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	Obsolete
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	52
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/atsam4cmp8cb-au

- Debug
  - Star Topology AHB-AP Debug Access Port Implementation with Common SW-DP / SWJ-DP Providing Higher Performance than Daisy-chain Topology
  - Debug Synchronization between both Cores (cross triggering to/from each core for Halt and Run Mode)
- I/O
  - Up to 57 I/O lines with External Interrupt Capability (edge or level sensitivity), Schmitt Trigger, Internal Pull-up/pull-down, Debouncing, Glitch Filtering and On-die Series Resistor Termination
- Package
  - 100-lead LQFP, 14 x 14 mm, pitch 0.5 mm

Note: 1. 120 MHz: -40°C/+85°C, VDDCORE = 1.2V

Each sector is organized in pages of 512 bytes.

For sector 0:

- The small sector 0 has 16 pages of 512 bytes, 8 Kbytes in total
- The small sector 1 has 16 pages of 512 bytes, 8 Kbytes in total
- The larger sector has 96 pages of 512 bytes, 48 Kbytes in total

From sector 1 to n:

The rest of the array is composed of 64-Kbyte sectors where each sector comprises 128 pages of 512 bytes. Refer to Figure 8-3 below.

Figure 8-3. Flash Sector Organization

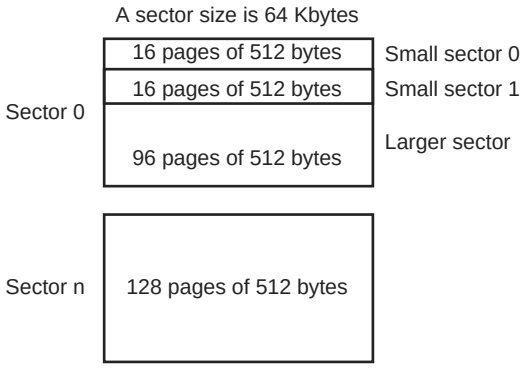


Table 8-1. SAM4CM Flash Size

Device	Flash (Kbytes)
SAM4CM4	256
SAM4CM8	512
SAM4CM16	1024
SAM4CM32	2048 (2 × 1024)

12.6.8.3 SXTA and UXTA

Signed and Unsigned Extend and Add

Syntax

```
op{cond} {Rd,} Rn, Rm {, ROR #n}
op{cond} {Rd,} Rn, Rm {, ROR #n}
```

where:

op is one of:

SXTAB Sign extends an 8-bit value to a 32-bit value and add.

SXTAH Sign extends a 16-bit value to a 32-bit value and add.

SXTAB16 Sign extends two 8-bit values to two 16-bit values and add.

UXTAB Zero extends an 8-bit value to a 32-bit value and add.

UXTAH Zero extends a 16-bit value to a 32-bit value and add.

UXTAB16 Zero extends two 8-bit values to two 16-bit values and add.

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

Rd is the destination register.

Rn is the first operand register.

Rm is the register holding the value to rotate and extend.

ROR #*n* is one of:

ROR #8 Value from *Rm* is rotated right 8 bits.

ROR #16 Value from *Rm* is rotated right 16 bits.

ROR #24 Value from *Rm* is rotated right 24 bits.

If ROR #*n* is omitted, no rotation is performed.

Operation

These instructions do the following:

1. Rotate the value from Rm right by 0, 8, 16 or 24 bits.
2. Extract bits from the resulting value:
   - SXTAB extracts bits[7:0] from *Rm* and sign extends to 32 bits.
   - UXTAB extracts bits[7:0] from *Rm* and zero extends to 32 bits.
   - SXTAH extracts bits[15:0] from *Rm* and sign extends to 32 bits.
   - UXTAH extracts bits[15:0] from *Rm* and zero extends to 32 bits.
   - SXTAB16 extracts bits[7:0] from *Rm* and sign extends to 16 bits, and extracts bits [23:16] from *Rm* and sign extends to 16 bits.
   - UXTAB16 extracts bits[7:0] from *Rm* and zero extends to 16 bits, and extracts bits [23:16] from *Rm* and zero extends to 16 bits.
3. Adds the signed or zero extended value to the word or corresponding halfword of *Rn* and writes the result in *Rd*.

Restrictions

Do not use SP and do not use PC.

12.6.11.8 VFMA, VFMS

Floating-point Fused Multiply Accumulate and Subtract.

