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	I²C, IrDA, Memory Card, SPI, SSC, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	47
Program Memory Size	1MB (1M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	160K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.6V
Data Converters	A/D 11x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	64-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsam4sd16bb-an

Figure 2-6. SAM4S4/S2 64-pin Version Block Diagram

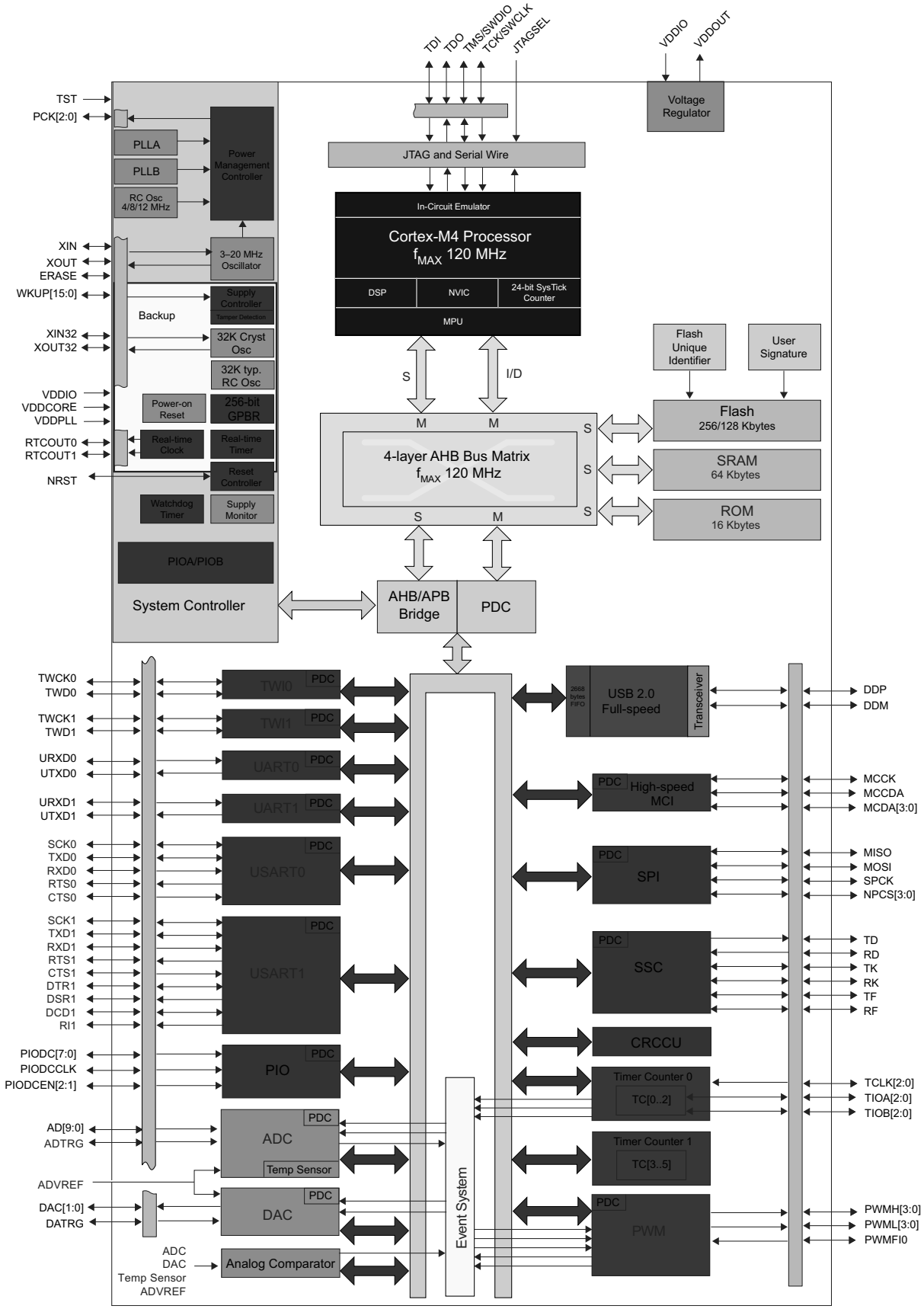


Table 5-1. Low-power Mode Configuration Summary

Mode	SUPC, 32 kHz Osc., RTC, RTT, GPBR, POR (Backup Region)	Regulator	Core Memory Peripherals	Mode Entry	Potential Wake Up Sources	Core at Wake Up	PIO State while in Low-Power Mode	PIO State at Wake Up	Consumption (1) (2)	Wake-up Time(3)
Backup Mode	ON	OFF	OFF (Not powered)	VROFF = 1 or WFE + SLEEPDEEP = 1	WKUP0-15 pins SM alarm RTC alarm RTT alarm	Reset	Previous state saved	PIOA & PIOB & PIOC Inputs with pull ups	1 µA typ(4)	< 1 ms
Wait Mode w/Flash in Standby Mode	ON	ON	Powered (Not clocked)	WAITMODE = 1 + FLPM = 0 or WFE + SLEEPDEEP = 0 + LPM = 1 + FLPM = 0	Any Event from: Fast startup through WKUP0-15 pins RTC alarm RTT alarm USB wake-up	Clocked back	Previous state saved	Unchanged	32.2 µA(5)	< 10 µs
Wait Mode w/Flash in Deep Power Down Mode	ON	ON	Powered (Not clocked)	WAITMODE = 1 + FLPM = 1 or WFE + SLEEPDEEP = 0 + LPM = 1 + FLPM = 1	Any Event from: Fast startup through WKUP0-15 pins RTC alarm RTT alarm USB wake-up	Clocked back	Previous state saved	Unchanged	27.6 µA	< 100 µs
Sleep Mode	ON	ON	Powered(6) (Not clocked)	WFE or WFI + SLEEPDEEP = 0 + LPM = 0	Entry mode =WFI Interrupt Only; Entry mode =WFE Any Enabled Interrupt and/or Any Event from: Fast start-up through WKUP0-15 pins RTC alarm RTT alarm USB wake-up	Clocked back	Previous state saved	Unchanged	(7)	(7)

Notes:
1. The external loads on PIOs are not taken into account in the calculation.
2. Supply Monitor current consumption is not included.
