



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, I ² C, 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	128KB (128K 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 ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TFBGA
Supplier Device Package	100-TFBGA (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsam4s2cb-cnr

- For all other instructions that use labels, the value of the PC is the address of the current instruction plus 4 bytes, with bit[1] of the result cleared to 0 to make it word-aligned.
- Your assembler might permit other syntaxes for PC-relative expressions, such as a label plus or minus a number, or an expression of the form [PC, #number].

12.6.3.7 Conditional Execution

Most data processing instructions can optionally update the condition flags in the *Application Program Status Register* (APSR) according to the result of the operation, see “Application Program Status Register” . Some instructions update all flags, and some only update a subset. If a flag is not updated, the original value is preserved. See the instruction descriptions for the flags they affect.

An instruction can be executed conditionally, based on the condition flags set in another instruction, either:

- Immediately after the instruction that updated the flags
- After any number of intervening instructions that have not updated the flags.

Conditional execution is available by using conditional branches or by adding condition code suffixes to instructions. See Table 12-16 for a list of the suffixes to add to instructions to make them conditional instructions. The condition code suffix enables the processor to test a condition based on the flags. If the condition test of a conditional instruction fails, the instruction:

- Does not execute
- Does not write any value to its destination register
- Does not affect any of the flags
- Does not generate any exception.

Conditional instructions, except for conditional branches, must be inside an If-Then instruction block. See “IT” for more information and restrictions when using the IT instruction. Depending on the vendor, the assembler might automatically insert an IT instruction if there are conditional instructions outside the IT block.

The CBZ and CBNZ instructions are used to compare the value of a register against zero and branch on the result.

This section describes:

- “Condition Flags”
- “Condition Code Suffixes” .

Condition Flags

The APSR contains the following condition flags:

N	Set to 1 when the result of the operation was negative, cleared to 0 otherwise.
Z	Set to 1 when the result of the operation was zero, cleared to 0 otherwise.
C	Set to 1 when the operation resulted in a carry, cleared to 0 otherwise.
V	Set to 1 when the operation caused overflow, cleared to 0 otherwise.

For more information about the APSR, see “Program Status Register” .

A carry occurs:

- If the result of an addition is greater than or equal to 2^{32}
- If the result of a subtraction is positive or zero
- As the result of an inline barrel shifter operation in a move or logical instruction.

An overflow occurs when the sign of the result, in bit[31], does not match the sign of the result, had the operation been performed at infinite precision, for example:

- If adding two negative values results in a positive value
- If adding two positive values results in a negative value
- If subtracting a positive value from a negative value generates a positive value

12.6.4.1 ADR

Load PC-relative address.

Syntax

`ADR{cond} Rd, label`

where:

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

Rd is the destination register.

label is a PC-relative expression. See “PC-relative Expressions” .

Operation

ADR determines the address by adding an immediate value to the PC, and writes the result to the destination register.

ADR produces position-independent code, because the address is PC-relative.

If ADR is used to generate a target address for a BX or BLX instruction, ensure that bit[0] of the address generated is set to 1 for correct execution.

Values of *label* must be within the range of –4095 to +4095 from the address in the PC.

Note: The user might have to use the .W suffix to get the maximum offset range or to generate addresses that are not word-aligned. See “Instruction Width Selection” .

Restrictions

Rd must not be SP and must not be PC.

Condition Flags

This instruction does not change the flags.

Examples

```
ADR    R1, TextMessage    ; Write address value of a location labelled as
                          ; TextMessage to R1
```

12.9.1.16 MemManage Fault Address Register

Name: SCB_MMFAR

Access: Read/Write

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

The SCB_MMFAR contains the address of the location that generated a memory management fault.

- **ADDRESS: Memory Management Fault Generation Location Address**

When the MMARVALID bit of the MMFSR subregister is set to 1, this field holds the address of the location that generated the memory management fault.

- Notes:
1. When an unaligned access faults, the address is the actual address that faulted. Because a single read or write instruction can be split into multiple aligned accesses, the fault address can be any address in the range of the requested access size.
 2. Flags in the MMFSR subregister indicate the cause of the fault, and whether the value in the SCB_MMFAR is valid. See “MMFSR: Memory Management Fault Status Subregister” .

12.10.1.1 SysTick Control and Status Register

Name: SYST_CSR

Access: Read/Write

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

The SysTick SYST_CSR enables the SysTick features.

- **COUNTFLAG: Count Flag**

Returns 1 if the timer counted to 0 since the last time this was read.

- **CLKSOURCE: Clock Source**

Indicates the clock source:

0: External Clock.

