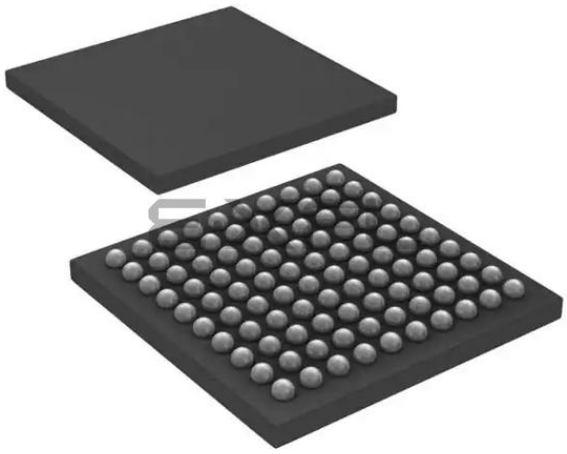


Welcome to E-XFL.COM

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	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 16x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	100-VFBGA
Supplier Device Package	100-VFBGA (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsam4s8cb-cfnr

12.6.5.20 UHSUB16 and UHSUB8

Unsigned Halving Subtract 16 and Unsigned Halving Subtract 8

Syntax

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

where:

op is any of:

UHSUB16 Performs two unsigned 16-bit integer additions, halves the results, and writes the results to the destination register.

UHSUB8 Performs four unsigned 8-bit integer additions, halves the results, and writes the results to the destination register.

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

Rd is the destination register.

Rn is the first register holding the operand.

Rm is the second register holding the operand.

Operation

Use these instructions to add 16-bit and 8-bit data and then to halve the result before writing the result to the destination register:

The UHSUB16 instruction:

1. Subtracts each halfword of the second operand from the corresponding halfword of the first operand.
2. Shuffles each halfword result to the right by one bit, halving the data.
3. Writes each unsigned halfword result to the corresponding halfwords in the destination register.

The UHSUB8 instruction:

1. Subtracts each byte of second operand from the corresponding byte of the first operand.
2. Shuffles each byte result by one bit to the right, halving the data.
3. Writes the unsigned byte results to the corresponding byte of the destination register.

Restrictions

Do not use SP and do not use PC.

Condition Flags

These instructions do not change the flags.

Examples

```
UHSUB16 R1, R0      ; Subtracts halfwords in R0 from corresponding halfword of
                    ; R1 and writes halved result to corresponding halfword in R1
UHSUB8  R4, R0, R5  ; Subtracts bytes of R5 from corresponding byte in R0 and
                    ; writes halved result to corresponding byte in R4.
```

12.6.5.21 SEL

Select Bytes. Selects each byte of its result from either its first operand or its second operand, according to the values of the GE flags.

Syntax

```
SEL{<c>}{<q>} {<Rd>,} <Rn>, <Rm>
```

where:

c, q are standard assembler syntax fields.

- **UNSTKERR: Bus Fault on Unstacking for a Return From Exception**

This is part of “BFSR: Bus Fault Status Subregister” .

0: No unstacking fault.

1: Unstack for an exception return has caused one or more bus faults.

This fault is chained to the handler. This means that when the processor sets this bit to 1, the original return stack is still present. The processor does not adjust the SP from the failing return, does not performed a new save, and does not write a fault address to the BFAR.

- **STKERR: Bus Fault on Stacking for Exception Entry**

This is part of “BFSR: Bus Fault Status Subregister” .

0: No stacking fault.

1: Stacking for an exception entry has caused one or more bus faults.

When the processor sets this bit to 1, the SP is still adjusted but the values in the context area on the stack might be incorrect. The processor does not write a fault address to the SCB_BFAR.

- **BFARVALID: Bus Fault Address Register (BFAR) Valid flag**

This is part of “BFSR: Bus Fault Status Subregister” .

0: The value in SCB_BFAR is not a valid fault address.

1: SCB_BFAR holds a valid fault address.

The processor sets this bit to 1 after a bus fault where the address is known. Other faults can set this bit to 0, such as a memory management fault occurring later.

If a bus fault occurs and is escalated to a hard fault because of priority, the hard fault handler must set this bit to 0. This prevents problems if returning to a stacked active bus fault handler whose SCB_BFAR value has been overwritten.

- **UNDEFINSTR: Undefined Instruction Usage Fault**

This is part of “UFSR: Usage Fault Status Subregister” .

0: No undefined instruction usage fault.

1: The processor has attempted to execute an undefined instruction.

When this bit is set to 1, the PC value stacked for the exception return points to the undefined instruction.

An undefined instruction is an instruction that the processor cannot decode.

- **INVSTATE: Invalid State Usage Fault**

This is part of “UFSR: Usage Fault Status Subregister” .

0: No invalid state usage fault.

1: The processor has attempted to execute an instruction that makes illegal use of the EPSR.

When this bit is set to 1, the PC value stacked for the exception return points to the instruction that attempted the illegal use of the EPSR.

This bit is not set to 1 if an undefined instruction uses the EPSR.

- **INVPC: Invalid PC Load Usage Fault**

This is part of “UFSR: Usage Fault Status Subregister” . It is caused by an invalid PC load by EXC_RETURN:

0: No invalid PC load usage fault.

1: The processor has attempted an illegal load of EXC_RETURN to the PC, as a result of an invalid context, or an invalid EXC_RETURN value.