3. When considering wake-up time, the time required to start the PLL is not taken into account. Once started, the device works with the 4/8/12 MHz fast RC oscillator. The user has to add the PLL start-up time if it is needed in the system. The wake-up time is defined as the time taken for wake up until the first instruction is fetched.
4. Total consumption 1 µA typ to 1.8V on VDDIO at 25°C.
5. 20.4 µA on VDDCORE, 32.2 µA for total current consumption.
6. Depends on MCK frequency.
7. Depends on MCK frequency. In this mode, the core is supplied but some peripherals can be clocked.

12.6.6.11 UMULL, UMLAL, SMULL, and SMLAL

Signed and Unsigned Long Multiply, with optional Accumulate, using 32-bit operands and producing a 64-bit result.

Syntax

```
op{cond} RdLo, RdHi, Rn, Rm
```

where:

op is one of:

- UMULL Unsigned Long Multiply.
- UMLAL Unsigned Long Multiply, with Accumulate.
- SMULL Signed Long Multiply.
- SMLAL Signed Long Multiply, with Accumulate.

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

RdHi, RdLo are the destination registers. For UMLAL and SMLAL they also hold the accumulating value.

Rn, Rm are registers holding the operands.

Operation

The UMULL instruction interprets the values from *Rn* and *Rm* as unsigned integers. It multiplies these integers and places the least significant 32 bits of the result in *RdLo*, and the most significant 32 bits of the result in *RdHi*.

The UMLAL instruction interprets the values from *Rn* and *Rm* as unsigned integers. It multiplies these integers, adds the 64-bit result to the 64-bit unsigned integer contained in *RdHi* and *RdLo*, and writes the result back to *RdHi* and *RdLo*.

The SMULL instruction interprets the values from *Rn* and *Rm* as two’s complement signed integers. It multiplies these integers and places the least significant 32 bits of the result in *RdLo*, and the most significant 32 bits of the result in *RdHi*.

The SMLAL instruction interprets the values from *Rn* and *Rm* as two’s complement signed integers. It multiplies these integers, adds the 64-bit result to the 64-bit signed integer contained in *RdHi* and *RdLo*, and writes the result back to *RdHi* and *RdLo*.

Restrictions

In these instructions:

- Do not use SP and do not use PC
- *RdHi* and *RdLo* must be different registers.