1: Processor Clock.

- **TICKINT: SysTick Exception Request Enable**

Enables a SysTick exception request:

0: Counting down to zero does not assert the SysTick exception request.

1: Counting down to zero asserts the SysTick exception request.

The software can use COUNTFLAG to determine if SysTick has ever counted to zero.

- **ENABLE: Counter Enable**

Enables the counter:

0: Counter disabled.

1: Counter enabled.

When ENABLE is set to 1, the counter loads the RELOAD value from the SYST_RVR and then counts down. On reaching 0, it sets the COUNTFLAG to 1 and optionally asserts the SysTick depending on the value of TICKINT. It then loads the RELOAD value again, and begins counting.

18.4 Functional Description

18.4.1 Overview

The device is divided into two power supply areas:

- VDDIO power supply: includes the Supply Controller, part of the Reset Controller, the slow clock switch, the general-purpose backup registers, the supply monitor and the clock which includes the Real-time Timer and the Real-time Clock.
- Core power supply: includes part of the Reset Controller, the Brownout Detector, the processor, the SRAM memory, the Flash memory and the peripherals.

The Supply Controller (SUPC) controls the supply voltage of the core power supply. The SUPC intervenes when the VDDIO power supply rises (when the system is starting) or when Backup mode is entered.

The SUPC also integrates the slow clock generator, which is based on a 32 kHz crystal oscillator, and an embedded 32 kHz RC oscillator. The slow clock defaults to the RC oscillator, but the software can enable the crystal oscillator and select it as the slow clock source.

The SUPC and the VDDIO power supply have a reset circuitry based on a zero-power power-on reset cell. The zero-power power-on reset allows the SUPC to start correctly as soon as the VDDIO voltage becomes valid.

At start-up of the system, once the backup voltage VDDIO is valid and the embedded 32 kHz RC oscillator is stabilized, the SUPC starts up the core by sequentially enabling the internal voltage regulator. The SUPC waits until the core voltage VDDCORE is valid, then releases the reset signal of the core `vddcore_nreset` signal.

Once the system has started, the user can program a supply monitor and/or a brownout detector. If the supply monitor detects a voltage level on VDDIO that is too low, the SUPC asserts the reset signal of the core `vddcore_nreset` signal until VDDIO is valid. Likewise, if the brownout detector detects a core voltage level VDDCORE that is too low, the SUPC asserts the reset signal `vddcore_nreset` until VDDCORE is valid.

When Backup mode is entered, the SUPC sequentially asserts the reset signal of the core power supply `vddcore_nreset` and disables the voltage regulator, in order to supply only the VDDIO power supply. Current consumption is reduced to a few microamps for the backup part retention. Exit from this mode is possible on multiple wake-up sources including an event on WKUP pins, or a clock alarm. To exit this mode, the SUPC operates in the same way as system start-up.

The erase sequence is the following:

1. Erase starts as soon as one of the erase commands and the FARG field are written in EEFC_FCR.
 - For the EPA command, the two lowest bits of the FARG field define the number of pages to be erased (FARG[1:0]):

Table 20-4. EEFC_FCR.FARG Field for EPA Command

FARG[1:0]	Number of pages to be erased with EPA command
0	4 pages (only valid for small 8 KB sectors)
1	8 pages
2	16 pages
3	32 pages (not valid for small 8 KB sectors)

2. When erasing is completed, the bit EEFC_FSR.FRDY rises. If an interrupt has been enabled by setting the bit EEFC_FMR.FRDY, the interrupt line of the interrupt controller is activated.

Three errors can be detected in EEFC_FSR after an erasing sequence:

- Command Error: A bad keyword has been written in EEFC_FCR.
- Lock Error: At least one page to be erased belongs to a locked region. The erase command has been refused, no page has been erased. A command must be run previously to unlock the corresponding region.
- Flash Error: At the end of the erase period, the EraseVerify test of the Flash memory has failed.

20.4.3.4 Lock Bit Protection

Lock bits are associated with several pages in the embedded Flash memory plane. This defines lock regions in the embedded Flash memory plane. They prevent writing/erasing protected pages.

The lock sequence is the following:

1. Execute the 'Set Lock Bit' command by writing EEFC_FCR.FCMD with the SLB command and EEFC_FCR.FARG with a page number to be protected.
2. When the locking completes, the bit EEFC_FSR.FRDY rises. If an interrupt has been enabled by setting the bit EEFC_FMR.FRDY, the interrupt line of the interrupt controller is activated.
3. The result of the SLB command can be checked running a 'Get Lock Bit' (GLB) command.

