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®-M3
Core Size	32-Bit Single-Core
Speed	96MHz
Connectivity	EBI/EMI, I²C, Memory Card, SPI, SSC, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	57
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	36K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.6V
Data Converters	A/D 4x10b, 4x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	100-LQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsam3u2ca-au

12.3.2 Stacks

The processor uses a full descending stack. This means the stack pointer indicates the last stacked item on the stack memory. When the processor pushes a new item onto the stack, it decrements the stack pointer and then writes the item to the new memory location. The processor implements two stacks, the *main stack* and the *process stack*, with independent copies of the stack pointer, see “Stack Pointer” on page 48.

In Thread mode, the CONTROL register controls whether the processor uses the main stack or the process stack, see “CONTROL register” on page 56. In Handler mode, the processor always uses the main stack. The options for processor operations are:

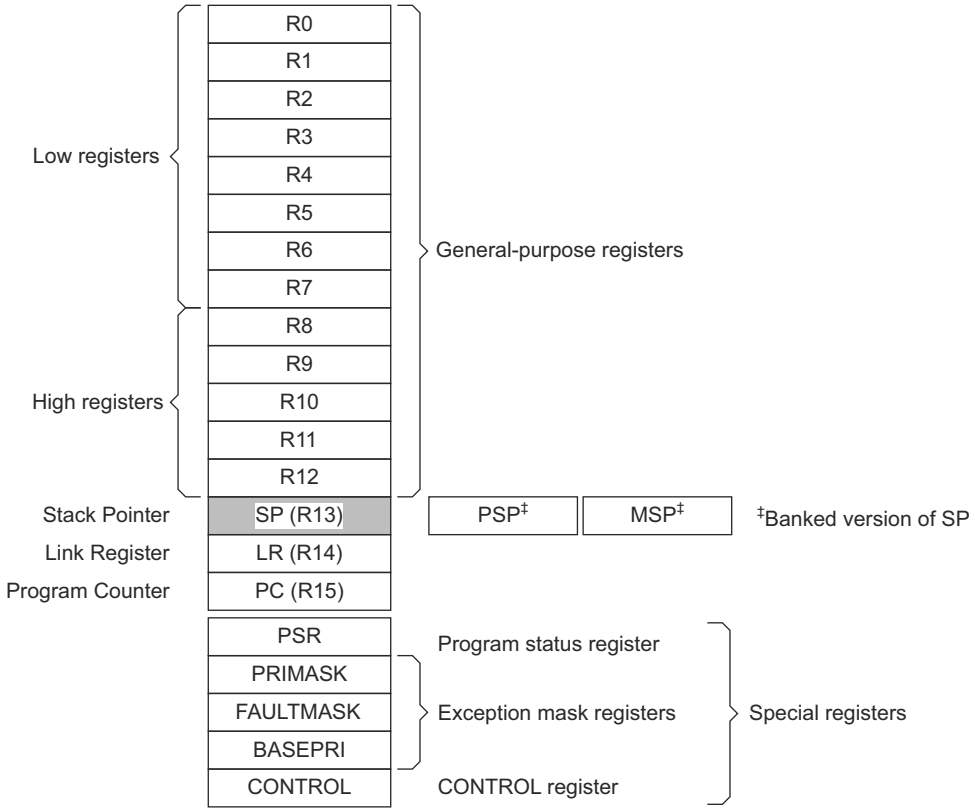
Table 12-1. Summary of processor mode, execution privilege level, and stack use options

Processor mode	Used to execute	Privilege level for software execution	Stack used
Thread	Applications	Privileged or unprivileged (1)	Main stack or process stack(1)
Handler	Exception handlers	Always privileged	Main stack

1. See “CONTROL register” on page 56.

12.3.3 Core registers

The processor core registers are:



- **C**

Carry or borrow flag:

0 = add operation did not result in a carry bit or subtract operation resulted in a borrow bit

1 = add operation resulted in a carry bit or subtract operation did not result in a borrow bit.

- **V**

Overflow flag:

0 = operation did not result in an overflow

1 = operation resulted in an overflow.

- **Q**

Sticky saturation flag:

0 = indicates that saturation has not occurred since reset or since the bit was last cleared to zero

1 = indicates when an SSAT or USAT instruction results in saturation.

This bit is cleared to zero by software using an MRS instruction.

12.3.3.7 Interrupt Program Status Register

The IPSR contains the exception type number of the current *Interrupt Service Routine* (ISR). See the register summary in Table 12-2 on page 48 for its attributes. The bit assignments are:

- **ISR_NUMBER**

This is the number of the current exception:

0 = Thread mode

1 = Reserved

2 = NMI

3 = Hard fault

4 = Memory management fault

5 = Bus fault

6 = Usage fault

7-10 = Reserved

11 = SVCall

12 = Reserved for Debug

13 = Reserved

14 = PendSV

15 = SysTick

16 = IRQ0

45 = IRQ29

see “Exception types” on page 67 for more information.

The memory types are:

12.4.1.1 Normal

The processor can re-order transactions for efficiency, or perform speculative reads.

