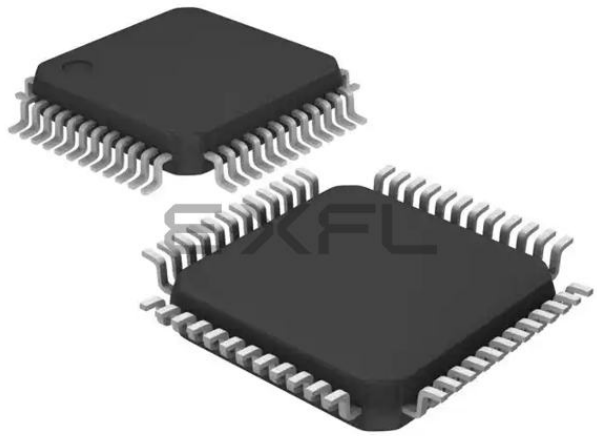

Welcome to E-XFL.COM

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	S12Z
Core Size	16-Bit
Speed	32MHz
Connectivity	CANbus, I²C, IrDA, LINbus, SCI, SPI, UART/USART
Peripherals	LVD, POR, PWM, WDT
Number of I/O	34
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	5.5V ~ 18V
Data Converters	A/D 10x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	48-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s912zvl64f0vlfr

18.4.3 Data Format . . . 521
18.4.4 Baud Rate Generation . . . 523
18.4.5 Transmitter . . . 524
18.4.6 Receiver . . . 529
18.4.7 Single-Wire Operation . . . 537
18.4.8 Loop Operation . . . 538
18.5 Initialization/Application Information . . . 538
18.5.1 Reset Initialization . . . 538
18.5.2 Modes of Operation . . . 539
18.5.3 Interrupt Operation . . . 539
18.5.4 Recovery from Wait Mode . . . 541
18.5.5 Recovery from Stop Mode . . . 541

Chapter 19
Serial Peripheral Interface (S12SPIV5)

19.1 Introduction . . . 543
19.1.1 Glossary of Terms . . . 543
19.1.2 Features . . . 543
19.1.3 Modes of Operation . . . 543
19.1.4 Block Diagram . . . 544
19.2 External Signal Description . . . 545
19.2.1 MOSI — Master Out/Slave In Pin . . . 545
19.2.2 MISO — Master In/Slave Out Pin . . . 546
19.2.3 $\overline{SS}$ — Slave Select Pin . . . 546
19.2.4 SCK — Serial Clock Pin . . . 546
19.3 Memory Map and Register Definition . . . 546
19.3.1 Module Memory Map . . . 546
19.3.2 Register Descriptions . . . 547
19.4 Functional Description . . . 555
19.4.1 Master Mode . . . 556
19.4.2 Slave Mode . . . 557
19.4.3 Transmission Formats . . . 558
19.4.4 SPI Baud Rate Generation . . . 563
19.4.5 Special Features . . . 564
19.4.6 Error Conditions . . . 565
19.4.7 Low Power Mode Options . . . 566

Chapter 20
Inter-Integrated Circuit (IICV3)

20.1 Introduction . . . 569
20.1.1 Features . . . 569
20.1.2 Modes of Operation . . . 570
20.1.3 Block Diagram . . . 570
20.2 External Signal Description . . . 570

same time, the highest ranked module according the predefined priority scheme in Table 2-1 will take precedence on the pin.

Table 2-28 shows the effect of enabled peripheral features on I/O state and enabled pull devices.

