Microchip Technology - ATSAM4CMP8CC-AUR Datasheet

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	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/atsam4cmp8cc-aur

Table 4-2. SAM4CMS32C/16C/8C/4C 100-lead LQFP Pinout

1	PB6	26	TDI/PB0	51	VDDIO	76	ADVREF
2	PB7	27	TCK/SWCLK/PB3	52	GND	77	GND
3	PB18	28	TMS/SWDIO/PB2	53	PA31	78	PB13/AD3
4	GND	29	ERASE/PC9	54	GND	79	PA5/AD2/PGMRDY
5	PB19	30	TDO/TRACESWO /PB1	55	VDDPLL	80	PA4/AD1/PGMNCMD
6	PB8	31	PC1	56	PA28	81	PA12/AD0/PGMD0
7	IN1	32	PC6	57	PA27/PGMD15	82	VDDIN
8	IP1	33	VDDIO	58	PA6/PGMNOE	83	VDDOUT
9	IN0	34	VDDBU	59	VDDCORE	84	PB21
10	IP0	35	FWUP	60	PA3	85	VP2
11	GND	36	JTAGSEL	61	PA21/PGMD9	86	VDDCORE
12	VDDCORE	37	SDHN	62	PA22/PGMD10	87	VP1
13	PB9	38	TST	63	VDDIO	88	PA0/PGMEN0
14	PB10	39	WKUP0/TMP0	64	VDDIN_AFE	89	VN
15	PB11	40	XIN32	65	–	90	VREF_AFE
16	PB12	41	XOUT32	66	PA23/PGMD11	91	GNDREF
17	PB14	42	GND	67	PA9/PGMM1	92	VDDLCD
18	PB15	43	PB4	68	PA10/PGMM2	93	GNDA
19	PA26/PGMD14	44	VDDCORE	69	PA11/PGMM3	94	VDDA
20	PA25/PGMD13	45	PB5	70	PA13/PGMD1	95	PB16/TMP1
21	PA24/PGMD12	46	PC7	71	PA14/PGMD2	96	PA1/PGMEN1
22	PA20/PGMD8	47	PC0	72	PA15/PGMD3	97	PB17
23	PA19/PGMD7	48	NRST	73	PA16/PGMD4	98	PA7/PGMNVALID
24	PA18/PGMD6	49	VDDIO	74	PA17/PGMD5	99	VDDIO
25	PA8/PGMM0	50	PA30	75	VDDIO	100	PA2

5. Power Supply and Power Control

5.1 Power Supplies

The SAM4CM has several types of power supply pins. In most cases, a single supply scheme for all power supplies (except VDDBU) is possible. Figure 5-1 shows power domains according to the different power supply pins.

Figure 5-1. Power Domains

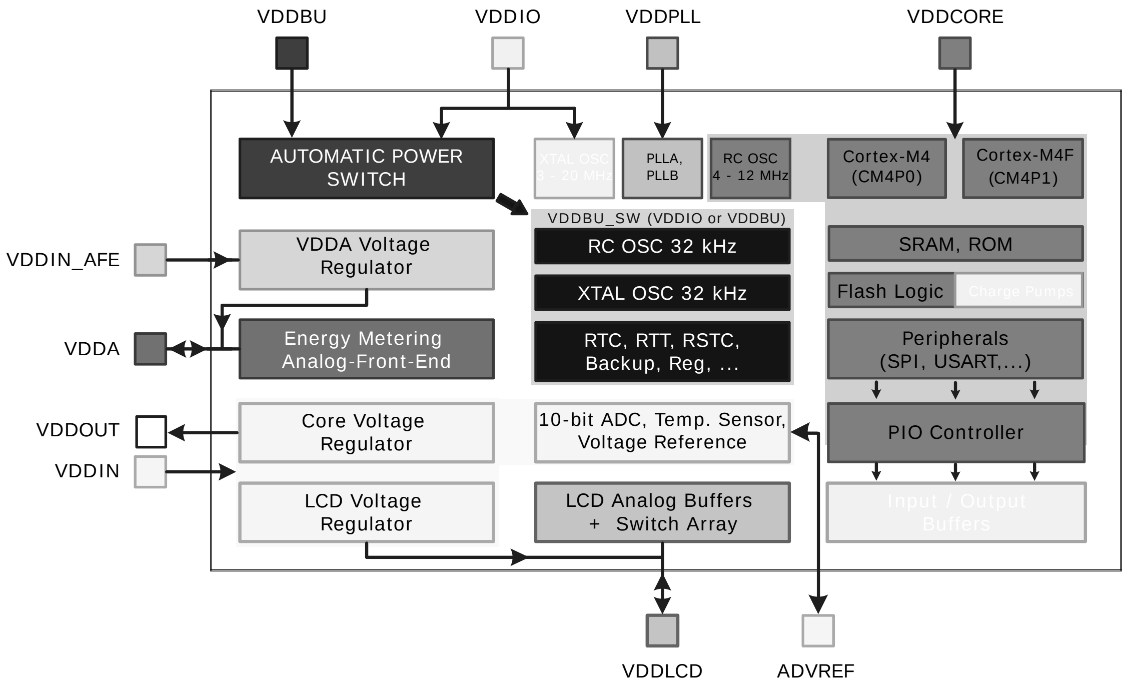


Table 5-1. Power Supply Voltage Ranges(1)

Power Supply	Range	Comments
VDDIO	1.6V to 3.6V	Flash memory charge pumps supply for erase and program operations, and read operation. Input/Output buffers supply. EMAFE digital functions supply. Restrictions on range may apply. Refer to Section 46. “Electrical Characteristics”.
VDDBU(2)	1.6V to 3.6V	Backup area power supply. VDDBU is automatically disconnected when VDDIO is present (> 1.9V).
VDDIN	1.6V to 3.6V	Core voltage regulator supply, LCD voltage regulator supply, ADC and programmable voltage reference supply. Restrictions on range may apply. Refer to Section 46. “Electrical Characteristics”.
VDDLCD	2.5V to 3.6V	LCD voltage regulator output. External LCD power supply input (LCD regulator not used). VDDIO/VDDIN must be supplied when the LCD Controller is used.