19.3.1 General Purpose Backup Register x

Name: SYS_GPBRx

Address: 0x400E1490

Access: Read/Write

31	30	29	28	27	26	25	24
GPBR_VALUE							

23	22	21	20	19	18	17	16
GPBR_VALUE							

15	14	13	12	11	10	9	8
GPBR_VALUE							

7	6	5	4	3	2	1	0
GPBR_VALUE							

These registers are reset at first power-up and on each loss of VVDIO.

- **GPBR_VALUE: Value of GPBR x**

If a Tamper event has been detected, it is not possible to write GPBR_VALUE as long as the LPDBCS0 or LPDBCS1 flag has not been cleared in the Supply Controller Status Register (SUPC_SR).

29.17.16PMC Status Register

Name: PMC_SR

Address: 0x400E0468

Access: Read-only

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

23	22	21	20	19	18	17	16
–	–	–	FOS	CFDS	CFDEV	MOSCRCS	MOSCSELS

15	14	13	12	11	10	9	8
–	–	–	–	–	PCKRDY2	PCKRDY1	PCKRDY0

7	6	5	4	3	2	1	0
OSCSELS	–	–	–	MCKRDY	LOCKB	LOCKA	MOSCXTS

- **MOSCXTS: Main Crystal Oscillator Status**

0: Main crystal oscillator is not stabilized.

1: Main crystal oscillator is stabilized.

- **LOCKA: PLLA Lock Status**

0: PLLA is not locked

1: PLLA is locked.

- **LOCKB: PLLB Lock Status**

0: PLLB is not locked

1: PLLB is locked.

- **MCKRDY: Master Clock Status**

0: Master Clock is not ready.

1: Master Clock is ready.

- **OSCSELS: Slow Clock Oscillator Selection**

0: Internal slow clock RC oscillator is selected.

1: External slow clock 32 kHz oscillator is selected.

- **PCKRDYx: Programmable Clock Ready Status**

0: Programmable Clock x is not ready.

1: Programmable Clock x is ready.

31.6.14 PIO Interrupt Enable Register

Name: PIO_IER

Address: 0x400E0E40 (PIOA), 0x400E1040 (PIOB), 0x400E1240 (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: Input Change Interrupt Enable**

0: No effect.

1: Enables the input change interrupt on the I/O line.

- **FSOS: Receive Frame Sync Output Selection**

Value	Name	Description
0	NONE	None, RF pin is an input
1	NEGATIVE	Negative Pulse, RF pin is an output
2	POSITIVE	Positive Pulse, RF pin is an output
3	LOW	Driven Low during data transfer, RF pin is an output
4	HIGH	Driven High during data transfer, RF pin is an output
5	TOGGLING	Toggling at each start of data transfer, RF pin is an output

- **FSEDGE: Frame Sync Edge Detection**

Determines which edge on Frame Sync will generate the interrupt RXSYN in the SSC Status Register.

Value	Name	Description
0	POSITIVE	Positive Edge Detection
1	NEGATIVE	Negative Edge Detection

- **FSLEN_EXT: FSLEN Field Extension**

Extends FSLEN field. For details, refer to FSLEN bit description on page 664.

- **START: Transmit Start Selection**

Value	Name	Description
0	CONTINUOUS	Continuous, as soon as a word is written in the SSC_THR (if Transmit is enabled), and immediately after the end of transfer of the previous data
1	RECEIVE	Receive start
2	TF_LOW	Detection of a low level on TF signal
3	TF_HIGH	Detection of a high level on TF signal
4	TF_FALLING	Detection of a falling edge on TF signal
5	TF_RISING	Detection of a rising edge on TF signal
6	TF_LEVEL	Detection of any level change on TF signal
7	TF_EDGE	Detection of any edge on TF signal

- **STTDLY: Transmit Start Delay**

If STTDLY is not 0, a delay of STTDLY clock cycles is inserted between the start event and the actual start of transmission of data. When the Transmitter is programmed to start synchronously with the Receiver, the delay is also applied.

Note: Note: STTDLY must be set carefully. If STTDLY is too short in respect to TAG (Transmit Sync Data) emission, data is emitted instead of the end of TAG.

- **PERIOD: Transmit Period Divider Selection**

This field selects the divider to apply to the selected Transmit Clock to generate a new Frame Sync Signal. If 0, no period signal is generated. If not 0, a period signal is generated at each 2 × (PERIOD + 1) Transmit Clock.

