**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, I²C, 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 | 256KB (256K 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/atsam4s4cb-cnr |

## 6.3 Test Pin

The TST pin is used for JTAG Boundary Scan Manufacturing Test or Fast Flash programming mode of the SAM4S series. The TST pin integrates a permanent pull-down resistor of about 15 kΩ to GND, so that it can be left unconnected for normal operations. To enter fast programming mode, see Section 21. “Fast Flash Programming Interface (FFPI)”. For more on the manufacturing and test mode, refer to Section 13. “Debug and Test Features”.

## 6.4 NRST Pin

The NRST pin is bidirectional. It is handled by the on-chip reset controller and can be driven low to provide a reset signal to the external components or asserted low externally to reset the microcontroller. It will reset the Core and the peripherals except the Backup region (RTC, RTT and Supply Controller). There is no constraint on the length of the reset pulse and the reset controller can guarantee a minimum pulse length. The NRST pin integrates a permanent pull-up resistor to VDDIO of about 100 kΩ. By default, the NRST pin is configured as an input.

## 6.5 ERASE Pin

The ERASE pin is used to reinitialize the Flash content (and some of its NVM bits) to an erased state (all bits read as logic level 1). The ERASE pin and the ROM code ensure an in-situ reprogrammability of the Flash content without the use of a debug tool. When the security bit is activated, the ERASE pin provides the capability to reprogram the Flash content. It integrates a pull-down resistor of about 100 kΩ to GND, so that it can be left unconnected for normal operations.

This pin is debounced by SCLK to improve the glitch tolerance. To avoid unexpected erase at power-up, a minimum ERASE pin assertion time is required. This time is defined in Table 44-74 “AC Flash Characteristics”.

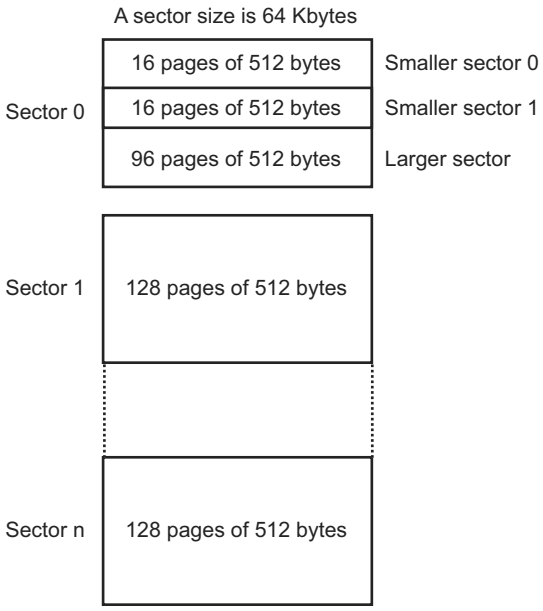
The ERASE pin is a system I/O pin and can be used as a standard I/O. At startup, the ERASE pin is not configured as a PIO pin. If the ERASE pin is used as a standard I/O, startup level of this pin must be low to prevent unwanted erasing. Refer to Section 11.2 “Peripheral Signal Multiplexing on I/O Lines” on page 51. Also, if the ERASE pin is used as a standard I/O output, asserting the pin to low does not erase the Flash.

## 6.6 Anti-tamper Pins/Low-power Tamper Detection

WKUP0 and WKUP1 generic wake-up pins can be used as anti-tamper pins. Anti-tamper pins detect intrusion, for example, into a housing box. Upon detection through a tamper switch, automatic, asynchronous and immediate clear of registers in the backup area will be performed. Anti-tamper pins can be used in all power modes (Backup/Wait/Sleep/Active). Anti-tampering events can be programmed so that half of the General Purpose Backup Registers (GPBR) are erased automatically. See "Supply Controller" section for further description.

RTCOUT0 and RTCOUT1 pins can be used to generate waveforms from the RTC in order to take advantage of the RTC inherent prescalers while the RTC is the only powered circuitry (low-power mode, Backup mode) or in any active mode. Entering backup or low-power modes does not affect the waveform generation outputs. Anti-tampering pin detection can be synchronized with this signal.

**Figure 8-2. Flash Sector Organization**

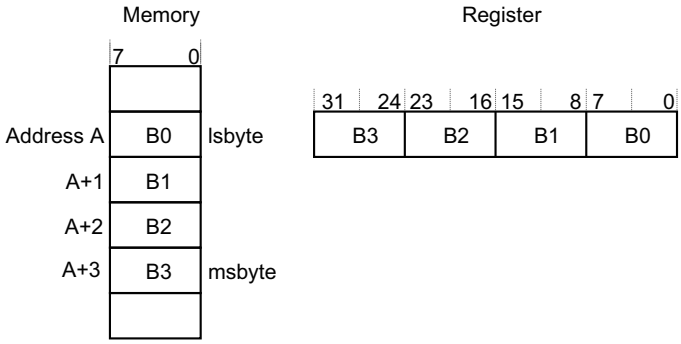



Flash size varies by product:

- SAM4S2: the Flash size is 128 Kbytes in a single plane
- SAM4S4: the Flash size is 256 Kbytes in a single plane
- SAM4S8/S16: the Flash size is 512 Kbytes in a single plane
  - Internal Flash address is 0x0040_0000
- SAM4SD16/SA16: the Flash size is 2 x 512 Kbytes
  - Internal Flash0 address is 0x0040_0000
  - Internal Flash1 address is 0x0048_0000
- SAM4SD32: the Flash size is 2 x 1024 Kbytes
  - Internal Flash0 address is 0x0040_0000
  - Internal Flash1 address is 0x0050_0000

Refer to Figure 8-3, "Flash Size" for the organization of the Flash depending on its size.