12.4.2.1 Memory Regions, Types and Attributes

The memory map and the programming of the MPU split the memory map into regions. Each region has a defined memory type, and some regions have additional memory attributes. The memory type and attributes determine the behavior of accesses to the region.

Memory Types

- **Normal**
  The processor can re-order transactions for efficiency, or perform speculative reads.
- **Device**
  The processor preserves transaction order relative to other transactions to Device or Strongly-ordered memory.
- **Strongly-ordered**
  The processor preserves transaction order relative to all other transactions.

The different ordering requirements for Device and Strongly-ordered memory mean that the memory system can buffer a write to Device memory, but must not buffer a write to Strongly-ordered memory.

Additional Memory Attributes

- **Shareable**
  For a shareable memory region, the memory system provides data synchronization between bus masters in a system with multiple bus masters, for example, a processor with a DMA controller.
  Strongly-ordered memory is always shareable.
  If multiple bus masters can access a non-shareable memory region, the software must ensure data coherency between the bus masters.
- **Execute Never (XN)**
  Means the processor prevents instruction accesses. A fault exception is generated only on execution of an instruction executed from an XN region.

12.4.2.2 Memory System Ordering of Memory Accesses

For most memory accesses caused by explicit memory access instructions, the memory system does not guarantee that the order in which the accesses complete matches the program order of the instructions, providing this does not affect the behavior of the instruction sequence. Normally, if correct program execution depends on two memory accesses completing in program order, the software must insert a memory barrier instruction between the memory access instructions, see “Software Ordering of Memory Accesses”.

However, the memory system does guarantee some ordering of accesses to Device and Strongly-ordered memory. For two memory access instructions A1 and A2, if A1 occurs before A2 in program order, the ordering of the memory accesses is described below.

12.6.5.5 CMP and CMN

Compare and Compare Negative.

Syntax

```
CMP{cond} Rn, Operand2
CMN{cond} Rn, Operand2
```

where:

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

Rn is the register holding the first operand.

Operand2 is a flexible second operand. See “Flexible Second Operand” for details of the options.

Operation

These instructions compare the value in a register with *Operand2*. They update the condition flags on the result, but do not write the result to a register.

The CMP instruction subtracts the value of *Operand2* from the value in *Rn*. This is the same as a SUBS instruction, except that the result is discarded.

The CMN instruction adds the value of *Operand2* to the value in *Rn*. This is the same as an ADDS instruction, except that the result is discarded.

Restrictions

In these instructions:

- Do not use PC
- *Operand2* must not be SP.

Condition Flags

These instructions update the N, Z, C and V flags according to the result.

Examples

```
CMP      R2, R9
CMN      R0, #6400
CMPGT    SP, R7, LSL #2
```

12.8.3.2 Interrupt Clear-enable Registers

Name: NVIC_ICERx [x=0..7]

Access: Read/Write

Reset: 0x000000000

31	30	29	28	27	26	25	24
CLRENA							

23	22	21	20	19	18	17	16
CLRENA							

15	14	13	12	11	10	9	8
CLRENA							

7	6	5	4	3	2	1	0
CLRENA							

These registers disable interrupts, and show which interrupts are enabled.

- **CLRENA: Interrupt Clear-enable**

Write:

0: No effect.

1: Disables the interrupt.

Read:

0: Interrupt disabled.

1: Interrupt enabled.

12.9.1.9 System Handler Priority Register 1

Name: SCB_SHPR1

Access: Read/Write

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

23	22	21	20	19	18	17	16
PRI_6							

15	14	13	12	11	10	9	8
PRI_5							

7	6	5	4	3	2	1	0
PRI_4							

- **PRI_6: Priority**

Priority of system handler 6, UsageFault.

- **PRI_5: Priority**

Priority of system handler 5, BusFault.

- **PRI_4: Priority**

Priority of system handler 4, MemManage.

12.11.1.5 Subregions

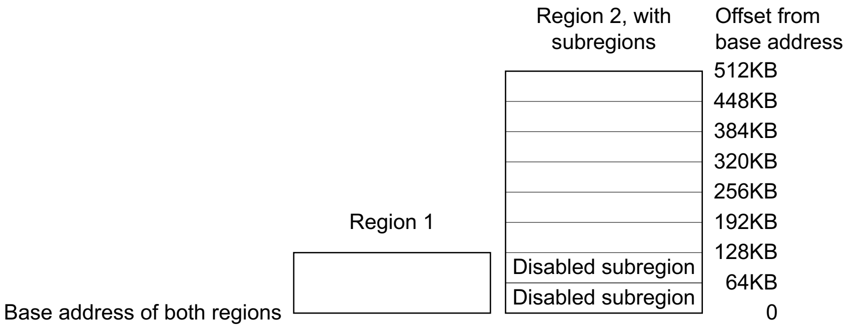
Regions of 256 bytes or more are divided into eight equal-sized subregions. Set the corresponding bit in the SRD field of the MPU_RASR field to disable a subregion. See “MPU Region Attribute and Size Register”. The least significant bit of SRD controls the first subregion, and the most significant bit controls the last subregion. Disabling a subregion means another region overlapping the disabled range matches instead. If no other enabled region overlaps the disabled subregion, the MPU issues a fault.

