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	256KB (256K 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/atsam4cms4cb-aur

Features

- Application/Master Core
  - ARM Cortex-M4 running at up to 120 MHz(1)
  - Memory Protection Unit (MPU)
  - DSP Instruction
  - Thumb®-2 instruction set
  - Instruction and Data Cache Controller with 2 Kbytes Cache Memory
  - Memories
    - Up to 2 Mbytes of Embedded Flash for Program Code (I-Code bus) and Program Data (D-Code bus) with Built-in ECC (2-bit error detection and 1-bit correction per 128 bits)
    - Up to 256 Kbytes of Embedded SRAM (SRAM0) for Program Data (System bus)
    - 8 Kbytes of ROM with embedded bootloader routines (UART) and In-Application Programming (IAP) routines
- Coprocessor (provides ability to separate application, communication or metrology functions)
  - ARM Cortex-M4F running at up to 120 MHz(1)
  - IEEE® 754 Compliant, Single-precision Floating-Point Unit (FPU)
  - DSP Instruction
  - Thumb-2 instruction set
  - Instruction and Data Cache Controller with 2 Kbytes of Cache Memory
  - Memories
    - Up to 32 Kbytes of Embedded SRAM (SRAM1) for Program Code (I-Code bus) and Program Data (D-Code bus and System bus)
    - Up to 16 Kbytes of Embedded SRAM (SRAM2) for Program Data (System bus)
- Symmetrical/Asynchronous Dual Core Architecture
  - Interrupt-based Interprocessor Communication
  - Asynchronous Clocking
  - One Interrupt Controller (NVIC) for each core
  - Each Peripheral IRQ routed to each NVIC Input
- Cryptography
  - High-performance AES 128 to 256 with various modes (GCM, CBC, ECB, CFB, CBC-MAC, CTR)
  - TRNG (up to 38 Mbit/s stream, with tested Diehard and FIPS)
  - Public Key Crypto accelerator and associated ROM library for RSA, ECC, DSA, ECDSA
  - Integrity Check Module (ICM) based on Secure Hash Algorithm (SHA1, SHA224, SHA256), DMA-assisted
- Safety
  - Up to two physical Anti-tamper Detection I/Os with Time Stamping and Immediate Clear of General Backup Registers
  - Security Bit for Device Protection from JTAG Accesses
- Shared System Controller
  - Power Supply
    - Embedded core and LCD voltage regulator for single-supply operation
    - Power-on-Reset (POR), Brownout Detector (BOD) and Dual Watchdog for safe operation
    - Ultra-low-power Backup mode (< 0.5 µA Typical @ 25°C)

1. Configuration Summary

The SAM4CM devices differ in memory size, package and features. Table 1-1 summarizes the different device configurations.

Table 1-1. Configuration Summary

Feature	SAM4CMP32C	SAM4CMP16C	SAM4CMP8C	SAM4CMS32C	SAM4CMS16C	SAM4CMS8C	SAM4CMS4C
Flash	2048 Kbytes	1024 Kbytes	512 Kbytes	2048 Kbytes	1024 Kbytes	512 Kbytes	256 Kbytes
SRAM	256 + 32 +16 Kbytes	128 + 16 + 8 Kbytes		256 + 32 +16 Kbytes	128 + 16 + 8 Kbytes		
Package	LQFP 100						
Number of PIOs	52			57			
External Bus Interface	8-bit data						
16-bit Timer	6 channels						
16-bit PWM	3 channels						
UART / USART	2/3			2/4			
SPI(1)	1/4 + 3			1/4 + 4			
TWI	2						
10-bit ADC Channels(2)	6						
Energy Metering Analog Front End	7 channels (3 voltages, 4 currents)			4 channels (2 voltages, 2 currents)			
Cryptography	AES, CPKCC, ICM (SHA), TRNG						
Segmented LCD	33 segments × 6 commons			38 segments × 6 commons			
Anti-Tampering Inputs	1			2			
Flash Page Size	512 bytes						
Flash Pages	2 × 2048	2048	1024	2 × 2048	2048	1024	512
Flash Lock Region Size	8 Kbytes						
Flash Lock Bits	2 × 128	128	64	2 × 128	128	64	32

Notes: 1. 1/4 + 3 = Number of SPI Controllers / Number of Chip Selects + Number of USARTs with SPI mode.
2. One channel is reserved for internal temperature sensor and one channel for VDDBU measurement.