Syntax

```
VFMA{cond}.F32 {Sd,} Sn, Sm
VFMS{cond}.F32 {Sd,} Sn, Sm
```

where:

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

Sd is the destination register.

Sn, Sm are the operand registers.

Operation

The VFMA instruction:

1. Multiplies the floating-point values in the operand registers.
2. Accumulates the results into the destination register.

The result of the multiply is not rounded before the accumulation.

The VFMS instruction:

1. Negates the first operand register.
2. Multiplies the floating-point values of the first and second operand registers.
3. Adds the products to the destination register.
4. Places the results in the destination register.

The result of the multiply is not rounded before the addition.

Restrictions

There are no restrictions.

Condition Flags

These instructions do not change the flags.

12.11.2.5 MPU Region Attribute and Size Register

Name: MPU_RASR

Access: Read/Write

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

23	22	21	20	19	18	17	16
–	–	TEX			S	C	B

15	14	13	12	11	10	9	8
SRD							

7	6	5	4	3	2	1	0
–	–	SIZE					ENABLE

The MPU_RASR defines the region size and memory attributes of the MPU region specified by the MPU_RNR, and enables that region and any subregions.

MPU_RASR is accessible using word or halfword accesses:

- The most significant halfword holds the region attributes.
- The least significant halfword holds the region size, and the region and subregion enable bits.

- **XN: Instruction Access Disable**

0: Instruction fetches enabled.

1: Instruction fetches disabled.

- **AP: Access Permission**

See Table 12-39.

- **TEX, C, B: Memory Access Attributes**

See Table 12-37.

- **S: Shareable**

See Table 12-37.

- **SRD: Subregion Disable**

For each bit in this field:

0: Corresponding subregion is enabled.

1: Corresponding subregion is disabled.

See “Subregions” for more information.

Region sizes of 128 bytes and less do not support subregions. When writing the attributes for such a region, write the SRD field as 0x00.

0b11: Round towards Zero (RZ) mode.

The specified rounding mode is used by almost all floating-point instructions.

- **IDC: Input Denormal Cumulative Exception**

IDC is a cumulative exception bit for floating-point exception; see also bits [4:0].

This bit is set to 1 to indicate that the corresponding exception has occurred since 0 was last written to it.

- **IXC: Inexact Cumulative Exception**

IXC is a cumulative exception bit for floating-point exception; see also bit [7].

This bit is set to 1 to indicate that the corresponding exception has occurred since 0 was last written to it.

- **UFC: Underflow Cumulative Exception**

UFC is a cumulative exception bit for floating-point exception; see also bit [7].

This bit is set to 1 to indicate that the corresponding exception has occurred since 0 was last written to it.

- **OFC: Overflow Cumulative Exception**

OFC is a cumulative exception bit for floating-point exception; see also bit [7].

This bit is set to 1 to indicate that the corresponding exception has occurred since 0 was last written to it.

- **DZC: Division by Zero Cumulative Exception**

DZC is a cumulative exception bit for floating-point exception; see also bit [7].

This bit is set to 1 to indicate that the corresponding exception has occurred since 0 was last written to it.

- **IOC: Invalid Operation Cumulative Exception**

IOC is a cumulative exception bit for floating-point exception; see also bit [7].

This bit is set to 1 to indicate that the corresponding exception has occurred since 0 was last written to it.

17.4 Product Dependencies

17.4.1 Power Management

The Real-time Clock is continuously clocked at 32.768 kHz. The Power Management Controller has no effect on RTC behavior.

17.4.2 Interrupt

RTC interrupt line is connected on one of the internal sources of the interrupt controller. RTC interrupt requires the interrupt controller to be programmed first.

Table 17-1. Peripheral IDs

Instance	ID
RTC	2

17.5 Functional Description

The RTC provides a full binary-coded decimal (BCD) clock that includes century (19/20), year (with leap years), month, date, day, hours, minutes and seconds reported in RTC Time Register (RTC_TIMR) and RTC Calendar Register (RTC_CALR).

The valid year range is up to 2099 in Gregorian mode (or 1300 to 1499 in Persian mode).

The RTC can operate in 24-hour mode or in 12-hour mode with an AM/PM indicator.

Corrections for leap years are included (all years divisible by 4 being leap years except 1900). This is correct up to the year 2099.