12.4.1.2 Device

The processor preserves transaction order relative to other transactions to Device or Strongly-ordered memory.

12.4.1.3 Strongly-ordered

The processor preserves transaction order relative to all other transactions.

The different ordering requirements for Device and Strongly-ordered memory mean that the memory system can buffer a write to Device memory, but must not buffer a write to Strongly-ordered memory.

The additional memory attributes include.

12.4.1.4 Shareable

For a shareable memory region, the memory system provides data synchronization between bus masters in a system with multiple bus masters, for example, a processor with a DMA controller.

Strongly-ordered memory is always shareable.

If multiple bus masters can access a non-shareable memory region, software must ensure data coherency between the bus masters.

12.4.1.5 Execute Never (XN)

Means the processor prevents instruction accesses. Any attempt to fetch an instruction from an XN region causes a memory management fault exception.

12.4.2 Memory system ordering of memory accesses

For most memory accesses caused by explicit memory access instructions, the memory system does not guarantee that the order in which the accesses complete matches the program order of the instructions, providing this does not affect the behavior of the instruction sequence. Normally, if correct program execution depends on two memory accesses completing in program order, software must insert a memory barrier instruction between the memory access instructions, see “Software ordering of memory accesses” on page 61.

However, the memory system does guarantee some ordering of accesses to Device and Strongly-ordered memory. For two memory access instructions A1 and A2, if A1 occurs before A2 in program order, the ordering of the memory accesses caused by two instructions is:

A1 \ A2	Normal access	Device access		Strongly-ordered access
		Non-shareable	Shareable	
Normal access	-	-	-	-
Device access, non-shareable	-	<	-	<
Device access, shareable	-	-	<	<
Strongly-ordered access	-	<	<	<

Where:

- Means that the memory system does not guarantee the ordering of the accesses.

< Means that accesses are observed in program order, that is, A1 is always observed before A2.

12.10.4.3 LSL

Logical shift left by *n* bits moves the right-hand 32-n bits of the register Rm, to the left by *n* places, into the left-hand 32-n bits of the result. And it sets the right-hand *n* bits of the result to 0. See Figure 12-6 on page 85.

You can use he LSL #n operation to multiply the value in the register *Rm* by 2^n, if the value is regarded as an unsigned integer or a two's complement signed integer. Overflow can occur without warning.

When the instruction is LSLS or when LSL #n, with non-zero *n*, is used in *Operand2* with the instructions MOVS, MVNS, ANDS, ORRS, ORNS, EORS, BICS, TEQ or TST, the carry flag is updated to the last bit shifted out, bit[32-*n*], of the register *Rm*. These instructions do not affect the carry flag when used with LSL #0.

- If *n* is 32 or more, then all the bits in the result are cleared to 0.
- If *n* is 33 or more and the carry flag is updated, it is updated to 0.

Figure 12-6. LSL #3

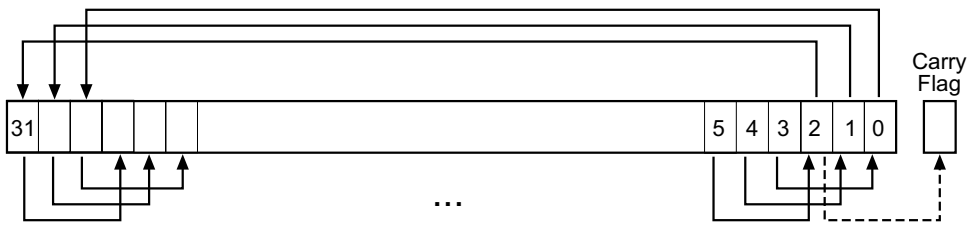
12.10.4.4 ROR

Rotate right by *n* bits moves the left-hand 32-*n* bits of the register *Rm*, to the right by *n* places, into the right-hand 32-*n* bits of the result. And it moves the right-hand *n* bits of the register into the left-hand *n* bits of the result. See Figure 12-7.

When the instruction is RORS or when ROR #*n* is used in *Operand2* with the instructions MOVS, MVNS, ANDS, ORRS, ORNS, EORS, BICS, TEQ or TST, the carry flag is updated to the last bit rotation, bit[*n*-1], of the register *Rm*.

- If *n* is 32, then the value of the result is same as the value in *Rm*, and if the carry flag is updated, it is updated to bit[31] of *Rm*.
- ROR with shift length, *n*, more than 32 is the same as ROR with shift length *n*-32.

Figure 12-7. ROR #3



12.11.1 ADR

Load PC-relative address.

12.11.1.1 Syntax

```
ADR{cond} Rd, label
```

where:

cond is an optional condition code, see “Conditional execution” on page 87.

Rd is the destination register.

label is a PC-relative expression. See “PC-relative expressions” on page 86.

12.11.1.2 Operation

ADR determines the address by adding an immediate value to the PC, and writes the result to the destination register.

ADR produces position-independent code, because the address is PC-relative.

If you use ADR to generate a target address for a BX or BLX instruction, you must ensure that bit[0] of the address you generate is set to1 for correct execution.

