TNCR(1) has a value other than 0.

1: Both SPI_TCR(1) and SPI_TNCR(1) have a value of 0.

- **NSSR: NSS Rising (cleared on read)**

0: No rising edge detected on NSS pin since the last read of SPI_SR.

1: A rising edge occurred on NSS pin since the last read of SPI_SR.

- **TXEMPTY: Transmission Registers Empty (cleared by writing SPI_TDR)**

0: As soon as data is written in SPI_TDR.

1: SPI_TDR and internal shift register are empty. If a transfer delay has been defined, TXEMPTY is set after the end of this delay.

- **UNDES: Underrun Error Status (Slave mode only) (cleared on read)**

0: No underrun has been detected since the last read of SPI_SR.

1: A transfer starts whereas no data has been loaded in SPI_TDR.

- **SPIENS: SPI Enable Status**

0: SPI is disabled.

1: SPI is enabled.

Note: 1. SPI_RCR, SPI_RNCR, SPI_TCR, SPI_TNCR are PDC registers.

Figure 34-21. Programmer Sends Data While the Bus is Busy

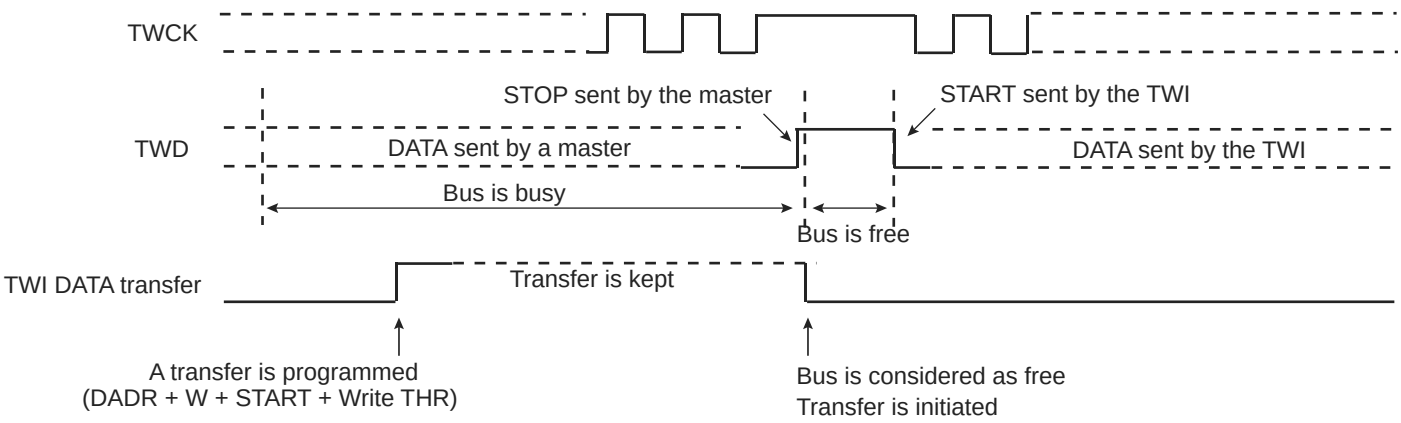
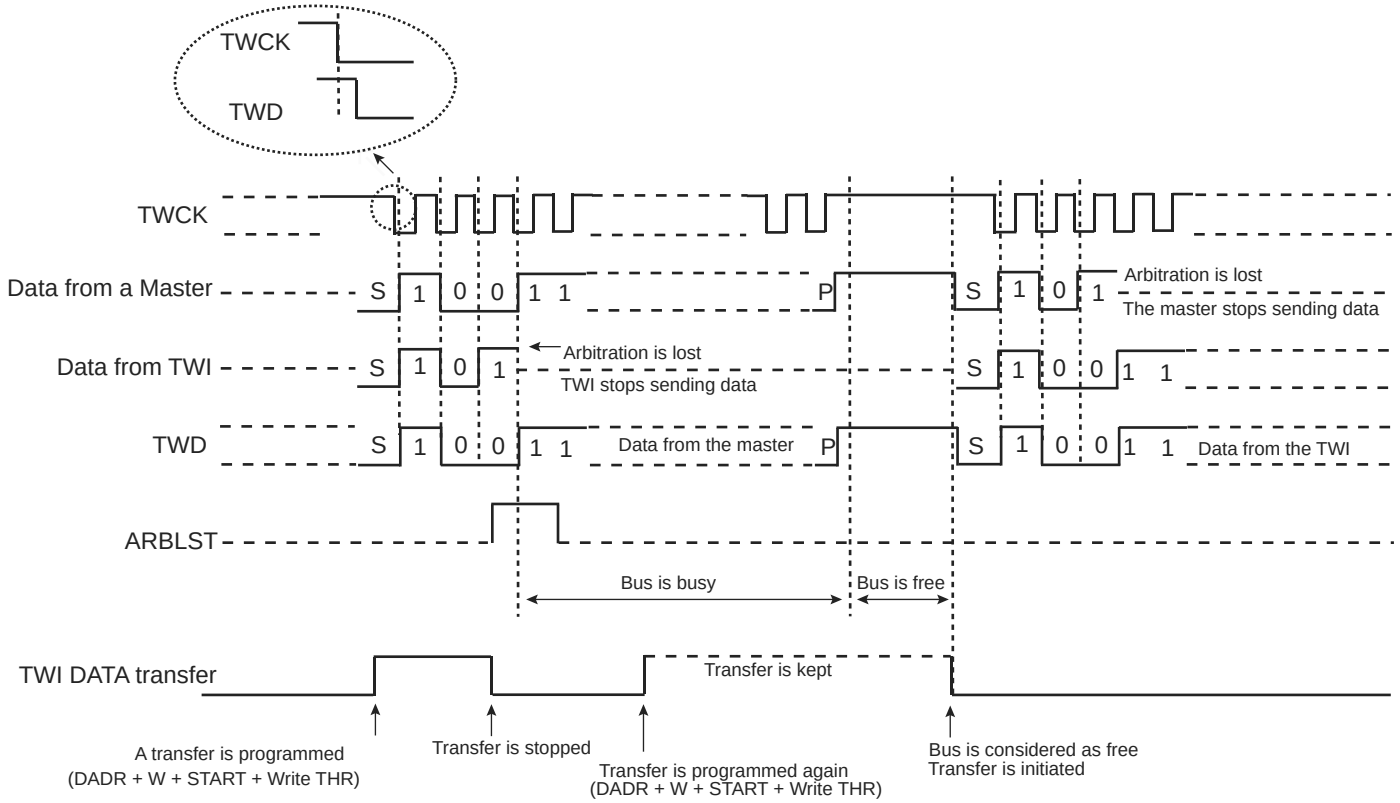