**Figure 12-5.** ***Little-endian Format***




#### 12.4.2.7 Synchronization Primitives

The Cortex-M4 instruction set includes pairs of *synchronization primitives*. These provide a non-blocking mechanism that a thread or process can use to obtain exclusive access to a memory location. The software can use them to perform a guaranteed read-modify-write memory update sequence, or for a semaphore mechanism.

A pair of synchronization primitives comprises:

**A Load-exclusive Instruction**, used to read the value of a memory location, requesting exclusive access to that location.

**A Store-Exclusive instruction**, used to attempt to write to the same memory location, returning a status bit to a register. If this bit is:

- 0: It indicates that the thread or process gained exclusive access to the memory, and the write succeeds,
- 1: It indicates that the thread or process did not gain exclusive access to the memory, and no write is performed.

The pairs of Load-Exclusive and Store-Exclusive instructions are:

- The word instructions LDREX and STREX
- The halfword instructions LDREXH and STREXH
- The byte instructions LDREXB and STREXB.

The software must use a Load-Exclusive instruction with the corresponding Store-Exclusive instruction.

To perform an exclusive read-modify-write of a memory location, the software must:

1. Use a Load-Exclusive instruction to read the value of the location.
2. Update the value, as required.
3. Use a Store-Exclusive instruction to attempt to write the new value back to the memory location
4. Test the returned status bit. If this bit is:

   0: The read-modify-write completed successfully.

   1: No write was performed. This indicates that the value returned at step 1 might be out of date. The software must retry the read-modify-write sequence.

The software can use the synchronization primitives to implement a semaphore as follows:

1. Use a Load-Exclusive instruction to read from the semaphore address to check whether the semaphore is free.
2. If the semaphore is free, use a Store-Exclusive instruction to write the claim value to the semaphore address.

### 12.6.4 Memory Access Instructions

The table below shows the memory access instructions.

**Table 12-17. Memory Access Instructions**

| Mnemonic | Description |
|---|---|
| ADR | Load PC-relative address |
| CLREX | Clear Exclusive |
| LDM{mode} | Load Multiple registers |
| LDR{type} | Load Register using immediate offset |
| LDR{type} | Load Register using register offset |
| LDR{type}T | Load Register with unprivileged access |
| LDR | Load Register using PC-relative address |
| LDRD | Load Register Dual |
| LDREX{type} | Load Register Exclusive |
| POP | Pop registers from stack |
| PUSH | Push registers onto stack |
| STM{mode} | Store Multiple registers |
| STR{type} | Store Register using immediate offset |
| STR{type} | Store Register using register offset |
| STR{type}T | Store Register with unprivileged access |
| STREX{type} | Store Register Exclusive |

#### 12.6.5.22 USAD8

Unsigned Sum of Absolute Differences

Syntax

```
USAD8{cond}{Rd,} Rn, Rm
```

where:

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

Rd is the destination register.

Rn is the first operand register.

Rm is the second operand register.

Operation

The USAD8 instruction:

1. Subtracts each byte of the second operand register from the corresponding byte of the first operand register.
2. Adds the absolute values of the differences together.
3. Writes the result to the destination register.

Restrictions

Do not use SP and do not use PC.

Condition Flags

These instructions do not change the flags.

Examples

```
USAD8 R1, R4, R0 ; Subtracts each byte in R0 from corresponding byte of R4
                 ; adds the differences and writes to R1
USAD8 R0, R5     ; Subtracts bytes of R5 from corresponding byte in R0,
                 ; adds the differences and writes to R0.
```

XN and Strongly-ordered rules always apply to the System Control Space regardless of the value of the ENABLE bit.

When the ENABLE bit is set to 1, at least one region of the memory map must be enabled for the system to function unless the PRIVDEFENA bit is set to 1. If the PRIVDEFENA bit is set to 1 and no regions are enabled, then only privileged software can operate.

When the ENABLE bit is set to 0, the system uses the default memory map. This has the same memory attributes as if the MPU is not implemented. The default memory map applies to accesses from both privileged and unprivileged software.

When the MPU is enabled, accesses to the System Control Space and vector table are always permitted. Other areas are accessible based on regions and whether PRIVDEFENA is set to 1.