12.6.7.7 UQADD and UQSUB

Saturating Add and Saturating Subtract Unsigned.

Syntax

```
op{cond} {Rd}, Rn, Rm
op{cond} {Rd}, Rn, Rm
```

where:

op is one of:

UQADD8 Saturating four unsigned 8-bit integer additions.

UQADD16 Saturating two unsigned 16-bit integer additions.

UDSUB8 Saturating four unsigned 8-bit integer subtractions.

UQSUB16 Saturating two unsigned 16-bit integer subtractions.

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

Rd is the destination register.

Rn, Rm are registers holding the first and second operands.

Operation

These instructions add or subtract two or four values and then writes an unsigned saturated value in the destination register.

The UQADD16 instruction:

- Adds the respective top and bottom halfwords of the first and second operands.
- Saturates the result of the additions for each halfword in the destination register to the unsigned range $0 \le x \le 2^{16}$-1, where x is 16.

The UQADD8 instruction:

- Adds each respective byte of the first and second operands.
- Saturates the result of the addition for each byte in the destination register to the unsigned range $0 \le x \le 2^{8}$-1, where x is 8.

The UQSUB16 instruction:

- Subtracts both halfwords of the second operand from the respective halfwords of the first operand.
- Saturates the result of the differences in the destination register to the unsigned range $0 \le x \le 2^{16}$-1, where x is 16.

The UQSUB8 instructions:

- Subtracts the respective bytes of the second operand from the respective bytes of the first operand.
- Saturates the results of the differences for each byte in the destination register to the unsigned range $0 \le x \le 2^{8}$-1, where x is 8.

Restrictions

Do not use SP and do not use PC.

12.6.11.24 VPOP

Floating-point extension register Pop.

Syntax

```
VPOP{cond}{.size} list
```

where:

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

size — is an optional data size specifier.
If present, it must be equal to the size in bits, 32 or 64, of the registers in *list*.

list — is the list of extension registers to be loaded, as a list of consecutively numbered doubleword or singleword registers, separated by commas and surrounded by brackets.

Operation

This instruction loads multiple consecutive extension registers from the stack.

Restrictions

The list must contain at least one register, and not more than sixteen registers.

Condition Flags

These instructions do not change the flags.

12.6.11.29 VSUB

Floating-point Subtract.

Syntax

```
VSUB{cond}.F32 {Sd,} Sn, Sm
```

where:

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

Sd is the destination floating-point value.

Sn, Sm are the operand floating-point value.

Operation

This instruction:

1. Subtracts one floating-point value from another floating-point value.
2. Places the results in the destination floating-point register.

Restrictions

There are no restrictions.

Condition Flags

These instructions do not change the flags.