Values of *label* must be within the range of –4095 to +4095 from the address in the PC.

You might have to use the .W suffix to get the maximum offset range or to generate addresses that are not word-aligned. See “Instruction width selection” on page 88.

12.11.1.3 Restrictions

Rd must not be SP and must not be PC.

12.11.1.4 Condition flags

This instruction does not change the flags.

12.11.1.5 Examples

```
ADR     R1, TextMessage   ; Write address value of a location labelled as
                          ; TextMessage to R1
```

12.22.1 MPU Type Register

The TYPE register indicates whether the MPU is present, and if so, how many regions it supports. See the register summary in Table 12-35 on page 195 for its attributes. The bit assignments are:

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
IREGION							

15	14	13	12	11	10	9	8
DREGION							

7	6	5	4	3	2	1	0
Reserved							SEPARATE

- **IREGION**

Indicates the number of supported MPU instruction regions.

Always contains 0x00. The MPU memory map is unified and is described by the DREGION field.

- **DREGION**

Indicates the number of supported MPU data regions:

0x08 = Eight MPU regions.

- **SEPARATE**

Indicates support for unified or separate instruction and date memory maps:

0 = unified.

24.5 Multiplexed Signals

Table 24-2. Static Memory Controller (SMC) Multiplexed Signals

Multiplexed Signals		Related Function
NWR0	NWE	Byte-write or byte-select access, see Figure 24-4 "Memory Connection for an 8-bit Data Bus" and Figure 24-5 "Memory Connection for a 16-bit Data Bus"
A0	NBS0	8-bit or 16-bit data bus, see Section 24.9.1 ”Data Bus Width”
A22	NANDCLE	NAND Flash Command Latch Enable
A21	NANDALE	NAND Flash Address Latch Enable
NWR1	NBS1	Byte-write or byte-select access, see Figure 24-4 and Figure 24-5
A1	–	8-/16-bit data bus, see Section 24.9.1 ”Data Bus Width” Byte-write or byte-select access, see Figure 24-4 and Figure 24-5

24.6 Application Example

24.6.1 Implementation Examples

For Hardware implementation examples, refer to ATSAM3U-EK schematics which show examples of connection to an LCD module, PSRAM and NAND Flash.