Table 2-28. Effect of Enabled Features

Enabled Feature[1]	Related Signal(s)	Effect on I/O state	Effect on enabled pull device
CPMU OSC	EXTAL, XTAL	CPMU takes control	Forced off
TIMx output compare	IOCx_y	Forced output	Forced off
TIMx input capture	IOCx_y	None[2]	None[3]
SPI0	MISO0, MOSI0, SCK0, $\overline{SS0}$	Controlled input/output	Forced off if output
SCIx transmitter	TXDx	Forced output	Forced off
SCIx receiver	RXDx	Forced input	None[3]
IIC0	SDA0, SCL0	Forced open-drain	Pull-down forced off
S12ZDBG	DBGEEV	None[2]	None[3]
PWM channel	PWMx	Forced output	Forced off
ADC0	ANx	None[2 4]	None[3]
	VRH, VRL		
ACMP	ACMP0_0, ACMP0_1	None[2 4]	None[3]
	ACMPO0	Forced output	Forced off
DAC	AMPP, AMPM	None[2 4]	None[3]
	AMP	Forced analog output, digital output forced off	Forced off
PGA	PGA_IN0, PGA_IN1, PGA_REF	None[2 4]	None[3]
IRQ	$\overline{IRQ}$	Forced input	None[3]
XIRQ	$\overline{XIRQ}$	Forced input	None[3]
MSCAN0	TXCAN0	Forced output	Forced off
	RXCAN0	Forced input	Pull-down forced off
LINPHY0	LPTXD0	Forced input	None[3]
	LPRXD0	Forced output	Forced off

1 If applicable the appropriate routing configuration must be set for the signals to take effect on the pins.

2 DDR maintains control

3 PER/PPS maintain control

4 To use the digital input function the related bit in Digital Input Enable Register (DIENADx) must be set to logic level “1”.

5.3.2.2 BDC Control Status Register Low (BDCCSRL)

Register Address: This register is not in the device memory map. It is accessible using BDC inherent addressing commands

	7	6	5	4	3	2	1	0
R W	WAIT	STOP	RAMWF	OVRUN	NORESP	RDINV	ILLACC	ILLCMD
Reset	0	0	0	0	0	0	0	0

Figure 5-4. BDC Control Status Register Low (BDCCSRL)

Read: BDC access only.

Write: Bits [7:5], [3:0] BDC access only, restricted to flag clearing by writing a “1” to the bit position.
Write: Bit 4 never. It can only be cleared by a SYNC pulse.

If ACK handshaking is enabled then BDC commands with ACK causing a BDCCSRL[3:1] flag setting condition also generate a long ACK pulse. Subsequent commands that are executed correctly generate a normal ACK pulse. Subsequent commands that are not correctly executed generate a long ACK pulse. The first ACK pulse after WAIT or STOP have been set also generates a long ACK. Subsequent ACK pulses are normal, whilst STOP and WAIT remain set.

Long ACK pulses are not immediately generated if an overrun condition is caused by the host driving the BKGD pin low whilst a target ACK is pending, because this would conflict with an attempted host transmission following the BKGD edge. When a whole byte has been received following the offending BKGD edge, the OVRUN bit is still set, forcing subsequent ACK pulses to be long.

Unimplemented BDC opcodes causing the ILLCMD bit to be set do not generate a long ACK because this could conflict with further transmission from the host. If the ILLCMD is set for another reason, then a long ACK is generated for the current command if it is a BDC command with ACK.

Table 5-6. BDCCSRL Field Descriptions

Field	Description
7 WAIT	**WAIT Indicator Flag** — Indicates that the device entered wait mode. Writing a “1” to this bit whilst in wait mode has no effect. Writing a “1” after exiting wait mode, clears the bit. 0 Device did not enter wait mode 1 Device entered wait mode.
6 STOP	**STOP Indicator Flag** — Indicates that the CPU requested stop mode following a STOP instruction. Writing a “1” to this bit whilst not in stop mode clears the bit. Writing a “1” to this bit whilst in stop mode has no effect. This bit can only be set when the BDC is enabled. 0 Device did not enter stop mode 1 Device entered stop mode.
5 RAMWF	**RAM Write Fault** — Indicates an ECC double fault during a BDC write access to RAM. Writing a “1” to this bit, clears the bit. 0 No RAM write double fault detected. 1 RAM write double fault detected.

Table 5-6. BDCCSRL Field Descriptions (continued)

Field	Description
0 ILLCMD	**Illegal Command Flag** — Indicates an illegal BDC command. This bit is set in the following cases: When an unimplemented BDC command opcode is received. When a DUMP_MEM{_WS}, FILL_MEM{_WS} or READ_SAME{_WS} is attempted in an illegal sequence. When an active BDM command is received whilst BDM is not active When a non Always-available command is received whilst the BDC is disabled or a flash mass erase is ongoing. When a non Always-available command is received whilst the device is secure Read commands return a value of 0xEE for each data byte Writing a “1” to this bit, clears the bit. 0 No illegal command detected. 1 Illegal BDC command detected.