Figure 34-22. Arbitration Cases



34.8.4 TWI Internal Address Register

Name: TWI_IADR

Address: 0x4001800C (0), 0x4001C00C (1)

Access: Read/Write

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

23	22	21	20	19	18	17	16
IADR							

15	14	13	12	11	10	9	8
IADR							

7	6	5	4	3	2	1	0
IADR							

- **IADR: Internal Address**

0, 1, 2 or 3 bytes depending on IADRSZ.

37.7.5 TC Counter Value Register

Name: TC_CVx [x=0..2]

Address: 0x40010010 (0)[0], 0x40010050 (0)[1], 0x40010090 (0)[2], 0x40014010 (1)[0], 0x40014050 (1)[1], 0x40014090 (1)[2]

Access: Read-only

31	30	29	28	27	26	25	24
CV							

23	22	21	20	19	18	17	16
CV							

15	14	13	12	11	10	9	8
CV							

7	6	5	4	3	2	1	0
CV							

- **CV: Counter Value**

CV contains the counter value in real time.

IMPORTANT: For 16-bit channels, CV field size is limited to register bits 15:0.

40.7.14 ADC Overrun Status Register

Name: ADC_OVER

Address: 0x4003803C

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
OVRE7	OVRE6	–	–	OVRE3	OVRE2	OVRE1	OVRE0

- **OVREx: Overrun Error x**

0: No overrun error on the corresponding channel since the last read of ADC_OVER.

1: There has been an overrun error on the corresponding channel since the last read of ADC_OVER.

Note: An overrun error does not always mean that the unread data has been replaced by a new valid data. Refer to Section 40.6.11 ”Enhanced Resolution Mode and Digital Averaging Function” for details.