Regions of 32, 64, and 128 bytes do not support subregions. With regions of these sizes, the SRD field must be set to 0x00, otherwise the MPU behavior is unpredictable.

12.11.1.6 Example of SRD Use

Two regions with the same base address overlap. Region 1 is 128 KB, and region 2 is 512 KB. To ensure the attributes from region 1 apply to the first 128 KB region, set the SRD field for region 2 to b00000011 to disable the first two subregions, as in Figure 12-13 below:

Figure 12-13. SRD Use



12.11.1.7 MPU Design Hints And Tips

To avoid unexpected behavior, disable the interrupts before updating the attributes of a region that the interrupt handlers might access.

Ensure the software uses aligned accesses of the correct size to access MPU registers:

- Except for the MPU_RASR, it must use aligned word accesses
- For the MPU_RASR, it can use byte or aligned halfword or word accesses.

The processor does not support unaligned accesses to MPU registers.

When setting up the MPU, and if the MPU has previously been programmed, disable unused regions to prevent any previous region settings from affecting the new MPU setup.

MPU Configuration for a Microcontroller

Usually, a microcontroller system has only a single processor and no caches. In such a system, program the MPU as follows:

Table 12-40. Memory Region Attributes for a Microcontroller

Memory Region	TEX	C	B	S	Memory Type and Attributes
Flash memory	b000	1	0	0	Normal memory, non-shareable, write-through
Internal SRAM	b000	1	0	1	Normal memory, shareable, write-through
External SRAM	b000	1	1	1	Normal memory, shareable, write-back, write-allocate
Peripherals	b000	0	1	1	Device memory, shareable

- **HFRDY: Hard Fault Ready**

0: The priority did not permit to set the HardFault handler to the pending state when the floating-point stack frame was allocated.

1: The priority permitted to set the HardFault handler to the pending state when the floating-point stack frame was allocated.

- **THREAD: Thread Mode**

0: The mode was not the Thread Mode when the floating-point stack frame was allocated.

1: The mode was the Thread Mode when the floating-point stack frame was allocated.

- **USER: User Privilege Level**

0: The privilege level was not User when the floating-point stack frame was allocated.

1: The privilege level was User when the floating-point stack frame was allocated.

- **LSPACT: Lazy State Preservation Active**

0: The lazy state preservation is not active.

1: The lazy state preservation is active. The floating-point stack frame has been allocated but saving the state to it has been deferred.

reuses some addresses of the memory plane for code, but the unique identifier area is physically different from the memory plane for code.

3. To stop reading the unique identifier area, execute the ‘Stop Read Unique Identifier’ command by writing EEFC_FCR.FCMD with the SPUI command. Field EEFC_FCR.FARG is meaningless.
4. When the SPUI command has been executed, the bit EEFC_FSR.FRDY rises. If an interrupt was enabled by setting the bit EEFC_FMR.FRDY, the interrupt line of the interrupt controller is activated.

Note that during the sequence, the software cannot be fetched from the Flash or from the second plane in case of dual plane.

22.4.3.9 User Signature Area

Each product contains a user signature area of 512 bytes. It can be used for storage. Read, write and erase of this area is allowed.

See Figure 22-1 "Flash Memory Areas".

The sequence to read the user signature area is the following:

1. Execute the ‘Start Read User Signature’ command by writing EEFC_FCR.FCMD with the STUS command. Field EEFC_FCR.FARG is meaningless.
2. Wait until the bit EEFC_FSR.FRDY falls to read the user signature area. The user signature area is located in the first 512 bytes of the Flash memory mapping. The ‘Start Read User Signature’ command reuses some addresses of the memory plane but the user signature area is physically different from the memory plane
3. To stop reading the user signature area, execute the ‘Stop Read User Signature’ command by writing EEFC_FCR.FCMD with the SPUS command. Field EEFC_FCR.FARG is meaningless.
4. When the SPUI command has been executed, the bit EEFC_FSR.FRDY rises. If an interrupt was enabled by setting the bit EEFC_FMR.FRDY, the interrupt line of the interrupt controller is activated.

Note that during the sequence, the software cannot be fetched from the Flash or from the second plane in case of dual plane.

One error can be detected in EEFC_FSR after this sequence:

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

The sequence to write the user signature area is the following:

1. Write the full page, at any page address, within the internal memory area address space.
2. Execute the ‘Write User Signature’ command by writing EEFC_FCR.FCMD with the WUS command. Field EEFC_FCR.FARG is meaningless.
3. When programming is completed, the bit EEFC_FSR.FRDY rises. If an interrupt has been enabled by setting the bit EEFC_FMR.FRDY, the corresponding interrupt line of the interrupt controller is activated.