Clock Synchronization/Stretching

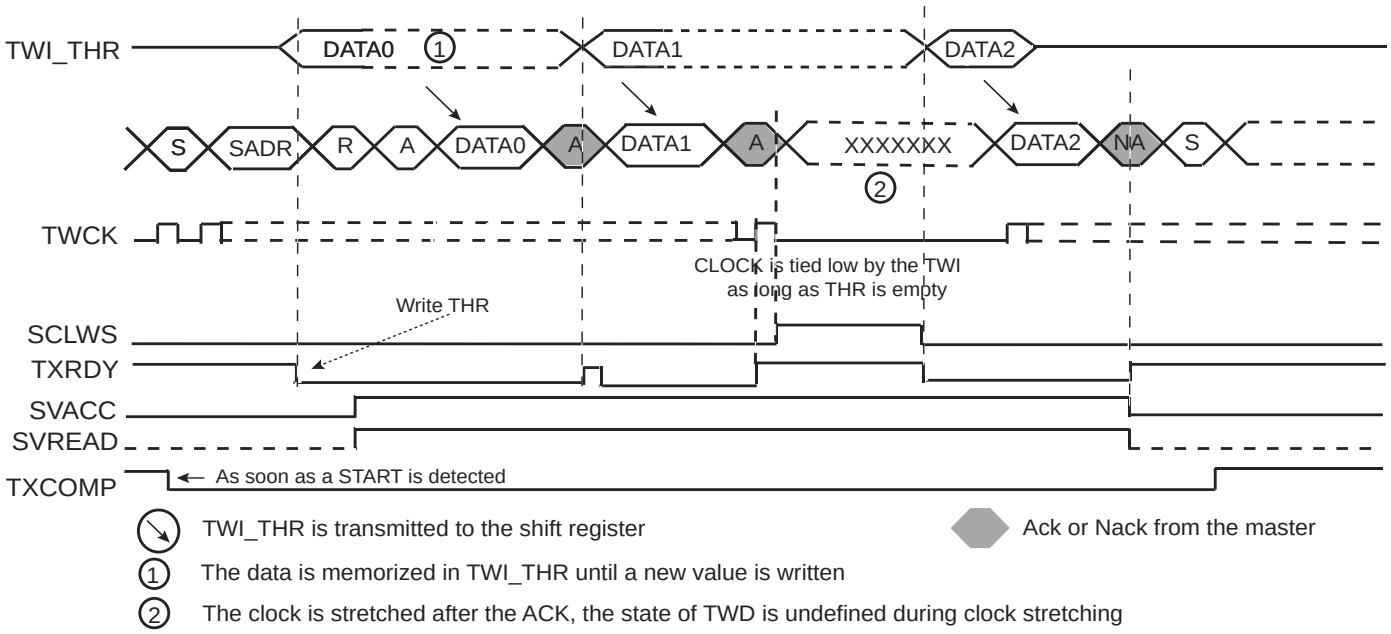
In both Read and Write modes, it may occur that TWI_THR/TWI_RHR buffer is not filled /emptied before transmission/reception of a new character. In this case, to avoid sending/receiving undesired data, a clock stretching/synchronization mechanism is implemented.

Clock Stretching in Read Mode

The clock is tied low during the acknowledge phase if the internal shifter is empty and if a STOP or REPEATED START condition was not detected. It is tied low until the internal shifter is loaded.

Figure 34-27 describes clock stretching in Read mode.

Figure 34-27. Clock Stretching in Read Mode



Notes:
1. TXRDY is reset when data has been written in the TWI_THR to the internal shifter and set when this data has been acknowledged or non acknowledged.
2. At the end of the read sequence, TXCOMP is set after a STOP or after a REPEATED_START + an address different from SADR.
3. SCLWS is automatically set when the clock stretching mechanism is started.

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**

One of the POSEN or SPEEDEN bits must be also enabled.

- **POSEN: Position Enabled**

0: Disable position.

1: Enables the position measure on channel 0 and 1.

- **SPEEDEN: Speed Enabled**

0: Disabled.

1: Enables the speed measure on channel 0, the time base being provided by channel 2.

- **QDTRANS: Quadrature Decoding Transparent**

0: Full quadrature decoding logic is active (direction change detected).

1: Quadrature decoding logic is inactive (direction change inactive) but input filtering and edge detection are performed.

- **EDGPHA: Edge on PHA Count Mode**

0: Edges are detected on PHA only.

1: Edges are detected on both PHA and PHB.

- **INVA: Inverted PHA**

0: PHA (TIOA0) is directly driving the QDEC.

1: PHA is inverted before driving the QDEC.

- **INVB: Inverted PHB**

0: PHB (TIOB0) is directly driving the QDEC.

1: PHB is inverted before driving the QDEC.

- **INVIDX: Inverted Index**

0: IDX (TIOA1) is directly driving the QDEC.

1: IDX is inverted before driving the QDEC.

- **SWAP: Swap PHA and PHB**

0: No swap between PHA and PHB.

1: Swap PHA and PHB internally, prior to driving the QDEC.

- **IDXPHB: Index Pin is PHB Pin**

0: IDX pin of the rotary sensor must drive TIOA1.

1: IDX pin of the rotary sensor must drive TIOB0.

- **MAXFILT: Maximum Filter**

1–63: Defines the filtering capabilities.

Pulses with a period shorter than MAXFILT+1 peripheral clock cycles are discarded.

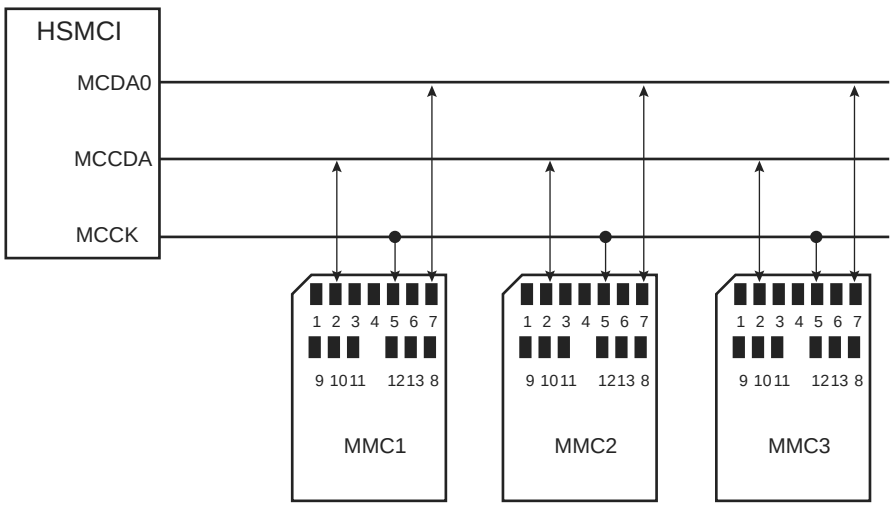
The High Speed MultiMedia Card communication is based on a 13-pin serial bus interface. It has three communication lines and four supply lines.

