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	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 16x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°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/atsam4sa16ca-cu

```
                        ; with top halfword of R5, subtracts second from
                        ; first, adds R6, writes to R0
SMLSDX  R1, R3, R2, R0 ; Multiplies bottom halfword of R3 with top
                        ; halfword of R2, multiplies top halfword of R3
                        ; with bottom halfword of R2, subtracts second from
                        ; first, adds R0, writes to R1
SMLSLD  R3, R6, R2, R7 ; Multiplies bottom halfword of R6 with bottom
                        ; halfword of R2, multiplies top halfword of R6
                        ; with top halfword of R2, subtracts second from
                        ; first, adds R6:R3, writes to R6:R3
SMLSLDX R3, R6, R2, R7 ; Multiplies bottom halfword of R6 with top
                        ; halfword of R2, multiplies top halfword of R6
                        ; with bottom halfword of R2, subtracts second from
                        ; first, adds R6:R3, writes to R6:R3.
```

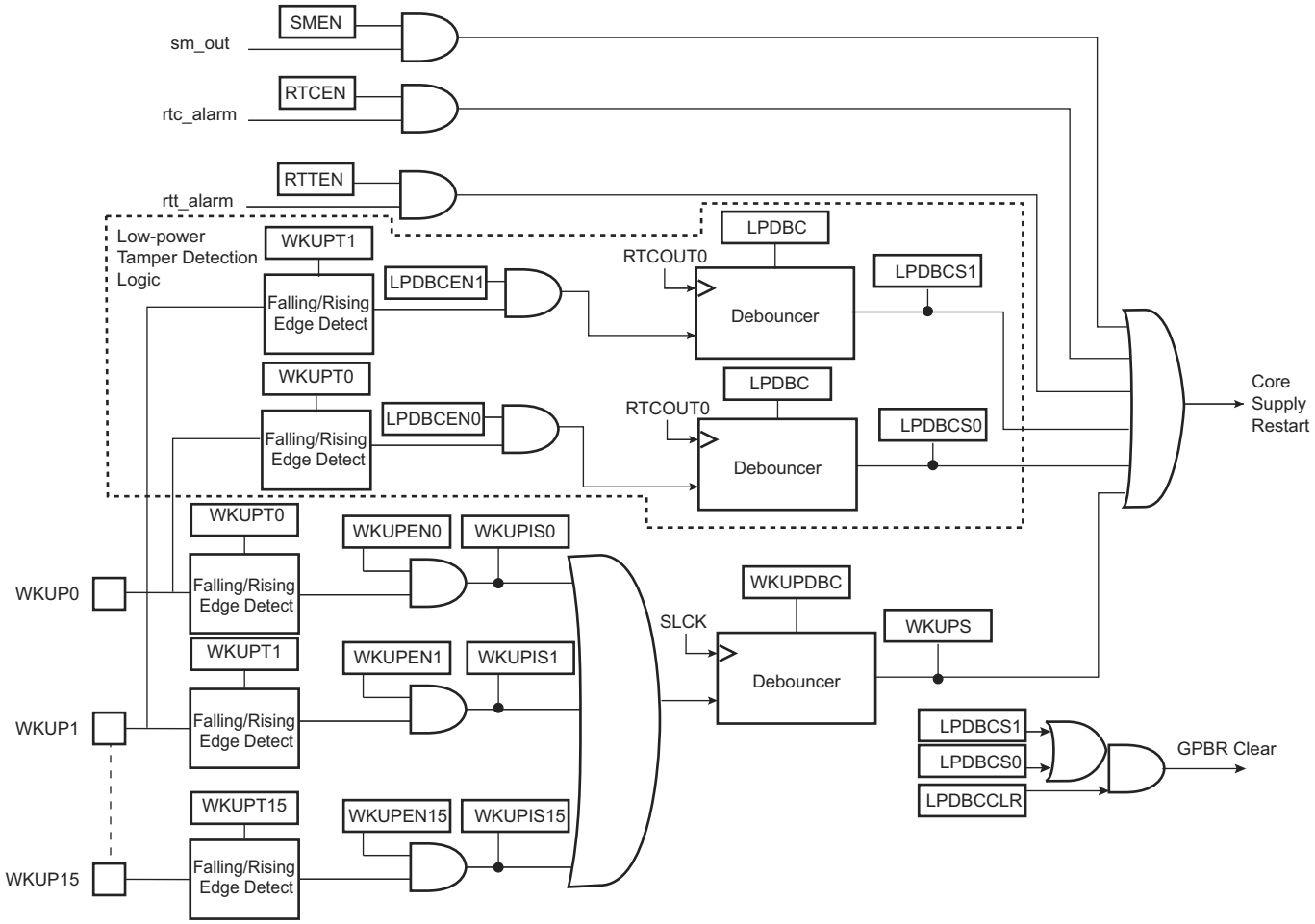
If BODRSTEN is set and the voltage regulation is lost (output voltage of the regulator too low), the vddcore_nreset signal is asserted for a minimum of one slow clock cycle and then released if bodcore_in has been reactivated. The BODRSTS bit in SUPC_SR indicates the source of the last reset.

Until bodcore_in is deactivated, the vddcore_nreset signal remains active.