12.6.12 Miscellaneous Instructions

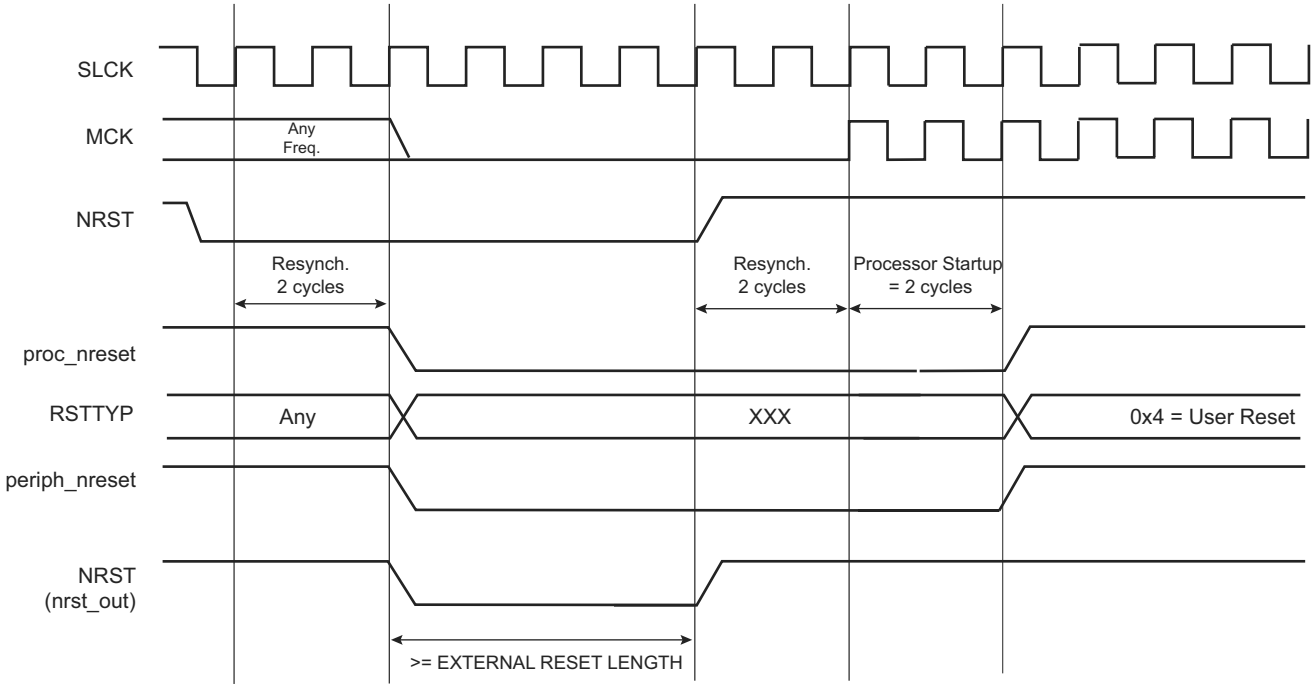
The table below shows the remaining Cortex-M4 instructions.

Table 12-28. Miscellaneous Instructions

Mnemonic	Description
BKPT	Breakpoint
CPSID	Change Processor State, Disable Interrupts
CPSIE	Change Processor State, Enable Interrupts
DMB	Data Memory Barrier
DSB	Data Synchronization Barrier
ISB	Instruction Synchronization Barrier
MRS	Move from special register to register
MSR	Move from register to special register
NOP	No Operation
SEV	Send Event
SVC	Supervisor Call
WFE	Wait For Event
WFI	Wait For Interrupt

Conditional execution	If the condition code flags indicate that the corresponding condition is true when the instruction starts executing, it executes normally. Otherwise, the instruction does nothing.
Context	The environment that each process operates in for a multitasking operating system. In ARM processors, this is limited to mean the physical address range that it can access in memory and the associated memory access permissions.
Coprocessor	A processor that supplements the main processor. Cortex-M4 does not support any coprocessors.
Debugger	A debugging system that includes a program, used to detect, locate, and correct software faults, together with custom hardware that supports software debugging.
Direct Memory Access (DMA)	An operation that accesses main memory directly, without the processor performing any accesses to the data concerned.
Doubleword	A 64-bit data item. The contents are taken as being an unsigned integer unless otherwise stated.
Doubleword-aligned	A data item having a memory address that is divisible by eight.
Endianness	Byte ordering. The scheme that determines the order that successive bytes of a data word are stored in memory. An aspect of the system's memory mapping. *See also* "Little-endian (LE)" and "Big-endian (BE)".
Exception	An event that interrupts program execution. When an exception occurs, the processor suspends the normal program flow and starts execution at the address indicated by the corresponding exception vector. The indicated address contains the first instruction of the handler for the exception. An exception can be an interrupt request, a fault, or a software-generated system exception. Faults include attempting an invalid memory access, attempting to execute an instruction in an invalid processor state, and attempting to execute an undefined instruction.
Exception service routine	*See* "Interrupt handler".
Exception vector	*See* "Interrupt vector".
Flat address mapping	A system of organizing memory in which each physical address in the memory space is the same as the corresponding virtual address.
Halfword	A 16-bit data item.
Illegal instruction	An instruction that is architecturally Undefined.
Implementation-defined	The behavior is not architecturally defined, but is defined and documented by individual implementations.