Two errors can be detected in EEFC_FSR after this sequence:

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

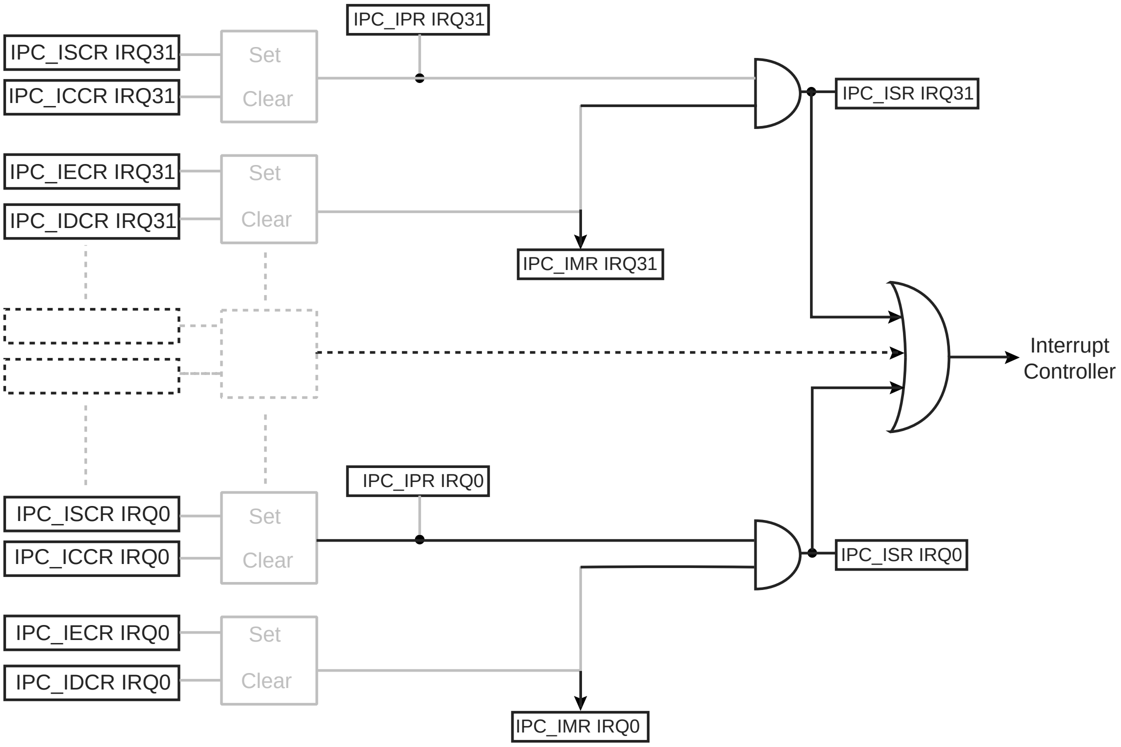
The sequence to erase the user signature area is the following:

1. Execute the ‘Erase User Signature’ command by writing EEFC_FCR.FCMD with the EUS command. Field EEFC_FCR.FARG is meaningless.
2. When programming is completed, the bit EEFC_FSR.FRDY rises. If an interrupt has been enabled by setting the bit EEFC_FMR.FRDY, the corresponding interrupt line of the interrupt controller is activated.

Two errors can be detected in EEFC_FSR after this sequence:

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

Figure 25-3. Interrupt Input Stage



32.6.17 PIO Interrupt Status Register

Name: PIO_ISR

Address: 0x400E0E4C (PIOA), 0x400E104C (PIOB), 0x4800C04C (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

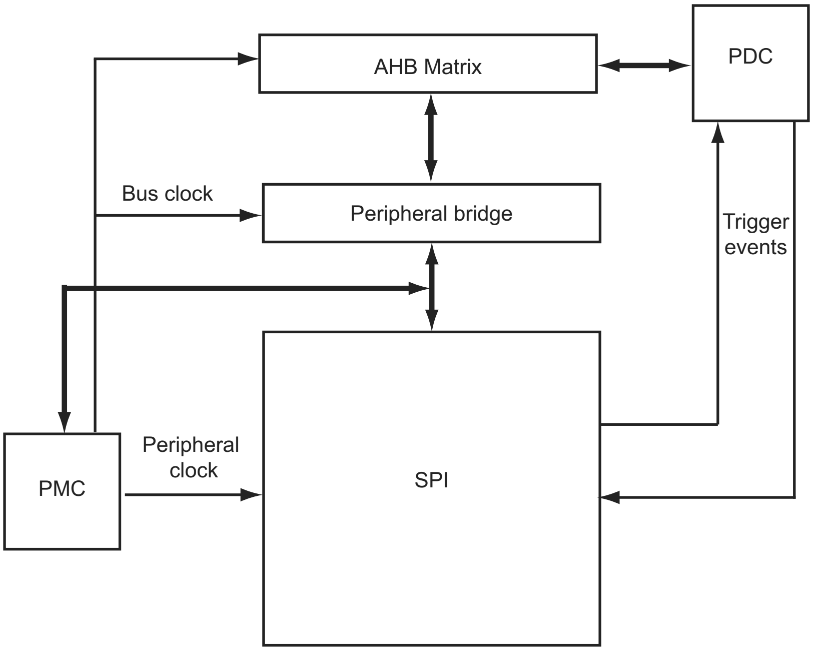
- **P0–P31: Input Change Interrupt Status**

0: No input change has been detected on the I/O line since PIO_ISR was last read or since reset.

1: At least one input change has been detected on the I/O line since PIO_ISR was last read or since reset.

