Welcome to E-XFL.COM

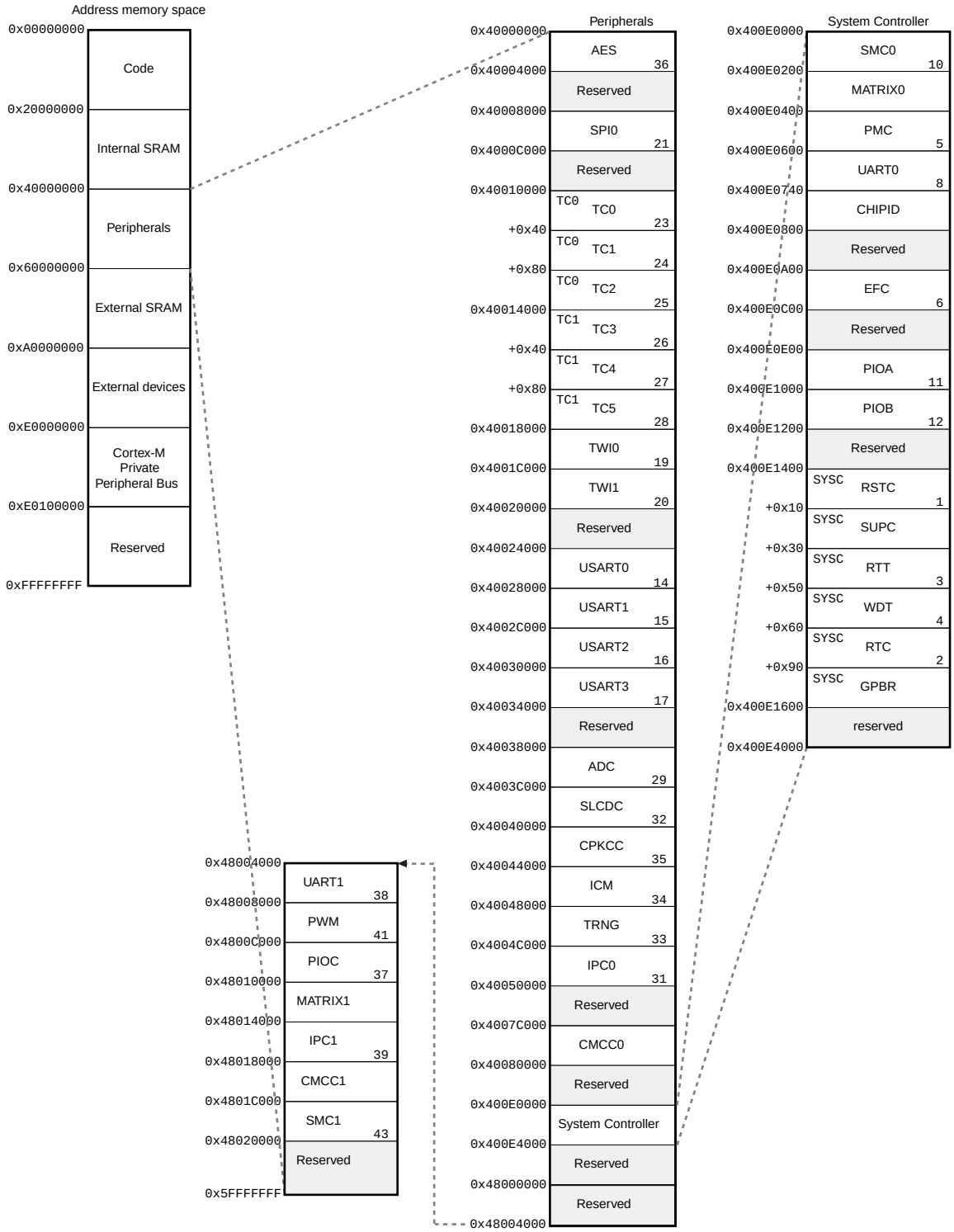
What is "Embedded - Microcontrollers"?

"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	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/atsam4cms8ca-au

Figure 7-4. SAM4CM16/8/4 Memory Mapping of the Peripherals Area



- All conditional instructions except B*cond* must be inside an IT block. B*cond* can be either outside or inside an IT block but has a larger branch range if it is inside one
- Each instruction inside the IT block must specify a condition code suffix that is either the same or logical inverse as for the other instructions in the block.

Your assembler might place extra restrictions on the use of IT blocks, such as prohibiting the use of assembler directives within them.

Condition Flags

This instruction does not change the flags.