5.4 Functional Description

5.4.1 Security

If the device resets with the system secured, the device clears the BDCCSR UNSEC bit. In the secure state BDC access is restricted to the BDCCSR register. A mass erase can be requested using the ERASE_FLASH command. If the mass erase is completed successfully, the device programs the security bits to the unsecure state and sets the BDC UNSEC bit. If the mass erase is unsuccessful, the device remains secure and the UNSEC bit is not set.

For more information regarding security, please refer to device specific security information.

5.4.2 Enabling BDC And Entering Active BDM

BDM can be activated only after being enabled. BDC is enabled by setting the ENBDC bit in the BDCCSR register, via the single-wire interface, using the command WRITE_BDCCSR.

After being enabled, BDM is activated by one of the following[1]:

- The BDC BACKGROUND command
- A CPU BGND instruction
- The DBG Breakpoint mechanism

Alternatively BDM can be activated directly from reset when resetting into Special Single Chip Mode.

The BDC is ready for receiving the first command 10 core clock cycles after the deassertion of the internal reset signal. This is delayed relative to the external pin reset as specified in the device reset documentation. On S12Z devices an NVM initialization phase follows reset. During this phase the BDC commands classified as always available are carried out immediately, whereas other BDC commands are subject to delayed response due to the NVM initialization phase.

NOTE

After resetting into SSC mode, the initial PC address must be supplied by the host using the WRITE_Rn command before issuing the GO command.

1. BDM active immediately out of special single-chip reset.

Table 7-4. DBGC2 Field Descriptions

Field	Description
1–0 ABCM[1:0]	**A and B Comparator Match Control** — These bits determine the A and B comparator match mapping as described in Table 7-5.

Table 7-5. ABCM Encoding

ABCM	Description
00	Match0 mapped to comparator A match..... Match1 mapped to comparator B match.
01	Match0 mapped to comparator A/B inside range..... Match1 disabled.
10	Match0 mapped to comparator A/B outside range..... Match1 disabled.
11	Reserved[1]

[1] Currently defaults to Match0 mapped to inside range: Match1 disabled

7.3.2.3 Debug State Control Register 1 (DBGSCR1)

Address: 0x0107

	7	6	5	4	3	2	1	0
R	C3SC1	C3SC0	0	0	C1SC1	C1SC0	C0SC1	C0SC0
W								
Reset	0	0	0	0	0	0	0	0

Figure 7-6. Debug State Control Register 1 (DBGSCR1)

Read: Anytime.

Write: If DBG is not armed.

The state control register 1 selects the targeted next state whilst in State1. The matches refer to the outputs of the comparator match control logic as depicted in Figure 7-1 and described in Section 7.3.2.8, “Debug Comparator A Control Register (DBGACTL)”. Comparators must be enabled by setting the comparator enable bit in the associated DBGXCTL control register.

Table 7-7. DBGSCR1 Field Descriptions

Field	Description
1–0 C0SC[1:0]	Channel 0 State Control. These bits select the targeted next state whilst in State1 following a match0.
3–2 C1SC[1:0]	Channel 1 State Control. These bits select the targeted next state whilst in State1 following a match1.
7–6 C3SC[1:0]	Channel 3 State Control. If EEVE !=10, these bits select the targeted next state whilst in State1 following a match3. If EEVE = 10, these bits select the targeted next state whilst in State1 following an external event.

10.5.2.2 ADC Control Register 1 (ADCCTL_1)

Module Base + 0x0001

	7	6	5	4	3	2	1	0
R	CSL_BMOD	RVL_BMOD	SMOD_ACC	AUT_RSTA	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 10-5. ADC Control Register 1 (ADCCTL_1)

Read: Anytime

Write:

- Bit CSL_BMOD and RVL_BMOD writable if bit ADC_EN clear or bit SMOD_ACC set
- Bit SMOD_ACC only writable in MCU Special Mode
- Bit AUT_RSTA writable anytime