– If the first block of the fragment is a block of Plaintext data, set IV in AES_IVRx with a value constructed as follows: ‘LSB96(J0) || CTR’ value, (96 bit LSB of J0 concatenated with saved CTR value from previous fragment).

4. Set AADLEN field in AES_AADLENR and CLEN field in AES_CLENR according to the length of the current fragment, or set the fields with the remaining message length, both configurations work.
5. Fill the GHASH field of AES_GHASHRx with the value stored after the previous fragment.
6. Fill the IDATA field of AES_IDATARx with the current fragment of the message to process (aligned on 16 byte boundary) according to the SMOD configuration used. If Manual Mode or Auto Mode is used, the DATRDY bit indicates when the data have been processed (however, no output data are generated when processing *AAD*).
7. Make sure the last output data have been read if the fragment ends in C phase (or wait for DATRDY if the fragment ends in *AAD* phase), then read the GHASH field of AES_GHASHRx to obtain the value of the hash after the last processed data and finally read the CTR field of the AES_CTR to obtain the value of the CTR encryption counter (not needed when the fragment ends in *AAD* phase).

Note: Step 1 and 2 are required only if the value of the concerned registers has been modified.

Once the last fragment has been processed, the GHASH value will allow manual generation of the GCM tag (see “Manual GCM Tag Generation” for details).

Manual GCM Tag Generation

This section describes the last steps of the GCM Tag generation.

The Manual GCM Tag Generation is used to complete the GCM Tag Generation when the message has been processed without Tag Generation.

Note: The Message Processing without Tag Generation must be finished before processing the Manual GCM Tag Generation.

To generate a GCM Tag manually, the sequence is as follows:

Processing $S = \mathrm{GHASH}_H (A \| 0^v \| C \| 0^u \| [\mathrm{len}(A)]_{64} \| [\mathrm{len}(C)]_{64})$:

1. In AES_MR set OPMOD to GCM and GTAGEN to ‘0’ (configuration as usual for the rest).
2. Set KEYW in AES_KEYWRx and wait for DATRDY bit of AES_ISR to be set (GCM hash subkey generation complete); use interrupt if needed. After the GCM hash subkey generation is complete the GCM hash subkey can be read or overwritten with specific value in the AES_GCMHRx (see Section 42.4.6.2 “Key Writing and Automatic Hash Subkey Calculation” for details).
3. Set AADLEN field to 0x10 (16 bytes) in AES_AADLENR and CLEN field to ‘0’ in AES_CLENR. This will allow running a single GHASH_H on a 16-byte input data (see Figure 42-7).
4. Fill the GHASH field of AES_GHASHRx with the state of the GHASH field stored at the end of the message processing.
5. Fill the IDATA field of AES_IDATARx according to the SMOD configuration used with ‘$\mathrm{len}(A)_{64} \| \mathrm{len}(C)_{64}$’ value as described in the NIST documentation and wait for DATRDY to be set; use interrupt if needed.
6. Read the GHASH field of AES_GHASHRx to obtain the current value of the hash.

Processing $T = \mathrm{GCTR}_K(J_0, S)$:

7. In AES_MR set OPMOD to CTR (configuration as usual for the rest).
8. Set the IV field in AES_IVRx with ‘J_0’ value.
9. Fill the IDATA field of AES_IDATARx with the GHASH value read at step 6 and wait for DATRDY to be set (use interrupt if needed).
10. Read the ODATA field of AES_ODATARx to obtain the GCM Tag value.

Note: Step 4 is optional if the GHASH field is to be filled with value ‘0’ (0 length packet for instance).

42.5.16 AES GCM H Word Register x

Name: AES_GCMHRx [x=0..3]

Address: 0x4000009C

Access: Read/Write

31	30	29	28	27	26	25	24
H							

23	22	21	20	19	18	17	16
H							

15	14	13	12	11	10	9	8
H							

7	6	5	4	3	2	1	0
H							

- **H: GCM H Word x**

The four 32-bit H Word registers contain the 128-bit GCM hash subkey *H* value.

Whenever a new key (AES_KEYWRx) is written to the hardware two automatic actions are processed:

- GCM hash subkey *H* generation
- AES_GHASHRx Clear

If the application software requires a specific hash subkey, the automatically generated H value can be overwritten in the AES_GCMHRx (see Section 42.4.6.2 “Key Writing and Automatic Hash Subkey Calculation” for details).