Table 38-4. Bus Topology

Pin Number	Name	Type[1]	Description	HSMCI Pin Name[2] (Slot z)
1	DAT[3]	I/O/PP	Data	MCDz3
2	CMD	I/O/PP/OD	Command/response	MCCDz
3	VSS1	S	Supply voltage ground	VSS
4	VDD	S	Supply voltage	VDD
5	CLK	I/O	Clock	MCCK
6	VSS2	S	Supply voltage ground	VSS
7	DAT[0]	I/O/PP	Data 0	MCDz0
8	DAT[1]	I/O/PP	Data 1	MCDz1
9	DAT[2]	I/O/PP	Data 2	MCDz2
10	DAT[4]	I/O/PP	Data 4	MCDz4
11	DAT[5]	I/O/PP	Data 5	MCDz5
12	DAT[6]	I/O/PP	Data 6	MCDz6
13	DAT[7]	I/O/PP	Data 7	MCDz7

Notes: 1. I: Input, O: Output, PP: Push/Pull, OD: Open Drain.
2. When several HSMCI (x HSMCI) are embedded in a product, MCCK refers to HSMCIx_CK, MCCDA to HSMCIx_CDA, MCDAy to HSMCIx_DAy.

Figure 38-4. MMC Bus Connections (One Slot)



Note: When several HSMCI (x HSMCI) are embedded in a product, MCCK refers to HSMCIx_CK, MCCDA to HSMCIx_CDA MCDAy to HSMCIx_DAy.

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

- To prevent any unexpected activation of the Fault Protection on the channel x, the bit FPEx[y] can be set to ‘1’ only if the FPOLy bit has been previously configured to its final value.

If a comparison unit is enabled (see Section 39.6.3 “PWM Comparison Units”) and if a fault is triggered in the channel 0, then the comparison cannot match.

As soon as the fault protection is triggered on a channel, an interrupt (different from the interrupt generated at the end of the PWM period) can be generated but only if it is enabled and not masked. The interrupt is reset by reading the interrupt status register, even if the fault which has caused the trigger of the fault protection is kept active.

39.6.2.7 Synchronous Channels

Some channels can be linked together as synchronous channels. They have the same source clock, the same period, the same alignment and are started together. In this way, their counters are synchronized together.

The synchronous channels are defined by the SYNCx bits in the PWM Sync Channels Mode Register (PWM_SCM). Only one group of synchronous channels is allowed.

When a channel is defined as a synchronous channel, the channel 0 is also automatically defined as a synchronous channel. This is because the channel 0 counter configuration is used by all the synchronous channels.

If a channel x is defined as a synchronous channel, the fields/bits for the channel 0 are used instead of those of channel x:

- CPRE in PWM_CMR0 instead of CPRE in PWM_CMRx (same source clock)
- CPRD in PWM_CPRD0 instead of CPRD in PWM_CPRDx (same period)
- CALG in PWM_CMR0 instead of CALG in PWM_CMRx (same alignment)

Modifying the fields CPRE, CPRD and CALG of for channels with index greater than 0 has no effect on output waveforms.

Because counters of synchronous channels must start at the same time, they are all enabled together by enabling the channel 0 (by the CHID0 bit in PWM_ENA register). In the same way, they are all disabled together by disabling channel 0 (by the CHID0 bit in PWM_DIS register). However, a synchronous channel x different from channel 0 can be enabled or disabled independently from others (by the CHIDx bit in PWM_ENA and PWM_DIS registers).