24.6.2 Hardware Interface

Figure 24-2. SMC Connections to Static Memory Devices

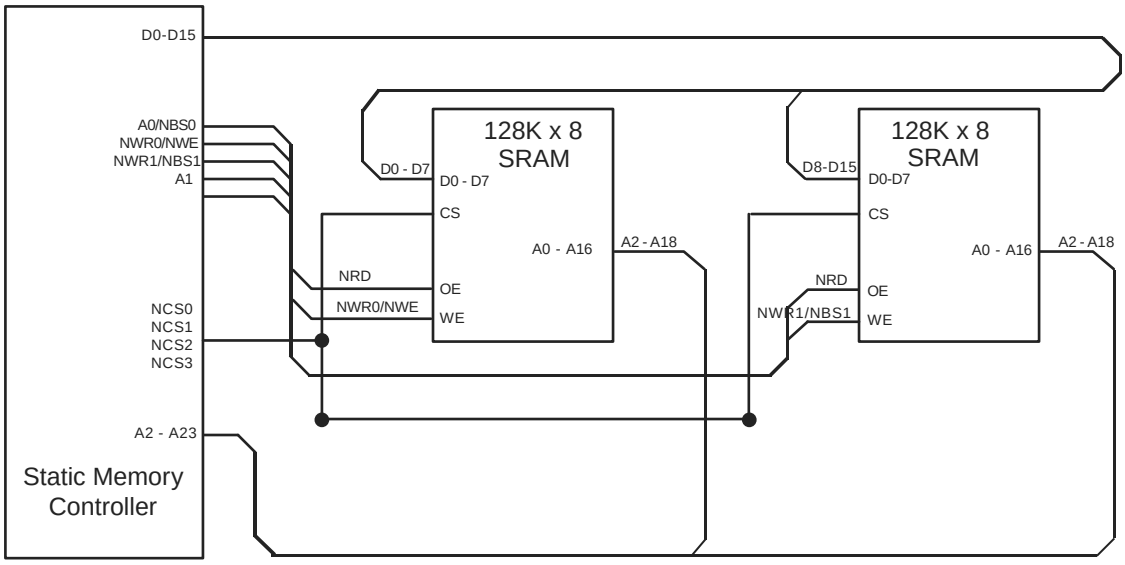
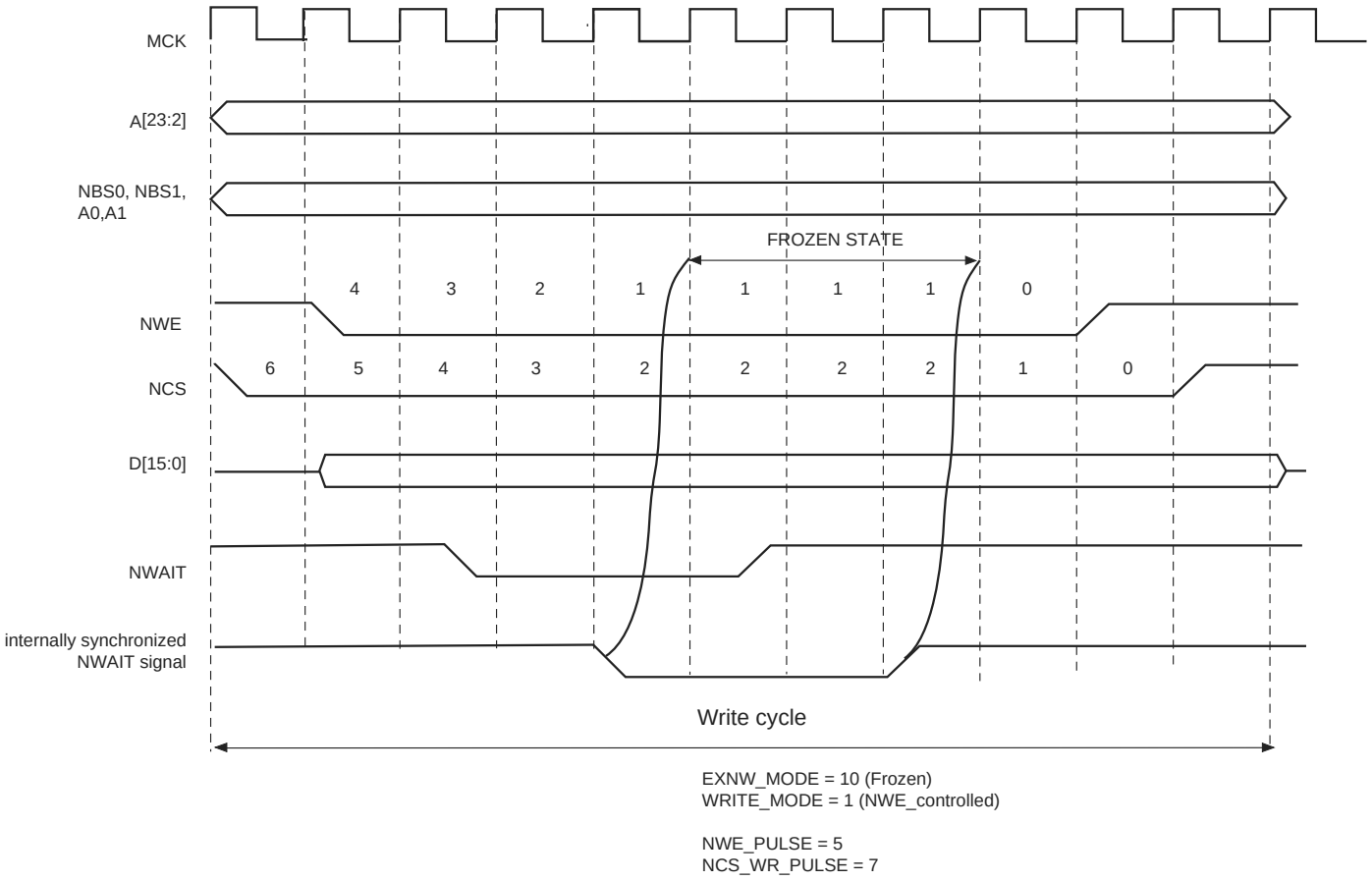


Figure 24-23. Write Access with NWAIT Assertion in Frozen Mode (EXNW_MODE = 10)



24.18.5 SMC NFC Interrupt Disable Register

Name: SMC_IDR

Address: 0x400E0010

Access: Write-only

Reset: 0x00000000

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	RB_EDGE0
23	22	21	20	19	18	17	16
NFCASE	AWB	UNDEF	DTOE	–	–	CMDDONE	XFRDONE
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	RB_FALL	RB_RISE	–	–	–	–

- **RB_RISE: Ready Busy Rising Edge Detection Interrupt Disable**

0: No effect.

1: Interrupt source is disabled.

- **RB_FALL: Ready Busy Falling Edge Detection Interrupt Disable**

0: No effect.

1: Interrupt source is disabled.

- **XFRDONE: Transfer Done Interrupt Disable**

0: No effect.

1: Interrupt source is disabled.

- **CMDDONE: Command Done Interrupt Disable**

0: No effect.

1: Interrupt source is disabled.

- **DTOE: Data Timeout Error Interrupt Disable**

0: No effect.

1: Interrupt source is disabled.

- **UNDEF: Undefined Area Access Interrupt Disable**

0: No effect.

1: Interrupt source is disabled.

- **AWB: Accessing While Busy Interrupt Disable**

0: No effect.

1: Interrupt source is disabled.