Unless HFNMIENA is set to 1, the MPU is not enabled when the processor is executing the handler for an exception with priority –1 or –2. These priorities are only possible when handling a hard fault or NMI exception, or when FAULTMASK is enabled. Setting the HFNMIENA bit to 1 enables the MPU when operating with these two priorities.

## 12.12 Glossary

This glossary describes some of the terms used in technical documents from ARM.

**Abort**

A mechanism that indicates to a processor that the value associated with a memory access is invalid. An abort can be caused by the external or internal memory system as a result of attempting to access invalid instruction or data memory.

**Aligned**

A data item stored at an address that is divisible by the number of bytes that defines the data size is said to be aligned. Aligned words and halfwords have addresses that are divisible by four and two respectively. The terms word-aligned and halfword-aligned therefore stipulate addresses that are divisible by four and two respectively.

**Banked register**

A register that has multiple physical copies, where the state of the processor determines which copy is used. The Stack Pointer, SP (R13) is a banked register.

**Base register**

In instruction descriptions, a register specified by a load or store instruction that is used to hold the base value for the instruction's address calculation. Depending on the instruction and its addressing mode, an offset can be added to or subtracted from the base register value to form the address that is sent to memory.

*See also* "Index register" .

**Big-endian (BE)**

Byte ordering scheme in which bytes of decreasing significance in a data word are stored at increasing addresses in memory.

*See also* "Byte-invariant" , "Endianness" , "Little-endian (LE)" .

**Big-endian memory**

Memory in which:
a byte or halfword at a word-aligned address is the most significant byte or halfword within the word at that address,
a byte at a halfword-aligned address is the most significant byte within the halfword at that address.

*See also* "Little-endian memory" .

**Breakpoint**

A breakpoint is a mechanism provided by debuggers to identify an instruction at which program execution is to be halted. Breakpoints are inserted by the programmer to enable inspection of register contents, memory locations, variable values at fixed points in the program execution to test that the program is operating correctly. Breakpoints are removed after the program is successfully tested.

- **OSCSEL: 32-kHz Oscillator Selection Status**

0 (RC): The slow clock, SLCK, is generated by the embedded 32 kHz RC oscillator.

1 (CRYST): The slow clock, SLCK, is generated by the 32 kHz crystal oscillator.

- **LPDBCS0: Low-power Debouncer Wake-up Status on WKUP0 (cleared on read)**

0 (NO): No wake-up due to the assertion of the WKUP0 pin has occurred since the last read of SUPC_SR.

1 (PRESENT): At least one wake-up due to the assertion of the WKUP0 pin has occurred since the last read of SUPC_SR.

- **LPDBCS1: Low-power Debouncer Wake-up Status on WKUP1 (cleared on read)**

0 (NO): No wake-up due to the assertion of the WKUP1 pin has occurred since the last read of SUPC_SR.

1 (PRESENT): At least one wake-up due to the assertion of the WKUP1 pin has occurred since the last read of SUPC_SR.

- **WKUPISx: WKUPx Input Status (cleared on read)**

0 (DIS): The corresponding wake-up input is disabled, or was inactive at the time the debouncer triggered a wake-up event.

1 (EN): The corresponding wake-up input was active at the time the debouncer triggered a wake-up event since the last read of SUPC_SR.

automatically forces the fast RC oscillator to be the source clock for MAINCK. If the fast RC oscillator is disabled when a clock failure detection occurs, it is automatically re-enabled by the clock failure detection mechanism.

It takes two slow RC oscillator clock cycles to detect and switch from the main oscillator, to the fast RC oscillator if the source master clock (MCK) is main clock (MAINCK), or three slow clock RC oscillator cycles if the source of MCK is PLLACKor PLLBCK.

A clock failure detection activates a fault output that is connected to the Pulse Width Modulator (PWM) Controller. With this connection, the PWM controller is able to force its outputs and to protect the driven device, if a clock failure is detected.

The user can know the status of the clock failure detector at any time by reading the FOS bit in PMC_SR.

This fault output remains active until the defect is detected and until it is cleared by the bit FOCLR in the PMC Fault Output Clear Register (PMC_FOCR).

registers results in setting or clearing the corresponding bit in the Pull-down Status Register (PIO_PPDSR). Reading a one in PIO_PPDSR means the pull-up is disabled and reading a zero means the pull-down is enabled.

Enabling the pull-down resistor while the pull-up resistor is still enabled is not possible. In this case, the write of PIO_PPDER for the relevant I/O line is discarded. Likewise, enabling the pull-up resistor while the pull-down resistor is still enabled is not possible. In this case, the write of PIO_PUER for the relevant I/O line is discarded.

Control of the pull-up resistor is possible regardless of the configuration of the I/O line.

After reset, depending on the I/O, pull-up or pull-down can be set.

### 31.5.2 I/O Line or Peripheral Function Selection

When a pin is multiplexed with one or two peripheral functions, the selection is controlled with the Enable Register (PIO_PER) and the Disable Register (PIO_PDR). The Status Register (PIO_PSR) is the result of the set and clear registers and indicates whether the pin is controlled by the corresponding peripheral or by the PIO Controller. A value of zero indicates that the pin is controlled by the corresponding on-chip peripheral selected in the ABCD Select registers (PIO_ABCDSR1 and PIO_ABCDSR2). A value of one indicates the pin is controlled by the PIO Controller.