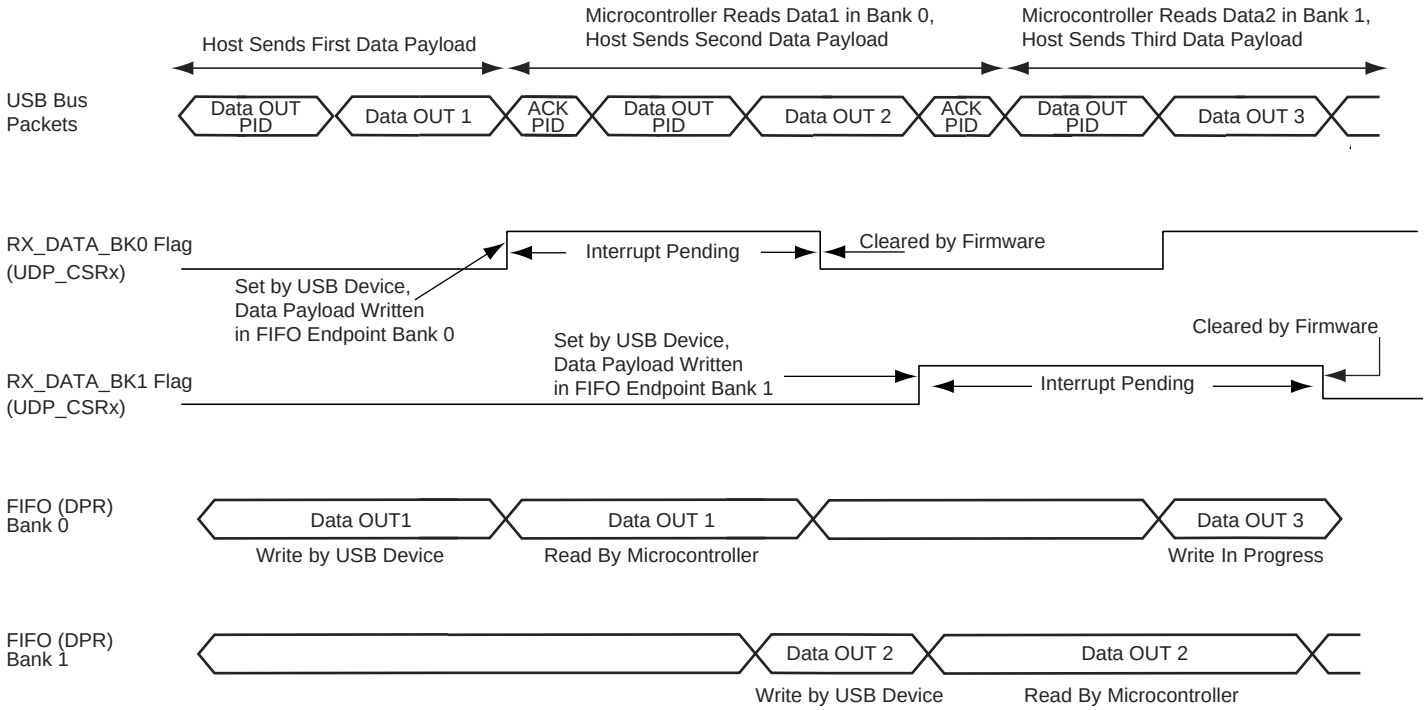
Defining a channel as a synchronous channel while it is an asynchronous channel (by writing the bit SYNCx to ‘1’ while it was at ‘0’) is allowed only if the channel is disabled at this time (CHIDx = 0 in PWM_SR). In the same way, defining a channel as an asynchronous channel while it is a synchronous channel (by writing the SYNCx bit to ‘0’ while it was ‘1’) is allowed only if the channel is disabled at this time.

The UPDM field (Update Mode) in the PWM_SCM register selects one of the three methods to update the registers of the synchronous channels:

- Method 1 (UPDM = 0): The period value, the duty-cycle values and the dead-time values must be written by the processor in their respective update registers (respectively PWM_CPRDUPDx, PWM_CDTYUPDx and PWM_DTUPDx).The update is triggered at the next PWM period as soon as the bit UPDULOCK in the PWM Sync Channels Update Control Register (PWM_SCUC) is set to ‘1’ (see “Method 1: Manual write of duty-cycle values and manual trigger of the update”).
- Method 2 (UPDM = 1): The period value, the duty-cycle values, the dead-time values and the update period value must be written by the processor in their respective update registers (respectively PWM_CPRDUPDx, PWM_CDTYUPDx and PWM_DTUPD). The update of the period value and of the dead-time values is triggered at the next PWM period as soon as the bit UPDULOCK in the PWM_SCUC register is set to ‘1’. The update of the duty-cycle values and the update period value is triggered automatically after an update period defined by the field UPR in the PWM Sync Channels Update Period Register (PWM_SCUP) (see “Method 2: Manual write of duty-cycle values and automatic trigger of the update”).

5. The number of bytes available in the FIFO is made available by reading RXBYTECNT in the endpoint's UDP_CSRx.
6. The microcontroller transfers out data received from the endpoint's memory to the microcontroller's memory. Data received is made available by reading the endpoint's UDP_FDRx.
7. The microcontroller notifies the USB peripheral device that it has finished the transfer by clearing RX_DATA_BK0 in the endpoint's UDP_CSRx.
8. A third Data OUT packet can be accepted by the USB peripheral device and copied in the FIFO Bank 0.
9. If a second Data OUT packet has been received, the microcontroller is notified by the flag RX_DATA_BK1 set in the endpoint's UDP_CSRx. An interrupt is pending for this endpoint while RX_DATA_BK1 is set.
10. The microcontroller transfers out data received from the endpoint's memory to the microcontroller's memory. Data received is available by reading the endpoint's UDP_FDRx.
11. The microcontroller notifies the USB device it has finished the transfer by clearing RX_DATA_BK1 in the endpoint's UDP_CSRx.
12. A fourth Data OUT packet can be accepted by the USB device and copied in the FIFO Bank 1.

Figure 40-11. Data OUT Transfer for Ping-pong Endpoint



Note: An interrupt is pending while the RX_DATA_BK0 or RX_DATA_BK1 flag is set.

Warning: When RX_DATA_BK0 and RX_DATA_BK1 are both set, there is no way to determine which one to clear first. Thus the software must keep an internal counter to be sure to clear alternatively RX_DATA_BK0 then RX_DATA_BK1. This situation may occur when the software application is busy elsewhere and the two banks are filled by the USB host. Once the application comes back to the USB driver, the two flags are set.