Table 10-5. ADCCTL_1 Field Descriptions

Field	Description
7 CSL_BMOD	**CSL Buffer Mode Select Bit** — This bit defines the CSL buffer mode. This bit is only writable if ADC_EN is clear. 0 CSL single buffer mode. 1 CSL double buffer mode.
6 RVL_BMOD	**RVL Buffer Mode Select Bit** — This bit defines the RVL buffer mode. 0 RVL single buffer mode 1 RVL double buffer mode
5 SMOD_ACC	**Special Mode Access Control Bit** — This bit controls register access rights in MCU Special Mode. This bit is automatically cleared when leaving MCU Special Mode. Note: When this bit is set also the ADCCMD register is writeable via the data bus to allow modification of the current command for debugging purpose. But this is only possible if the current command is not already processed (conversion not started). Please see access details given for each register. Care must be taken when modifying ADC registers while bit SMOD_ACC is set to not corrupt a possible ongoing conversion. 0 Normal user access - Register write restrictions exist as specified for each bit. 1 Special access - Register write restrictions are lifted.
4 AUT_RSTA	**Automatic Restart Event after exit from MCU Stop and Wait Mode (SWAI set)** — This bit controls if a Restart Event is automatically generated after exit from MCU Stop Mode or Wait Mode with bit SWAI set. It can be configured for ADC conversion flow control mode “Trigger Mode” and “Restart Mode” (anytime during application runtime). 0 No automatic Restart Event after exit from MCU Stop Mode. 1 Automatic Restart Event occurs after exit from MCU Stop Mode.

Table 12-6. Amplifier Gain

PGAGAIN[3:0]	Gain A_{PGA}
0000	10x
0001	20x
0010	40x
0011	80x
others	Reserved

12.3.2.4 PGA Offset Register (PGAOFFSET)

Module Base + 0x0003 — Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	0		PGAOFFSET[5:0]					
W								
Reset	0	0	0	0	0	0	0	0

Figure 12-6. PGA Offset Register (PGAOFFSET

[1] Read: Anytime
Write: Anytime

Table 12-7. PGAOFFSET Field Description

Field	Description
5:0 PGAOFFSET [5:0]	**PGA Offset** — These register bits select the offset correction for the PGA, see Table 12-8., "Offset Compensation and 12.4.2, "Offset Compensation.

Table 12-8. Offset Compensation

PGAOFFSET[5:3]	ΔV_{OUT}
0x011	- 3*V_{step_H}
0x010	- 2*V_{step_H}
0x001	- 1*V_{step_H}
0x000	0
0x111	+ 1*V_{step_H}
0x110	+ 2*V_{step_H}
0x101	+ 3*V_{step_H}
0x100	0

PGAOFFSET[2:0]	ΔV_{OUT}
0x011	- 3*V_{step_L}
0x010	- 2*V_{step_L}
0x001	- 1*V_{step_L}
0x000	0
0x111	+ 1*V_{step_L}
0x110	+ 2*V_{step_L}
0x101	+ 3*V_{step_L}
0x100	0

NOTE

The CANTBSEL register is held in the reset state when the initialization mode is active (INITRQ = 1 and INITAK=1). This register is writable when not in initialization mode (INITRQ = 0 and INITAK = 0).

Table 13-17. CANTBSEL Register Field Descriptions

Field	Description
2-0 TX[2:0]	**Transmit Buffer Select** — The lowest numbered bit places the respective transmit buffer in the CANTXFG register space (e.g., TX1 = 1 and TX0 = 1 selects transmit buffer TX0; TX1 = 1 and TX0 = 0 selects transmit buffer TX1). Read and write accesses to the selected transmit buffer will be blocked, if the corresponding TXEx bit is cleared and the buffer is scheduled for transmission (see Section 13.3.2.7, “MSCAN Transmitter Flag Register (CANTFLG)”). 0 The associated message buffer is deselected 1 The associated message buffer is selected, if lowest numbered bit

The following gives a short programming example of the usage of the CANTBSEL register:

To get the next available transmit buffer, application software must read the CANTFLG register and write this value back into the CANTBSEL register. In this example Tx buffers TX1 and TX2 are available. The value read from CANTFLG is therefore 0b0000_0110. When writing this value back to CANTBSEL, the Tx buffer TX1 is selected in the CANTXFG because the lowest numbered bit set to 1 is at bit position 1. Reading back this value out of CANTBSEL results in 0b0000_0010, because only the lowest numbered bit position set to 1 is presented. This mechanism eases the application software’s selection of the next available Tx buffer.