If a pin is used as a general-purpose I/O line (not multiplexed with an on-chip peripheral), PIO_PER and PIO_PDR have no effect and PIO_PSR returns a one for the corresponding bit.

After reset, the I/O lines are controlled by the PIO Controller, i.e., PIO_PSR resets at one. However, in some events, it is important that PIO lines are controlled by the peripheral (as in the case of memory chip select lines that must be driven inactive after reset, or for address lines that must be driven low for booting out of an external memory). Thus, the reset value of PIO_PSR is defined at the product level and depends on the multiplexing of the device.

### 31.5.3 Peripheral A or B or C or D Selection

The PIO Controller provides multiplexing of up to four peripheral functions on a single pin. The selection is performed by writing PIO_ABCDSR1 and PIO_ABCDSR2.

For each pin:

- The corresponding bit at level zero in PIO_ABCDSR1 and the corresponding bit at level zero in PIO_ABCDSR2 means peripheral A is selected.
- The corresponding bit at level one in PIO_ABCDSR1 and the corresponding bit at level zero in PIO_ABCDSR2 means peripheral B is selected.
- The corresponding bit at level zero in PIO_ABCDSR1 and the corresponding bit at level one in PIO_ABCDSR2 means peripheral C is selected.
- The corresponding bit at level one in PIO_ABCDSR1 and the corresponding bit at level one in PIO_ABCDSR2 means peripheral D is selected.

Note that multiplexing of peripheral lines A, B, C and D only affects the output line. The peripheral input lines are always connected to the pin input (see Figure 31-2).

Writing in PIO_ABCDSR1 and PIO_ABCDSR2 manages the multiplexing regardless of the configuration of the pin. However, assignment of a pin to a peripheral function requires a write in PIO_ABCDSR1 and PIO_ABCDSR2 in addition to a write in PIO_PDR.

After reset, PIO_ABCDSR1 and PIO_ABCDSR2 are zero, thus indicating that all the PIO lines are configured on peripheral A. However, peripheral A generally does not drive the pin as the PIO Controller resets in I/O line mode.

If the software selects a peripheral A, B, C or D which does not exist for a pin, no alternate functions are enabled for this pin and the selection is taken into account. The PIO Controller does not carry out checks to prevent selection of a peripheral which does not exist.

### 33.8.7 SPI Interrupt Disable Register

**Name:** SPI_IDR

**Address:** 0x40008018

**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 |
| – | – | – | – | – | UNDES | TXEMPTY | NSSR |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TXBUFE | RXBUFF | ENDTX | ENDRX | OVRES | MODF | TDRE | RDRF |

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

0: No effect.

1: Disables the corresponding interrupt.

- **RDRF: Receive Data Register Full Interrupt Disable**
- **TDRE: SPI Transmit Data Register Empty Interrupt Disable**
- **MODF: Mode Fault Error Interrupt Disable**
- **OVRES: Overrun Error Interrupt Disable**
- **ENDRX: End of Receive Buffer Interrupt Disable**
- **ENDTX: End of Transmit Buffer Interrupt Disable**
- **RXBUFF: Receive Buffer Full Interrupt Disable**
- **TXBUFE: Transmit Buffer Empty Interrupt Disable**
- **NSSR: NSS Rising Interrupt Disable**
- **TXEMPTY: Transmission Registers Empty Disable**
- **UNDES: Underrun Error Interrupt Disable**

**Table 36-12. IrDA Baud Rate Error (Continued)**

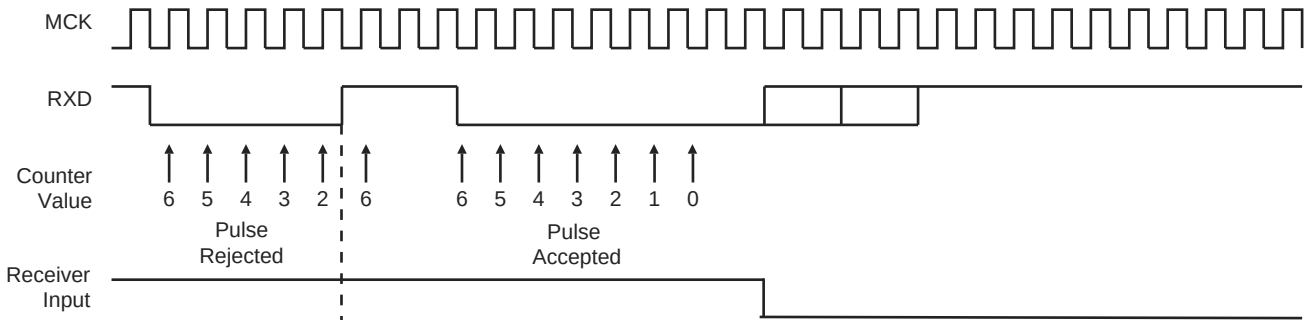
| Peripheral Clock | Baud Rate (bit/s) | CD | Baud Rate Error | Pulse Time (µs) |
|---|---|---|---|---|
| 3,686,400 | 19,200 | 12 | 0.00% | 9.77 |
| 20,000,000 | 19,200 | 65 | 0.16% | 9.77 |
| 32,768,000 | 19,200 | 107 | 0.31% | 9.77 |
| 40,000,000 | 19,200 | 130 | 0.16% | 9.77 |
| 3,686,400 | 9,600 | 24 | 0.00% | 19.53 |
| 20,000,000 | 9,600 | 130 | 0.16% | 19.53 |
| 32,768,000 | 9,600 | 213 | 0.16% | 19.53 |
| 40,000,000 | 9,600 | 260 | 0.16% | 19.53 |
| 3,686,400 | 2,400 | 96 | 0.00% | 78.13 |
| 20,000,000 | 2,400 | 521 | 0.03% | 78.13 |
| 32,768,000 | 2,400 | 853 | 0.04% | 78.13 |