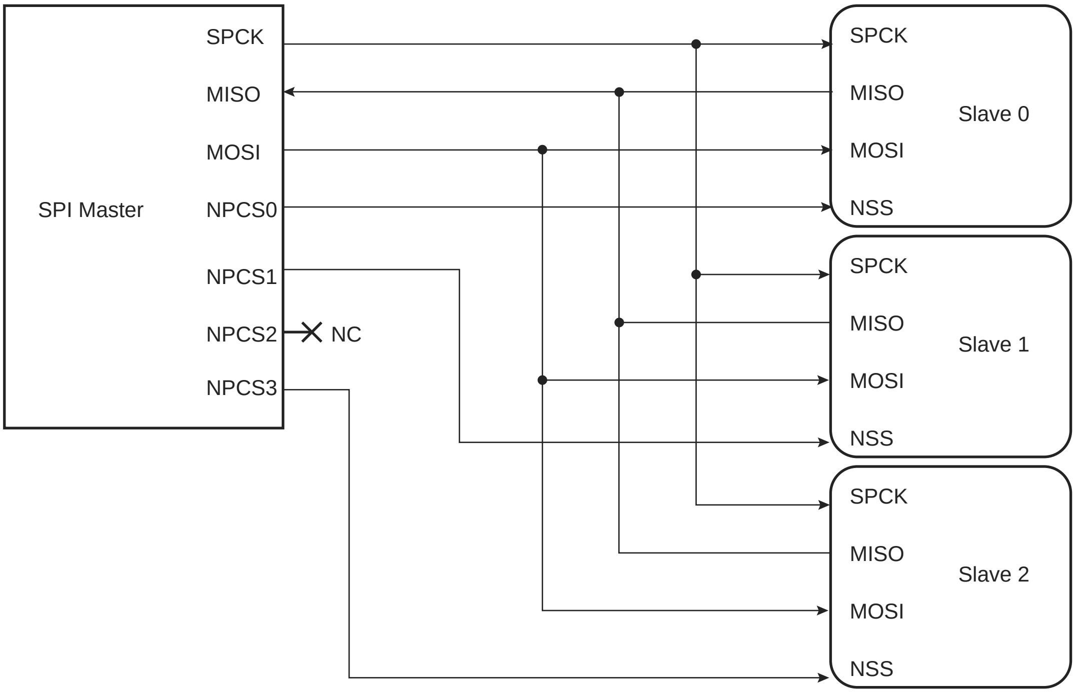
33.3 Block Diagram

Figure 33-1. Block Diagram



33.4 Application Block Diagram

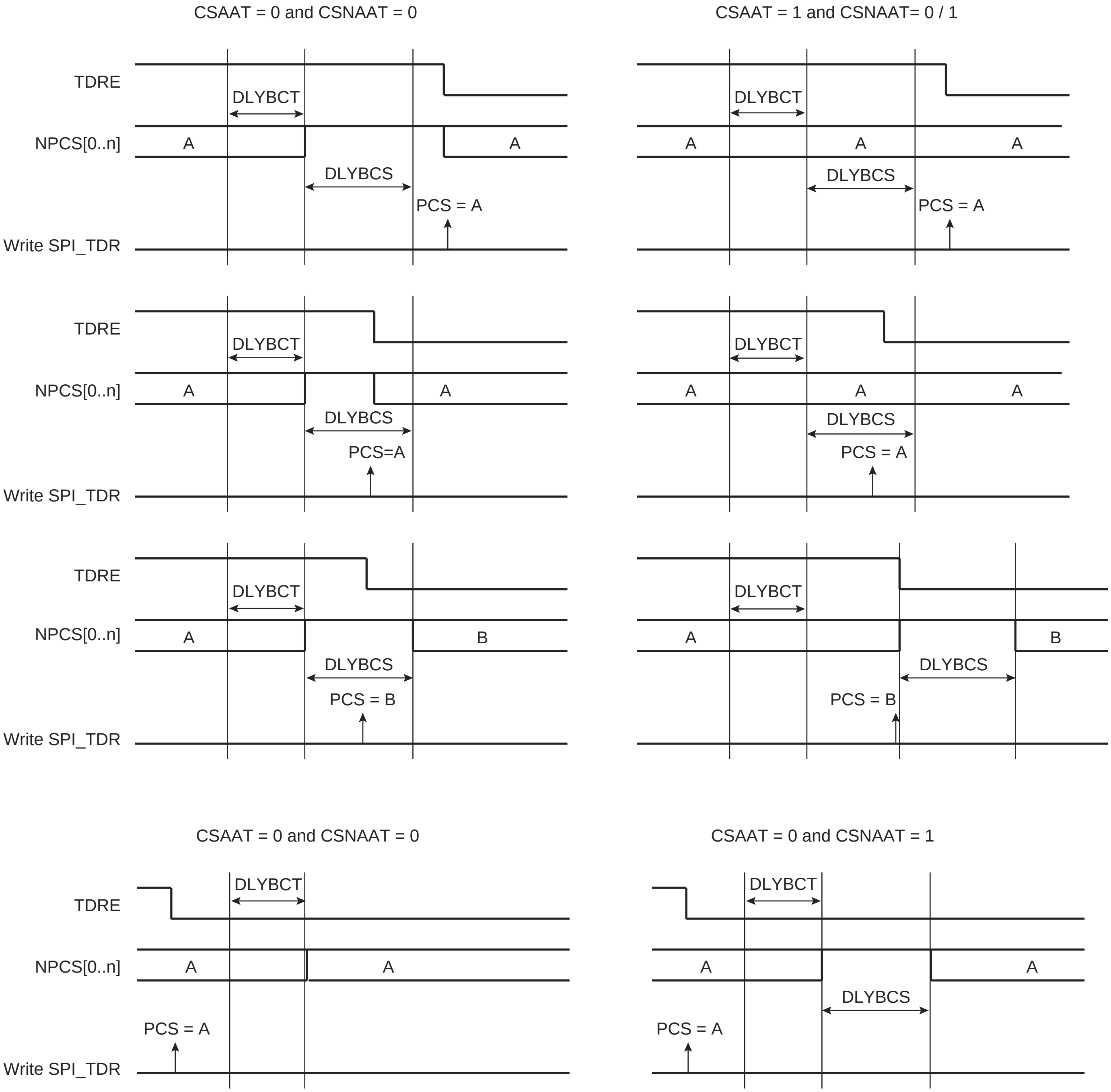
Figure 33-2. Application Block Diagram: Single Master/Multiple Slave Implementation