Condition Flags

These instructions do not affect the condition code flags.

Examples

```
UMULL       R0, R4, R5, R6   ; Unsigned (R4,R0) = R5 x R6
SMLAL       R4, R5, R3, R8   ; Signed (R5,R4) = (R5,R4) + R3 x R8
```

12.11.2.3 MPU Region Number Register

Name: MPU_RNR

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
REGION							

The MPU_RNR selects which memory region is referenced by the MPU_RBAR and MPU_RASRs.

- **REGION: MPU Region Referenced by the MPU_RBAR and MPU_RASRs**

Indicates the MPU region referenced by the MPU_RBAR and MPU_RASRs.

The MPU supports 8 memory regions, so the permitted values of this field are 0–7.

Normally, the required region number is written to this register before accessing the MPU_RBAR or MPU_RASR. However, the region number can be changed by writing to the MPU_RBAR with the VALID bit set to 1; see “MPU Region Base Address Register” . This write updates the value of the REGION field.

14.5 Reset Controller (RSTC) User Interface

Table 14-1. Register Mapping

Offset	Register	Name	Access	Reset
0x00	Control Register	RSTC_CR	Write-only	–
0x04	Status Register	RSTC_SR	Read-only	0x0000_0000[1]
0x08	Mode Register	RSTC_MR	Read/Write	0x0000 0001

Note: 1. This value assumes that a general reset has been performed, subject to change if other types of reset are generated.

22.5.1 Cache Controller Type Register

Name: CMCC_TYPE

Address: 0x4007C000

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
–		CLSIZE			CSIZE		

7	6	5	4	3	2	1	0
LCKDOWN	WAYNUM		RRP	LRUP	RANDP	GCLK	AP

- **AP: Access Port Access Allowed**

0: Access Port Access is disabled.

1: Access Port Access is enabled.

- **GCLK: Dynamic Clock Gating Supported**

0: Cache controller does not support clock gating.

1: Cache controller uses dynamic clock gating.

- **RANDP: Random Selection Policy Supported**

0: Random victim selection is not supported.

1: Random victim selection is supported.

- **LRUP: Least Recently Used Policy Supported**

0: Least Recently Used Policy is not supported.

1: Least Recently Used Policy is supported.

- **RRP: Random Selection Policy Supported**

0: Random Selection Policy is not supported.

1: Random Selection Policy is supported.

- **WAYNUM: Number of Ways**

Value	Name	Description
0	DMAPPED	Direct Mapped Cache
1	ARCH2WAY	2-way set associative
2	ARCH4WAY	4-way set associative
3	ARCH8WAY	8-way set associative

- **MOSCRCF: Main On-Chip RC Oscillator Frequency Selection**

At startup, the main on-chip RC oscillator frequency is 4 MHz.

Value	Name	Description
0x0	4_MHz	The fast RC oscillator frequency is at 4 MHz (default)
0x1	8_MHz	The fast RC oscillator frequency is at 8 MHz
0x2	12_MHz	The fast RC oscillator frequency is at 12 MHz

Note: MOSCRCF must be changed only if MOSCRCS is set in the PMC_SR. Therefore MOSCRCF and MOSCRCEN cannot be changed at the same time.

- **MOSCXTST: Main Crystal Oscillator Start-up Time**

Specifies the number of slow clock cycles multiplied by 8 for the main crystal oscillator start-up time.

- **KEY: Write Access Password**

Value	Name	Description
0x37	PASSWD	Writing any other value in this field aborts the write operation. Always reads as 0.

- **MOSCSEL: Main Oscillator Selection**

0: The main on-chip RC oscillator is selected.

1: The main crystal oscillator is selected.

- **CFDEN: Clock Failure Detector Enable**

0: The clock failure detector is disabled.

1: The clock failure detector is enabled.

Note:
1. The slow RC oscillator must be enabled when the CFDEN is enabled.
2. The clock failure detection must be enabled only when system clock MCK selects the fast RC oscillator.
3. Then the status register must be read 2 slow clock cycles after enabling.

Before using the SSC transmitter, the PIO controller must be configured to dedicate the SSC transmitter I/O lines to the SSC peripheral mode.