#### 36.6.5.3 IrDA Demodulator

The demodulator is based on the IrDA Receive filter comprised of an 8-bit down counter which is loaded with the value programmed in US_IF. When a falling edge is detected on the RXD pin, the Filter Counter starts counting down at the peripheral clock speed. If a rising edge is detected on the RXD pin, the counter stops and is reloaded with US_IF. If no rising edge is detected when the counter reaches 0, the input of the receiver is driven low during one bit time.

Figure 36-34 illustrates the operations of the IrDA demodulator.

**Figure 36-34. IrDA Demodulator Operations**




The programmed value in the US_IF register must always meet the following criteria:

$$t_{peripheral\ clock} \times (IRDA\_FILTER + 3) < 1.41\ \mu s$$

As the IrDA mode uses the same logic as the ISO7816, note that the FI_DI_RATIO field in US_FIDI must be set to a value higher than 0 in order to make sure IrDA communications operate correctly.

### 36.6.6 RS485 Mode

The USART features the RS485 mode to enable line driver control. While operating in RS485 mode, the USART behaves as though in Asynchronous or Synchronous mode and configuration of all the parameters is possible. The difference is that the RTS pin is driven high when the transmitter is operating. The behavior of the RTS pin is controlled by the TXEMPTY bit. A typical connection of the USART to an RS485 bus is shown in Figure 36-35.

## 36.7 Universal Synchronous Asynchronous Receiver Transmitter (USART) User Interface

**Table 36-15. Register Mapping**

| Offset | Register | Name | Access | Reset |
|---|---|---|---|---|
| 0x0000 | Control Register | US_CR | Write-only | – |
| 0x0004 | Mode Register | US_MR | Read/Write | 0x0 |
| 0x0008 | Interrupt Enable Register | US_IER | Write-only | – |
| 0x000C | Interrupt Disable Register | US_IDR | Write-only | – |
| 0x0010 | Interrupt Mask Register | US_IMR | Read-only | 0x0 |
| 0x0014 | Channel Status Register | US_CSR | Read-only | 0x0 |
| 0x0018 | Receive Holding Register | US_RHR | Read-only | 0x0 |
| 0x001C | Transmit Holding Register | US_THR | Write-only | – |
| 0x0020 | Baud Rate Generator Register | US_BRGR | Read/Write | 0x0 |
| 0x0024 | Receiver Time-out Register | US_RTOR | Read/Write | 0x0 |
| 0x0028 | Transmitter Timeguard Register | US_TTGR | Read/Write | 0x0 |
| 0x002C–0x003C | Reserved | – | – | – |
| 0x0040 | FI DI Ratio Register | US_FIDI | Read/Write | 0x174 |
| 0x0044 | Number of Errors Register | US_NER | Read-only | 0x0 |
| 0x0048 | Reserved | – | – | – |
| 0x004C | IrDA Filter Register | US_IF | Read/Write | 0x0 |
| 0x0050 | Manchester Configuration Register | US_MAN | Read/Write | 0x30011004 |
| 0x0054–0x005C | Reserved | – | – | – |
| 0x0060–0x00E0 | Reserved | – | – | – |
| 0x00E4 | Write Protection Mode Register | US_WPMR | Read/Write | 0x0 |
| 0x00E8 | Write Protection Status Register | US_WPSR | Read-only | 0x0 |
| 0x00EC–0x00FC | Reserved | – | – | – |
| 0x0100–0x0128 | Reserved for PDC Registers | – | – | – |

### 36.7.6 USART Interrupt Enable Register (SPI_MODE)

**Name:** US_IER (SPI_MODE)

**Address:** 0x40024008 (0), 0x40028008 (1)

**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 |
| – | – | – | RXBUFF | TXBUFE | UNRE | TXEMPTY | – |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| – | – | OVRE | ENDTX | ENDRX | – | TXRDY | RXRDY |

This configuration is relevant only if USART_MODE = 0xE or 0xF in the USART Mode Register.