18.4.7 Wake-up Sources

The wake-up events allow the device to exit Backup mode. When a wake-up event is detected, the SUPC performs a sequence that automatically reenables the core power supply.

Figure 18-4. Wake-up Sources



18.4.7.1 Wake-up Inputs

The wake-up inputs, WKUPx, can be programmed to perform a wake-up of the core power supply. Each input can be enabled by writing a 1 to the corresponding bit, WKUPENx, in the Wake-up Inputs register (SUPC_WUIR). The wake-up level can be selected with the corresponding polarity bit, WKUPTx, also located in SUPC_WUIR.

The resulting signals are wired-ORed to trigger a debounce counter, which is programmed with the WKUPDBC field in SUPC_WUMR. The WKUPDBC field selects a debouncing period of 3, 32, 512, 4,096 or 32,768 slow clock cycles. The duration of these periods corresponds, respectively, to about 100 µs, about 1 ms, about 16 ms, about 128 ms and about 1 second (for a typical slow clock frequency of 32 kHz). Programming WKUPDBC to 0x0 selects an immediate wake-up, i.e., an enabled WKUP pin must be active according to its polarity during a minimum of one slow clock period to wake up the core power supply.

The **Memory Write** command **(WRAM)** is optimized for consecutive writes. Write handshaking can be chained; an internal address buffer is automatically increased.

Table 21-14. Write Command

Step	Handshake Sequence	MODE[3:0]	DATA[15:0]
1	Write handshaking	CMDE	WRAM
2	Write handshaking	ADDR0	Memory Address LSB
3	Write handshaking	ADDR1	Memory Address
4	Write handshaking	DATA	*Memory Address++
5	Write handshaking	DATA	*Memory Address++
...	...	...	...
n	Write handshaking	ADDR0	Memory Address LSB
n+1	Write handshaking	ADDR1	Memory Address
n+2	Write handshaking	DATA	*Memory Address++
n+3	Write handshaking	DATA	*Memory Address++
...	...	...	...

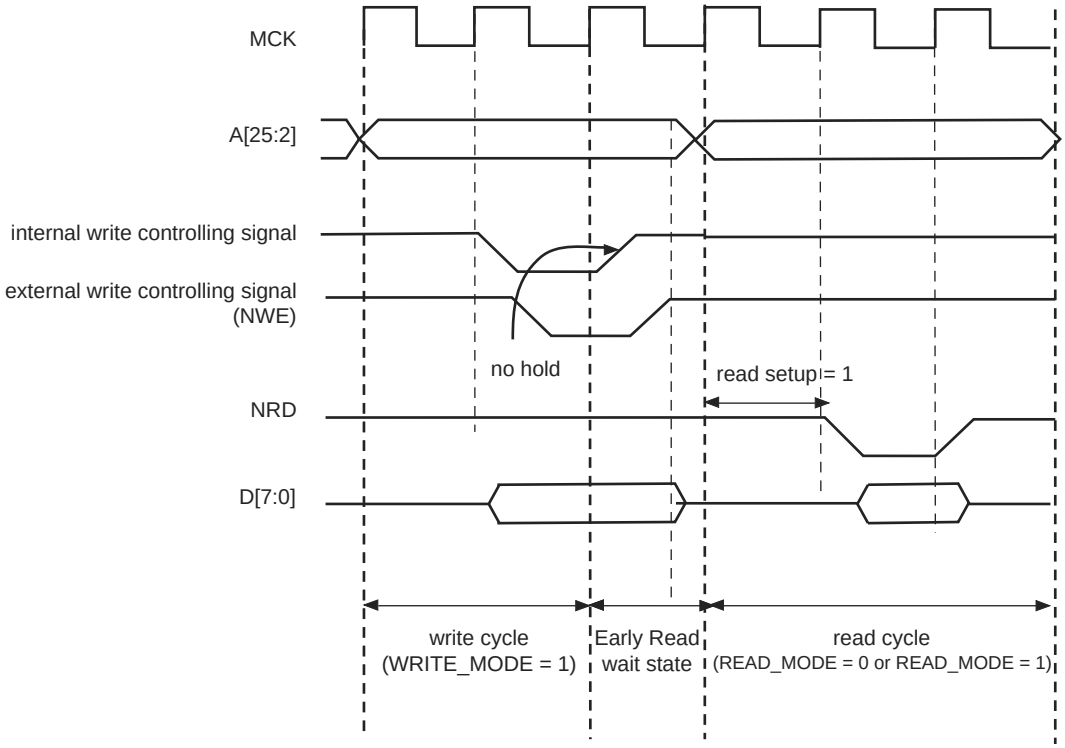
21.3.5.8 Get Version Command

The **Get Version** (GVE) command retrieves the version of the FFPI interface.

Table 21-15. Get Version Command

Step	Handshake Sequence	MODE[3:0]	DATA[15:0]
1	Write handshaking	CMDE	GVE
2	Read handshaking	DATA	Version

Figure 26-16. Early Read Wait State: NWE-controlled Write with No Hold Followed by a Read with one Set-up Cycle



26.11.3 Reload User Configuration Wait State

The user may change any of the configuration parameters by writing the SMC user interface.

When detecting that a new user configuration has been written in the user interface, the SMC inserts a wait state before starting the next access. This “Reload User Configuration Wait State” is used by the SMC to load the new set of parameters to apply to next accesses.