interfacing with such devices, the SPI_CSR can be programmed with the Chip Select Not Active After Transfer (CSNAAT) bit to 1. This allows the chip select lines to be de-asserted systematically during a time “DLYBCS” (the value of the CSNAAT bit is processed only if the CSAAT bit is configured to 0 for the same chip select).

Figure 33-12 shows different peripheral deselection cases and the effect of the CSAAT and CSNAAT bits.

Figure 33-12. Peripheral Deselection



General Call

The general call is performed in order to change the address of the slave.

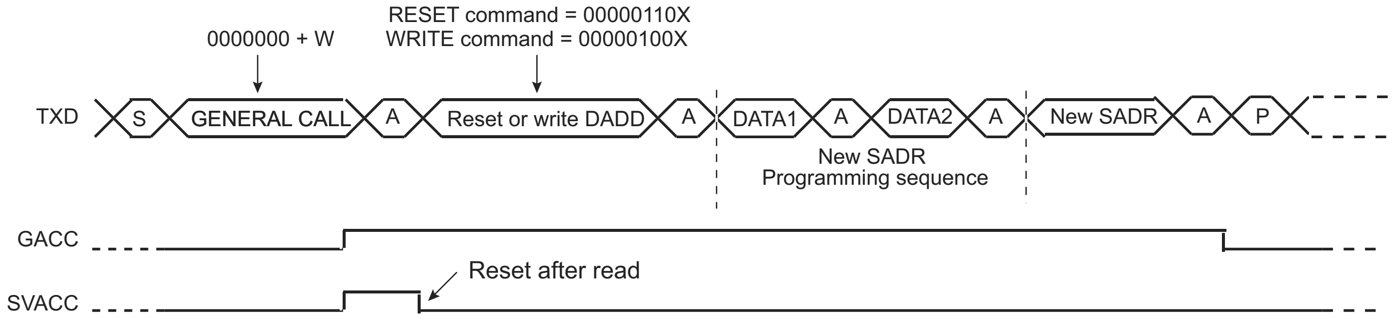
If a GENERAL CALL is detected, GACC is set.

After the detection of GENERAL CALL, it is up to the programmer to decode the commands which come afterwards.

In case of a WRITE command, the programmer has to decode the programming sequence and program a new SADR if the programming sequence matches.

Figure 34-27 describes the GENERAL CALL access.

Figure 34-27. Master Performs a General Call



Note: This method allows the user to create a personal programming sequence by choosing the programming bytes and the number of them. The programming sequence has to be provided to the master.

- **CPOL: SPI Clock Polarity**

Applicable if USART operates in SPI mode (slave or master, USART_MODE = 0xE or 0xF):

0: The inactive state value of SPCK is logic level zero.

1: The inactive state value of SPCK is logic level one.

CPOL is used to determine the inactive state value of the serial clock (SPCK). It is used with CPHA to produce the required clock/data relationship between master and slave devices.

- **CLKO: Clock Output Select**

0: The USART does not drive the SCK pin.

1: The USART drives the SCK pin if USCLKS does not select the external clock SCK.