24.18.20 SMC Timings Register

Name: SMC_TIMINGSx [x=0..3]

Address: 0x400E007C [0], 0x400E0090 [1], 0x400E00A4 [2], 0x400E00B8 [3]

Access: Read-write

Reset: 0x00000000

31	30	29	28	27	26	25	24
NFSEL	RBNSEL			TWB			
23	22	21	20	19	18	17	16
–	–	–	–	TRR			
15	14	13	12	11	10	9	8
–	–	–	OCMS	TAR			
7	6	5	4	3	2	1	0
TADL				TCLR			

- **TCLR: CLE to REN Low Delay**

Command Latch Enable falling edge to Read Enable falling edge timing.

Latch Enable Falling to Read Enable Falling = (TCLR[3] * 64) + TCLR[2:0] clock cycles.

- **TADL: ALE to Data Start**

Last address latch cycle to the first rising edge of WEN for data input.

Last address latch to first rising edge of WEN = (TADL[3] * 64) + TADL[2:0] clock cycles.

- **TAR: ALE to REN Low Delay**

Address Latch Enable falling edge to Read Enable falling edge timing.

Address Latch Enable to Read Enable = (TAR[3] * 64) + TAR[2:0] clock cycles.

- **OCMS: Off Chip Memory Scrambling Enable**

When set to one, the memory scrambling is activated.

- **TRR: Ready to REN Low Delay**

Ready/Busy signal to Read Enable falling edge timing.

Read to REN = (TRR[3] * 64) + TRR[2:0] clock cycles.

- **TWB: WEN High to REN to Busy**

Write Enable rising edge to Ready/Busy falling edge timing.

Write Enable to Read/Busy = (TWB[3] * 64) + TWB[2:0] clock cycles.

- **RBNSEL: Ready/Busy Line Selection**

This field indicates the selected Ready/Busy Line from the RBN bundle.

- **NFSEL: NAND Flash Selection**

If this bit is set to one, the chip select is assigned to NAND Flash write enable and read enable lines drive the Error Correcting Code module.

27.14.21 PMC Write Protection Status Register

Name: PMC_WPSR

Address: 0x400E04E8

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 PMC_WPSR.

1: A write protection violation has occurred since the last read of the PMC_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.

29.7.19 PIO Multi-driver Disable Register

Name: PIO_MDDR

Address: 0x400E0C54 (PIOA), 0x400E0E54 (PIOB), 0x400E1054 (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

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

- **P0-P31: Multi Drive Disable.**

0 = No effect.

1 = Disables Multi Drive on the I/O line.