The Reload Configuration Wait State is not applied in addition to the Chip Select Wait State. If accesses before and after re-programming the user interface are made to different devices (Chip Selects), then one single Chip Select Wait State is applied.

On the other hand, if accesses before and after writing the user interface are made to the same device, a Reload Configuration Wait State is inserted, even if the change does not concern the current Chip Select.

26.11.3.1 User Procedure

To insert a Reload Configuration Wait State, the SMC detects a write access to any SMC_MODE register of the user interface. If the user only modifies timing registers (SMC_SETUP, SMC_PULSE, SMC_CYCLE registers) in the user interface, he must validate the modification by writing the SMC_MODE, even if no change was made on the mode parameters.

The user must not change the configuration parameters of an SMC Chip Select (Setup, Pulse, Cycle, Mode) if accesses are performed on this CS during the modification. Any change of the Chip Select parameters, while fetching the code from a memory connected on this CS, may lead to unpredictable behavior. The instructions used to modify the parameters of an SMC Chip Select can be executed from the internal RAM or from a memory connected to another CS.

31.6.6 PIO Output Status Register

Name: PIO_OSR

Address: 0x400E0E18 (PIOA), 0x400E1018 (PIOB), 0x400E1218 (PIOC)

Access: Read-only

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

- **P0–P31: Output Status**

0: The I/O line is a pure input.

1: The I/O line is enabled in output.

31.6.11 PIO Clear Output Data Register

Name: PIO_CODR

Address: 0x400E0E34 (PIOA), 0x400E1034 (PIOB), 0x400E1234 (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: Clear Output Data**

0: No effect.

1: Clears the data to be driven on the I/O line.

31.6.13 PIO Pin Data Status Register

Name: PIO_PDSR

Address: 0x400E0E3C (PIOA), 0x400E103C (PIOB), 0x400E123C (PIOC)

Access: Read-only

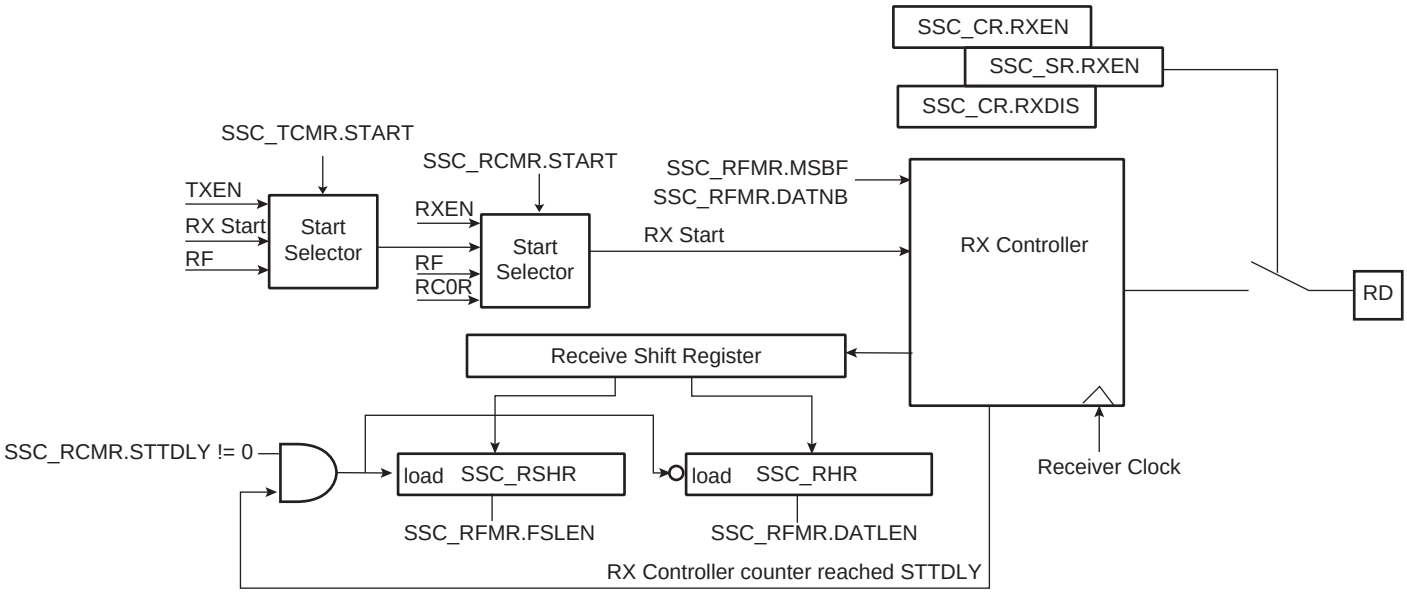
31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24
23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16
15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8
7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

- **P0–P31: Output Data Status**

0: The I/O line is at level 0.

1: The I/O line is at level 1.

Figure 32-12. Receiver Block Diagram



32.8.4 Start

The transmitter and receiver can both be programmed to start their operations when an event occurs, respectively in the Transmit Start Selection (START) field of SSC_TCMR and in the Receive Start Selection (START) field of SSC_RCMR.