Note: The value of the FARG argument passed together with SLB command must not exceed the higher lock bit index available in the product.

Two errors can be detected in EEFC_FSR after a programming sequence:

- Command Error: A bad keyword has been written in EEFC_FCR.
- Flash Error: At the end of the programming, the EraseVerify or WriteVerify test of the Flash memory has failed.

It is possible to clear lock bits previously set. After the lock bits are cleared, the locked region can be erased or programmed. The unlock sequence is the following:

1. Execute the 'Clear Lock Bit' command by writing EEFC_FCR.FCMD with the CLB command and EEFC_FCR.FARG with a page number to be unprotected.
2. When the unlock completes, the bit EEFC_FSR.FRDY rises. If an interrupt has been enabled by setting the bit EEFC_FMR.FRDY, the interrupt line of the interrupt controller is activated.

Note: The value of the FARG argument passed together with CLB command must not exceed the higher lock bit index available in the product.

Two errors can be detected in EEFC_FSR after a programming sequence:

- Command Error: A bad keyword has been written in EEFC_FCR.

20.5.4 EEFC Flash Result Register

Name: EEFC_FRR
Address: 0x400E0A0C (0), 0x400E0C0C (1)
Access: Read-only

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

• **FVALUE: Flash Result Value**

The result of a Flash command is returned in this register. If the size of the result is greater than 32 bits, the next resulting value is accessible at the next register read.

29.17.21PMC Write Protection Mode Register

Name: PMC_WPMR

Address: 0x400E04E4

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 0x504D43 (“PMC” in ASCII).

1: Enables the write protection if WPKEY corresponds to 0x504D43 (“PMC” in ASCII).

See Section 29.16 “Register Write Protection” for the list of registers that can be write-protected.

- **WPKEY: Write Protection Key**

Value	Name	Description
0x504D43	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

31.6.6 PIO Output Status Register

Name: PIO_OSR

Address: 0x400E0E18 (PIOA), 0x400E1018 (PIOB), 0x400E1218 (PIOC)

Access: Read-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

- **P0–P31: Output Status**

0: The I/O line is a pure input.

1: The I/O line is enabled in output.

31.6.10 PIO Set Output Data Register

Name: PIO_SODR

Address: 0x400E0E30 (PIOA), 0x400E1030 (PIOB), 0x400E1230 (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