Example

```
ITTE   NE           ; Next 3 instructions are conditional
ANDNE  R0, R0, R1   ; ANDNE does not update condition flags
ADDSNE R2, R2, #1   ; ADDSNE updates condition flags
MOVEQ  R2, R3       ; Conditional move


CMP    R0, #9       ; Convert R0 hex value (0 to 15) into ASCII
                    ; ('0'-'9', 'A'-'F')
ITE    GT           ; Next 2 instructions are conditional
ADDGT  R1, R0, #55  ; Convert 0xA -> 'A'
ADDLE  R1, R0, #48  ; Convert 0x0 -> '0'


IT     GT           ; IT block with only one conditional instruction
ADDGT  R1, R1, #1   ; Increment R1 conditionally


ITTEE  EQ           ; Next 4 instructions are conditional
MOVEQ  R0, R1       ; Conditional move
ADDEQ  R2, R2, #10  ; Conditional add
ANDNE  R3, R3, #1   ; Conditional AND
BNE.W  dloop        ; Branch instruction can only be used in the last
                    ; instruction of an IT block


IT     NE           ; Next instruction is conditional
ADD    R0, R0, R1   ; Syntax error: no condition code used in IT block
```

12.6.10.4 TBB and TBH

Table Branch Byte and Table Branch Halfword.

Syntax

```
TBB [Rn, Rm]
TBH [Rn, Rm, LSL #1]
```

where:

Rn is the register containing the address of the table of branch lengths.

If *Rn* is PC, then the address of the table is the address of the byte immediately following the TBB or TBH instruction.

Rm is the index register. This contains an index into the table. For halfword tables, LSL #1 doubles the value in *Rm* to form the right offset into the table.

Operation

These instructions cause a PC-relative forward branch using a table of single byte offsets for TBB, or halfword offsets for TBH. *Rn* provides a pointer to the table, and *Rm* supplies an index into the table. For TBB the branch offset is twice the unsigned value of the byte returned from the table. and for TBH the branch offset is twice the unsigned value of the halfword returned from the table. The branch occurs to the address at that offset from the address of the byte immediately after the TBB or TBH instruction.

Restrictions

The restrictions are:

- *Rn* must not be SP
- *Rm* must not be SP and must not be PC
- When any of these instructions is used inside an IT block, it must be the last instruction of the IT block.

Condition Flags

These instructions do not change the flags.

12.6.11.7 VDIV

Divides floating-point values.

Syntax

```
VDIV{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

This instruction:

1. Divides one floating-point value by another floating-point value.
2. Writes the result to the floating-point destination register.

Restrictions

There are no restrictions.

Condition Flags

These instructions do not change the flags.

12.9.1.14 Configurable Fault Status Register (Byte Access)

Name: SCB_CFSR (BYTE)

Access: Read/Write

31	30	29	28	27	26	25	24
UFSR							

23	22	21	20	19	18	17	16
UFSR							

15	14	13	12	11	10	9	8
BFSR							

7	6	5	4	3	2	1	0
MMFSR							

- **MMFSR: Memory Management Fault Status Subregister**

The flags in the MMFSR subregister indicate the cause of memory access faults. See bitfield [7..0] description in Section 12.9.1.13.

- **BFSR: Bus Fault Status Subregister**

The flags in the BFSR subregister indicate the cause of a bus access fault. See bitfield [14..8] description in Section 12.9.1.13.

- **UFSR: Usage Fault Status Subregister**

The flags in the UFSR subregister indicate the cause of a usage fault. See bitfield [31..15] description in Section 12.9.1.13.

Note: The UFSR bits are sticky. This means that as one or more fault occurs, the associated bits are set to 1. A bit that is set to 1 is cleared to 0 only by writing a 1 to that bit, or by a reset.

The SCB_CFSR indicates the cause of a memory management fault, bus fault, or usage fault. It is byte accessible. The user can access the SCB_CFSR or its subregisters as follows:

- Access complete SCB_CFSR with a word access to 0xE000ED28
- Access MMFSR with a byte access to 0xE000ED28
- Access MMFSR and BFSR with a halfword access to 0xE000ED28
- Access BFSR with a byte access to 0xE000ED29
- Access UFSR with a halfword access to 0xE000ED2A.

12.12 Floating Point Unit (FPU)

The Cortex-M4F FPU implements the FPv4-SP floating-point extension.

The FPU fully supports single-precision add, subtract, multiply, divide, multiply and accumulate, and square root operations. It also provides conversions between fixed-point and floating-point data formats, and floating-point constant instructions.

The FPU provides floating-point computation functionality that is compliant with the ANSI/IEEE Std 754-2008, IEEE Standard for Binary Floating-Point Arithmetic, referred to as the IEEE 754 standard.

The FPU contains 32 single-precision extension registers, which can also be accessed as 16 doubleword registers for load, store, and move operations.

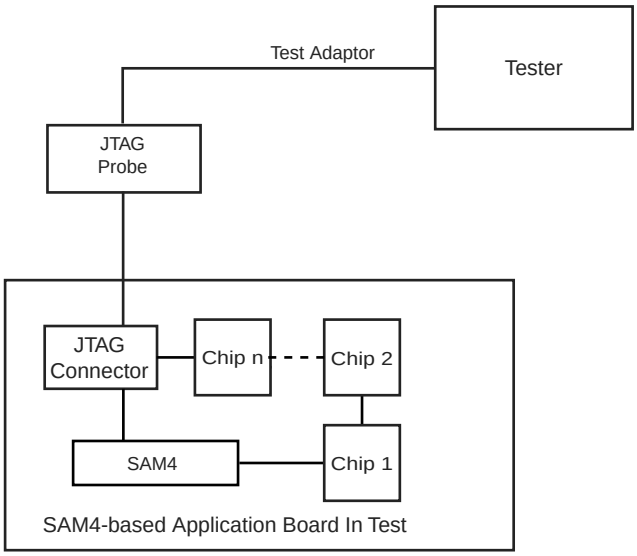
12.12.1 Enabling the FPU

The FPU is disabled from reset. It must be enabled before any floating-point instructions can be used. Example 4-1 shows an example code sequence for enabling the FPU in both privileged and user modes. The processor must be in privileged mode to read from and write to the CPACR.

Example of Enabling the FPU:

```
; CPACR is located at address 0xE000ED88
LDR.W R0, =0xE000ED88
; Read CPACR
LDR R1, [R0]
; Set bits 20-23 to enable CP10 and CP11 coprocessors
ORR R1, R1, #(0xF << 20)
; Write back the modified value to the CPACR
STR R1, [R0]; wait for store to complete
DSB
;reset pipeline now the FPU is enabled
ISB
```

Figure 13-4. Application Test Environment Example



13.6 Debug and Test Pin Description

Table 13-1. Debug and Test Signal List

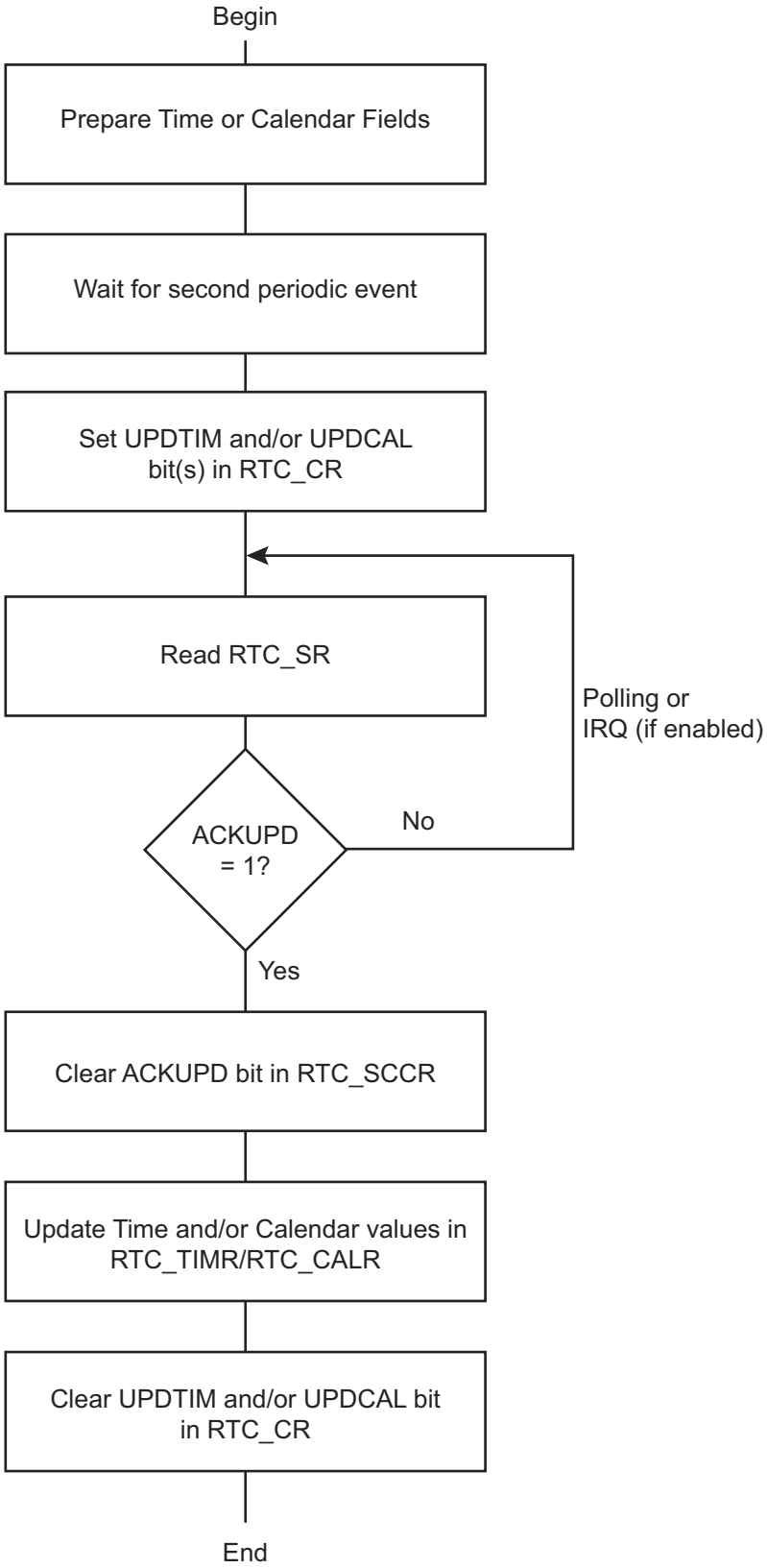
Signal Name	Function	Type	Active Level
Reset/Test			
NRST	Microcontroller Reset	Input/Output	Low
TST	Test Select	Input	–
SWD/JTAG			
TCK/SWCLK	Test Clock/Serial Wire Clock	Input	–
TDI	Test Data In	Input	–
TDO/TRACESWO	Test Data Out/Trace Asynchronous Data Out	Output	–
TMS/SWDIO	Test Mode Select/Serial Wire Input/Output	Input	–
JTAGSEL	JTAG Selection	Input	High

13.7 Functional Description

13.7.1 Test Pin

One dedicated pin, TST, is used to define the device operating mode. When this pin is at low level during power-up, the device is in Normal operating mode. When at high level, the device is in Test mode or FFPI mode. The TST pin integrates a permanent pull-down resistor of about 15 kΩ, so that it can be left unconnected for normal operation. Note that when setting the TST pin to low or high level at power-up, the pin must remain in the same state for the duration of the operation.

Figure 17-3. Gregorian and Persian Modes Update Sequence



24.5 Cortex-M Cache Controller (CMCC) User Interface