The choice of a GCM hash subkey *H* by a write in the AES_GCMHRx permits

- selection of the GCM hash subkey *H* for GHASH operations
- selection of one operand to process a single GF128 multiply

46.6 Embedded Flash Characteristics

46.6.1 Embedded Flash DC Characteristics

Table 46-50. DC Flash Characteristics

Symbol	Parameter	Conditions	Typ	Max	Unit
I_{CC}	Active current	Random 128-bit Read:			
		Maximum Read Frequency onto V_{DDCORE} = 1.2V @ 25°C	16	25	mA
		Maximum Read Frequency onto V_{DDIO} = 3.3V @ 25°C	3	5	
		Random 64-bit Read:			
		Maximum Read Frequency onto V_{DDCORE} = 1.2V @ 25°C	10	18	mA
		Maximum Read Frequency onto V_{DDIO} = 3.3V @ 25°C	3	5	
		Program:			
		- Onto V_{DDCORE} = 1.2V @ 25°C	3	5	mA
		- Onto V_{DDIO} = 3.3V @ 25°C	10	15	
		Erase:			
		- Onto V_{DDCORE} = 1.2V @ 25°C	3	5	mA
		- Onto V_{DDIO} = 3.3V @ 25°C	10	15	

46.6.2 Embedded Flash AC Characteristics

Table 46-51. AC Flash Characteristics

Parameter	Conditions	Min	Typ	Max	Unit
Program/ Erase Operation Cycle Time	Write page (512 bytes)	–	1.5	3	ms
	Erase page	–	10	50	ms
	Erase block (4 Kbytes)	–	50	200	ms
	Erase sector	–	400	950	ms
	Full chip erase - 1 Mbyte - 512 Kbytes	–	 9 5.5	 18 11	s
	Lock/Unlock time per region	–	1.5	3	ms
Data Retention	Not powered or powered	–	20	–	Years
Endurance	Write/Erase cycles per page, block or sector @ 85°C	10K	–	–	Cycles

46.7.4.1 Test Setup 1: CoreMark™

- CoreMark on Core 0 (CM4P0) running out of flash in 128-bit or 64-bit Access mode with and without Cache Enabled. Cache is enabled above 0 WS.
- Sub-system 1 Master Clock (CPBMCK) and Core Clock (CPHCLK) stopped and in reset state

Table 46-62. SAM4CM4/8/16 Test Setup 1 Current Consumption

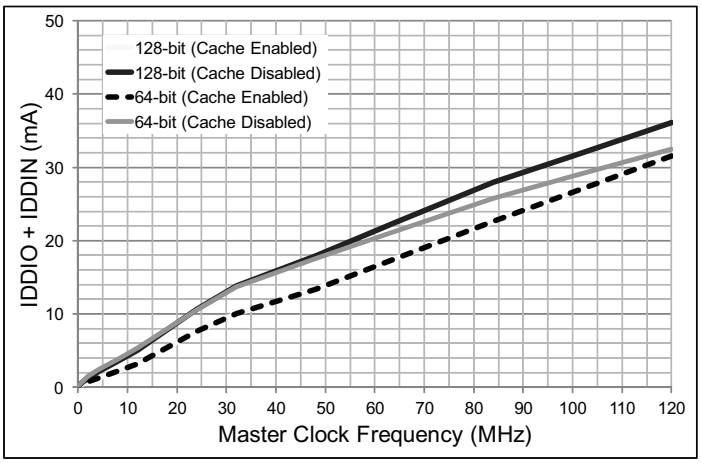
Clock (MHz)	128-bit Flash Access						64-bit Flash Access						Unit
	Cache Enabled			Cache Disabled			Cache Enabled			Cache Disabled			
	IDD_IN (AMP1)	IDD_I0 (AMP2)	IDD_CORE (AMP3)	IDD_IN (AMP1)	IDD_I0 (AMP2)	IDD_CORE (AMP3)	IDD_IN (AMP1)	IDD_I0 (AMP2)	IDD_CORE (AMP3)	IDD_IN (AMP1)	IDD_I0 (AMP2)	IDD_CORE (AMP3)	
120	21.8	0.27	18.5	24.4	2.0	21.1	21.5	0.27	18.3	21.2	1.9	17.9	mA
100	18.1	0.27	15.4	21.6	1.8	18.9	18.1	0.27	15.4	19.0	1.8	16.3	
84	15.3	0.27	13.0	18.8	1.7	16.6	15.3	0.27	13.0	16.8	1.7	14.5	
64	11.8	0.27	10.1	15.2	1.5	13.5	11.8	0.27	10.1	14.1	1.4	12.5	
48	9.2	0.27	7.9	11.7	1.4	10.5	9.2	0.27	7.9	11.3	1.3	10.0	
32	7.2	0.27	5.6	9.5	1.2	7.9	7.2	0.27	5.6	9.3	1.2	7.7	
24	5.6	0.27	4.3	7.5	1.1	6.2	5.6	0.27	4.3	7.2	1.2	5.9	
12	2.4	0.09	2.4	3.1	0.9	3.1	2.4	0.09	2.4	3.1	1.0	3.1	
8	1.6	0.09	1.6	2.1	0.7	2.1	1.6	0.09	1.6	2.1	0.9	2.1	
4	1.0	0.09	1.0	1.4	0.5	1.4	1.0	0.09	1.0	1.4	0.8	1.4	
2	0.70	0.09	0.69	0.90	0.40	0.90	0.70	0.09	0.69	0.70	0.70	0.70	
1	0.54	0.09	0.53	0.65	0.30	0.65	0.55	0.09	0.54	0.65	0.40	0.65	
0.5	0.47	0.09	0.46	0.50	0.20	0.50	0.47	0.09	0.46	0.60	0.20	0.60	
0.25	0.25	0.09	0.24	0.26	0.10	0.25	0.25	0.09	0.24	0.36	0.10	0.25	