Under the following conditions the start event is independently programmable:

- Continuous. In this case, the transmission starts as soon as a word is written in SSC_THR and the reception starts as soon as the Receiver is enabled.
- Synchronously with the transmitter/receiver
- On detection of a falling/rising edge on TF/RF
- On detection of a low level/high level on TF/RF
- On detection of a level change or an edge on TF/RF

A start can be programmed in the same manner on either side of the Transmit/Receive Clock Register (SSC_RCMR/SSC_TCMR). Thus, the start could be on TF (Transmit) or RF (Receive).

Moreover, the Receiver can start when data is detected in the bit stream with the Compare Functions.

Detection on TF/RF input/output is done by the field FSOS of the Transmit/Receive Frame Mode Register (SSC_TFMR/SSC_RFMR).

36.6.8 SPI Mode

The Serial Peripheral Interface (SPI) mode is a synchronous serial data link that provides communication with external devices in Master or Slave mode. It also enables communication between processors if an external processor is connected to the system.

The Serial Peripheral Interface is essentially a shift register that serially transmits data bits to other SPIs. During a data transfer, one SPI system acts as the “master” which controls the data flow, while the other devices act as “slaves" which have data shifted into and out by the master. Different CPUs can take turns being masters and one master may simultaneously shift data into multiple slaves. (Multiple master protocol is the opposite of single master protocol, where one CPU is always the master while all of the others are always slaves.) However, only one slave may drive its output to write data back to the master at any given time.

A slave device is selected when its NSS signal is asserted by the master. The USART in SPI Master mode can address only one SPI slave because it can generate only one NSS signal.

The SPI system consists of two data lines and two control lines:

- Master Out Slave In (MOSI): This data line supplies the output data from the master shifted into the input of the slave.
- Master In Slave Out (MISO): This data line supplies the output data from a slave to the input of the master.
- Serial Clock (SCK): This control line is driven by the master and regulates the flow of the data bits. The master may transmit data at a variety of baud rates. The SCK line cycles once for each bit that is transmitted.
- Slave Select (NSS): This control line allows the master to select or deselect the slave.

36.6.8.1 Modes of Operation

The USART can operate in SPI Master mode or in SPI Slave mode.

Operation in SPI Master mode is programmed by writing 0xE to the USART_MODE field in US_MR. In this case the SPI lines must be connected as described below:

- The MOSI line is driven by the output pin TXD
- The MISO line drives the input pin RXD
- The SCK line is driven by the output pin SCK
- The NSS line is driven by the output pin RTS

Operation in SPI Slave mode is programmed by writing to 0xF the USART_MODE field in US_MR. In this case the SPI lines must be connected as described below:

- The MOSI line drives the input pin RXD
- The MISO line is driven by the output pin TXD
- The SCK line drives the input pin SCK
- The NSS line drives the input pin CTS

In order to avoid unpredictable behavior, any change of the SPI mode must be followed by a software reset of the transmitter and of the receiver (except the initial configuration after a hardware reset). (See Section 36.6.8.4).

36.6.8.2 Baud Rate

In SPI mode, the baud rate generator operates in the same way as in USART Synchronous mode. See Section 36.6.1.3 ”Baud Rate in Synchronous Mode or SPI Mode”. However, there are some restrictions:

In SPI Master mode:

- The external clock SCK must not be selected (USCLKS ≠ 0x3), and the bit CLKO must be set to 1 in the US_MR, in order to generate correctly the serial clock on the SCK pin.
- To obtain correct behavior of the receiver and the transmitter, the value programmed in CD must be superior or equal to 6.

- **DSRIC: Data Set Ready Input Change Enable**
- **DCDIC: Data Carrier Detect Input Change Interrupt Enable**
- **CTSIC: Clear to Send Input Change Interrupt Enable**
- **MANE: Manchester Error Interrupt Enable**

36.7.23 USART Write Protection Status Register

Name: US_WPSR

Address: 0x400240E8 (0), 0x400280E8 (1)

Access: Read-only

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

23	22	21	20	19	18	17	16
WPVSRC							

15	14	13	12	11	10	9	8
WPVSRC							

7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	WPVS

- **WPVS: Write Protection Violation Status**

0: No write protection violation has occurred since the last read of the US_WPSR.

1: A write protection violation has occurred since the last read of the US_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

- **WPVSRC: Write Protection Violation Source**

When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

37.7.1 TC Channel Control Register

Name: TC_CCRx [x=0..2]

Address: 0x40010000 (0)[0], 0x40010040 (0)[1], 0x40010080 (0)[2], 0x40014000 (1)[0], 0x40014040 (1)[1], 0x40014080 (1)[2]

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
–	–	–	–	–	SWTRG	CLKDIS	CLKEN

- **CLKEN: Counter Clock Enable Command**

0: No effect.

1: Enables the clock if CLKDIS is not 1.

- **CLKDIS: Counter Clock Disable Command**

0: No effect.

1: Disables the clock.

- **SWTRG: Software Trigger Command**

0: No effect.

1: A software trigger is performed: the counter is reset and the clock is started.

- 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”).