- LDAA CANTFLG; value read is 0b0000_0110
- STAA CANTBSEL; value written is 0b0000_0110
- LDAA CANTBSEL; value read is 0b0000_0010

If all transmit message buffers are deselected, no accesses are allowed to the CANTXFG registers.

13.3.2.12 MSCAN Identifier Acceptance Control Register (CANIDAC)

The CANIDAC register is used for identifier acceptance control as described below.

Module Base + 0x000B — Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	0	0	IDAM1	IDAM0	0	IDHIT2	IDHIT1	IDHIT0
W								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 13-15. MSCAN Identifier Acceptance Control Register (CANIDAC)

1 Read: Anytime
Write: Anytime in initialization mode (INITRQ = 1 and INITAK = 1), except bits IDHITx, which are read-only

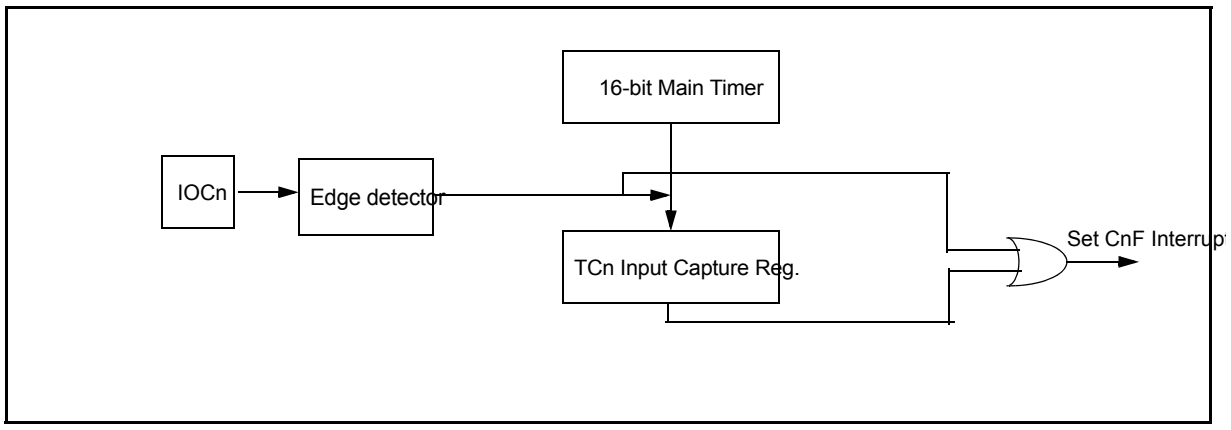

Figure 15-2. Interrupt Flag Setting

15.2 External Signal Description

The TIM16B6CV3 module has a selected number of external pins. Refer to device specification for exact number.

15.2.1 IOC5 - IOC0 — Input Capture and Output Compare Channel 5-0

Those pins serve as input capture or output compare for TIM16B6CV3 channel .

NOTE

For the description of interrupts see Section 15.6, “Interrupts”.

15.3 Memory Map and Register Definition

This section provides a detailed description of all memory and registers.

15.3.1 Module Memory Map

The memory map for the TIM16B6CV3 module is given below in Figure 15-3. The address listed for each register is the address offset. The total address for each register is the sum of the base address for the TIM16B6CV3 module and the address offset for each register.

15.3.2 Register Descriptions

This section consists of register descriptions in address order. Each description includes a standard register diagram with an associated figure number. Details of register bit and field function follow the register diagrams, in bit order.

The Prescaler can be calculated as follows depending on logical value of the PTPS[7:0] and PRNT bit:

PRNT = 1 : Prescaler = PTPS[7:0] + 1

Table 15-17. Precision Timer Prescaler Selection Examples when PRNT = 1

PTPS7	PTPS6	PTPS5	PTPS4	PTPS3	PTPS2	PTPS1	PTPS0	Prescale Factor
0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1	2
0	0	0	0	0	0	1	0	3
0	0	0	0	0	0	1	1	4
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
0	0	0	1	0	0	1	1	20
0	0	0	1	0	1	0	0	21
0	0	0	1	0	1	0	1	22
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
1	1	1	1	1	1	0	0	253
1	1	1	1	1	1	0	1	254
1	1	1	1	1	1	1	0	255
1	1	1	1	1	1	1	1	256

15.4 Functional Description

This section provides a complete functional description of the timer TIM16B6CV3 block. Please refer to the detailed timer block diagram in Figure 15-22 as necessary.