The RTC can generate configurable waveforms on RTCOUT0 output.

17.5.1 Reference Clock

The reference clock is the Slow Clock (SLCK). It can be driven internally or by an external 32.768 kHz crystal.

During low power modes of the processor, the oscillator runs and power consumption is critical. The crystal selection has to take into account the current consumption for power saving and the frequency drift due to temperature effect on the circuit for time accuracy.

17.5.2 Timing

The RTC is updated in real time at one-second intervals in Normal mode for the counters of seconds, at one-minute intervals for the counter of minutes and so on.

Due to the asynchronous operation of the RTC with respect to the rest of the chip, to be certain that the value read in the RTC registers (century, year, month, date, day, hours, minutes, seconds) are valid and stable, it is necessary to read these registers twice. If the data is the same both times, then it is valid. Therefore, a minimum of two and a maximum of three accesses are required.

17.5.3 Alarm

The RTC has five programmable fields: month, date, hours, minutes and seconds.

Each of these fields can be enabled or disabled to match the alarm condition:

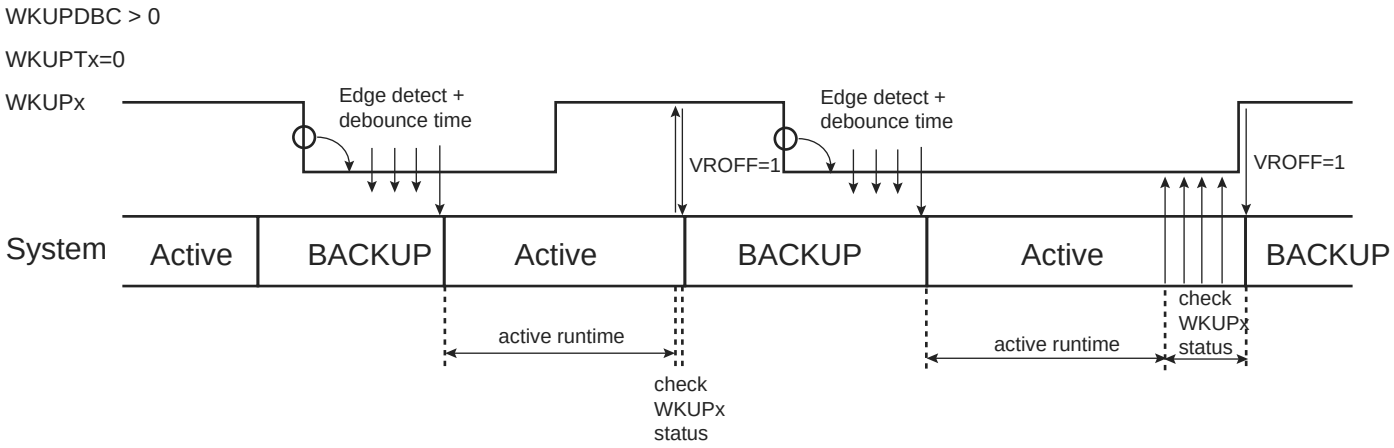
- If all the fields are enabled, an alarm flag is generated (the corresponding flag is asserted and an interrupt generated if enabled) at a given month, date, hour/minute/second.
- If only the “seconds” field is enabled, then an alarm is generated every minute.

respectively, to about 100 µs, about 1 ms, about 16 ms, about 128 ms and about 1 second (for a typical slow clock frequency of 32 kHz). Configuring WKUPDBC to 0 selects an immediate wakeup, i.e., an enabled WKUP pin must be active according to its polarity during a minimum of one slow clock period to wake up the core power supply.

If an enabled WKUPx pin holds the active polarity for a time longer than the debouncing period, a system wakeup is started and flags WKUPISx, as shown in Figure 20-4 and Figure 20-5, are reported in SUPC_SR. This enables the user to identify the source of the wakeup. However, if a new wakeup condition occurs, the primary information is lost. No new wakeup can be detected since the primary wakeup condition has disappeared.

Prior to instructing the system to enter Backup mode, if the field WKUPDBC > 0, it must be verified that none of the WKUPx pins, enabled for a wakeup (exit of Backup mode), holds an active polarity. The verification can be made by reading the pin status in the PIO controller. If WKUPENx=1 and the pin WKUPx holds an active polarity, the system must not be instructed to enter Backup mode.