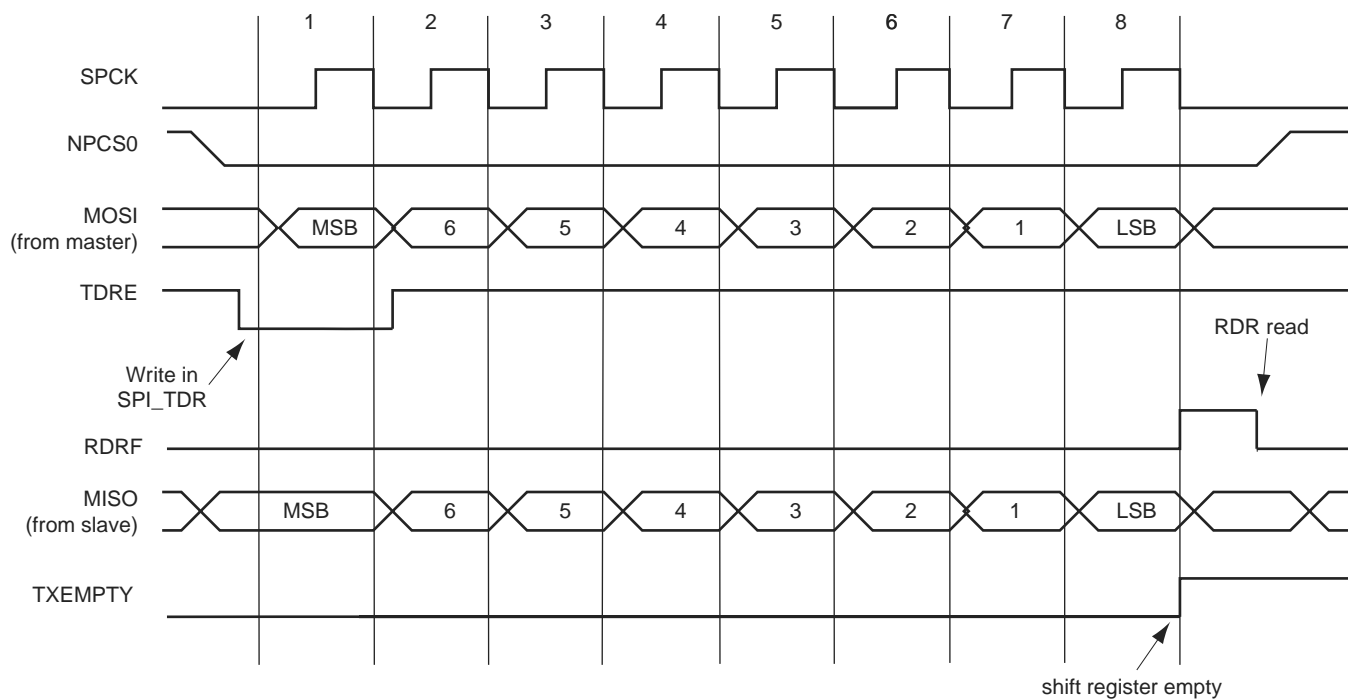
- **P0–P31: Set Output Data**

0: No effect.

1: Sets the data to be driven on the I/O line.

Figure 33-8 shows the behavior of Transmit Data Register Empty (TDRE), Receive Data Register (RDRF) and Transmission Register Empty (TXEMPTY) status flags within the SPI_SR during an 8-bit data transfer in Fixed mode without the PDC involved.

Figure 33-8. Status Register Flags Behavior



37.7.9 TC Status Register

Name: TC_SRx [x=0..2]

Address: 0x40010020 (0)[0], 0x40010060 (0)[1], 0x400100A0 (0)[2], 0x40014020 (1)[0], 0x40014060 (1)[1], 0x400140A0 (1)[2]

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	MTIOB	MTIOA	CLKSTA
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS

- **COVFS: Counter Overflow Status (cleared on read)**

0: No counter overflow has occurred since the last read of the Status Register.

1: A counter overflow has occurred since the last read of the Status Register.

- **LOVRS: Load Overrun Status (cleared on read)**

0: Load overrun has not occurred since the last read of the Status Register or TC_CMRx.WAVE = 1.

1: RA or RB have been loaded at least twice without any read of the corresponding register since the last read of the Status Register, if TC_CMRx.WAVE = 0.

- **CPAS: RA Compare Status (cleared on read)**

0: RA Compare has not occurred since the last read of the Status Register or TC_CMRx.WAVE = 0.

1: RA Compare has occurred since the last read of the Status Register, if TC_CMRx.WAVE = 1.

- **CPBS: RB Compare Status (cleared on read)**

0: RB Compare has not occurred since the last read of the Status Register or TC_CMRx.WAVE = 0.

1: RB Compare has occurred since the last read of the Status Register, if TC_CMRx.WAVE = 1.

- **CPCS: RC Compare Status (cleared on read)**

0: RC Compare has not occurred since the last read of the Status Register.

1: RC Compare has occurred since the last read of the Status Register.

- **LDRAS: RA Loading Status (cleared on read)**

0: RA Load has not occurred since the last read of the Status Register or TC_CMRx.WAVE = 1.

1: RA Load has occurred since the last read of the Status Register, if TC_CMRx.WAVE = 0.

- **LDRBS: RB Loading Status (cleared on read)**

0: RB Load has not occurred since the last read of the Status Register or TC_CMRx.WAVE = 1.

1: RB Load has occurred since the last read of the Status Register, if TC_CMRx.WAVE = 0.

37.7.12 TC Interrupt Mask Register

Name: TC_IMRx [x=0..2]

Address: 0x4001002C (0)[0], 0x4001006C (0)[1], 0x400100AC (0)[2], 0x4001402C (1)[0], 0x4001406C (1)[1], 0x400140AC (1)[2]

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
ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS

- **COVFS: Counter Overflow**

0: The Counter Overflow Interrupt is disabled.

1: The Counter Overflow Interrupt is enabled.

- **LOVRS: Load Overrun**

0: The Load Overrun Interrupt is disabled.

1: The Load Overrun Interrupt is enabled.

- **CPAS: RA Compare**

0: The RA Compare Interrupt is disabled.

1: The RA Compare Interrupt is enabled.

- **CPBS: RB Compare**

0: The RB Compare Interrupt is disabled.

1: The RB Compare Interrupt is enabled.

- **CPCS: RC Compare**

0: The RC Compare Interrupt is disabled.

1: The RC Compare Interrupt is enabled.

- **LDRAS: RA Loading**

0: The Load RA Interrupt is disabled.

1: The Load RA Interrupt is enabled.

- **LDRBS: RB Loading**

0: The Load RB Interrupt is disabled.