Table 17-10. PWMCTL Field Descriptions

Note: Bits related to available channels have functional significance. Writing to unavailable bits has no effect. Read from unavailable bits return a zero

Field	Description
7 CON67	Concatenate Channels 6 and 7 0 Channels 6 and 7 are separate 8-bit PWMs. 1 Channels 6 and 7 are concatenated to create one 16-bit PWM channel. Channel 6 becomes the high order byte and channel 7 becomes the low order byte. Channel 7 output pin is used as the output for this 16-bit PWM (bit 7 of port PWMP). Channel 7 clock select control-bit determines the clock source, channel 7 polarity bit determines the polarity, channel 7 enable bit enables the output and channel 7 center aligned enable bit determines the output mode.
6 CON45	Concatenate Channels 4 and 5 0 Channels 4 and 5 are separate 8-bit PWMs. 1 Channels 4 and 5 are concatenated to create one 16-bit PWM channel. Channel 4 becomes the high order byte and channel 5 becomes the low order byte. Channel 5 output pin is used as the output for this 16-bit PWM (bit 5 of port PWMP). Channel 5 clock select control-bit determines the clock source, channel 5 polarity bit determines the polarity, channel 5 enable bit enables the output and channel 5 center aligned enable bit determines the output mode.
5 CON23	Concatenate Channels 2 and 3 0 Channels 2 and 3 are separate 8-bit PWMs. 1 Channels 2 and 3 are concatenated to create one 16-bit PWM channel. Channel 2 becomes the high order byte and channel 3 becomes the low order byte. Channel 3 output pin is used as the output for this 16-bit PWM (bit 3 of port PWMP). Channel 3 clock select control-bit determines the clock source, channel 3 polarity bit determines the polarity, channel 3 enable bit enables the output and channel 3 center aligned enable bit determines the output mode.
4 CON01	Concatenate Channels 0 and 1 0 Channels 0 and 1 are separate 8-bit PWMs. 1 Channels 0 and 1 are concatenated to create one 16-bit PWM channel. Channel 0 becomes the high order byte and channel 1 becomes the low order byte. Channel 1 output pin is used as the output for this 16-bit PWM (bit 1 of port PWMP). Channel 1 clock select control-bit determines the clock source, channel 1 polarity bit determines the polarity, channel 1 enable bit enables the output and channel 1 center aligned enable bit determines the output mode.
3 PSWAI	PWM Stops in Wait Mode — Enabling this bit allows for lower power consumption in wait mode by disabling the input clock to the prescaler. 0 Allow the clock to the prescaler to continue while in wait mode. 1 Stop the input clock to the prescaler whenever the MCU is in wait mode.
2 PFRZ	**PWM Counters Stop in Freeze Mode** — In freeze mode, there is an option to disable the input clock to the prescaler by setting the PFRZ bit in the PWMCTL register. If this bit is set, whenever the MCU is in freeze mode, the input clock to the prescaler is disabled. This feature is useful during emulation as it allows the PWM function to be suspended. In this way, the counters of the PWM can be stopped while in freeze mode so that once normal program flow is continued, the counters are re-enabled to simulate real-time operations. Since the registers can still be accessed in this mode, to re-enable the prescaler clock, either disable the PFRZ bit or exit freeze mode. 0 Allow PWM to continue while in freeze mode. 1 Disable PWM input clock to the prescaler whenever the part is in freeze mode. This is useful for emulation.

17.3.2.7 PWM Clock A/B Select Register (PWMCLKAB)

Each PWM channel has a choice of four clocks to use as the clock source for that channel as described below.

On the front end of the PWM timer, the clock is enabled to the PWM circuit by the PWMEx bit being high. There is an edge-synchronizing circuit to guarantee that the clock will only be enabled or disabled at an edge. When the channel is disabled (PWMEx = 0), the counter for the channel does not count.