Table 32-2. I/O Lines

Instance	Signal	I/O Line	Peripheral
SSC	RD	PA18	A
SSC	RF	PA20	A
SSC	RK	PA19	A
SSC	TD	PA17	A
SSC	TF	PA15	A
SSC	TK	PA16	A

32.7.2 Power Management

The SSC is not continuously clocked. The SSC interface may be clocked through the Power Management Controller (PMC), therefore the programmer must first configure the PMC to enable the SSC clock.

32.7.3 Interrupt

The SSC interface has an interrupt line connected to the interrupt controller. Handling interrupts requires programming the interrupt controller before configuring the SSC.

All SSC interrupts can be enabled/disabled configuring the SSC Interrupt Mask Register. Each pending and unmasked SSC interrupt will assert the SSC interrupt line. The SSC interrupt service routine can get the interrupt origin by reading the SSC Interrupt Status Register.

Table 32-3. Peripheral IDs

Instance	ID
SSC	22

32.8.5 Frame Sync

The Transmitter and Receiver Frame Sync pins, TF and RF, can be programmed to generate different kinds of frame synchronization signals. The Frame Sync Output Selection (FSOS) field in the Receive Frame Mode Register (SSC_RFMR) and in the Transmit Frame Mode Register (SSC_TFMR) are used to select the required waveform.

- Programmable low or high levels during data transfer are supported.
- Programmable high levels before the start of data transfers or toggling are also supported.

If a pulse waveform is selected, the Frame Sync Length (FSLEN) field in SSC_RFMR and SSC_TFMR programs the length of the pulse, from 1 bit time up to 256 bit times.

The periodicity of the Receive and Transmit Frame Sync pulse output can be programmed through the Period Divider Selection (PERIOD) field in SSC_RCMR and SSC_TCMR.

32.8.5.1 Frame Sync Data

Frame Sync Data transmits or receives a specific tag during the Frame Sync signal.

During the Frame Sync signal, the Receiver can sample the RD line and store the data in the Receive Sync Holding Register and the transmitter can transfer Transmit Sync Holding Register in the shift register. The data length to be sampled/shifted out during the Frame Sync signal is programmed by the FSLEN field in SSC_RFMR/SSC_TFMR and has a maximum value of 256.

Concerning the Receive Frame Sync Data operation, if the Frame Sync Length is equal to or lower than the delay between the start event and the actual data reception, the data sampling operation is performed in the Receive Sync Holding Register through the receive shift register.

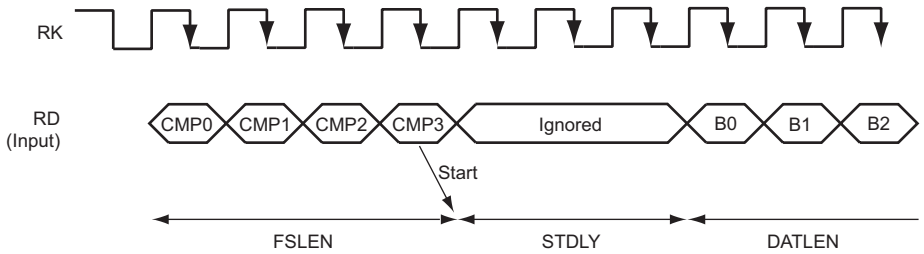
The Transmit Frame Sync Operation is performed by the transmitter only if the bit Frame Sync Data Enable (FSDEN) in SSC_TFMR is set. If the Frame Sync length is equal to or lower than the delay between the start event and the actual data transmission, the normal transmission has priority and the data contained in the Transmit Sync Holding Register is transferred in the Transmit Register, then shifted out.

32.8.5.2 Frame Sync Edge Detection

The Frame Sync Edge detection is programmed by the FSEDGE field in SSC_RFMR/SSC_TFMR. This sets the corresponding flags RXSYN/TXSYN in the SSC Status Register (SSC_SR) on frame synchro edge detection (signals RF/TF).

32.8.6 Receive Compare Modes

Figure 32-15. Receive Compare Modes



32.8.6.1 Compare Functions

The length of the comparison patterns (Compare 0, Compare 1) and thus the number of bits they are compared to is defined by FSLEN, but with a maximum value of 256 bits. Comparison is always done by comparing the last bits received with the comparison pattern. Compare 0 can be one start event of the Receiver. In this case, the receiver compares at each new sample the last bits received at the Compare 0 pattern contained in the Compare 0 Register (SSC_RC0R). When this start event is selected, the user can program the Receiver to start a new data

33.6.3 Interrupt

The SPI interface has an interrupt line connected to the interrupt controller. Handling the SPI interrupt requires programming the interrupt controller before configuring the SPI.