1: The Load RB Interrupt is enabled.

38.14.15HSMCI Interrupt Mask Register

Name: HSMCI_IMR

Address: 0x4000004C

Access: Read-only

31	30	29	28	27	26	25	24
UNRE	OVRE	ACKRCVE	ACKRCV	XFRDONE	FIFOEMPTY	–	–
23	22	21	20	19	18	17	16
CSTOE	DTOE	DCRCE	RTOE	RENDE	RCRCE	RDIRE	RINDE
15	14	13	12	11	10	9	8
TXBUFE	RXBUFF	CSRCV	SDIOWAIT	–	–	–	SDIOIRQA
7	6	5	4	3	2	1	0
ENDTX	ENDRX	NOTBUSY	DTIP	BLKE	TXRDY	RXRDY	CMDRDY

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

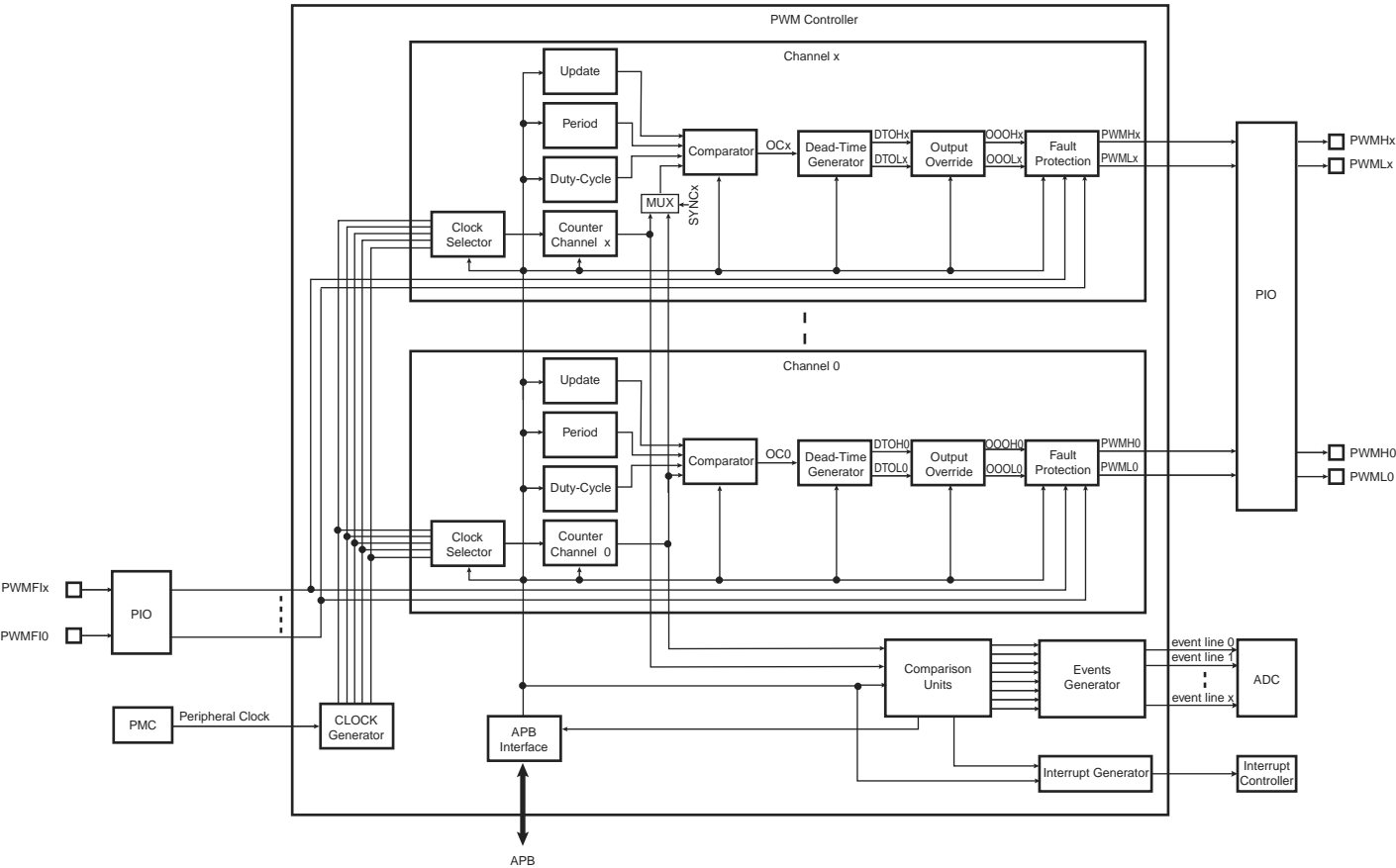
0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