Figure 15-6. User Reset State



15.4.4 Reset State Priorities

The reset state manager manages the priorities among the different reset sources. The resets are listed in order of priority as follows:

1. General reset
2. Backup reset
3. Watchdog reset
4. Software reset
5. User reset

Particular cases are listed below:

- When in user reset:
  - A watchdog event is impossible because the Watchdog Timer is being reset by the proc_nreset signal.
  - A software reset is impossible, since the processor reset is being activated.
- When in software reset:
  - A watchdog event has priority over the current state.
  - The NRST has no effect.
- When in watchdog reset:
  - The processor reset is active and so a software reset cannot be programmed.
  - A user reset cannot be entered.

15.4.5 Managing Reset at Application Level

As described, the device embeds only one system and power management controller shared by the two sub-systems, i.e.:

- Sub-system 0 (SUB-S0 - Application Master/Core)
- Sub-system 1 (SUB-S1 - Metrology/Co-Processor Core)

17.6.1 RTC Control Register

Name: RTC_CR

Address: 0x400E1460

Access: Read/Write

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

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

- **UPDTIM: Update Request Time Register**

0: No effect or, if UPDTIM has been previously written to 1, stops the update procedure.

1: Stops the RTC time counting.

Time counting consists of second, minute and hour counters. Time counters can be programmed once this bit is set and acknowledged by the bit ACKUPD of the RTC_SR.

- **UPDCAL: Update Request Calendar Register**

0: No effect or, if UPDCAL has been previously written to 1, stops the update procedure.

1: Stops the RTC calendar counting.

Calendar counting consists of day, date, month, year and century counters. Calendar counters can be programmed once this bit is set and acknowledged by the bit ACKUPD of the RTC_SR.

- **TIMEVSEL: Time Event Selection**

The event that generates the flag TIMEV in RTC_SR depends on the value of TIMEVSEL.

Value	Name	Description
0	MINUTE	Minute change
1	HOUR	Hour change
2	MIDNIGHT	Every day at midnight
3	NOON	Every day at noon

25.5.6 IPC Interrupt Mask Register

Name: IPC_IMR

Address: 0x4004C014 (0), 0x48014014 (1)

Access: Read-only

31	30	29	28	27	26	25	24
IRQ31	IRQ30	IRQ29	IRQ28	IRQ27	IRQ26	IRQ25	IRQ24
23	22	21	20	19	18	17	16
IRQ23	IRQ22	IRQ21	IRQ20	IRQ19	IRQ18	IRQ17	IRQ16
15	14	13	12	11	10	9	8
IRQ15	IRQ14	IRQ13	IRQ12	IRQ11	IRQ10	IRQ9	IRQ8
7	6	5	4	3	2	1	0
IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0

- **IRQ0-IRQ31: Interrupt Mask**

0: The corresponding interrupt is disabled.

1: The corresponding interrupt is enabled.

26.9.1 Bus Matrix Master Configuration Registers

Name: MATRIX_MCFGx [x=0..6]

Address: 0x400E0200 (0), 0x48010000 (1)

Access: Read/Write

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
–	–	–	–	–	ULBT		

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

- **ULBT: Undefined Length Burst Type**

0: Unlimited Length Burst

No predicted end of burst is generated, therefore INCR bursts coming from this master can only be broken if the Slave Slot Cycle Limit is reached. If the Slot Cycle Limit is not reached, the burst is normally completed by the master, at the latest, on the next AHB 1 KByte address boundary, allowing up to 256-beat word bursts or 128-beat double-word bursts.

This value should not be used in the very particular case of a master capable of performing back-to-back undefined length bursts on a single slave, since this could indefinitely freeze the slave arbitration and thus prevent another master from accessing this slave.

1: Single Access

The undefined length burst is treated as a succession of single accesses, allowing re-arbitration at each beat of the INCR burst or bursts sequence.

2: 4-beat Burst

The undefined length burst or bursts sequence is split into 4-beat bursts or less, allowing re-arbitration every 4 beats.

3: 8-beat Burst

The undefined length burst or bursts sequence is split into 8-beat bursts or less, allowing re-arbitration every 8 beats.

4: 16-beat Burst

The undefined length burst or bursts sequence is split into 16-beat bursts or less, allowing re-arbitration every 16 beats.