Table 33-3. Peripheral IDs

Instance	ID
SPI	21

33.6.4 Peripheral DMA Controller (PDC)

The SPI interface can be used in conjunction with the PDC in order to reduce processor overhead. For a full description of the PDC, refer to the corresponding section in the full datasheet.

34.8.3 TWI Slave Mode Register

Name: TWI_SMR

Address: 0x40018008 (0), 0x4001C008 (1)

Access: Read/Write

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

23	22	21	20	19	18	17	16
–	SADR						

15	14	13	12	11	10	9	8
–	–	–	–	–	–		

7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

- **SADR: Slave Address**

The slave device address is used in Slave mode in order to be accessed by master devices in Read or Write mode.

SADR must be programmed before enabling the Slave mode or after a general call. Writes at other times have no effect.

The USART cannot operate concurrently in both Receiver and Transmitter modes as the communication is unidirectional at a time. It has to be configured according to the required mode by enabling or disabling either the receiver or the transmitter as desired. Enabling both the receiver and the transmitter at the same time in ISO7816 mode may lead to unpredictable results.

The ISO7816 specification defines an inverse transmission format. Data bits of the character must be transmitted on the I/O line at their negative value.

36.6.4.2 Protocol T = 0

In T = 0 protocol, a character is made up of one start bit, eight data bits, one parity bit and one guard time, which lasts two bit times. The transmitter shifts out the bits and does not drive the I/O line during the guard time.

If no parity error is detected, the I/O line remains at 1 during the guard time and the transmitter can continue with the transmission of the next character, as shown in Figure 36-30.

If a parity error is detected by the receiver, it drives the I/O line to 0 during the guard time, as shown in Figure 36-31. This error bit is also named NACK, for Non Acknowledge. In this case, the character lasts 1 bit time more, as the guard time length is the same and is added to the error bit time which lasts 1 bit time.

When the USART is the receiver and it detects an error, it does not load the erroneous character in the Receive Holding register (US_RHR). It appropriately sets the PARE bit in the Status register (US_SR) so that the software can handle the error.

Figure 36-30. T = 0 Protocol without Parity Error

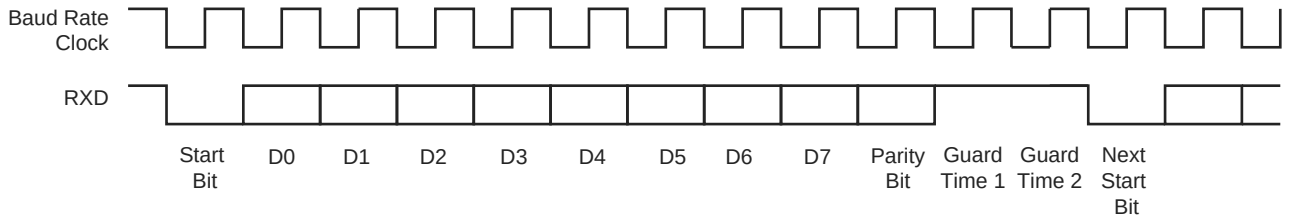
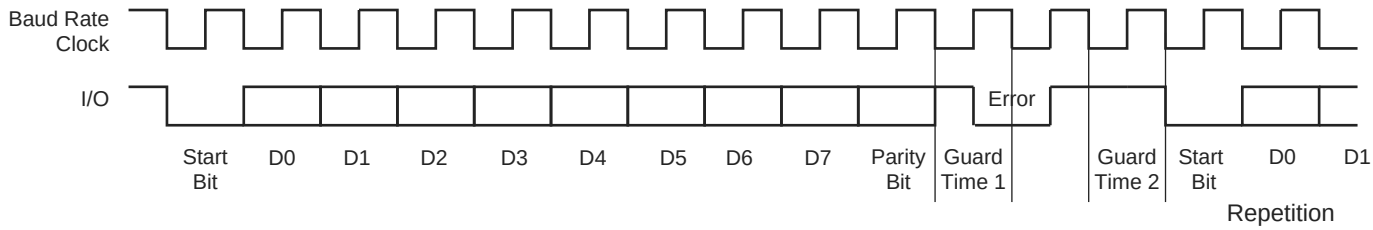


Figure 36-31. T = 0 Protocol with Parity Error



Receive Error Counter

The USART receiver also records the total number of errors. This can be read in the Number of Error (US_NER) register. The NB_ERRORS field can record up to 255 errors. Reading US_NER automatically clears the NB_ERRORS field.