39.7.21 PWM Output Selection Set Update Register

Name: PWM_OSSUPD

Address: 0x40020054

Access: Write-only

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

23	22	21	20	19	18	17	16
–	–	–	–	OSSUPL3	OSSUPL2	OSSUPL1	OSSUPL0

15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–

7	6	5	4	3	2	1	0
–	–	–	–	OSSUPH3	OSSUPH2	OSSUPH1	OSSUPH0

- **OSSUPHx: Output Selection Set for PWMH output of the channel x**

0: No effect.

1: Output override value OOVHx selected as PWMH output of channel x at the beginning of the next channel x PWM period.

- **OSSUPLx: Output Selection Set for PWML output of the channel x**

0: No effect.

1: Output override value OOVLx selected as PWML output of channel x at the beginning of the next channel x PWM period.

39.7.30 PWM Write Protection Control Register

Name: PWM_WPCR

Address: 0x400200E4

Access: Write-only

31	30	29	28	27	26	25	24
WPKEY							

23	22	21	20	19	18	17	16
WPKEY							

15	14	13	12	11	10	9	8
WPKEY							

7	6	5	4	3	2	1	0
WPRG5	WPRG4	WPRG3	WPRG2	WPRG1	WPRG0	WPCMD	

See Section 39.6.6 “Register Write Protection” for the list of registers that can be write-protected.

- **WPCMD: Write Protection Command**

This command is performed only if the WPKEY corresponds to 0x50574D (“PWM” in ASCII).

Value	Name	Description
0	DISABLE_SW_PROT	Disables the software write protection of the register groups of which the bit WPRGx is at ‘1’.
1	ENABLE_SW_PROT	Enables the software write protection of the register groups of which the bit WPRGx is at ‘1’.
2	ENABLE_HW_PROT	Enables the hardware write protection of the register groups of which the bit WPRGx is at ‘1’. Only a hardware reset of the PWM controller can disable the hardware write protection. Moreover, to meet security requirements, the PIO lines associated with the PWM can not be configured through the PIO interface.

- **WPRGx: Write Protection Register Group x**

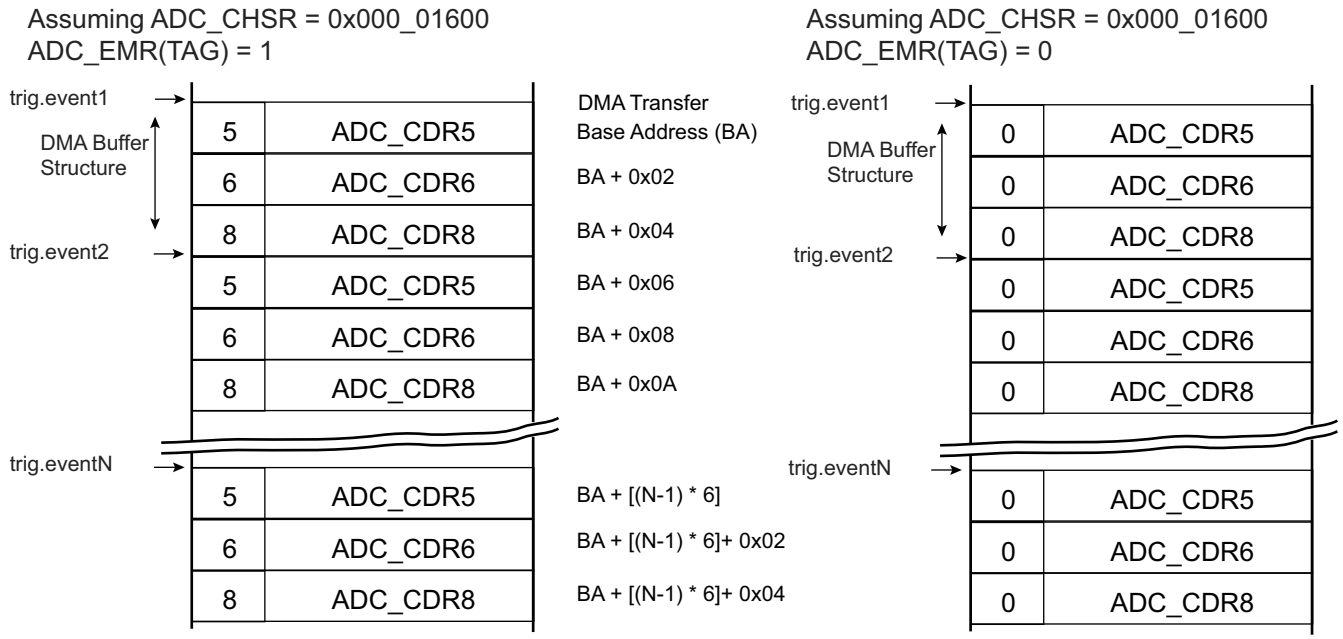
0: The WPCMD command has no effect on the register group x.

1: The WPCMD command is applied to the register group x.

- **WPKEY: Write Protection Key**

Value	Name	Description
0x50574D	PASSWD	Writing any other value in this field aborts the write operation of the WPCMD field. Always reads as 0

Figure 42-8. Buffer Structure



42.6.14 Fault Output

The ADC Controller internal fault output is directly connected to PWM fault input. Fault output may be asserted depending on the configuration of ADC_EMR and ADC_CWR and converted values. When the compare occurs, the ADC fault output generates a pulse of one peripheral clock cycle to the PWM fault input. This fault line can be enabled or disabled within PWM. Should it be activated and asserted by the ADC Controller, the PWM outputs are immediately placed in a safe state (pure combinational path). Note that the ADC fault output connected to the PWM is not the COMPE bit. Thus the Fault mode (FMOD) within the PWM configuration must be FMOD = 1.