Figure 42-2. Sequence of ADC Conversions When Tracking Time > Conversion Time

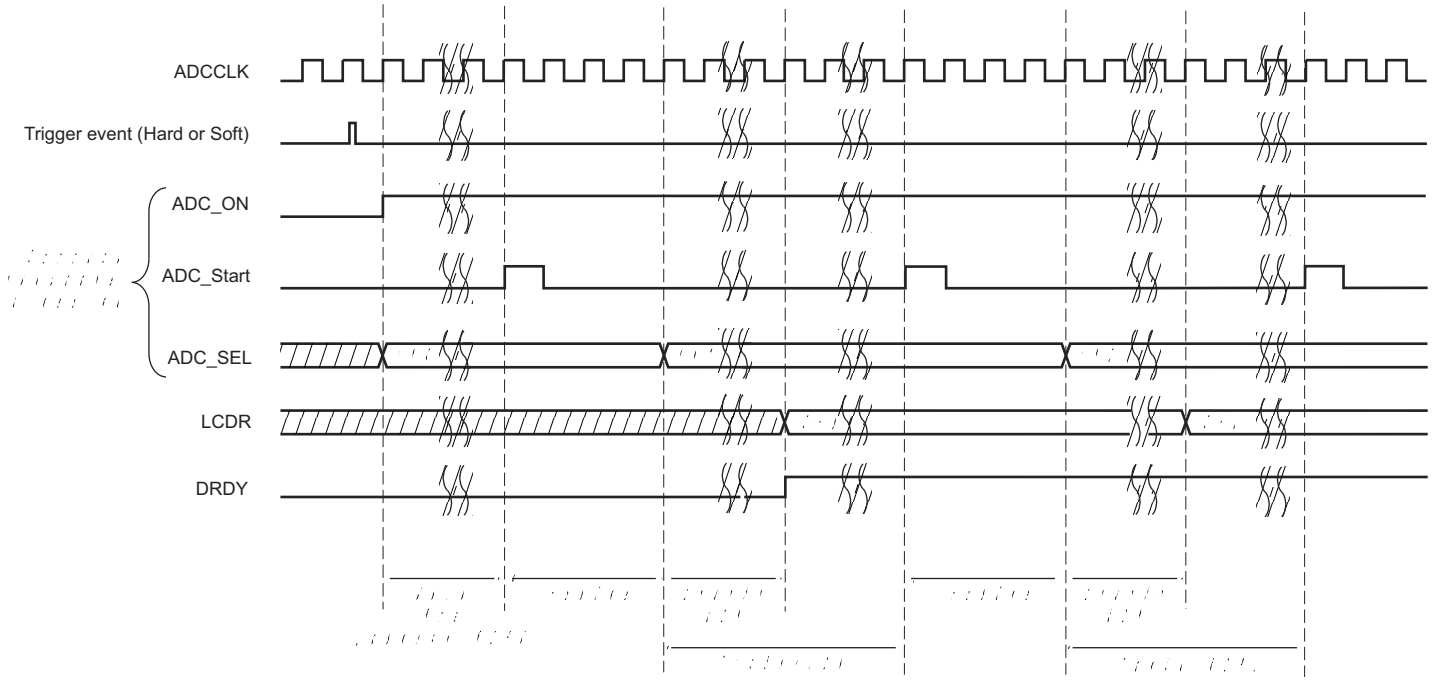
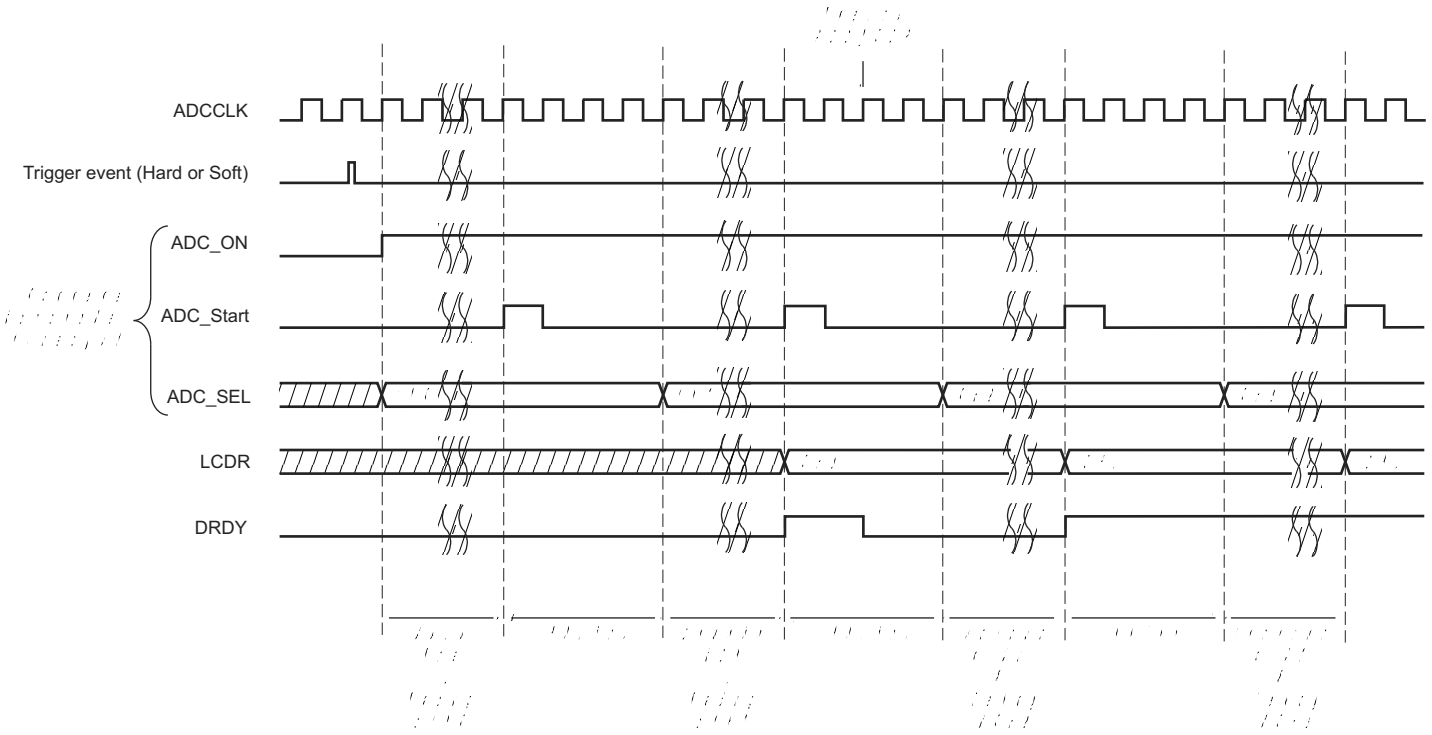