Table 24-1. Register Mapping

Offset	Register	Name	Access	Reset
0x00	Cache Controller Type Register	CMCC_TYPE	Read-only	0x000011D7
0x04	Reserved	–	–	–
0x08	Cache Controller Control Register	CMCC_CTRL	Write-only	–
0x0C	Cache Controller Status Register	CMCC_SR	Read-only	0x00000001
0x10–0x1C	Reserved	–	–	–
0x20	Cache Controller Maintenance Register 0	CMCC_MAINT0	Write-only	–
0x24	Cache Controller Maintenance Register 1	CMCC_MAINT1	Write-only	–
0x28	Cache Controller Monitor Configuration Register	CMCC_MCFG	Read/Write	0x00000000
0x2C	Cache Controller Monitor Enable Register	CMCC_MEN	Read/Write	0x00000000
0x30	Cache Controller Monitor Control Register	CMCC_MCTRL	Write-only	–
0x34	Cache Controller Monitor Status Register	CMCC_MSR	Read-only	0x00000000
0x38–0xFC	Reserved	–	–	–

– Wait for the MCKRDY bit to be set in PMC_SR.

If at some stage, parameters CSS or PRES are modified, the MCKRDY bit goes low to indicate that the master clock and the processor clock are not yet ready. The user must wait for MCKRDY bit to be set again before using the master and processor clocks.

Note: IF PLLx clock was selected as the master clock and the user decides to modify it by writing in CKGR_PLLxR, the MCKRDY flag will go low while PLLx is unlocked. Once PLLx is locked again, LOCKx goes high and MCKRDY is set. While PLLx is unlocked, the master clock selection is automatically changed to slow clock for PLLA and main clock for PLLB. For further information, see Section 30.16.2 "Clock Switching Waveforms".

Code Example:

```
write_register(PMC_MCKR,0x00000001)
wait (MCKRDY=1)
write_register(PMC_MCKR,0x00000011)
wait (MCKRDY=1)
```

The master clock is main clock divided by 2.

8. Select the programmable clocks

   Programmable clocks are controlled via registers, PMC_SCER, PMC_SCDR and PMC_SCSR.

   Programmable clocks can be enabled and/or disabled via PMC_SCER and PMC_SCDR. Three programmable clocks can be used. PMC_SCSR indicates which programmable clock is enabled. By default all programmable clocks are disabled.

   PMC_PCKx registers are used to configure programmable clocks.

   The CSS field is used to select the programmable clock divider source. Several clock options are available: main clock, slow clock, master clock, PLLACK, PLLBCK. The slow clock is the default clock source.

   The PRES field is used to control the programmable clock prescaler. It is possible to choose between different values (1, 2, 4, 8, 16, 32, 64). Programmable clock output is prescaler input divided by PRES parameter. By default, the PRES value is cleared which means that PCKx is equal to slow clock.

   Once PMC_PCKx register has been configured, the corresponding programmable clock must be enabled and the user is constrained to wait for the PCKRDYx bit to be set in the PMC_SR. This can be done either by polling PCKRDYx in PMC_SR or by waiting for the interrupt line to be raised if the associated interrupt source (PCKRDYx) has been enabled in PMC_IER. All parameters in PMC_PCKx can be programmed in a single write operation.

   If the CSS and PRES parameters are to be modified, the corresponding programmable clock must be disabled first. The parameters can then be modified. Once this has been done, the user must re-enable the programmable clock and wait for the PCKRDYx bit to be set.

9. Enable the peripheral clocks

   Once all of the previous steps have been completed, the peripheral clocks can be enabled and/or disabled via registers PMC_PCER0, PMC_PCER, PMC_PCDR0 and PMC_PCDR.

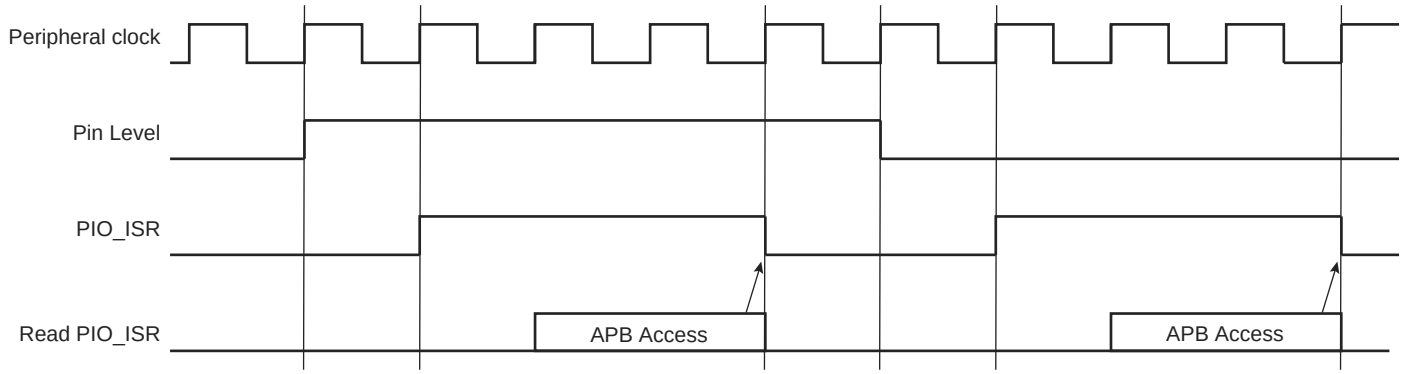
- Falling edge on PIO line 6
- Rising edge on PIO line 7
- Any edge on the other lines

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

32.6.39 PIO Edge Select Register

Name: PIO_ESR

Address: 0x400E0EC0 (PIOA), 0x400E10C0 (PIOB), 0x4800C0C0 (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

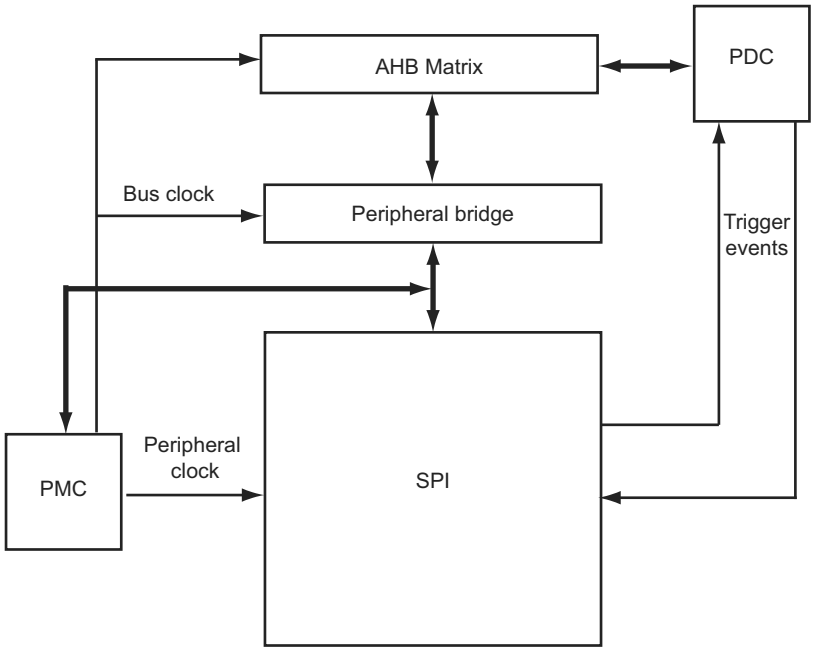
- **P0–P31: Edge Interrupt Selection**

0: No effect.

1: The interrupt source is an edge-detection event.

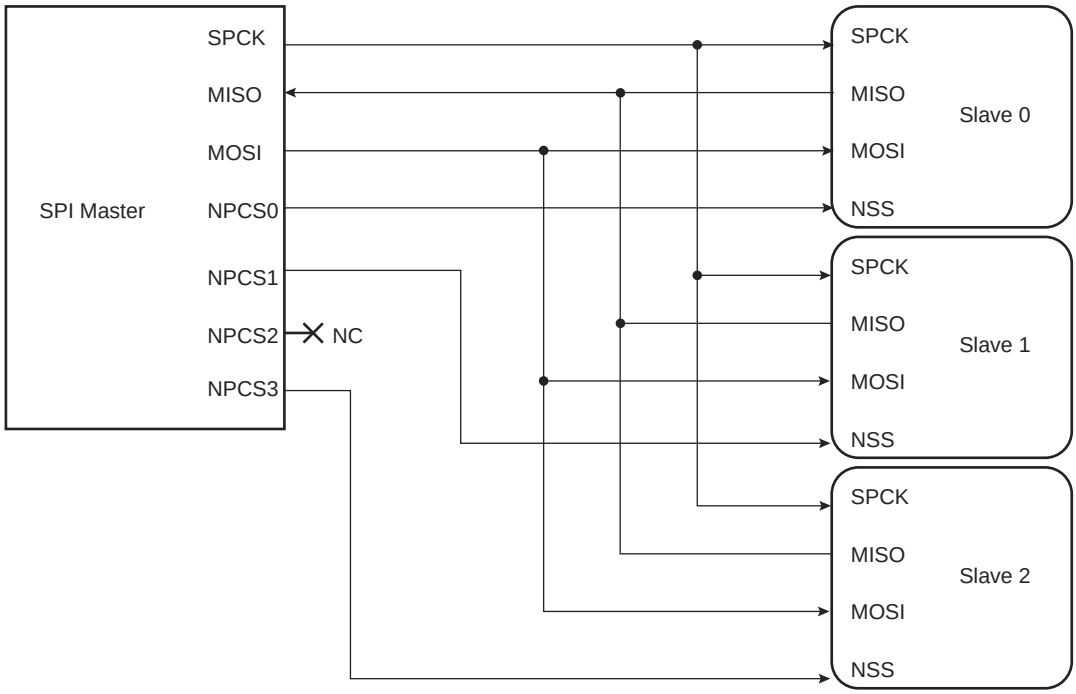
33.3 Block Diagram

Figure 33-1. Block Diagram



33.4 Application Block Diagram

Figure 33-2. Application Block Diagram: Single Master/Multiple Slave Implementation



- **DLYBCT: Delay Between Consecutive Transfers**