Figure 20-6. Entering and Exiting Backup Mode with a WKUP Pin



20.4.9.3 Low-power Debouncer Inputs (Tamper Detection Pins)

Low-power debouncer inputs are dedicated to tamper detection. If the tamper sensor is biased through a resistor and constantly driven by the power supply, this leads to power consumption as long as the tamper detection switch is in its active state. To prevent power consumption when the switch is in active state, the tamper sensor circuitry can be intermittently powered, thus, a specific waveform must be generated.

The waveform can be generated using pin RTCOUT0 in all modes, including Backup mode. Refer to the section “Real-time Counter (RTC)” for waveform generation.

For SAM4CM devices, separate debouncers are embedded, one for each wakeup/tamper input. See Figure 20-4 and Figure 20-5.

The WKUP0/TMP0 and/or WKUP10/TMP1 inputs can be programmed to perform a system wakeup with a debouncing done by RTCOUT0. This can be enabled by setting LPDBCEN0/1 in SUPC_WUMR.

These inputs can be also used when VDDCORE is powered to obtain the tamper detection function with a low power debounce function and to raise an interrupt.

The low-power debounce mode of operation requires the RTC output (RTCOUT0) to be configured to generate a duty cycle programmable pulse (i.e., OUT0 = 0x7 in RTC_MR) in order to create the sampling points of both debouncers. The sampling point is the falling edge of the RTCOUT0 waveform.

Figure 20-7 shows an example of an application where two tamper switches are used. RTCOUT0 powers the external pullup used by the tampers.

21.3.1 General Purpose Backup Register x

Name: SYS_GPBRx

Address: 0x400E1490

Access: Read/Write

31	30	29	28	27	26	25	24
GPBR_VALUE							

23	22	21	20	19	18	17	16
GPBR_VALUE							

15	14	13	12	11	10	9	8
GPBR_VALUE							

7	6	5	4	3	2	1	0
GPBR_VALUE							

These registers are reset at first power-up and on each loss of VDDBU_SW.

- **GPBR_VALUE: Value of GPBR x**

If a Tamper event has been detected, it is not possible to write GPBR_VALUE as long as the LPDBCS0 or LPDBCS3 flag has not been cleared in the Supply Controller Status Register (SUPC_SR).

23. Fast Flash Programming Interface (FFPI)

23.1 Description

The Fast Flash Programming Interface (FFPI) provides parallel high-volume programming using a standard gang programmer. The parallel interface is fully handshaked and the device is considered to be a standard EEPROM. Additionally, the parallel protocol offers an optimized access to all the embedded Flash functionalities.

Although the Fast Flash Programming mode is a dedicated mode for high volume programming, this mode is not designed for in-situ programming.

23.2 Embedded Characteristics

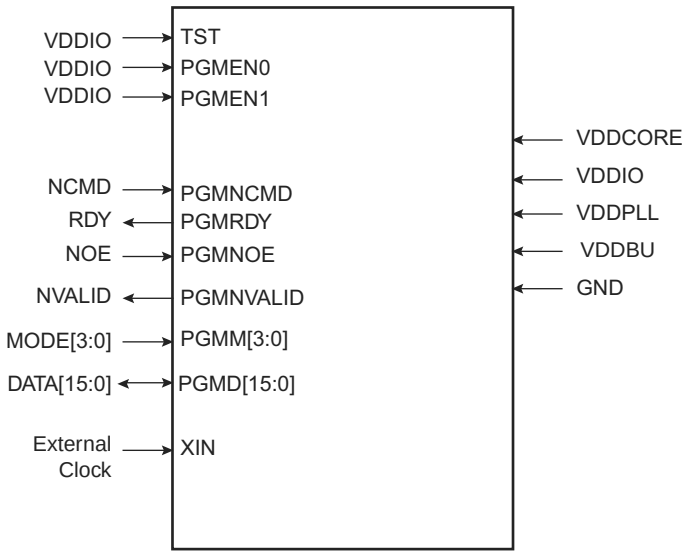
- Programming Mode for High-volume Flash Programming Using Gang Programmer
  - Offers Read and Write Access to the Flash Memory Plane
  - Enables Control of Lock Bits and General-purpose NVM Bits
  - Enables Security Bit Activation
  - Disabled Once Security Bit is Set
- Parallel Fast Flash Programming Interface
  - Provides an 16-bit Parallel Interface to Program the Embedded Flash
  - Full Handshake Protocol

23.3 Parallel Fast Flash Programming

23.3.1 Device Configuration

In Fast Flash Programming mode, the device is in a specific test mode. Only a certain set of pins is significant. The rest of the PIOs are used as inputs with a pull-up. The crystal oscillator is in bypass mode. Other pins must be left unconnected.