Figure 42-3. Sequence of ADC Conversions When Tracking Time < Conversion Time



- **TRANSFER: Hold Time**

  The TRANSFER field should be set to 2 to guarantee the optimal hold time.

- **USEQ: Use Sequence Enable**

Value	Name	Description
0	NUM_ORDER	Normal Mode: The controller converts channels in a simple numeric order depending only on the channel index.
1	REG_ORDER	User Sequence Mode: The sequence respects what is defined in ADC_SEQR1 and ADC_SEQR2 registers and can be used to convert the same channel several times.

Table 44-12. SAM4S4/S2 Typical Sleep Mode Current Consumption vs Master Clock (MCK) Variation with PLLA

Core Clock/MCK (MHz)	Typical Value @ 25°C		Unit
	VDDCORE Consumption (AMP1)	Total Consumption (AMP2)	
120	5.1	6.9	mA
100	4.3	5.8	
84	3.7	5.0	
64	2.8	3.9	
32	1.5	2.2	
24	1.2	1.8	

Table 44-13. SAM4S4/2 Typical Sleep Mode Current Consumption vs Master Clock (MCK) Variation with Fast RC

Core Clock/MCK (MHz)	Typical Value @ 25°C		Unit
	VDDCORE Consumption (AMP1)	Total Consumption (AMP2)	
12	0.6	0.8	mA
8	0.4	0.7	
4	0.2	0.5	
2	0.17	0.41	
1	0.13	0.34	
0.5	0.11	0.35	

44.12.9 Embedded Flash Characteristics

The maximum operating frequency given in Table 44-73 is limited by the embedded Flash access time when the processor is fetching code out of it. The table provides the device maximum operating frequency defined by the value of the field FWS in the EEFC_FMR. This field defines the number of wait states required to access the embedded Flash memory.

The embedded Flash is fully tested during production test. The Flash contents are not set to a known state prior to shipment. Therefore, the Flash contents should be erased prior to programming an application.

Table 44-73. Embedded Flash Wait State at 105°C

FWS	Read Operations	Maximum Operating Frequency (MHz)			
		VDDCORE 1.08V		VDDCORE 1.2V	
		VDDIO 1.62–3.6 V	VDDIO 2.7–3.6 V	VDDIO 1.62–3.6 V	VDDIO 2.7–3.6 V
0	1 cycle	16	20	17	21
1	2 cycles	33	40	34	42
2	3 cycles	50	60	52	63
3	4 cycles	67	80	69	84
4	5 cycles	84	100	87	105
5	6 cycles	100	–	104	120

Table 44-74. AC Flash Characteristics

Parameter	Conditions[1]	Min	Typ	Max	Unit
Program Cycle Time	Erase Page Mode	–	10	50	ms
	Erase Block Mode (by 4 Kbytes)	–	50	200	ms
	Erase Sector Mode	–	400	950	ms
Erase Pin Assertion Time	Erase pin high	220	–	–	ms
Full Chip Erase	1 Mbyte	–	9	18	s
	512 Kbytes	–	5.5	11	
	256 Kbytes	–	3	6	
	128 Kbytes	–	2	4	
Data Retention	Not powered or powered	–	20	–	years
Page Program Time[2]	1 word changed in the page	–	–	75	µs
	2 words changed in the page	–	–	120	µs
	4 words changed in the page	–	–	210	µs
	16 words changed in the page	–	–	740	µs
	32 words changed in the page	–	–	1.45	ms
	Full page	–	–	3	ms
Endurance	Write/Erase cycles per page, block or sector @ 85°C	10k	–	–	cycles
	Write/Erase cycles per page, block or sector @ 50°C	50k	–	–	

Notes: 1. Only the read operation is characterized between -40 and 105 °C. Other operations are characterized between -40 and 85 °C.
2. All bits in the word(s) are set to 0.