This field defines the delay between two consecutive transfers with the same peripheral without removing the chip select. The delay is always inserted after each transfer and before removing the chip select if needed.

When DLYBCT = 0, no delay between consecutive transfers is inserted and the clock keeps its duty cycle over the character transfers.

Otherwise, the following equation determines the delay:

DLYBCT = Delay Between Consecutive Transfers × $f_{peripheral\ clock}$ / 32

37.7 Timer Counter (TC) User Interface

Table 37-6. Register Mapping

Offset[1]	Register	Name	Access	Reset
0x00 + channel * 0x40 + 0x00	Channel Control Register	TC_CCR	Write-only	–
0x00 + channel * 0x40 + 0x04	Channel Mode Register	TC_CMR	Read/Write	0
0x00 + channel * 0x40 + 0x08	Stepper Motor Mode Register	TC_SMMR	Read/Write	0
0x00 + channel * 0x40 + 0x0C	Reserved	–	–	–
0x00 + channel * 0x40 + 0x10	Counter Value	TC_CV	Read-only	0
0x00 + channel * 0x40 + 0x14	Register A	TC_RA	Read/Write[2]	0
0x00 + channel * 0x40 + 0x18	Register B	TC_RB	Read/Write[2]	0
0x00 + channel * 0x40 + 0x1C	Register C	TC_RC	Read/Write	0
0x00 + channel * 0x40 + 0x20	Status Register	TC_SR	Read-only	0
0x00 + channel * 0x40 + 0x24	Interrupt Enable Register	TC_IER	Write-only	–
0x00 + channel * 0x40 + 0x28	Interrupt Disable Register	TC_IDR	Write-only	–
0x00 + channel * 0x40 + 0x2C	Interrupt Mask Register	TC_IMR	Read-only	0
0xC0	Block Control Register	TC_BCR	Write-only	–
0xC4	Block Mode Register	TC_BMR	Read/Write	0
0xC8	QDEC Interrupt Enable Register	TC_QIER	Write-only	–
0xCC	QDEC Interrupt Disable Register	TC_QIDR	Write-only	–
0xD0	QDEC Interrupt Mask Register	TC_QIMR	Read-only	0
0xD4	QDEC Interrupt Status Register	TC_QISR	Read-only	0
0xD8	Reserved	–	–	–
0xE4	Write Protection Mode Register	TC_WPMR	Read/Write	0
0xE8–0xFC	Reserved	–	–	–

Notes: 1. Channel index ranges from 0 to 2.
2. Read-only if TC_CMRx.WAVE = 0

38.7.4 PWM Status Register

Name: PWM_SR

Address: 0x4800800C

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
–	–	–	–	CHID3	CHID2	CHID1	CHID0

- **CHIDx: Channel ID**

0 = PWM output for channel x is disabled.

1 = PWM output for channel x is enabled.

40.7.1 ADC Control Register

Name: ADC_CR

Address: 0x40038000

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
–	–	–	–	–	–	START	SWRST

- **SWRST: Software Reset**

0: No effect.

1: Resets the ADC simulating a hardware reset.

- **START: Start Conversion**

0: No effect.

1: Begins analog-to-digital conversion.

Table 46-4. Recommended Operating Conditions on Power Supply Inputs at Powerup

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
RR_{VDDBU}	Rise Rate on VDDBU	(1)	660	–	300k	V/s
V_{ST_VDDBU}	VDDBU voltage at powerup	(1)	3.0	–	–	V
V_{ST_VDDIO}	VDDIO and VDDIN voltage at powerup	–	3.0	–	–	V