42.6.15 Register Write Protection

To prevent any single software error from corrupting ADC behavior, certain registers in the address space can be write-protected by setting the WPEN bit in the “ADC Write Protection Mode Register” (ADC_WPMR).

If a write access to the protected registers is detected, the WPVS flag in the “ADC Write Protection Status Register” (ADC_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS flag is automatically reset by reading the ADC_WPSR.

The following registers can be write-protected:

- ADC Mode Register
- ADC Channel Sequence 1 Register
- ADC Channel Sequence 2 Register
- ADC Channel Enable Register
- ADC Channel Disable Register
- ADC Extended Mode Register
- ADC Compare Window Register
- ADC Channel Gain Register
- ADC Channel Offset Register
- ADC Analog Control Register

Table 44-18. SAM4S4/S2 Typical Current Consumption in Wait Mode

Conditions	@ 25°C		@ 85°C	@ 105°C	Unit
	VDDOUT Consumption (AMP1)	Total Consumption (AMP2)	Total Consumption (AMP2)	Total Consumption (AMP2	
See Figure 44-9 on page 1155 There is no activity on the I/Os of the device. With the Flash in standby mode	14.9	28.4	211	436	µA
See Figure 44-9 on page 1155 There is no activity on the I/Os of the device. With the Flash in deep power down mode	14.9	24.1	205	432	

Table 44-19. SAM4S16/S8 Typical Current Consumption in Wait Mode

Conditions	@ 25°C		@ 85°C	@ 105°C	Unit
	VDDOUT Consumption (AMP1)	Total Consumption (AMP2)	Total Consumption (AMP2)	Total Consumption (AMP2	
See Figure 44-9 on page 1155 There is no activity on the I/Os of the device. With the Flash in standby mode	20.5	32.7	344	654	µA
See Figure 44-9 on page 1155 There is no activity on the I/Os of the device. With the Flash in deep power down mode	20.5	27.8	438	589	

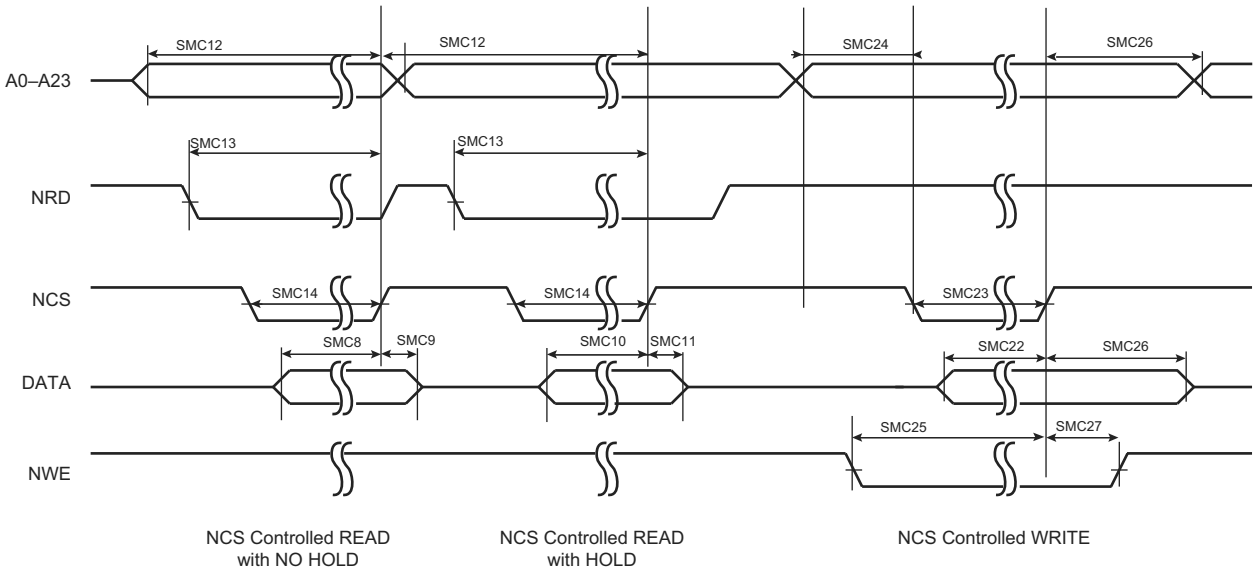
Table 44-20. SAM4SD32/SD16/SA16 Typical Current Consumption in Wait Mode

Conditions	@ 25°C		@ 85°C	@ 105°C	Unit
	VDDOUT Consumption (AMP1)	Total Consumption (AMP2)	Total Consumption (AMP2	Total Consumption (AMP2	
See Figure 44-9 on page 1155 There is no activity on the I/Os of the device. With the Flash in standby mode	–	42.1	633	1105	µA
See Figure 44-9 on page 1155 There is no activity on the I/Os of the device. With the Flash in deep power down mode	–	35.3	608	1085	