Table 49-4. SAM4S Datasheet Rev. 11100H Revision History (Continued)

Doc. Date	Changes
08-Jan-15	Added “Symbol” column to Table 44-57 ”Static Performance Characteristics”, Table 44-58 ”Dynamic Performance Characteristics”, Table 44-59 ”Analog Outputs”, and Table 44-60 ”Analog Comparator Characteristics” Section 44.11 “Temperature Sensor”: specified instances of “27°C” as ambient temperature Table 44-63 ”I/O Characteristics”: added parameter “Maximum I/O skew” Section 44.12.3.1 “Maximum SPI Frequency”: - under “Master Write Mode”, replaced “the maximum SPI frequency is the one from the pad” with “the maximum SPI frequency is defined by the pin FreqMax value - updated content under “Master Read Mode” Table 44-65 ”SSC Timings”: in Min/Max values for SSC_4 and SSC_7, corrected links to footnote 2 Section 44.12.9 “Embedded Flash Characteristics”: in first paragraph, corrected “field FWS of the MC_FMR” to “field FWS of the EEFC_FMR”
	Section 45. “Mechanical Characteristics” Inserted heading Section 45.1 “100-lead LQFP Mechanical Characteristics” Inserted heading Section 45.2 “100-ball TFBGA Mechanical Characteristics” Inserted heading Section 45.3 “100-ball VFBGA Mechanical Characteristics” Inserted heading Section 45.4 “64-lead LQFP Mechanical Characteristics” Table 45-16 ”LQFP Package Characteristics”: corrected title (was “LQFP and QFN Package Characteristics”) Inserted heading Section 45.5 “64-lead QFN Mechanical Characteristics” Inserted heading Section 45.6 “64-ball WLCSP Mechanical Characteristics” Inserted heading Section 45.7 “48-lead LQFP Mechanical Characteristics” and added sentence “This package respects the recommendations of the NEMI User Group.” Inserted heading Section 45.8 “48-lead QFN Mechanical Characteristics” and added sentence “This package respects the recommendations of the NEMI User Group.” Table 45-29 ”48-lead QFN Package Characteristics”: corrected title (was “48-lead LQFP Package Characteristics”) and changed Moisture Sensitivity Level from 1 to 3 Table 45-30 ”48-lead QFN Package Reference”: corrected title (was “48-lead LQFP Package Reference”)
	Added Section 46. “Marking”
	Section 47. “Ordering Information”: Table 47-1 ”Ordering Codes for SAM4S Devices”: added ordering codes for MRL ‘B’
	Section 48. “Errata” Section 48.1 “Errata SAM4SD32/SD16/SA16/S16/S8 Rev. A Parts”: added Section 48.1.5 “Low-power Mode” Added Section 48.2 “Errata SAM4SD32/SD16/SA16/S16/S8 Rev. B Parts” Section 48.3 “Errata SAM4S4/S2 Rev. A Parts”: added Section 48.3.4 “Low-power Mode” Added Section 48.4 “Errata SAM4S4/S2 Rev. B Parts”

.

Table 49-10. SAM4S Datasheet Rev. 11100B 31-Jul-12 Revision History (Continued)

Doc. Rev. 11100B	Comments	Change Request Ref.
	RTC	
	In Section 16.6.2 “RTC Mode Register” on page 303, formulas associated with conditions HIGHPPM = 1 and HIGHPPM = 0 have been swapped, text has been clarified.	7950
	In Section 16.5.7 “RTC Accurate Clock Calibration” on page 299, paragraph describing RTC clock calibration circuitry correction updated with mention of crystal drift.	7952
	SUPC	
	References to WFE instructions deleted in Section 18.3.3 “Core Voltage Regulator Control/Backup Low-power Mode” on page 328.	rfo
	Supply monitor threshold values modified in Section 18.3.4 “Supply Monitor” on page 328.	
	SMTH bit table replaced by a cross-reference to Electrical characteristics in Section 18.4.4 “Supply Controller Supply Monitor Mode Register” on page 338.	
	Typo in Section 18.4.8 “Supply Controller Status Register” on page 343 is now fixed.	8024
	“half” replaced with “first half” in Section 18.4.6 “Supply Controller Wake-up Mode Register” on page 340 and in Section 18.4.7.2 “Low Power Debouncer Inputs” on page 295.	8067
	Figure 18-4 on page 331 modified.	
	Push-to-Break figure example Figure 18-6 on page 333 added, title of Figure 18-5 on page 333 modified.	8064, 8082
	“square waveform ..” changed to “duty cycle ..” in Section 18.4.7.2 “Low Power Debouncer Inputs” on page 295.	8082 8226
	Switching time of slow crystal oscillator updated in Section 18.3.2 “Slow Clock Generator” on page 328.	8266
	EEFC	
	Added GPNVM command line in Section • “FARG: Flash Command Argument” on page 368.	8076
	Unique identifier address changed in Section 20.4.3.8 “Unique Identifier” on page 363.	8274
	User Signature address changed in Section 20.4.3.9 “User Signature” on page 363.	
	Changed the System Controller base address from 0x400E0800 to 0x400E0A00 in Section 20.5 “Enhanced Embedded Flash Controller (EEFC) User Interface” on page 365.	rfo
	FFPI	
	All references, tables, figures related to 48-bit devices cleared in this whole chapter.	rfo
	CMCC	
	New chapter.	
	CRCCU	
	Typos: CCIT802 corrected to CCITT802, CCIT16 corrected to CCITT16 in Section 23.5.1 “CRC Calculation Unit” on page 399 and Section 23.7.10 “CRCCU Mode Register” on page 414. TRC_RC corrected to TR_CRC in Section 23.7.10 “CRCCU Mode Register” on page 414.	7803
	SMC	
	“turned out” changed to “switched to output mode” in Section 26.8.4 “Write Mode” on page 450.	7925
	Removed DBW which is not required for 8-bit only in Section 26.15.4 “SMC MODE Register” on page 476.	8307