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

0: No effect

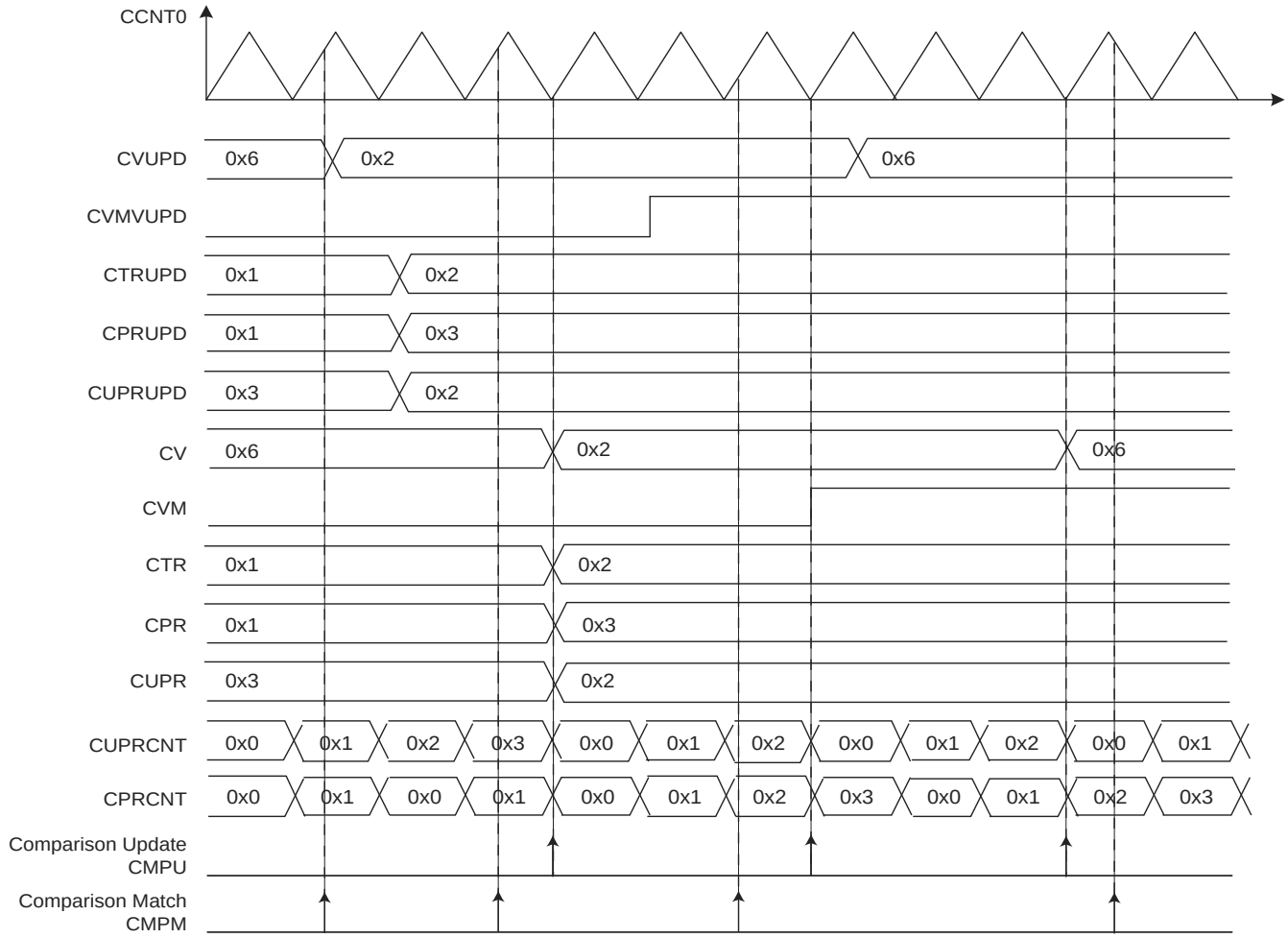
1: Enables the corresponding interrupt.

- **RXRDY: RXRDY Interrupt Enable**
- **TXRDY: TXRDY Interrupt Enable**
- **ENDRX: End of Receive Buffer Interrupt Enable**
- **ENDTX: End of Transmit Buffer Interrupt Enable**
- **OVRE: Overrun Error Interrupt Enable**
- **TXEMPTY: TXEMPTY Interrupt Enable**
- **UNRE: SPI Underrun Error Interrupt Enable**
- **TXBUFE: Transmit Buffer Empty Interrupt Enable**
- **RXBUFF: Receive Buffer Full Interrupt Enable**