Table 46-67. SAM4CM32 Test Setup 3 Current Consumption

Clock (MHz)	128-bit Flash Access						64-bit Flash Access						Unit
	Cache Enabled			Cache Disabled			Cache Enabled			Cache Disabled			
	IDD_IN (AMP1)	IDD_I0 (AMP2)	IDD_CORE (AMP3)	IDD_IN (AMP1)	IDD_I0 (AMP2)	IDD_CORE (AMP3)	IDD_IN (AMP1)	IDD_I0 (AMP2)	IDD_CORE (AMP3)	IDD_IN (AMP1)	IDD_I0 (AMP2)	IDD_CORE (AMP3)	
120	35.0	0.23	31.7	38.4	2.1	35.1	34.9	0.23	31.6	33.8	1.8	30.5	mA
100	29.5	0.23	26.8	33.8	2.0	31.0	29.4	0.23	26.7	29.5	1.7	27.0	
84	25.1	0.23	22.8	29.4	1.8	27.1	24.9	0.23	22.7	26.6	1.7	24.3	
64	19.3	0.23	17.7	23.2	1.5	21.5	19.2	0.23	17.6	21.8	1.5	20.1	
48	14.7	0.23	13.4	18.0	1.3	16.8	14.6	0.23	13.4	17.7	1.5	16.5	
32	10.9	0.23	9.2	13.3	1.1	11.7	10.8	0.23	9.2	13.5	1.3	11.8	
24	8.2	0.23	7.0	10.5	1.0	9.3	8.2	0.22	7.0	10.3	1.2	9.0	
12	3.5	0.02	3.5	4.8	0.86	4.7	3.5	0.02	3.5	4.7	1.1	4.6	
8	2.4	0.02	2.4	3.2	0.74	3.2	2.4	0.02	2.4	3.1	1.0	3.1	
4	1.3	0.02	1.3	1.7	0.42	1.7	1.3	0.02	1.3	1.7	0.87	1.7	
2	0.72	0.02	0.71	0.92	0.40	0.89	0.71	0.02	0.81	0.94	0.56	0.94	
1	0.43	0.02	0.42	0.52	0.18	0.52	0.43	0.02	0.42	0.55	0.36	0.54	
0.5	0.29	0.02	0.28	0.36	0.09	0.36	0.29	0.02	0.28	0.35	0.18	0.34	
0.25	0.16	0.02	0.15	0.18	0.02	0.16	0.16	0.02	0.15	0.17	0.06	0.16	

Figure 46-29. Typical Current Consumption in Active Mode (Test Setup 3)

SAM4Cx4/8/16



SAM4Cx32