31.7.3.2 Master Mode Flow Diagram

Figure 31-6. Master Mode Flow Diagram

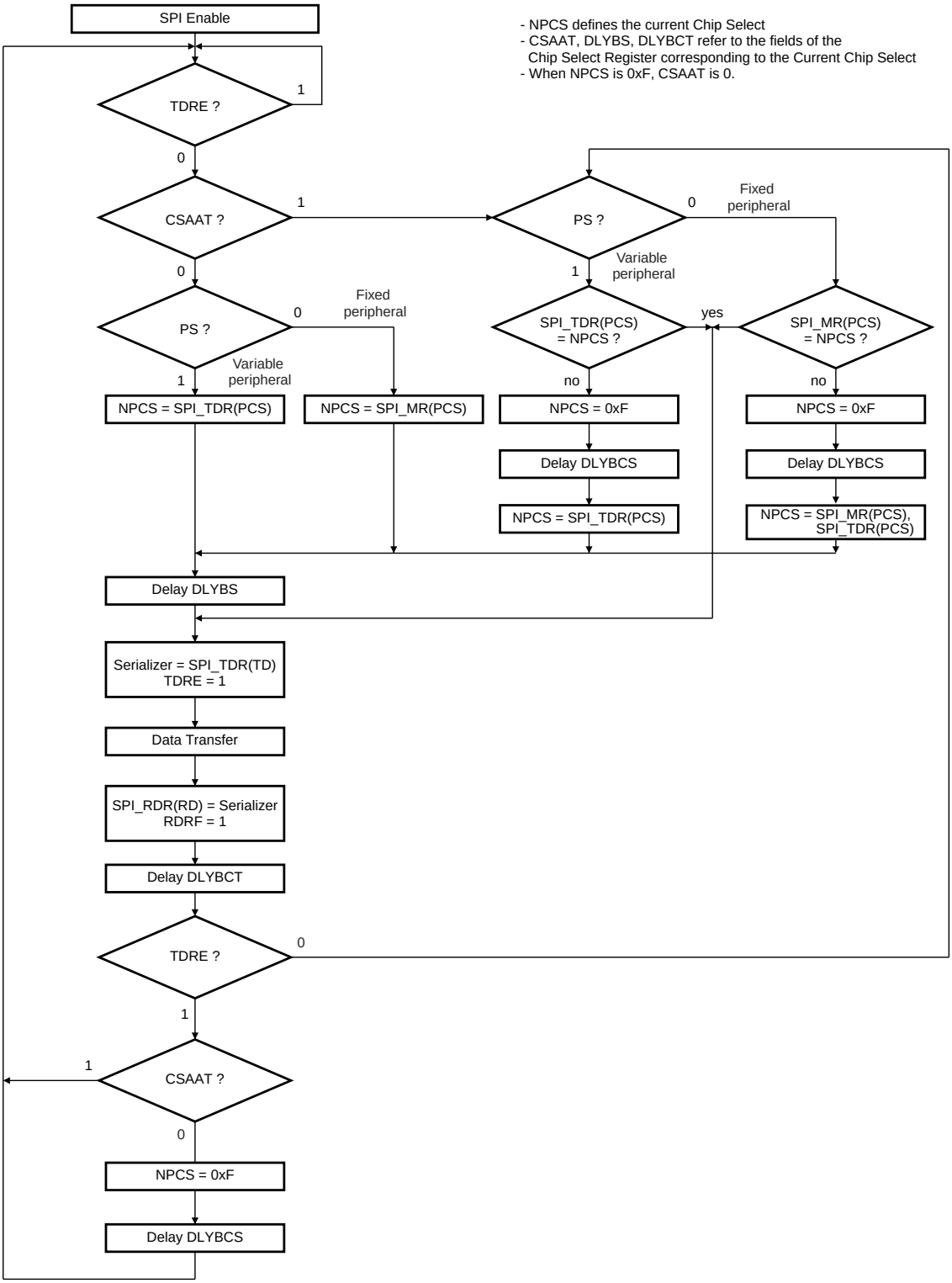
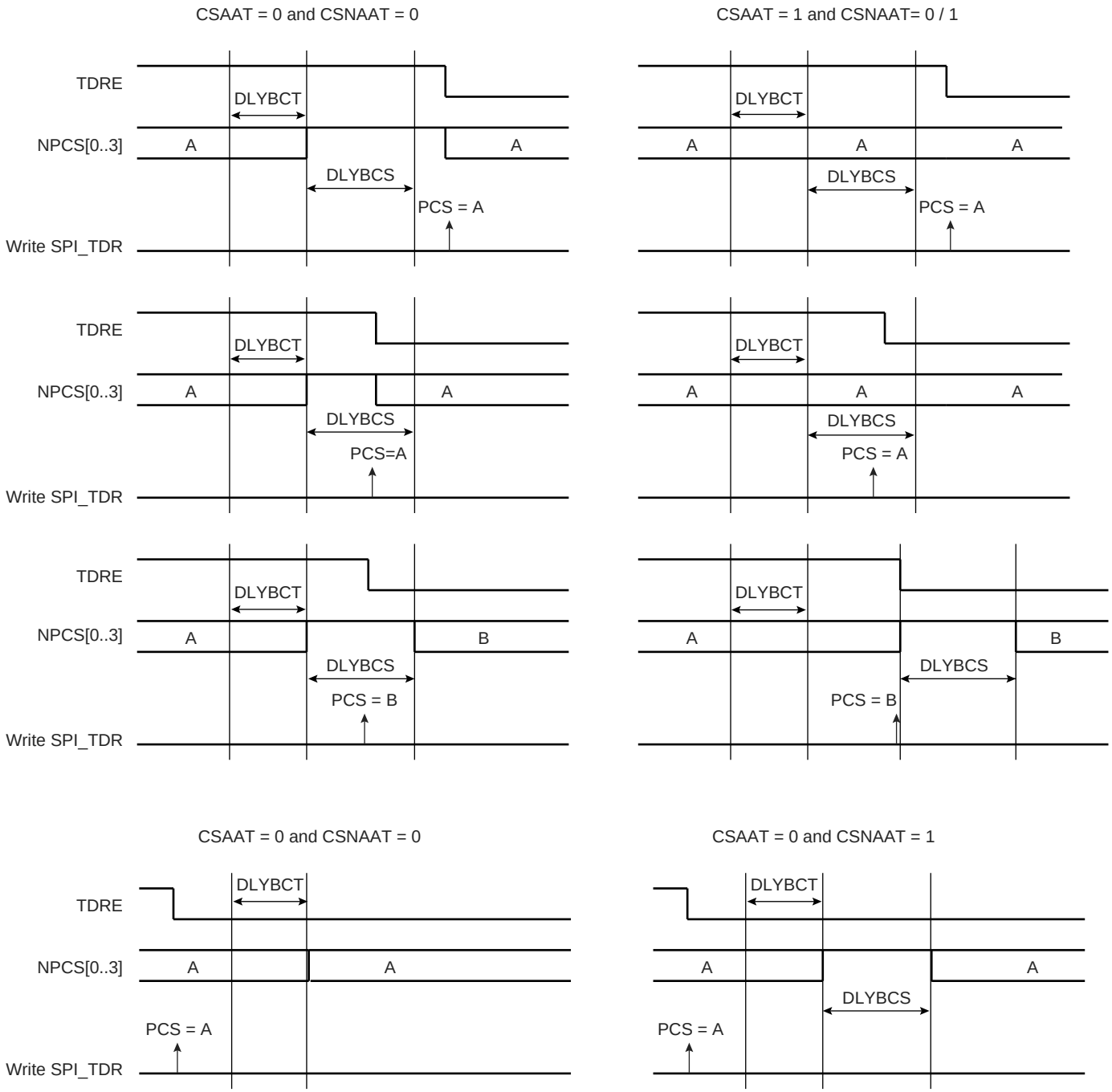


Figure 31-10. Peripheral Deselection



31.7.3.10Mode Fault Detection

A mode fault is detected when the SPI is programmed in Master Mode and a low level is driven by an external master on the NPCS0/NSS signal. In this case, multi-master configuration, NPCS0, MOSI, MISO and SPCK pins must be configured in open drain (through the PIO controller). When a mode fault is detected, the MODF bit in the SPI_SR is set until the SPI_SR is read and the SPI is automatically disabled until re-enabled by writing the SPIEN bit in the SPI_CR (Control Register) at 1.