5: 32-beat Burst

The undefined length burst or bursts sequence is split into 32-beat bursts or less, allowing re-arbitration every 32 beats.

6: 64-beat Burst

The undefined length burst or bursts sequence is split into 64-beat bursts or less, allowing re-arbitration every 64 beats.

7: 128-beat Burst

The undefined length burst or bursts sequence is split into 128-beat bursts or less, allowing re-arbitration every 128 beats.

Unless duly needed, the ULBT should be left at its default 0 value for power saving.

32.6.36 PIO Additional Interrupt Modes Enable Register

Name: PIO_AIMER

Address: 0x400E0EB0 (PIOA), 0x400E10B0 (PIOB), 0x4800C0B0 (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: Additional Interrupt Modes Enable**

0: No effect.

1: The interrupt source is the event described in PIO_ELSR and PIO_FRLHSR.

32.6.37 PIO Additional Interrupt Modes Disable Register

Name: PIO_AIMDR

Address: 0x400E0EB4 (PIOA), 0x400E10B4 (PIOB), 0x4800C0B4 (PIOC)

Access: Write-only

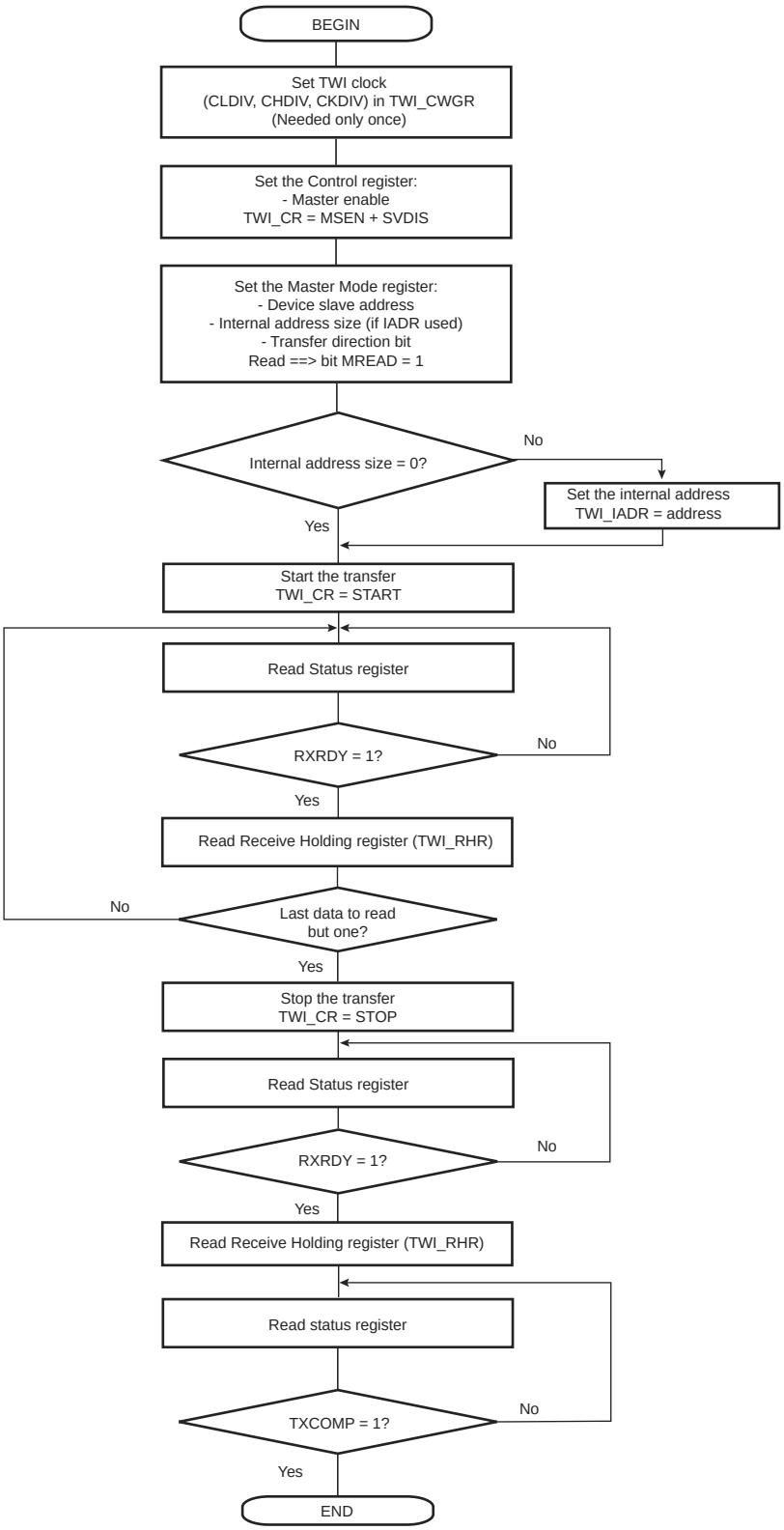
31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24
23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16
15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8
7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