Table 44-70. SMC Write Signals - NCS Controlled (WRITE_MODE = 0)

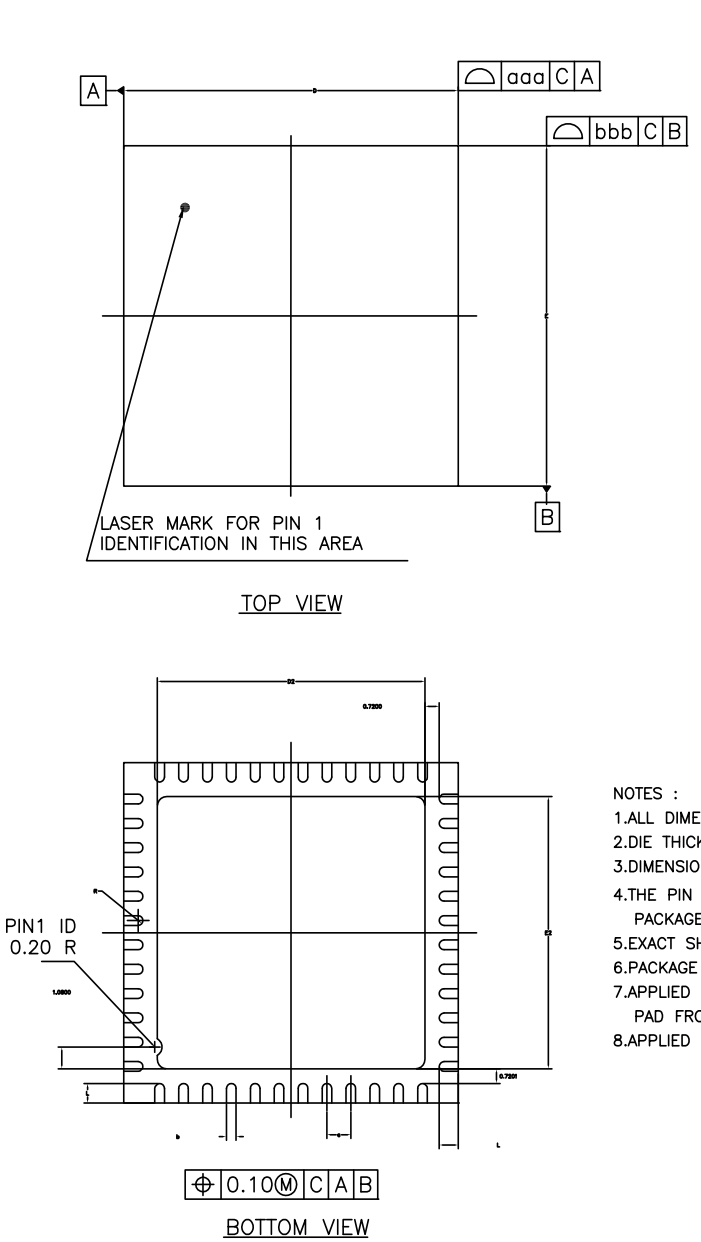
Symbol	Parameter	Min		Max		Unit
		1.8V Domain	3.3V Domain	1.8V Domain	3.3V Domain	
SMC_{22}	Data Out Valid before NCS High	NCS_WR_PULSE $\times t_{CPMCK}$ - 6.3	NCS_WR_PULSE $\times t_{CPMCK}$ - 6.2	–	–	ns
SMC_{23}	NCS Pulse Width	NCS_WR_PULSE $\times t_{CPMCK}$ - 7.7	NCS_WR_PULSE $\times t_{CPMCK}$ - 6.7	–	–	ns
SMC_{24}	A0–A22 Valid before NCS Low	NCS_WR_SETUP $\times t_{CPMCK}$ - 6.5	NCS_WR_SETUP $\times t_{CPMCK}$ - 6.3	–	–	ns
SMC_{25}	NWE Low before NCS High	(NCS_WR_SETUP - NWE_SETUP + NCS pulse) $\times t_{CPMCK}$ - 5.1	(NCS_WR_SETUP - NWE_SETUP + NCS pulse) $\times t_{CPMCK}$ - 4.9	–	–	ns
SMC_{26}	NCS High to Data Out, A0–A25, Change	NCS_WR_HOLD $\times t_{CPMCK}$ - 10.2	NCS_WR_HOLD $\times t_{CPMCK}$ - 8.4	–	–	ns
SMC_{27}	NCS High to NWE Inactive	(NCS_WR_HOLD - NWE_HOLD) $\times t_{CPMCK}$ - 2.1	(NCS_WR_HOLD - NWE_HOLD) $\times t_{CPMCK}$ - 1.6	–	–	ns

Figure 44-35. SMC Timings - NCS Controlled Read and Write



45.8 48-lead QFN Mechanical Characteristics

Figure 45-10. 48-lead QFN Package Drawing



* CONTROLLING DIMENSION : MM

SYMBOL	MILLIMETER			INCH		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	---	---	0.90	---	---	0.035
A1	---	---	0.050	---	---	0.002
A2	---	0.65	0.70	---	0.026	0.028
A3	0.20 REF.			0.008 REF.		
b	0.18	0.20	0.23	0.007	0.008	0.009
D	7.00 bsc			0.276 bsc		
D2	5.45	5.60	5.75	0.215	0.220	0.226
E	7.00 bsc			0.276 bsc		
E2	5.45	5.60	5.75	0.215	0.220	0.226
L	0.35	0.40	0.45	0.014	0.016	0.018
e	0.50 bsc			0.020 bsc		
R	0.09	---	---	0.004	---	---
TOLERANCES OF FORM AND POSITION						
aaa	0.10			0.004		
bbb	0.10			0.004		
ccc	0.05			0.002		