Figure 23-1. 16-bit Parallel Programming Interface



29.4 Slow Clock

The Supply Controller embeds a slow clock generator that is supplied with the VDDBU power supply. As soon as VDDBU is supplied, both the 32.768 kHz crystal oscillator and the embedded 32 kHz (typical) RC oscillator are powered up, but only the RC oscillator is enabled. This allows the slow clock to be valid in a short time (about 100 µs).

The slow clock is generated either by the 32.768 kHz crystal oscillator or by the embedded 32 kHz (typical) RC oscillator.

The selection of the slow clock source is made via the XTALSEL bit in the Supply Controller Control Register (SUPC_CR).

The OSCSEL bit of the Supply Controller Status Register (SUPC_SR) and the OSCSEL bit of the PMC Status Register (PMC_SR) report which oscillator is selected as the slow clock source. PMC_SR.OSCSEL informs when the switch sequence initiated by a new value written in SUPC_CR.XTALSEL is done.

29.4.1 Embedded 32 kHz (typical) RC Oscillator

By default, the embedded 32 kHz (typical) RC oscillator is enabled and selected. The user has to take into account the possible drifts of this oscillator. More details are given in the section “DC Characteristics”.

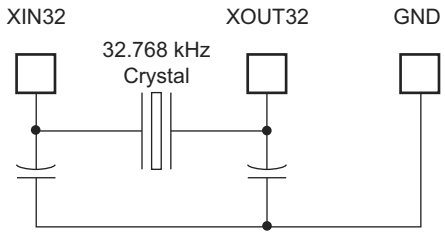
This oscillator is disabled by clearing the SUPC_CR.XTALSEL.

29.4.2 32.768 kHz Crystal Oscillator

The Clock Generator integrates a low-power 32.768 kHz crystal oscillator. To use this oscillator, the XIN32 and XOUT32 pins must be connected to a 32.768 kHz crystal. Two external capacitors must be wired as shown in Figure 29-2. More details are given in the section “DC Characteristics”.

Note that the user is not obliged to use the 32.768 kHz crystal oscillator and can use the 32 kHz (typical) RC oscillator instead.

Figure 29-2. Typical 32768 Crystal Oscillator Connection



The 32.768 kHz crystal oscillator provides a more accurate frequency than the 32 kHz (typical) RC oscillator.

To select the 32.768 kHz crystal oscillator as the source of the slow clock, the bit SUPC_CR.XTALSEL must be set. This results in a sequence which enables the 32.768 kHz crystal oscillator and then disables the 32 kHz (typical) RC oscillator to save power. The switch of the slow clock source is glitch-free.

Reverting to the 32 kHz (typical) RC oscillator is only possible by shutting down the VDDBU power supply. If the user does not need the 32.768 kHz crystal oscillator, the XIN32 and XOUT32 pins can be left unconnected.

The user can also set the 32.768 kHz crystal oscillator in Bypass mode instead of connecting a crystal. In this case, the user must provide the external clock signal on XIN32. The input characteristics of the XIN32 pin are given in the section “Electrical Characteristics”. To enter Bypass mode, the OSCBYPASS bit of the Supply Controller Mode Register (SUPC_MR) must be set prior to setting SUPC_CR.XTALSEL.

However, when the clock is disabled, not all of the features of the PIO Controller are available, including glitch filtering. Note that the input change interrupt, the interrupt modes on a programmable event and the read of the pin level require the clock to be validated.

After a hardware reset, the peripheral clock is disabled by default.

The user must configure the Power Management Controller before any access to the input line information.

32.4.3 Interrupt Sources

For interrupt handling, the PIO Controllers are considered as user peripherals. This means that the PIO Controller interrupt lines are connected among the interrupt sources. Refer to the PIO Controller peripheral identifier in Table 11-1 “Peripheral Identifiers” to identify the interrupt sources dedicated to the PIO Controllers. Using the PIO Controller requires the Interrupt Controller to be programmed first.

The PIO Controller interrupt can be generated only if the peripheral clock is enabled.