By default, the Mode Fault detection circuitry is enabled. The user can disable Mode Fault detection by setting the MODFDIS bit in the SPI Mode Register (SPI_MR).

- **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.

37.7.12 PWM Sync Channels Update Period Update Register

Name: PWM_SCUPUPD

Address: 0x4008C030

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
–	–	–	–	UPRUPD			

This register acts as a double buffer for the UPR value. This prevents an unexpected automatic trigger of the update of synchronous channels.

- **UPRUPD: Update Period Update**

Defines the wanted time between each update of the synchronous channels if automatic trigger of the update is activated (UPDM = 1 or UPDM = 2 in “PWM Sync Channels Mode Register” on page 904). This time is equal to UPR+1 periods of the synchronous channels.

37.7.30 PWM Write Protect Status Register

Name: PWM_WPSR

Address: 0x4008C0E8

Access: Read-only

31	30	29	28	27	26	25	24
WPVSRC							

23	22	21	20	19	18	17	16
WPVSRC							

15	14	13	12	11	10	9	8
–	–	WPHWS5	WPHWS4	WPHWS3	WPHWS2	WPHWS1	WPHWS0

7	6	5	4	3	2	1	0
WPVS	–	WPSWS5	WPSWS4	WPSWS3	WPSWS2	WPSWS1	WPSWS0

- **WPSWSx: Write Protect SW Status**

0 = The Write Protect SW x of the register group x is disabled.

1 = The Write Protect SW x of the register group x is enabled.

- **WPHWSx: Write Protect HW Status**

0 = The Write Protect HW x of the register group x is disabled.

1 = The Write Protect HW x of the register group x is enabled.

- **WPVS: Write Protect Violation Status**

0 = No Write Protect violation has occurred since the last read of the PWM_WPSR register.

1 = At least one Write Protect violation has occurred since the last read of the PWM_WPSR register. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

- **WPVSRC: Write Protect Violation Source**

When WPVS is active, this field indicates the write-protected register (through address offset) in which a write access has been attempted.

Note: The two LSBs of the address offset of the write-protected register are not reported

Note: Reading PWM_WPSR automatically clears WPVS and WPVSRC fields.

42. Electrical Characteristics

42.1 Absolute Maximum Ratings

Table 42-1. Absolute Maximum Ratings*

Operating Temperature (Industrial).....-40°C to + 85°C
Storage Temperature.....-60°C to + 150°C
Voltage on Input Pins with Respect to Ground.....-0.3V to + 4.0V
Maximum Operating Voltage (VDDCORE) 2.0V
Maximum Operating Voltage (VDDIO) 4.0V
Total DC Output Current on all I/O lines 100-lead LQFP.....100 mA 144-lead LQFP .....130 mA 100-ball TFBGA .....100 mA 144-ball LFBGA .....130 mA

*NOTICE: Stresses beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. **Exposure to absolute maximum rating conditions for extended periods may affect device reliability.**

Figure 42-7. Current Consumption in Sleep Mode (AMP1) versus Master Clock Ranges (refer to Table 42-10)