- **CMDRDY: Command Ready Interrupt Mask**
- **RXRDY: Receiver Ready Interrupt Mask**
- **TXRDY: Transmit Ready Interrupt Mask**
- **BLKE: Data Block Ended Interrupt Mask**
- **DTIP: Data Transfer in Progress Interrupt Mask**
- **NOTBUSY: Data Not Busy Interrupt Mask**
- **ENDRX: End of Receive Buffer Interrupt Mask**
- **ENDTX: End of Transmit Buffer Interrupt Mask**
- **SDIOIRQA: SDIO Interrupt for Slot A Interrupt Mask**
- **SDIOWAIT: SDIO Read Wait Operation Status Interrupt Mask**
- **CSRCV: Completion Signal Received Interrupt Mask**
- **RXBUFF: Receive Buffer Full Interrupt Mask**
- **TXBUFE: Transmit Buffer Empty Interrupt Mask**
- **RINDE: Response Index Error Interrupt Mask**
- **RDIRE: Response Direction Error Interrupt Mask**
- **RCRCE: Response CRC Error Interrupt Mask**
- **RENDE: Response End Bit Error Interrupt Mask**

39.3 Block Diagram

Figure 39-1. Pulse Width Modulation Controller Block Diagram



39.4 I/O Lines Description

Each channel outputs two complementary external I/O lines.

Table 39-1. I/O Line Description

Name	Description	Type
PWMHx	PWM Waveform Output High for channel x	Output
PWMLx	PWM Waveform Output Low for channel x	Output
PWMFx	PWM Fault Input x	Input

39.5 Product Dependencies

39.5.1 I/O Lines

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

All of the PWM outputs may or may not be enabled. If an application requires only four channels, then only four PIO lines will be assigned to PWM outputs.

Table 39-2. I/O Lines

Instance	Signal	I/O Line	Peripheral
PWM	PWMFIO	PA9	C
PWM	PWMF11	PA10	C
PWM	PWMF12	PA18	D
PWM	PWMH0	PA0	A
PWM	PWMH0	PA11	B
PWM	PWMH0	PA23	B
PWM	PWMH0	PB0	A
PWM	PWMH0	PC18	B
PWM	PWMH1	PA1	A
PWM	PWMH1	PA12	B
PWM	PWMH1	PA24	B
PWM	PWMH1	PB1	A
PWM	PWMH1	PC19	B
PWM	PWMH2	PA2	A
PWM	PWMH2	PA13	B
PWM	PWMH2	PA25	B
PWM	PWMH2	PB4	B
PWM	PWMH2	PC20	B
PWM	PWMH3	PA7	B
PWM	PWMH3	PA14	B
PWM	PWMH3	PA17	C
PWM	PWMH3	PB14	B
PWM	PWMH3	PC21	B
PWM	PWML0	PA19	B
PWM	PWML0	PB5	B
PWM	PWML0	PC0	B
PWM	PWML0	PC13	B
PWM	PWML1	PA20	B
PWM	PWML1	PB12	A
PWM	PWML1	PC1	B
PWM	PWML1	PC15	B
PWM	PWML2	PA16	C
PWM	PWML2	PA30	A
PWM	PWML2	PB13	A
PWM	PWML2	PC2	B

40.5.1 USB Device Transceiver

The USB device transceiver is embedded in the product. However, discrete components are required for each of the following actions:

- to monitor VBUS voltage
- for line termination
- to disconnect the host for reduced power consumption

40.5.2 VBUS Monitoring

VBUS monitoring is required to detect host connection. VBUS monitoring is done using a standard PIO with internal pull-up disabled. When the host is switched off, it should be considered as a disconnect, the pull-up must be disabled in order to prevent powering the host through the pull-up resistor.

When the host is disconnected and the transceiver is enabled, then DDP and DDM are floating. This may lead to over consumption. A solution is to enable the integrated pull-down by disabling the transceiver (UDP_TXVC.TXVDIS = 1) and then remove the pull-up (UDP_TXVC.PUON = 0).

A termination serial resistor must be connected to DDP and DDM. The resistor value is defined in the electrical specification of the product (R_{EXT}).