NOTES :
1.ALL DIMENSIONS ARE IN MILLIMETERS.
2.DIE THICKNESS ALLOWABLE IS 0.305 mm MAXIMUM(.012 INCHES MAXIMUM)
3.DIMENSIONING & TOLERANCES CONFORM TO ASME Y14.5M. −1994.
4.THE PIN #1 IDENTIFIER MUST BE PLACED ON THE TOP SURFACE OF THE PACKAGE BY USING INDENTATION MARK OR OTHER FEATURE OF PACKAGE BODY.
5.EXACT SHAPE AND SIZE OF THIS FEATURE IS OPTIONAL.
6.PACKAGE WARPAGE MAX 0.08 mm.
7.APPLIED FOR EXPOSED PAD AND TERMINALS. EXCLUDE EMBEDDING PART OF EXPOSED PAD FROM MEASURING.
8.APPLIED ONLY TO TERMINALS.

Table 45-28. Device and 48-lead LQFP Package Maximum Weight

SAM4S	143	mg

Table 45-29. 48-lead QFN Package Characteristics

Moisture Sensitivity Level	3

Table 45-30. 48-lead QFN Package Reference

JEDEC Drawing Reference	
JESD97 Classification	e3

Table 49-3. SAM4S Datasheet Rev. 11100I Revision History (Continued)

Doc. Date	Changes
03-Apr-15	**Section 23., “Cyclic Redundancy Check Calculation Unit (CRCCU)”** Modified Section 23.1, “Description” and Section 23.2, “Embedded Characteristics” Updated Figure 23-1, “Block Diagram”Section 23.5.2, “CRC Calculation Unit Operation”: reworded text describing how to start the CRCCU Table 23-3 “Register Mapping”: deleted reset value 0x00000000 from all write-only registers Section 23.6, “Transfer Control Registers Memory Mapping”: added reset values to Table 23-2 “Transfer Control Register Memory Mapping” Section 23.7, “Cyclic Redundancy Check Calculation Unit (CRCCU) User Interface”: Modified Section 23.7.10, “CRCCU Mode Register” and Section 23.7.11, “CRCCU Status Register”
	Section 26., “Static Memory Controller (SMC)” Added Latch Enables for NAND Flash in Table 26-1 “I/O Line Description”. Updated Section 26.9.5, “Register Write Protection” and added information on write protection to Section 26.16.1, “SMC Setup Register”, Section 26.16.2, “SMC Pulse Register”, Section 26.16.3, “SMC Cycle Register” and Section 26.16.4, “SMC MODE Register”. Updated Section 26.16.8, “SMC Write Protection Mode Register”, Section 26.16.9, “SMC Write Protection Status Register”and Section 26.10, “Scrambling/Unscrambling Function”.
	Section 27., “Peripheral DMA Controller (PDC)” Added Section 27.3, “Peripheral DMA Controller Connections” Table 27-2 "Register Mapping": removed reset value from Transfer Control Register / PERIPH_PTCR (register access is write-only) Modified Section 27.6.10, “Transfer Status Register”
	Section 28., “Clock Generator” Section 28.2, “Embedded Characteristics” : removed bullet “Write Protected Registers”. Section 28.4.2, “Slow Clock Crystal Oscillator”: deleted redundant content from end of section; appended “OSCBYPASS bit of the Supply Controller Mode Register (SUPC_MR) needs to be set at 1” to read “OSCBYPASS bit of the Supply Controller Mode Register (SUPC_MR) needs to be set at 1 prior to writing a 1 in bit XTALSEL”. Added Section 28.5.5, “Bypassing the Main Crystal Oscillator”. Updated Section 28.6.1, “Divider and Phase Lock Loop Programming”.
	Section 29., “Power Management Controller (PMC)” Updated Section 29.2, “Embedded Characteristics” and Section 29.7, “USB Clock Controller”: Modified Section 29.11, “Fast Startup” Table 29-3 "Register Mapping": changed PMC_SR reset value to 0x0003_0008 (was 0x0001_0008) Modified Section 29.17.7, “PMC Clock Generator Main Oscillator Register”, Section 29.17.8, “PMC Clock Generator Main Clock Frequency Register”, Section 29.17.9, “PMC Clock Generator PLLA Register” and Section 29.17.10, “PMC Clock Generator PLLB Register”
	Section 31., “Parallel Input/Output Controller (PIO)” Replaced all instances of “PIO clock” and “PIO controller clock” with “peripheral clock” Renamed “MCK” or “master clock” to “Peripheral clock” as needed Section 31.2, “Embedded Characteristics”: replaced bullet “Write Protect Registers” with “Register Write Protection” Updated Figure 31-2, “I/O Line Control Logic” and Figure 31-5, “Input Debouncing Filter Timing”