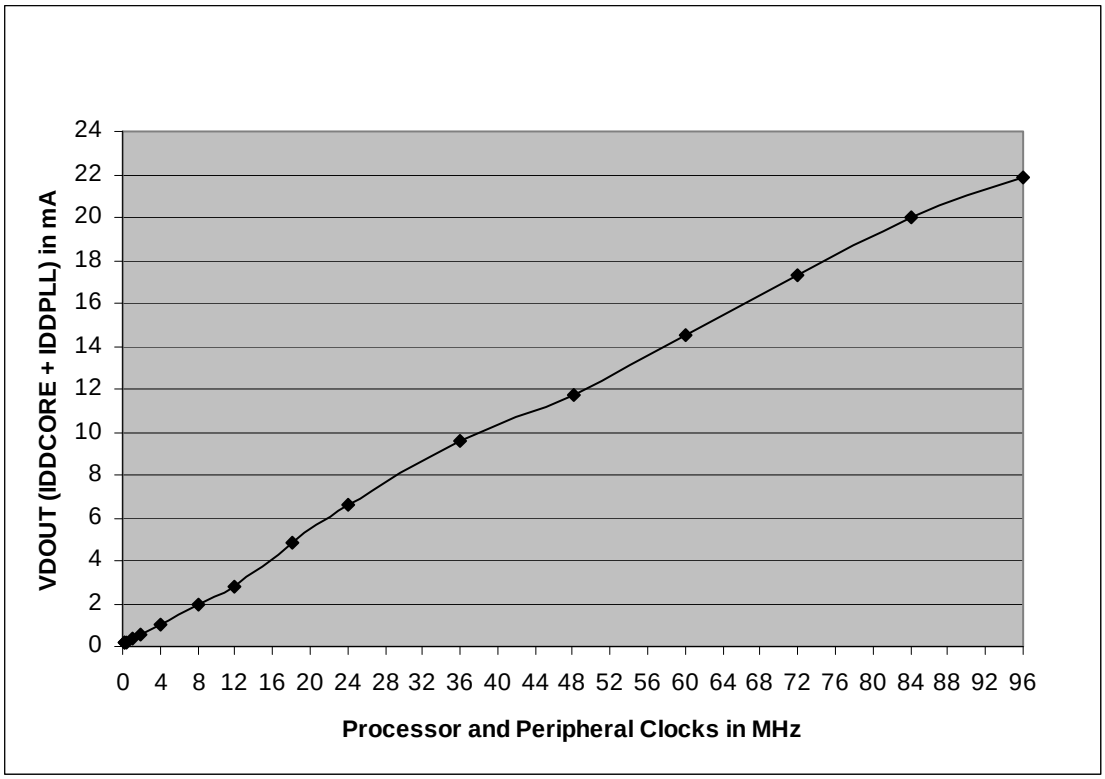


Table 42-11. Sleep Mode Current Consumption Versus Master Clock (MCK) Variation

Core Clock/MCK (MHz)	AMP1 (VDDOUT) Consumption	Unit
96	21.9	mA
84	20	
72	17.3	
60	14.5	
48	11.72	
36	9.6	
24	6.56	
18	4.8	
12	2.835	
8	1.937	
4	1.013	
2	0.567	
1	0.343	
0.5	0.23	
0.25	0.174	
0.125	0.146	

46.2.2 12-bit Analog-to-Digital Converter (ADC12B)

46.2.2.1 ADC12B: Current Consumption in Backup Mode on VDDANA

In Backup mode, the current consumption on VDDANA is around 1.0 mA instead of 0.1 µA

Problem Fix/Workaround

Four workarounds are possible:

1. Do not supply VDDANA and VDDIO in Backup mode using an external switch managed by SHDN pin.
2. Do not supply VDDANA in Backup mode using an external switch managed by the SHDN and set all PIOs with ADC inputs (PA22, PA30, PB3–PB8, PC15–PC18, PC28–C21) at low level (either externally or by software).
3. Do not supply VDDANA in Backup mode using an external switch managed by any PIO and set all PIOs with ADC inputs (PA22, PA30, PB3–PB8, PC15–PC18, PC28–C21) at low level (either externally or by software). Since the PIO state is preserved when in backup mode, any free PIO line can be used to switch off the external switch by driving the PIO line at low level (PIO is input, pull-up enabled after backup reset).
4. Use Wait mode instead of Backup mode.

46.2.2.2 ADC: Trigger Launches only One Conversion

A start command initiates a conversion sequence of one channel but not all activated channels as expected.

Problem Fix/Workaround

Send as many start commands as the number of activated channels, or use free run mode.

46.2.2.3 ADC: Wrong First Conversions

The first conversions done by the ADC may be erroneous if the maximum gain (x4 in single ended or x2 in differential mode) is not used. The issue appears after the power-up or if a conversion has not occured for 1 minute.

Problem Fix/Workaround

Three workarounds are possible:

1. Perform 16 dummy conversions on one channel (whatever conditions used in term of setup of gain, single/differential, offset, and channel selected). The next conversions will be correct for any channels and any settings. Note that these dummy conversions need to be performed if no conversion has occured for 1 minute or for a new chip startup.
2. Perform a dummy conversion on a single ended channel on which an external voltage of ADVREF/2 (±10%) is applied. Use the following conditions for this conversion: gain at 4, offset set at 1. The next conversions will be correct for any channels and any settings. Note that this dummy conversion needs to be performed if no conversion has occured for 1 minute or for a new chip startup.
3. Perform a dummy conversion on a differential channel on which the two inputs are connected together and connected to any voltage (from 0 to ADVREF). Use the following conditions for this conversion: gain at 4, offset set at 1. The next conversions will be correct for any channels and any settings. Note that this dummy conversion needs to be performed if no conversion has occured for 1 minute or for a new chip startup.

46.2.3 Power Management Controller (PMC)

46.2.3.1 PMC: Main Oscillator Frequency selection if the Main On-chip RC Oscillator is OFF

When the 4/8/12 MHz RC Oscillator is off, the frequency selection (MOSCRCF field in CKGR_MOR) can not be changed. The register can be written but the modification to MOSCRCF will not be taken into account.

Problem Fix/Workaround

Modify MOSCRCF while 4/8/12 MHz RC Oscillator is on (MOSCREN = 1).