Receive NACK Inhibit

The USART can also be configured to inhibit an error. This can be achieved by setting the INACK bit in US_MR. If INACK is to 1, no error signal is driven on the I/O line even if a parity bit is detected.

Moreover, if INACK is set, the erroneous received character is stored in the Receive Holding register, as if no error occurred and the RXRDY bit does rise.

- If the divided peripheral clock is selected, the value programmed in CD must be even to ensure a 50:50 mark/space ratio on the SCK pin, this value can be odd if the peripheral clock is selected.

In SPI Slave mode:

- The external clock (SCK) selection is forced regardless of the value of the USCLKS field in the US_MR. Likewise, the value written in US_BRGR has no effect, because the clock is provided directly by the signal on the USART SCK pin.
- To obtain correct behavior of the receiver and the transmitter, the external clock (SCK) frequency must be at least 6 times lower than the system clock.

36.6.8.3 Data Transfer

Up to nine data bits are successively shifted out on the TXD pin at each rising or falling edge (depending of CPOL and CPHA) of the programmed serial clock. There is no Start bit, no Parity bit and no Stop bit.

The number of data bits is selected by the CHRL field and the MODE 9 bit in the US_MR. The nine bits are selected by setting the MODE 9 bit regardless of the CHRL field. The MSB data bit is always sent first in SPI mode (Master or Slave).

Four combinations of polarity and phase are available for data transfers. The clock polarity is programmed with the CPOL bit in the US_MR. The clock phase is programmed with the CPHA bit. These two parameters determine the edges of the clock signal upon which data is driven and sampled. Each of the two parameters has two possible states, resulting in four possible combinations that are incompatible with one another. Thus, a master/slave pair must use the same parameter pair values to communicate. If multiple slaves are used and fixed in different configurations, the master must reconfigure itself each time it needs to communicate with a different slave.

Table 36-14. SPI Bus Protocol Mode

SPI Bus Protocol Mode	CPOL	CPHA
0	0	1
1	0	0
2	1	1
3	1	0

38.14.5 HSMCI Argument Register

Name: HSMCI_ARGR

Address: 0x40000010

Access: Read/Write

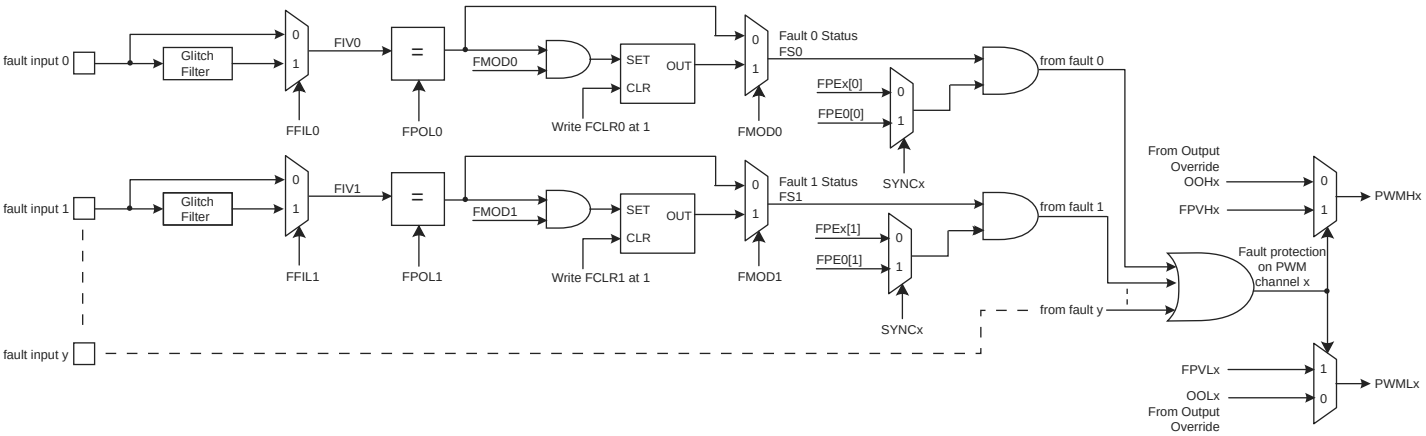
31	30	29	28	27	26	25	24
ARG							
23	22	21	20	19	18	17	16
ARG							
15	14	13	12	11	10	9	8
ARG							
7	6	5	4	3	2	1	0
ARG							