## 39.2 Embedded Characteristics

- 4 Channels
- Common Clock Generator Providing Thirteen Different Clocks
  - A Modulo n Counter Providing Eleven Clocks
  - Two Independent Linear Dividers Working on Modulo n Counter Outputs
- Independent Channels
  - Independent 16-bit Counter for Each Channel
  - Independent Complementary Outputs with 12-bit Dead-Time Generator (Also Called Dead-Band or Non-Overlapping Time) for Each Channel
  - Independent Enable Disable Command for Each Channel
  - Independent Clock Selection for Each Channel
  - Independent Period, Duty-Cycle and Dead-Time for Each Channel
  - Independent Double Buffering of Period, Duty-Cycle and Dead-Times for Each Channel
  - Independent Programmable Selection of The Output Waveform Polarity for Each Channel
  - Independent Programmable Center- or Left-aligned Output Waveform for Each Channel
  - Independent Output Override for Each Channel
  - Independent Interrupt for Each Channel, at Each Period for Left-Aligned or Center-Aligned Configuration
- 2 2-bit Gray Up/Down Channels for Stepper Motor Control
- Synchronous Channel Mode
  - Synchronous Channels Share the Same Counter
  - Mode to Update the Synchronous Channels Registers after a Programmable Number of Periods
  - Synchronous Channels Supports Connection of one Peripheral DMA Controller Channel Which Offers Buffer Transfer Without Processor Intervention To Update Duty-Cycle Registers
- 2 Independent Events Lines Intended to Synchronize ADC Conversions
  - Programmable delay for Events Lines to delay ADC measurements
- 8 Comparison Units Intended to Generate Interrupts, Pulses on Event Lines and Peripheral DMA Controller Transfer Requests
- 8 Programmable Fault Inputs Providing an Asynchronous Protection of PWM Outputs
  - 3 User Driven through PIO Inputs
  - PMC Driven when Crystal Oscillator Clock Fails
  - ADC Controller Driven through Configurable Comparison Function
  - Analog Comparator Controller Driven
  - Timer/Counter Driven through Configurable Comparison Function
- Register Write Protection

**Figure 39-15. Comparison Waveform**




### 39.6.4 PWM Event Lines

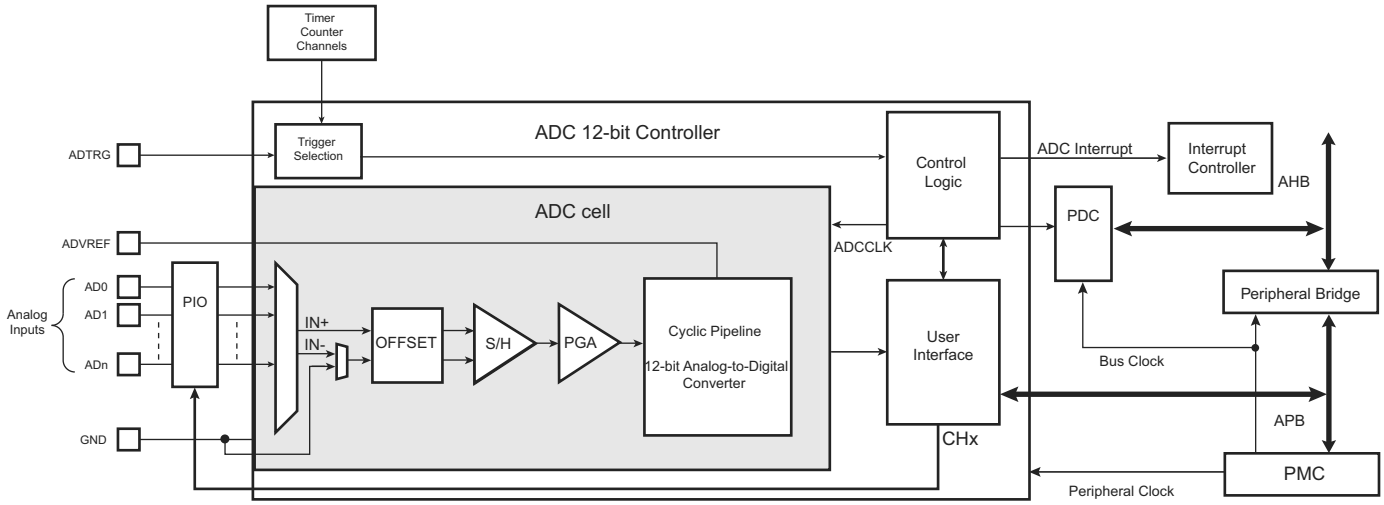
The PWM provides 2 independent event lines intended to trigger actions in other peripherals (e.g., for the Analog-to-Digital Converter (ADC)).

A pulse (one cycle of the peripheral clock) is generated on an event line, when at least one of the selected comparisons is matching. The comparisons can be selected or unselected independently by the CSEL bits in the PWM Event Line x Register (PWM_ELMRx for the Event Line x).

An example of event generation is provided in Figure 39-17.

## 42.3 Block Diagram

**Figure 42-1. Analog-to-Digital Converter Block Diagram**




Note: DMA may be referred to as PDC (Peripheral DMA Controller).

## 42.4 Signal Description

**Table 42-1. ADC Pin Description**

| Pin Name | Description |
|---|---|
| ADVREF | reference voltage |
| AD0–AD15[1] | Analog input channels |
| ADTRG | External trigger |

Note: 1. AD15 is not an actual pin but is internally connected to a temperature sensor.