- **WRDBT: Wait Read Data Before Transfer**

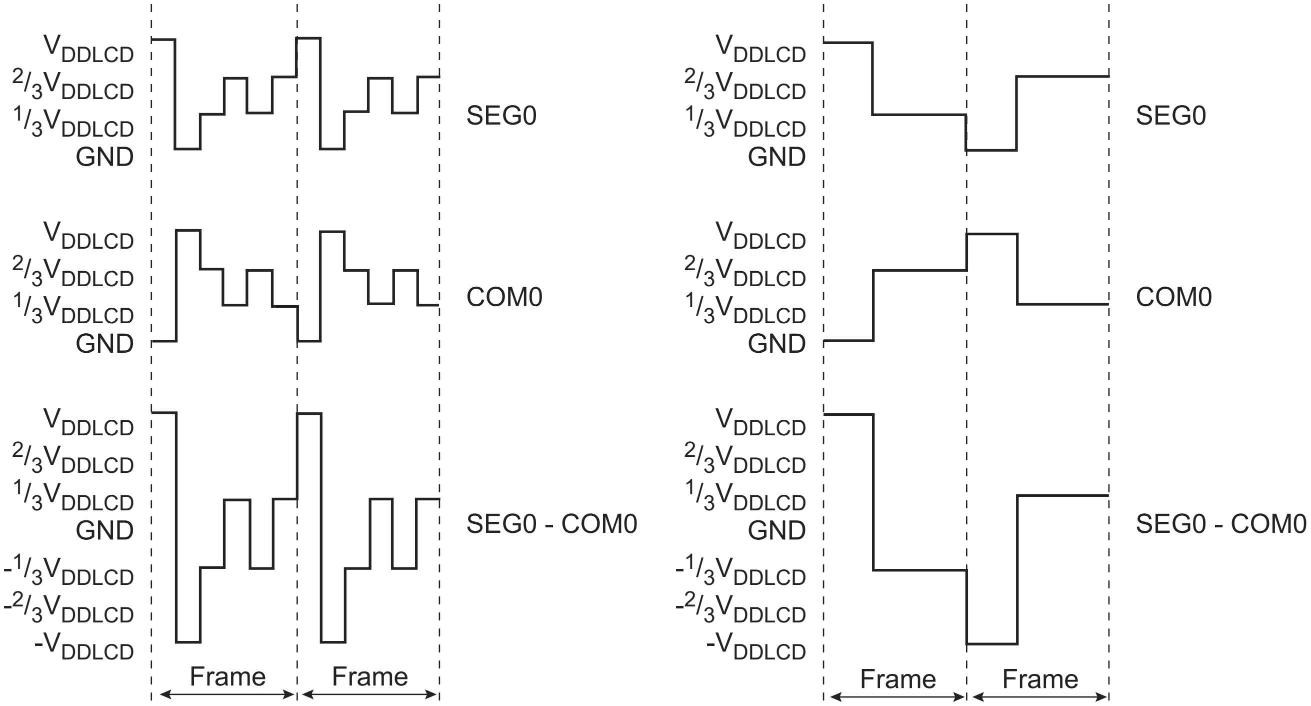
0: The character transmission starts as soon as a character is written into US_THR (assuming TXRDY was set).

1: The character transmission starts when a character is written and only if RXRDY flag is cleared (Receive Holding Register has been read).

39.6.2.5 Low Power Waveform

To reduce toggle activity and hence power consumption, a low power waveform can be selected by writing LPMODE to one. The default and low power waveform is shown in Figure 39-7 for 1/3 duty and 1/3 bias. For other selections of duty and bias, the effect is similar.

Figure 39-7. Default and Low Power Waveform



Note: Refer to the LCD specification to verify that low power waveforms are supported.

39.6.2.6 Frame Rate

The Frame Rate register (SLCDC_FRR) enables the generation of the frequency used by the SLCDC. It is done by a prescaler (division by 8, 16, 32, 64, 128, 256, 512 and 1024) followed by a finer divider (division by 1, 2, 3, 4, 5, 6, 7 or 8).

To calculate the needed frame frequency, the equation below must be used:

$$f_{frame} = \frac{fSLCK}{(PRESC \cdot DIV \cdot NCOM)}$$

where:

f_{SLCK} = slow clock frequency

f_{frame} = frame frequency

PRESC = prescaler value (8, 16, 32, 64, 128, 256, 512 or 1024)

DIV = divider value (1, 2, 3, 4, 5, 6, 7, or 8)

NCOM = depends of number of commons and is defined in Table 39-5.

NCOM is automatically provided by the SLCDC.

For example, if COMSEL is programmed to 0 (1 common terminal on display device), the SLCDC introduces a divider by 16 so that NCOM = 16. If COMSEL is programmed to 3 (3 common terminals on display device), the

39.8.11 SLCDC Segment Map Register 1

Name: SLCDC_SMR1

Address: 0x4003C034

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	LCD49	LCD48
15	14	13	12	11	10	9	8
LCD47	LCD46	LCD45	LCD44	LCD43	LCD42	LCD41	LCD40
7	6	5	4	3	2	1	0
LCD39	LCD38	LCD37	LCD36	LCD35	LCD34	LCD33	LCD32

- **LCDx: LCD Segment Mapped on SEGx I/O Pin**

(For safety reasons, can be configured when SLCDC is disabled)

0: The corresponding I/O pin is driven either by SLCDC or digital function, depending on the SEGSEL field configuration in the SLCDC_MR.

1: An LCD segment is driven on the corresponding I/O pin.

44. Classical Public Key Cryptography Controller (CPKCC)

44.1 Description

The Classical Public Key Cryptography Controller (CPKCC) is an Atmel macrocell that processes public key cryptography algorithm calculus in both GF(p) and GF(2^n) fields. The ROMed CPKCL, the Classical Public Key Cryptography Library, is the library built on the top of the CPKCC.