- **ARG: Command Argument**

39.6.2.6 Fault Protection

8 inputs provide fault protection which can force any of the PWM output pairs to a programmable value. This mechanism has priority over output overriding.

Figure 39-9. Fault Protection



The polarity level of the fault inputs is configured by the FPOL field in the PWM Fault Mode Register (PWM_FMR). For fault inputs coming from internal peripherals such as ADC or Timer Counter, the polarity level must be FPOL = 1. For fault inputs coming from external GPIO pins the polarity level depends on the user's implementation.

The configuration of the Fault Activation mode (FMOD field in PWMC_FMR) depends on the peripheral generating the fault. If the corresponding peripheral does not have “Fault Clear” management, then the FMOD configuration to use must be FMOD = 1, to avoid spurious fault detection. Refer to the corresponding peripheral documentation for details on handling fault generation.

Fault inputs may or may not be glitch-filtered depending on the FFIL field in the PWM_FMR. When the filter is activated, glitches on fault inputs with a width inferior to the PWM peripheral clock period are rejected.

A fault becomes active as soon as its corresponding fault input has a transition to the programmed polarity level. If the corresponding bit FMOD is set to ‘0’ in the PWM_FMR, the fault remains active as long as the fault input is at this polarity level. If the corresponding FMOD field is set to ‘1’, the fault remains active until the fault input is no longer at this polarity level and until it is cleared by writing the corresponding bit FCLR in the PWM Fault Clear Register (PWM_FCR). In the PWM Fault Status Register (PWM_FSR), the field FIV indicates the current level of the fault inputs and the field FIS indicates whether a fault is currently active.

Each fault can be taken into account or not by the fault protection mechanism in each channel. To be taken into account in the channel x, the fault y must be enabled by the bit FPEx[y] in the PWM Fault Protection Enable registers (PWM_FPE1). However, synchronous channels (see Section 39.6.2.7 “Synchronous Channels”) do not use their own fault enable bits, but those of the channel 0 (bits FPE0[y]).

The fault protection on a channel is triggered when this channel is enabled and when any one of the faults that are enabled for this channel is active. It can be triggered even if the PWM peripheral clock is not running but only by a fault input that is not glitch-filtered.

When the fault protection is triggered on a channel, the fault protection mechanism resets the counter of this channel and forces the channel outputs to the values defined by the fields FPVHx and FPVLx in the PWM Fault Protection Value Register (PWM_FPV) . The output forcing is made asynchronously to the channel counter.

CAUTION:

- To prevent any unexpected activation of the status flag FSy in the PWM_FSR, the FMODy bit can be set to ‘1’ only if the FPOLy bit has been previously configured to its final value.

43.7.7 DACC Interrupt Enable Register

Name: DACC_IER

Address: 0x4003C024

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
–	–	–	–	TXBUFE	ENDTX	EOC	TXRDY

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

0: No effect

1: Enables the corresponding interrupt

- **TXRDY: Transmit Ready Interrupt Enable**
- **EOC: End of Conversion Interrupt Enable**
- **ENDTX: End of Transmit Buffer Interrupt Enable**
- **TXBUFE: Transmit Buffer Empty Interrupt Enable**

43.7.8 DACC Interrupt Disable Register

Name: DACC_IDR

Address: 0x4003C028

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
–	–	–	–	TXBUFE	ENDTX	EOC	TXRDY

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

0: No effect

1: Disables the corresponding interrupt

- **TXRDY: Transmit Ready Interrupt Disable.**
- **EOC: End of Conversion Interrupt Disable**
- **ENDTX: End of Transmit Buffer Interrupt Disable**
- **TXBUFE: Transmit Buffer Empty Interrupt Disable**

44.8.4 ADC Transfer Function

The first operation of the ADC is a sampling function relative to a common mode voltage. The common mode voltage (V_{CM}) is equal to $V_{ADVREF}/2$ when the bits OFFx = 1, in Differential and in Single-ended mode. When the bits OFFx = 0, sampling is done versus $V_{ADVREF}/4$ for gain = 2, and $V_{ADVREF}/8$ for gain = 4, in Single-ended mode only.