- **P0–P31: Additional Interrupt Modes Disable**

0: No effect.

1: The interrupt mode is set to the default interrupt mode (both-edge detection).

Figure 34-20. TWI Read Operation with Multiple Data Bytes with or without Internal Address



35.4 Product Dependencies

35.4.1 I/O Lines

The UART pins are multiplexed with PIO lines. The user must first configure the corresponding PIO Controller to enable I/O line operations of the UART.

Table 35-2. I/O Lines

Instance	Signal	I/O Line	Peripheral
UART0	URXD0	PB4	A
UART0	UTXD0	PB5	A
UART1	URXD1	PC1	A
UART1	UTXD1	PC0	A

35.4.2 Power Management

The UART clock can be controlled through the Power Management Controller (PMC). In this case, the user must first configure the PMC to enable the UART clock. Usually, the peripheral identifier used for this purpose is 1.

35.4.3 Interrupt Sources

The UART interrupt line is connected to one of the interrupt sources of the Interrupt Controller. Interrupt handling requires programming of the Interrupt Controller before configuring the UART.

Table 35-3. Peripheral IDs

Instance	ID
UART0	8
UART1	38

35.4.4 Optical Interface

The UART optical interface requires configuration of the PMC to generate 4096 kHz or 8192 kHz on the PLLA prior to any transfer.

35.5 Functional Description

The UART operates in Asynchronous mode only and supports only 8-bit character handling (with parity). It has no clock pin.

The UART is made up of a receiver and a transmitter that operate independently, and a common baud rate generator. Receiver timeout and transmitter time guard are not implemented. However, all the implemented features are compatible with those of a standard USART.

35.5.1 Baud Rate Generator

The baud rate generator provides the bit period clock named baud rate clock to both the receiver and the transmitter.

The baud rate clock is the peripheral clock divided by 16 times the clock divisor (CD) value written in the Baud Rate Generator register (UART_BRGR). If UART_BRGR is set to 0, the baud rate clock is disabled and the UART remains inactive. The maximum allowable baud rate is peripheral clock divided by 16. The minimum allowable baud rate is peripheral clock divided by (16 x 65536).

36.7.15 USART Baud Rate Generator Register

Name: US_BRGR

Address: 0x40024020 (0), 0x40028020 (1), 0x4002C020 (2), 0x40030020 (3), 0x40034020 (4)

Access: Read/Write

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

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

15	14	13	12	11	10	9	8
CD							

7	6	5	4	3	2	1	0
CD							

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

- **CD: Clock Divider**

	USART_MODE ≠ ISO7816			
	SYNC = 0		**SYNC = 1 or USART_MODE = SPI (Master or Slave)**	**USART_MODE = ISO7816**
CD	**OVER = 0**	**OVER = 1**		
0	Baud Rate Clock Disabled			
1 to 65535	CD = Selected Clock / (16 × Baud Rate)	CD = Selected Clock / (8 × Baud Rate)	CD = Selected Clock / Baud Rate	CD = Selected Clock / (FI_DI_RATIO × Baud Rate)

- **FP: Fractional Part**

0: Fractional divider is disabled.

1–7: Baud rate resolution, defined by FP × 1/8.

38.7.7 PWM Interrupt Mask Register

Name: PWM_IMR

Address: 0x48008018

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 = Interrupt for PWM channel x is disabled.

1 = Interrupt for PWM channel x is enabled.