V_{VDDIO_VDDBU}	Voltage on VDDIO and VDDIN while VDDBU < 1.6V	(1)	–	–	V_{VDDBU}	V
RR_{VDDIO}	Rise Rate on VDDIO and VDDIN	–	330	–	300k	V/s

Note: 1. Applies whenever VDDBU is restarted from below 1.35V.

46.2.3 Recommended Operating Conditions on Input Pins

Table 46-5. Recommended Operating Conditions on Input Pins

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$AD[x]_{IN}$	Input voltage range on 10-bit ADC analog inputs	On AD[0..x]	0	–	Min (VDDIN, VDDIO)	V
$EMAFE_{IN}$	Input voltage range on EMAFE input pins	On $I_{P\{0,1,2,3\}}$, $I_{N\{0,1,2,3\}}$ and $V_{P\{1,2,3\}}$	-0.25	–	0.25	V
V_{GPIO_IN}	Input voltage range on GPIOs referenced to VDDIO	On any pin configured as a digital input	0	–	VDDIO	V
V_{VDDBU_IN}	Input voltage range on inputs referenced to VDDBU	On FWUP, TMP0 and XIN32 inputs	0	–	VDDBU	V

46.2.4 Recommended Thermal Operating Conditions

Table 46-6. Recommended Thermal Operating Conditions

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
T_A	Ambient temperature range	–		-40	–	+85	°C
T_J	Junction temperature range	–		-40	–	100	
R_{JA}	Junction-to-ambient thermal resistance	LQFP100 (SAM4CM16/8/4)		–	43	–	°C/W
		LQFP100 (SAM4CM32)		–	41	–	
P_D	Power dissipation	LQFP100 (SAM4CM16/8/4)	T_A = 70°C	–	–	700	mW
			T_A = 85°C	–	–	350	
		LQFP100 (SAM4CM32)	T_A = 70°C	–	–	730	
			T_A = 85°C	–	–	365	

46.3 Electrical Parameters Usage

The tables that follow further on in Section 46.4 “I/O Characteristics”, Section 46.5 “Embedded Analog Peripherals Characteristics”, Section 46.6 “Embedded Flash Characteristics”, and Section 46.7 “Power Supply Current Consumption” define the limiting values for several electrical parameters. Unless otherwise noted, these values are valid over the ambient temperature range T_A= [-40°C + 85°C]. Note that these limits may be affected by the board on which the MCU is mounted. Particularly, noisy supply and ground conditions must be avoided and care must be taken to provide:

- a PCB with a low impedance ground plane (unbroken ground planes are strongly recommended)

46.4.4 SMC Timings

Timings are given in the following domains:

- 1.8V domain: VDDIO from 1.65V to 1.95V, maximum external capacitor = 10 pF
- 3.3V domain: VDDIO from 2.85V to 3.6V, maximum external capacitor = 10 pF

Timings are given assuming a capacitance load on data, control and address pads.

In the tables that follow, t_{CPMCK} is the MCK period.

46.4.4.1 Read Timings

Table 46-11. SMC Read Signals - NRD Controlled (READ_MODE = 1)

Symbol	Parameter	Min		Max		Unit
	VDDIO Supply	1.8V(1)	3.3V(2)	–	–	
NO HOLD SETTINGS (nrd hold = 0)						
SMC_1	Data setup before NRD high	18	18	–	–	ns
SMC_2	Data hold after NRD high	-6.7	-6.7	–	–	ns
HOLD SETTINGS (nrd hold ≠ 0)						