Table 32-1. Peripheral IDs

Instance	ID
PIOA	11
PIOB	12
PIOC	37

32.6.30 PIO Pad Pull-Down Disable Register

Name: PIO_PPDDR

Address: 0x400E0E90 (PIOA), 0x400E1090 (PIOB), 0x4800C090 (PIOC)

Access: Write-only

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24
23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16
15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8
7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

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

- **P0–P31: Pull-Down Disable**

0: No effect.

1: Disables the pull-down resistor on the I/O line.

36.7.21 USART Manchester Configuration Register

Name: US_MAN

Address: 0x40024050 (0), 0x40028050 (1), 0x4002C050 (2), 0x40030050 (3), 0x40034050 (4)

Access: Read/Write

31	30	29	28	27	26	25	24
–	DRIFT	ONE	RX_MPOL	–	–	RX_PP	

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

15	14	13	12	11	10	9	8
–	–	–	TX_MPOL	–	–	TX_PP	

7	6	5	4	3	2	1	0
–	–	–	–	TX_PL			

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

- **TX_PL: Transmitter Preamble Length**

0: The transmitter preamble pattern generation is disabled

1–15: The preamble length is TX_PL × Bit Period

- **TX_PP: Transmitter Preamble Pattern**

The following values assume that TX_MPOL field is not set:

Value	Name	Description
0	ALL_ONE	The preamble is composed of ‘1’s
1	ALL_ZERO	The preamble is composed of ‘0’s
2	ZERO_ONE	The preamble is composed of ‘01’s
3	ONE_ZERO	The preamble is composed of ‘10’s

- **TX_MPOL: Transmitter Manchester Polarity**

0: Logic zero is coded as a zero-to-one transition, Logic one is coded as a one-to-zero transition.

1: Logic zero is coded as a one-to-zero transition, Logic one is coded as a zero-to-one transition.

- **RX_PL: Receiver Preamble Length**

0: The receiver preamble pattern detection is disabled

1–15: The detected preamble length is RX_PL × Bit Period

- **RX_PP: Receiver Preamble Pattern detected**

The following values assume that RX_MPOL field is not set:

Value	Name	Description
00	ALL_ONE	The preamble is composed of ‘1’s
01	ALL_ZERO	The preamble is composed of ‘0’s
10	ZERO_ONE	The preamble is composed of ‘01’s
11	ONE_ZERO	The preamble is composed of ‘10’s

39.8.6 SLCDC Interrupt Enable Register

Name: SLCDC_IER

Address: 0x4003C020

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
–	–	–	–	–	DIS	–	ENDFRAME

- **ENDFRAME: End of Frame Interrupt Enable**

0: No effect.

1: Enables the corresponding interrupt.

- **DIS: SLCDC Disable Completion Interrupt Enable**

0: No effect.

1: Enables the corresponding interrupt.

40.7.5 ADC Channel Disable Register

Name: ADC_CHDR

Address: 0x40038014

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
CH7	CH6	–	–	CH3	CH2	CH1	CH0

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

- **CHx: Channel x Disable**

0: No effect.

1: Disables the corresponding channel.

Warning: If the corresponding channel is disabled during a conversion or if it is disabled and then reenabled during a conversion, the associated data and corresponding EOCx and GOVRE flags in ADC_ISR and OVREx flags in ADC_OVER are unpredictable.

40.7.6 ADC Channel Status Register

Name: ADC_CHSR

Address: 0x40038018

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
CH7	CH6	–	–	CH3	CH2	CH1	CH0

- **CHx: Channel x Status**

0: The corresponding channel is disabled.

1: The corresponding channel is enabled.

42. Advanced Encryption Standard (AES)

42.1 Description

The Advanced Encryption Standard (AES) is compliant with the American *FIPS (Federal Information Processing Standard) Publication 197* specification.

The AES supports all five confidentiality modes of operation for symmetrical key block cipher algorithms (ECB, CBC, OFB, CFB and CTR), as specified in the *NIST Special Publication 800-38A Recommendation,* as well as Galois/Counter Mode (GCM) as specified in the *NIST Special Publication 800-38D Recommendation*. It is compatible with all these modes via Peripheral DMA Controller channels, minimizing processor intervention for large buffer transfers.

The 128-bit/192-bit/256-bit key is stored in four/six/eight 32-bit write-only AES Key Word Registers (AES_KEYWR0–3).