The Classical Public Key Cryptography Library includes complete implementation of the following public key cryptography algorithms:

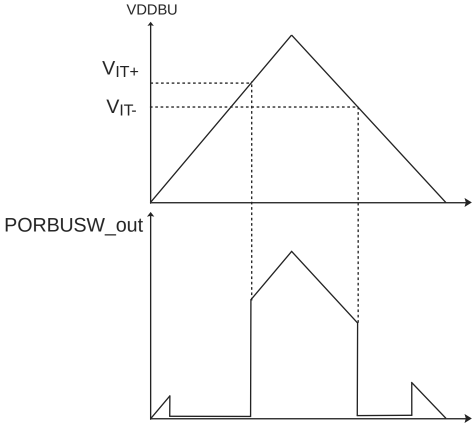
- RSA, DSA:
  - Modular Exponentiation with CRT up to 6144 bits
  - Modular Exponentiation without CRT up to 5408 bits
  - Prime generation
  - Utilities: GCD/modular Inverse, Divide, Modular reduction, Multiply, …
- Elliptic Curves:
  - ECDSA up to 1504 bits
  - Point Multiply,
  - Point Add/Doubling
  - Elliptic Curves in GF(p) or GF(2^n)
  - Choice of the curves parameters so compatibility with NIST Curves or others.
- Deterministic Random Number Generation (DRNG ANSI X9.31) for DSA

46.5.7 VDDBU Power-On-Reset

Table 46-25. Zero-Power-On POR (Backup POR) Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{TH+}	Positive-going input threshold voltage (VDDBU)	At startup	1.45	1.53	1.59	V
V_{TH-}	Negative-going input threshold voltage (VDDBU)	–	1.35	1.45	1.55	V
I_{DDBU}	Current consumption	Enabled	–	300	700	nA
t_{res}	Reset time-out period	–	100	240	500	µs

Figure 46-15. Zero-Power-On Reset Characteristics



46.5.8 VDDIO Power-On-Reset

Table 46-26. Zero-Power-On POR (VDDIO POR) Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{TH+}	Positive-going input threshold voltage (VDDIO)	At startup	1.45	1.53	1.59	V
V_{TH-}	Negative-going input threshold voltage (VDDIO)	–	1.35	1.45	1.55	V
I_{DDIO}	Current consumption	–	–	300	700	nA
t_{res}	Reset time-out period	–	100	240	500	µs

Table 55-2. SAM4CM Datasheet Rev. 11203D Revision History

Doc. Rev. 11203D	Changes
27-Mar-15 (cont'd)	Section 35. "Universal Asynchronous Receiver Transmitter (UART)" Modified Figure 35-2 "Baud Rate Generator" Section 35.6.9 "UART Baud Rate Generator Register": updated CD bit description
	Section 36. "Universal Synchronous Asynchronous Receiver Transceiver (USART)" Removed all references to bit RXIDLEV Updated Section 36.5.1 "I/O Lines", Section 36.6.1 "Baud Rate Generator", Section 36.6.3.15 "Hardware Handshaking", Section 36.6.9 "Register Write Protection" Section 36.7.1 "USART Control Register", Section 36.7.3 "USART Mode Register", Section 36.7.6 "USART Interrupt Enable Register (SPI_MODE)", Section 36.7.8 "USART Interrupt Disable Register (SPI_MODE)", Section 36.7.10 "USART Interrupt Mask Register (SPI_MODE)", Section 36.7.11 "USART Channel Status Register", Section 36.7.12 "USART Channel Status Register (SPI_MODE)", Section 36.7.15 "USART Baud Rate Generator Register", Section 36.7.16 "USART Receiver Time-out Register", Section 36.7.17 "USART Transmitter Timeguard Register": updated bit descriptions Updated Table 36-14 "Register Mapping"
	Section 37. "Timer Counter (TC)" Updated Section 37.1 "Description", Section 37.5.2 "Power Management", Section 37.5.3 "Interrupt Sources", Section 37.6.14.4 "Position and Rotation Measurement", Section 37.6.14.5 "Speed Measurement" and Section 37.6.16 "Register Write Protection" Section 37.6.14 "Quadrature Decoder": removed subsection "Missing Pulse Detection and Auto-correction" Section 37.7.2 "TC Channel Mode Register: Capture Mode": in 'Name' line, replaced "(WAVE = 0)" with "(CAPTURE_MODE)" Section 37.7.3 "TC Channel Mode Register: Waveform Mode": in 'Name' line, replaced "(WAVE = 1)" with "(WAVEFORM_MODE)" Section 37.7.5 "TC Counter Value Register", Section 37.7.6 "TC Register A", Section 37.7.7 "TC Register B", Section 37.7.8 "TC Register C": added 'IMPORTANT' note Section 37.7.9 "TC Status Register", Section 37.7.19 "TC Write Protection Mode Register": updated bit descriptions Section 37.7.14 "TC Block Mode Register": removed AUTOC bit and MAXCMP field Section 37.7.18 "TC QDEC Interrupt Status Register": removed MPE bit
	Section 42. "Advanced Encryption Standard (AES)" Added Section 42.4.1 "AES Register Endianism" Section 42.5.6 "AES Interrupt Status Register": updated bit descriptions
	Section 43. "Integrity Check Monitor (ICM)" Updated Section 43.5.1 "Overview", Section 43.5.4 "Using ICM as SHA Engine" Section 43.5.2.2 "ICM Region Configuration Structure Member", Section 43.6.1 "ICM Configuration Register", Section 43.6.3 "ICM Status Register": updated bit descriptions
	Section 45. "True Random Number Generator (TRNG)" Updated Section 45.5 "Functional Description" and Table 45-2 "Register Mapping"
	Section 46. "Electrical Characteristics" Updated Table 46-1 "Absolute Maximum Ratings*" and Table 46-3 "Recommended Operating Conditions on Input Pins" Updated Figure 46-19 "Typical Current Consumption in Backup Mode for Configurations C and D" Added footnotes (4) and (5) in Table 46-5 "I/O DC Characteristics" Updated VDDIN min. value in Table 46-41 "Programmable Voltage Reference Characteristics"