SMC_3	Data setup before NRD high	11.7	11.7	–	–	ns
SMC_4	Data hold after NRD high	-6.5	-6.5	–	–	ns
HOLD or NO HOLD SETTINGS (nrd hold ≠ 0, nrd hold = 0)						
SMC_5	NBS0/A0, NBS1, A1 - A23 Valid before NRD high	(nrd setup + nrd pulse) * t_{CPMCK} - 5.2	(nrd setup + nrd pulse) * t_{CPMCK} - 5.2	–	–	ns
SMC_6	NCS low before NRD high	(nrd setup + nrd pulse - ncs rd setup) * t_{CPMCK} - 1.1	(nrd setup + nrd pulse - ncs rd setup) * t_{CPMCK} - 1.1	–	–	ns
SMC_7	NRD pulse width	nrd pulse * t_{CPMCK} - 3.2	nrd pulse * t_{CPMCK} - 3.2	–	–	ns

Notes: 1. 1.8V domain: VDDIO from 1.65V to 1.95V, maximum external capacitor = 10 pF.
2. 3.3V domain: VDDIO from 3.0V to 3.6V, maximum external capacitor = 10 pF.

Table 46-12. SMC Read Signals - NCS Controlled (READ_MODE= 0)

Symbol	Parameter	Min		Max		Unit
	VDDIO supply	1.8V(1)	3.3V(2)	–	–	
NO HOLD SETTINGS (ncs rd hold = 0)						
SMC_8	Data setup before NCS high	31.3	31.3	–	–	ns
SMC_9	Data hold after NCS high	-6.9	-6.9	–	–	ns
HOLD SETTINGS (ncs rd hold ≠ 0)						
SMC_{10}	Data setup before NCS high	23	23	–	–	ns
SMC_{11}	Data hold after NCS high	-6.6	-6.6	–	–	ns
HOLD or NO HOLD SETTINGS (ncs rd hold ≠ 0, ncs rd hold = 0)						
SMC_{12}	NBS0/A0, NBS1, A1–A23 valid before NCS high	(ncs rd setup + ncs rd pulse)* t_{CPMCK} - 4.9	(ncs rd setup + ncs rd pulse)* t_{CPMCK} - 4.9	–	–	ns
SMC_{13}	NRD low before NCS high	(ncs rd setup + ncs rd pulse - nrd setup)* t_{CPMCK} - 1.5	(ncs rd setup + ncs rd pulse - nrd setup)* t_{CPMCK} - 1.5	–	–	ns
SMC_{14}	NCS pulse width	ncs rd pulse length * t_{CPMCK} - 5.4	ncs rd pulse length * t_{CPMCK} - 5.4	–	–	ns

Notes: 1. 1.8V domain: VDDIO from 1.65V to 1.95V, maximum external capacitor = 10 pF.
2. 3.3V domain: VDDIO from 3.0V to 3.6V, maximum external capacitor = 10 pF.

47.2 Soldering Profile

Table 47-4 gives the recommended soldering profile from J-STD-020C.

Table 47-4. Soldering Profile

Profile Feature	Green Package
Average Ramp-up Rate (217°C to Peak)	3°C/sec. max.
Preheat Temperature 175°C ± 25°C	180 sec. max.
Temperature Maintained Above 217°C	60 sec. to 150 sec.
Time within 5°C of Actual Peak Temperature	20 sec. to 40 sec.
Peak Temperature Range	260°C
Ramp-down Rate	6°C/sec. max.
Time 25°C to Peak Temperature	8 min. max.

Note: The package is certified to be backward compatible with Pb/Sn soldering profile.

A maximum of three reflow passes is allowed per component.

47.3 Packaging Resources

This section provides land pattern definition.

Refer to the following IPC standards:

- IPC-7351A and IPC-782 (*Generic Requirements for Surface Mount Design and Land Pattern Standards*) http://landpatterns.ipc.org/default.asp
- Atmel Green and RoHS Policy and Package Material Declaration Data Sheet http://www.atmel.com/about/quality/package.aspx