The 128-bit input data and initialization vector (for some modes) are each stored in four 32-bit write-only AES Input Data Registers (AES_IDATAR0–3) and AES Initialization Vector Registers (AES_IVR0–3).

As soon as the initialization vector, the input data and the key are configured, the encryption/decryption process may be started. Then the encrypted/decrypted data are ready to be read out on the four 32-bit AES Output Data Registers (AES_ODATAR0–3) or through the PDC channels.

42.2 Embedded Characteristics

- Compliant with *FIPS Publication 197, Advanced Encryption Standard (AES)*
- 128-bit/192-bit/256-bit Cryptographic Key
- 12/14/16 Clock Cycles Encryption/Decryption Processing Time with a 128-bit/192-bit/256-bit Cryptographic Key
- Double Input Buffer Optimizes Runtime
- Support of the Modes of Operation Specified in the *NIST Special Publication 800-38A* and *NIST Special Publication 800-38D*:
  - Electronic Code Book (ECB)
  - Cipher Block Chaining (CBC) including CBC-MAC
  - Cipher Feedback (CFB)
  - Output Feedback (OFB)
  - Counter (CTR)
  - Galois/Counter Mode (GCM)
- 8, 16, 32, 64 and 128-bit Data Sizes Possible in CFB Mode
- Last Output Data Mode Allows Optimized Message Authentication Code (MAC) Generation
- Connection to PDC Channel Capabilities Optimizes Data Transfers for all Operating Modes
  - One Channel for the Receiver, One Channel for the Transmitter
  - Next Buffer Support

42.3 Product Dependencies

42.3.1 Power Management

The AES may be clocked through the Power Management Controller (PMC), so the programmer must first to configure the PMC to enable the AES clock.

Figure 46-6. SPI Slave Mode with (CPOL=0 and NCPHA=1) or (CPOL=1 and NCPHA=0)

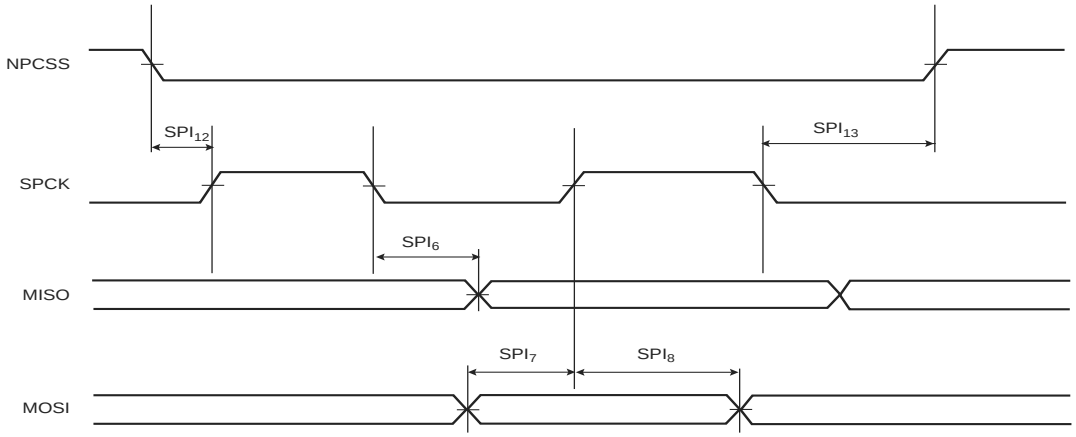
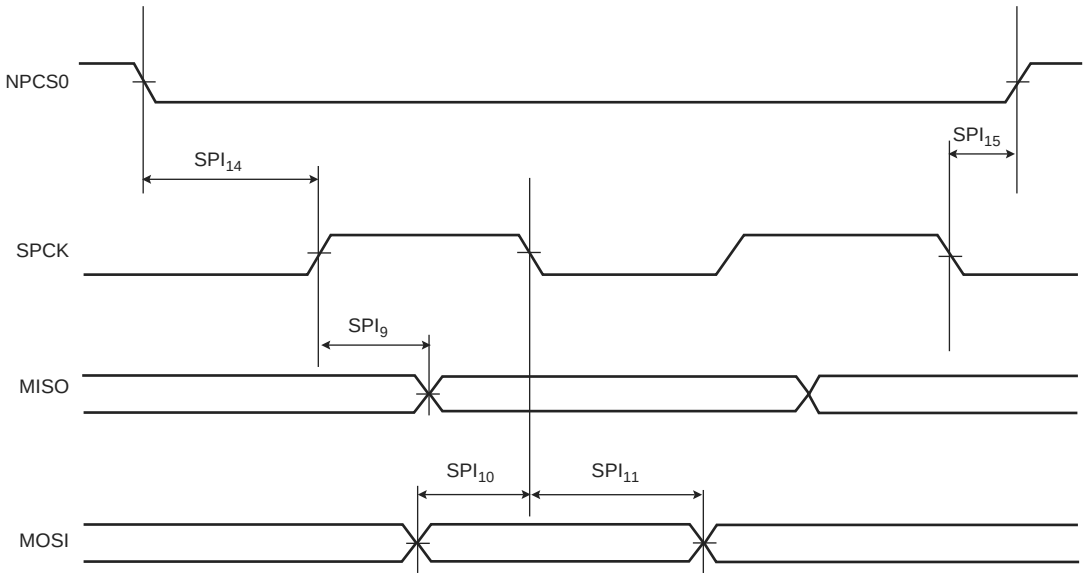