39.6.9 User Buffer Organization

The pixels to be displayed are written into SLCDC_LMEMRx and SLCDC_MMEMRx registers. There are up to two 32-bit registers for each common terminal. Table 39-6 provides the address mapping of all commons/segments to be displayed.

If the segment map registers (SLCDC_SMR0/1) are cleared and the number of segments to handle (SEGSEL field in SLCDC_MR) is lower or equal to 32, the registers SLCDC_MMEMRx are not required to be programmed and can be left cleared (default value).

In case segments are remapped, the SLCDC_MMEMRx registers are not required to be programmed if SLCDC_SMR1 register is cleared (i.e., no segment remapped on SEG32 to SEG39 I/O pins). In this case SLCDC_MMEMRx registers must be cleared.

In the same way if all segments are remapped on the upper part of the SEG terminals (SEG32 to SEG39) there is no need to program SLCDC_LMEMRx registers (they must be cleared).

When segment remap is used (SLCDC_SMR0/1 registers differ from 0), the unmapped segments must be kept cleared to limit internal signal switching.

Table 39-6. Commons/Segments Address Mapping

Register	Common Terminal	SEG0	--	SEG31	SEG32	--	SEG39	Memory address
SLCDC_MMEMR5	COM5				X	--	X	0x22C
SLCDC_LMEMR5	COM5	X	--	X				0x228
SLCDC_MMEMR4	COM4				X	--	X	0x224
SLCDC_LMEMR4	COM4	X	--	X				0x220
SLCDC_MMEMR3	COM3				X	--	X	0x21C
SLCDC_LMEMR3	COM3	X	--	X				0x218
SLCDC_MMEMR2	COM2				X	--	X	0x214
SLCDC_LMEMR2	COM2	X	--	X				0x210
SLCDC_MMEMR1	COM1				X	--	X	0x20C
SLCDC_LMEMR1	COM1	X	--	X				0x208
SLCDC_MMEMR0	COM0				X	--	X	0x204
SLCDC_LMEMR0	COM0	X	--	X				0x200

39.6.10 Segments Mapping Function

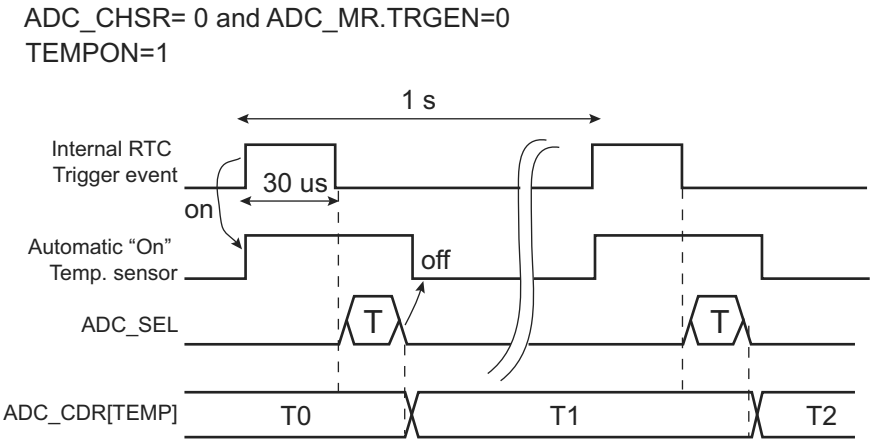
By default the segments pins (SEG0:39) are automatically assigned according to the SEGSEL configuration in the SLCDC_MR. The unused SEG I/O pins are forced to be driven by a digital peripheral or can be used as I/O through the PIO controller.

The automatic assignment is performed if the segment mapping function is not used (SLCDC_SMR0/1 registers are cleared). The following table provides such assignments.

Table 39-7. Segment Pin Assignments

SEGSEL	I/O Port in Use as Segment Driver	I/O Port Pin if SLCDC_SMR0/1 = 0
0	SEG0	SEG1:39
1	SEG0:1	SEG2:39
...	...	...
37	SEG0:37	SEG39
39	SEG0:39	None

Figure 40-8. Temperature Conversion Only



Notes: ADC_SEL: Command to the ADC cell
C: Classic ADC Conversion Sequence
T: Temperature Sensor Channel

It is possible to raise a flag only if there is a predefined change in the temperature. The user can define a range of temperature or a threshold in the Temperature Compare Window register (ADC_TEMPCWR), and the mode of comparison that can be programmed into the TEMPCMPMOD field into ADC_TEMPMR. These values define how the TEMPCHG flag is raised in ADC_ISR.