The code in ADC_CDRx is a 12-bit positive integer.

44.8.4.1 Differential Mode

A differential input voltage $V_I = V_{I+} - V_{I-}$ can be applied between two selected differential pins, e.g., AD0 and AD1. The ideal code Ci is calculated by using the following formula and rounding the result to the nearest positive integer.

$$Ci = \frac{4096}{V_{ADVREF}} \times V_I \times Gain + 2047$$

Table 44-42 is a computation example for the above formula, where V_{ADVREF} = 3V.

Table 44-42. Input Voltage Values in Differential Mode

Ci	Gain = 0.5	Gain = 1	Gain = 2
0	-3	-1.5	-0.75
2047	0	0	0
4095	3	1.5	0.75

44.8.4.2 Single-ended Mode

A single input voltage V_I can be applied to selected pins, e.g., AD0 or AD1. The ideal code Ci is calculated by using the following formula and rounding the result to the nearest positive integer.

The single-ended ideal code conversion formula for OFFx = 1 is:

$$Ci = \frac{4096}{V_{ADVREF}} \times \left(V_I - \frac{V_{ADVREF}}{2}\right) \times Gain + 2047$$

Table 44-43 is a computation example for the above formula, where V_{ADVREF} = 3V.

Table 44-43. Input Voltage Values in Single-ended Mode, OFFx = 1

Ci	Gain = 1	Gain = 2	Gain = 4
0	0	0.75	1.125
2047	1.5	1.5	1.5
4095	3	2.25	1.875

The single-ended ideal code conversion formula for OFFx = 0 is:

$$Ci = V_I \times Gain \times \frac{4096}{V_{ADVREF}} - 1$$

Table 49-6. SAM4S Datasheet Rev. 11100F 29-Jan-14 Revision History

Doc. Date	Changes
29-Jan-14	Update to add SAM4S4 and SAM4S2 devices.
	“Description” 48-pin package option added at end of section.
	“Features” ”System”: Added bullet on tamper detection and anti-tampering feature “Packages” : Modified dimensions of 64-ball WLCSP package used for SAM4S16/S8. Added 48-lead package options.
	Section 1. “Configuration Summary” Table 1-2, “Configuration Summary for SAM4S8/S4/S2 Devices” added.
	Section 2. “Block Diagram” Figure 2-5, “SAM4S4/S2 100-pin Version Block Diagram”, Figure 2-6, “SAM4S4/S2 64-pin Version Block Diagram” and Figure 2-7, “SAM4S4/S2 48-pin Version Block Diagram” added.
	Section 3. “Signal Description” Added fault input number and added comment for “PWMFI0–2” (“PWM Fault Input”) signal in Table 3-1 ”Signal Description List”: Available on SAM4S4/S2 only.
	Section 4. “Package and Pinout” Added Table 4-6, “SAM4S4/S2 64-ball WLCSP Pinout”. Added Section 4.3 “48-lead Packages and Pinouts”.
	Section 5. “Power Considerations” Figure 5-2, “Single Supply”: Modified note with restrictions after the figure. Figure 5-3, “Core Externally Supplied”: Changed voltage for ADC, DAC, Analog Comparator Supply. Modified note with restrictions after the figure. Removed redundant Figure 5-4. Wake-up Source.
	Section 6. “Input/Output Lines” Added Section 6.6 “Anti-tamper Pins/Low-power Tamper Detection”.
	Section 8. “Memories” Section 8.1.1 “Internal SRAM”, Section 8.1.3 “Embedded Flash”, Table 8-2, “Lock Bit Number”, Section 8.1.3.11 “GPNVM Bits” Added SAM4S4 and SAM4S2 devices.
	Section 10. “System Controller” Removed redundant Figure 10-1. System Controller Block Diagram.
	Section 11. “Peripherals” Table 11-2, “Multiplexing on PIO Controller A (PIOA)” updated with Peripheral D information and note on PWMFI signals available for SAM4S4/S2 only.