## 43.7 Digital-to-Analog Converter Controller (DACC) User Interface

**Table 43-3. Register Mapping**

| Offset | Register | Name | Access | Reset |
|---|---|---|---|---|
| 0x00 | Control Register | DACC_CR | Write-only | – |
| 0x04 | Mode Register | DACC_MR | Read/Write | 0x00000000 |
| 0x08–0x0C | Reserved | – | – | – |
| 0x10 | Channel Enable Register | DACC_CHER | Write-only | – |
| 0x14 | Channel Disable Register | DACC_CHDR | Write-only | – |
| 0x18 | Channel Status Register | DACC_CHSR | Read-only | 0x00000000 |
| 0x1C | Reserved | – | – | – |
| 0x20 | Conversion Data Register | DACC_CDR | Write-only | – |
| 0x24 | Interrupt Enable Register | DACC_IER | Write-only | – |
| 0x28 | Interrupt Disable Register | DACC_IDR | Write-only | – |
| 0x2C | Interrupt Mask Register | DACC_IMR | Read-only | 0x00000000 |
| 0x30 | Interrupt Status Register | DACC_ISR | Read-only | 0x00000000 |
| 0x34–0x90 | Reserved | – | – | – |
| 0x94 | Analog Current Register | DACC_ACR | Read/Write | 0x00000000 |
| 0x98–0xE0 | Reserved | – | – | – |
| 0xE4 | Write Protection Mode Register | DACC_WPMR | Read/Write | 0x00000000 |
| 0xE8 | Write Protection Status Register | DACC_WPSR | Read-only | 0x00000000 |
| 0xEC–0xFC | Reserved | – | – | – |

**Figure 44-28. SSC Transmitter, TK as Output and TF as Input**

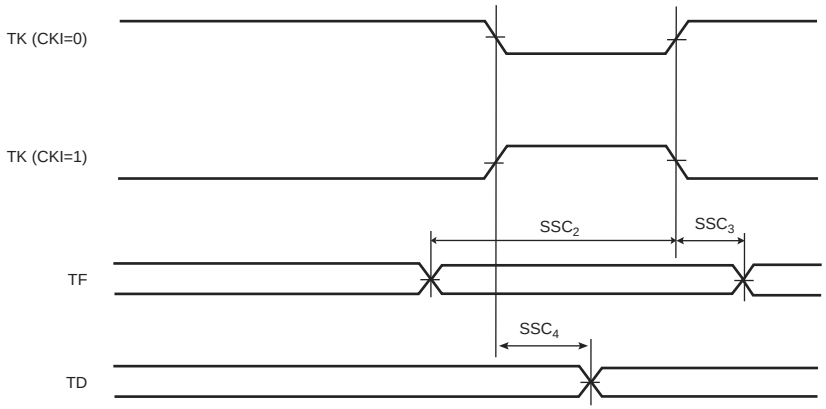



**Figure 44-29. SSC Transmitter, TK and TF as Input**

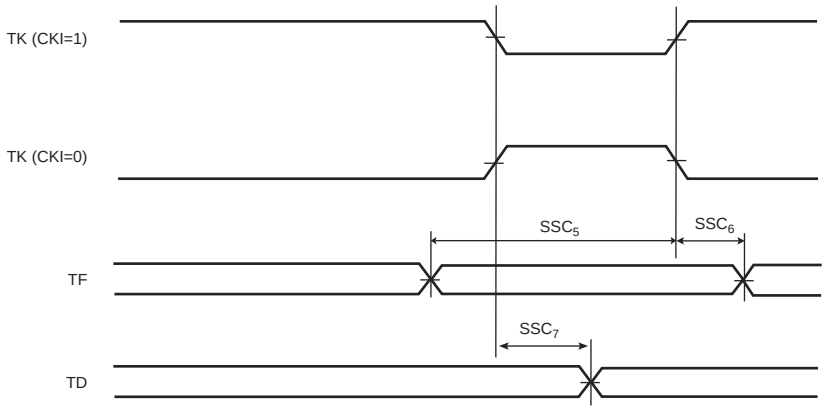



**Figure 44-30. SSC Receiver RK and RF as Input**

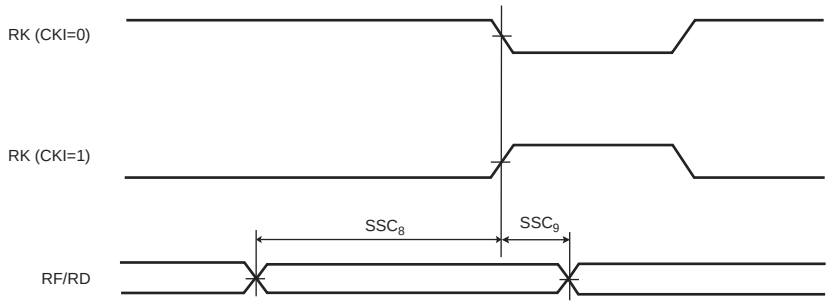


**Figure 44-39. USART SPI Slave Mode: (Mode 0 or 3)**