The TEMPCHG flag can be used to generate a temperature-dependent interrupt instead of the end-of-conversion interrupt. More specifically, the interrupt is generated only if the temperature sensor as measured by the ADC reports a temperature value below, above, inside or outside programmable thresholds (see “ADC Temperature Sensor Mode Register”).

In any case, if TEMPON is set, the temperature can be read at anytime in ADC_CDR7 without any specific software intervention.

40.6.10 VDDBU Measurement

The seventh ADC channel (CH6) of the ADC Controller is reserved for measurement of the VDDBU power supply pin. For this channel, setting up, starting conversion, and other tasks must be performed the same way as for all other channels. VDDBU is measured without any attenuation. This means that for VDDBU greater than the voltage reference applied to the ADC, the digital output clamps to the maximum value.

40.6.11 Enhanced Resolution Mode and Digital Averaging Function

The Enhanced Resolution mode is enabled if LOWRES is cleared in ADC_MR, and the OSR field is set to 1 or 2 in ADC_EMR. The enhancement is based on a digital averaging function.

FREERUN in ADC_MR must be cleared when digital averaging is used (OSR not equal to 0 in ADC_EMR).

There is no averaging on the last index channel if the measure is triggered by an RTC event (see Section 40.6.9 ”Temperature Sensor”).

In Enhanced Resolution mode, the ADC Controller trades conversion speed for quantization noise by averaging multiple samples, thus providing a digital low-pass filter function.

If 1-bit enhancement resolution is selected (OSR = 1 in ADC_EMR), the ADC real sample rate is the maximum ADC sample rate divided by 4. Thus, the oversampling ratio is 4.

When the 2-bit enhancement resolution is selected (OSR = 2 in ADC_EMR), the ADC real sample rate is the maximum ADC sample rate divided by 16 (oversampling ratio is 16).

46.6.2.1 SAM4CM4/8/16 Flash Wait States and Operating Frequency

The maximum operating frequency given in Table 46-52 below is limited by the Embedded Flash access time when the processor is fetching code out of it. The table gives the device maximum operating frequency depending on the FWS field of the EFC_FMR register. This field defines the number of wait states required to access the Embedded Flash Memory.

Table 46-52. SAM4CM4/8/16 Flash Wait State Versus Operating Frequency

FWS (Flash Wait State)	Maximum Operating Frequency (MHz) @ T_A = 85°C			
	VDDCORE = 1.08V VDDIO = 1.62V to 3.6V	VDDCORE = 1.2V VDDIO = 1.62V to 3.6V	VDDCORE = 1.08V VDDIO = 2.7V to 3.6V	VDDCORE = 1.2V VDDIO = 2.7V to 3.6V
0	16	17	20	21
1	33	35	40	42
2	51	52	61	63
3	67	70	81	85
4	85	87	98	106
5	100	105	–	120
6	–	121	–	–

46.6.2.2 SAM4CM32 Flash Wait States and Operating Frequency

The maximum operating frequency given in Table 46-53 below is limited by the Embedded Flash access time when the processor is fetching code out of it. The table gives the device maximum operating frequency depending on the FWS field of the EFC_FMR register. This field defines the number of wait states required to access the Embedded Flash Memory.

Table 46-53. SAM4CM32 Flash Wait State Versus Operating Frequency

FWS (Flash Wait State)	Maximum Operating Frequency (MHz) @ T_A = 85°C			
	VDDCORE = 1.08V VDDIO = 1.62V to 3.6V	VDDCORE = 1.2V VDDIO = 1.62V to 3.6V	VDDCORE = 1.08V VDDIO = 2.7V to 3.6V	VDDCORE = 1.2V VDDIO = 2.7V to 3.6V
0	16	17	20	21
1	33	34	40	42
2	50	52	60	63
3	67	69	80	83
4	84	86	91	104
5	91	104	–	118
6	–	114	–	–