41.7.3 ACC Interrupt Enable Register

Name: ACC_IER

Address: 0x40040024

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
–	–	–	–	–	–	–	CE

- **CE: Comparison Edge**

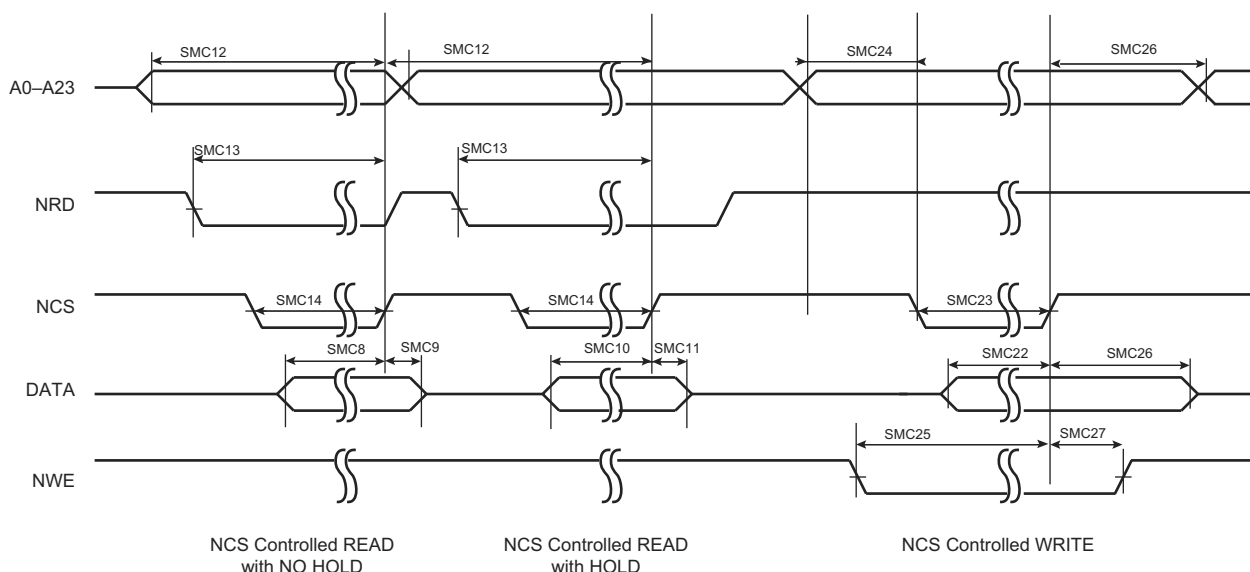
0: No effect.

1: Enables the interrupt when the selected edge (defined by EDGETYP) occurs.

Table 44-70. SMC Write Signals - NCS Controlled (WRITE_MODE = 0)

Symbol	Parameter	Min		Max		Unit
		1.8V Domain	3.3V Domain	1.8V Domain	3.3V Domain	
SMC ₂₂	Data Out Valid before NCS High	$\text{NCS_WR_PULSE} \times t_{\text{CPMCK}} - 6.3$	$\text{NCS_WR_PULSE} \times t_{\text{CPMCK}} - 6.2$	—	—	ns
SMC ₂₃	NCS Pulse Width	$\text{NCS_WR_PULSE} \times t_{\text{CPMCK}} - 7.7$	$\text{NCS_WR_PULSE} \times t_{\text{CPMCK}} - 6.7$	—	—	ns
SMC ₂₄	A0–A22 Valid before NCS Low	$\text{NCS_WR_SETUP} \times t_{\text{CPMCK}} - 6.5$	$\text{NCS_WR_SETUP} \times t_{\text{CPMCK}} - 6.3$	—	—	ns
SMC ₂₅	NWE Low before NCS High	$(\text{NCS_WR_SETUP} - \text{NWE_SETUP} + \text{NCS pulse}) \times t_{\text{CPMCK}} - 5.1$	$(\text{NCS_WR_SETUP} - \text{NWE_SETUP} + \text{NCS pulse}) \times t_{\text{CPMCK}} - 4.9$	—	—	ns
SMC ₂₆	NCS High to Data Out, A0–A25, Change	$\text{NCS_WR_HOLD} \times t_{\text{CPMCK}} - 10.2$	$\text{NCS_WR_HOLD} \times t_{\text{CPMCK}} - 8.4$	—	—	ns
SMC ₂₇	NCS High to NWE Inactive	$(\text{NCS_WR_HOLD} - \text{NWE_HOLD}) \times t_{\text{CPMCK}} - 2.1$	$(\text{NCS_WR_HOLD} - \text{NWE_HOLD}) \times t_{\text{CPMCK}} - 1.6$	—	—	ns