Figure 46-7. SPI Slave Mode with (CPOL = NCPHA = 0) or (CPOL= NCPHA= 1)



46.4.3.1 Maximum SPI Frequency

The formulas that follow give the maximum SPI frequency in Master Write and Read modes, and in Slave Read and Write modes.

Master Write Mode

The SPI is only sending data to a slave device such as an LCD, for example. The limit is given by SPI_2 (or SPI_5) timing. Since it gives a maximum frequency above the maximum pad speed (refer to Section 46.4.2 “I/O AC Characteristics”), the max SPI frequency is the one from the pad.

46.4.5.1 USART SPI TImings

Table 46-15. USART SPI Timings

Symbol	Parameter	Conditions	Min	Max	Unit
Master Mode					
SPI_0	SCK period	1.8V domain 3.3V domain	6 / MCK	–	–
SPI_1	Input data setup time	1.8V domain 3.3V domain	0.5 * MCK + 1.1 0.5 * MCK + 1.1	–	–
SPI_2	Input data hold time	1.8V domain 3.3V domain	1.5 * MCK + 4.8 1.5 * MCK + 4.8	–	–
SPI_3	Chip select active to serial clock	1.8V domain 3.3V domain	1.5 * SPCK + 0.9 1.5 * SPCK + 0.9	–	–
SPI_4	Output data setup time	1.8V domain 3.3V domain	- 6.7 - 6.7	7.1 7.1	ns
SPI_5	Serial clock to chip select inactive	1.8V domain 3.3V domain	1 * SPCK - 6.0 1 * SPCK - 6.0	–	ns
Slave Mode					
SPI_6	SCK falling to MISO	1.8V domain 3.3V domain	6.8 6.8	20.7 20.7	ns
SPI_7	MOSI setup time before SCK rises	1.8V domain 3.3V domain	2 * MCK + 0.2 2 * MCK + 0.2	–	ns
SPI_8	MOSI hold time after SCK rises	1.8V domain 3.3V domain	4.2 4.2	–	ns
SPI_9	SCK rising to MISO	1.8V domain 3.3V domain	8.1 8.1	19.8 19.8	ns
SPI_{10}	MOSI setup time before SCK falls	1.8V domain 3.3V domain	2 * MCK + 1 2 * MCK + 1	–	ns
SPI_{11}	MOSI hold time after SCK falls	1.8V domain 3.3V domain	5.2 5.2	–	ns
SPI_{12}	NPCS0 setup to SCK rising	1.8V domain 3.3V domain	2.5 * MCK - 0.4 2.5 * MCK - 0.4	–	ns
SPI_{13}	NPCS0 hold after SCK falling	1.8V domain 3.3V domain	1.5 * MCK + 5.5 1.5 * MCK + 5.5	–	ns
SPI_{14}	NPCS0 setup to SCK falling	1.8V domain 3.3V domain	2.5 * MCK + 0.2 2.5 * MCK + 0.2	–	ns
SPI_{15}	NPCS0 hold after SCK rising	1.8V domain 3.3V domain	1.5 * MCK + 4.5 1.5 * MCK + 4.5	–	ns