Figure 44-35. SMC Timings - NCS Controlled Read and Write

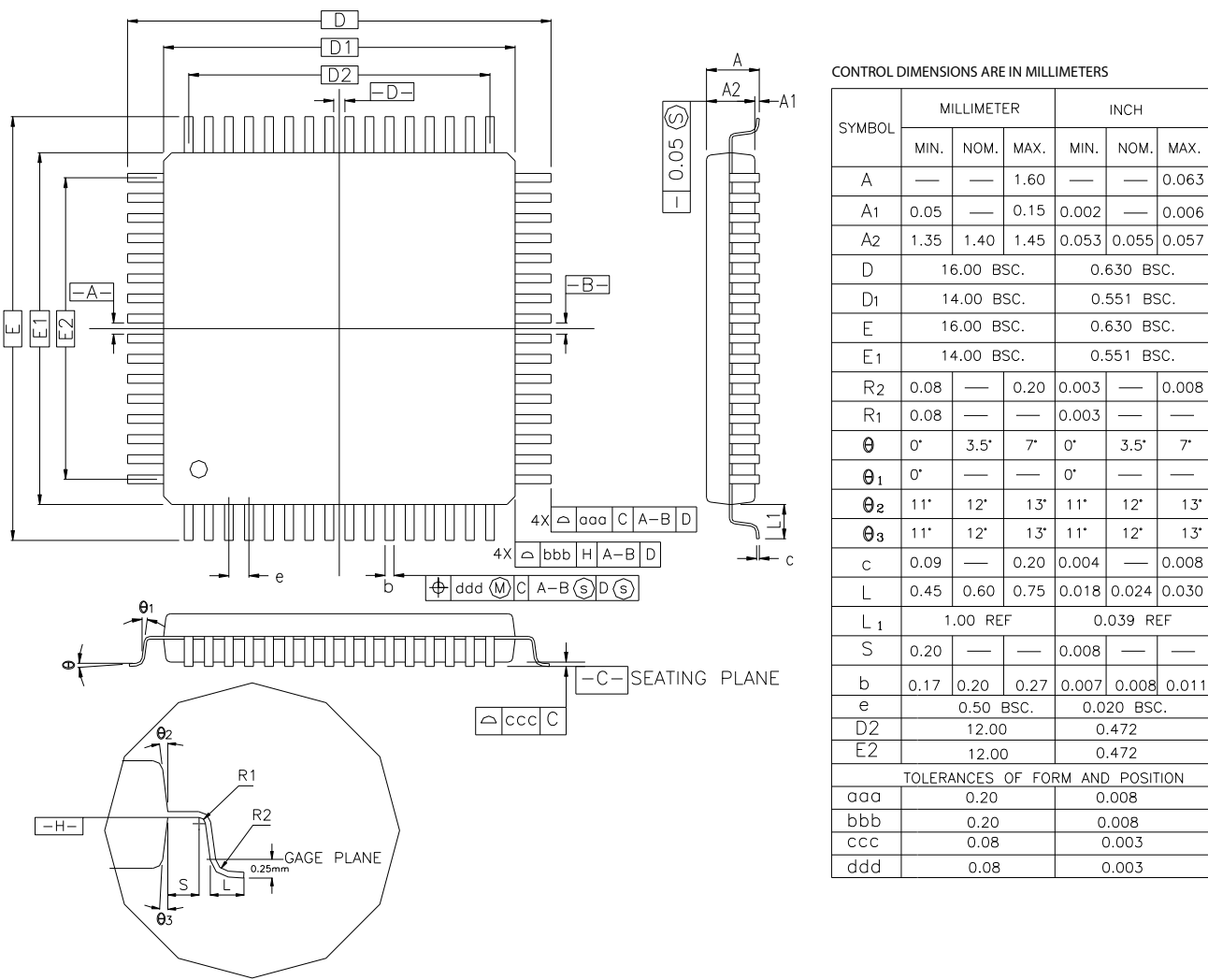


45. Mechanical Characteristics

All packages of the SAM4S devices respect the recommendations of the NEMI User Group.

45.1 100-lead LQFP Mechanical Characteristics

Figure 45-1. 100-lead LQFP Package Mechanical Drawing



Note : 1. This drawing is for general information only. Refer to JEDEC Drawing MS-026 for additional information.

Table 45-1. Device and LQFP Package Maximum Weight

SAM4S	800	mg
-------	-----	----

Table 45-2. Package Reference

JEDEC Drawing Reference	MS-026
JESD97 Classification	e3

Table 45-3. LQFP Package Characteristics

Moisture Sensitivity Level	3
----------------------------	---