17.4.2.2 PWM Polarity

Each channel has a polarity bit to allow starting a waveform cycle with a high or low signal. This is shown on the block diagram Figure 17-16 as a mux select of either the Q output or the $\overline{Q}$ output of the PWM output flip flop. When one of the bits in the PWMPOL register is set, the associated PWM channel output is high at the beginning of the waveform, then goes low when the duty count is reached. Conversely, if the polarity bit is zero, the output starts low and then goes high when the duty count is reached.

17.4.2.3 PWM Period and Duty

Dedicated period and duty registers exist for each channel and are double buffered so that if they change while the channel is enabled, the change will NOT take effect until one of the following occurs:

- The effective period ends
- The counter is written (counter resets to $00)
- The channel is disabled

In this way, the output of the PWM will always be either the old waveform or the new waveform, not some variation in between. If the channel is not enabled, then writes to the period and duty registers will go directly to the latches as well as the buffer.

A change in duty or period can be forced into effect “immediately” by writing the new value to the duty and/or period registers and then writing to the counter. This forces the counter to reset and the new duty and/or period values to be latched. In addition, since the counter is readable, it is possible to know where the count is with respect to the duty value and software can be used to make adjustments

NOTE

When forcing a new period or duty into effect immediately, an irregular PWM cycle can occur.

Depending on the polarity bit, the duty registers will contain the count of either the high time or the low time.

17.4.2.4 PWM Timer Counters

Each channel has a dedicated 8-bit up/down counter which runs at the rate of the selected clock source (see Section 17.4.1, “PWM Clock Select” for the available clock sources and rates). The counter compares to two registers, a duty register and a period register as shown in Figure 17-16. When the PWM counter matches the duty register, the output flip-flop changes state, causing the PWM waveform to also change state. A match between the PWM counter and the period register behaves differently depending on what output mode is selected as shown in Figure 17-16 and described in Section 17.4.2.5, “Left Aligned Outputs” and Section 17.4.2.6, “Center Aligned Outputs”.

When the transmit shift register is not transmitting a frame, the TXD pin goes to the idle condition, logic 1. If at any time software clears the TE bit in SCI control register 2 (SCICR2), the transmitter enable signal goes low and the transmit signal goes idle.

If software clears TE while a transmission is in progress (TC = 0), the frame in the transmit shift register continues to shift out. To avoid accidentally cutting off the last frame in a message, always wait for TDRE to go high after the last frame before clearing TE.

To separate messages with preambles with minimum idle line time, use this sequence between messages:

1. Write the last byte of the first message to SCIDRH/L.
2. Wait for the TDRE flag to go high, indicating the transfer of the last frame to the transmit shift register.
3. Queue a preamble by clearing and then setting the TE bit.
4. Write the first byte of the second message to SCIDRH/L.

18.4.5.3 Break Characters

Writing a logic 1 to the send break bit, SBK, in SCI control register 2 (SCICR2) loads the transmit shift register with a break character. A break character contains all logic 0s and has no start, stop, or parity bit. Break character length depends on the M bit in SCI control register 1 (SCICR1). As long as SBK is at logic 1, transmitter logic continuously loads break characters into the transmit shift register. After software clears the SBK bit, the shift register finishes transmitting the last break character and then transmits at least one logic 1. The automatic logic 1 at the end of a break character guarantees the recognition of the start bit of the next frame.

The SCI recognizes a break character when there are 10 or 11(M = 0 or M = 1) consecutive zero received. Depending if the break detect feature is enabled or not receiving a break character has these effects on SCI registers.

If the break detect feature is disabled (BKDFE = 0):

- Sets the framing error flag, FE
- Sets the receive data register full flag, RDRF
- Clears the SCI data registers (SCIDRH/L)
- May set the overrun flag, OR, noise flag, NF, parity error flag, PE, or the receiver active flag, RAF (see 3.4.4 and 3.4.5 SCI Status Register 1 and 2)

If the break detect feature is enabled (BKDFE = 1) there are two scenarios[1]

The break is detected right from a start bit or is detected during a byte reception.

- Sets the break detect interrupt flag, BKDIF
- Does not change the data register full flag, RDRF or overrun flag OR
- Does not change the framing error flag FE, parity error flag PE.
- Does not clear the SCI data registers (SCIDRH/L)
- May set noise flag NF, or receiver active flag RAF.

1. A Break character in this context are either 10 or 11 consecutive zero received bits

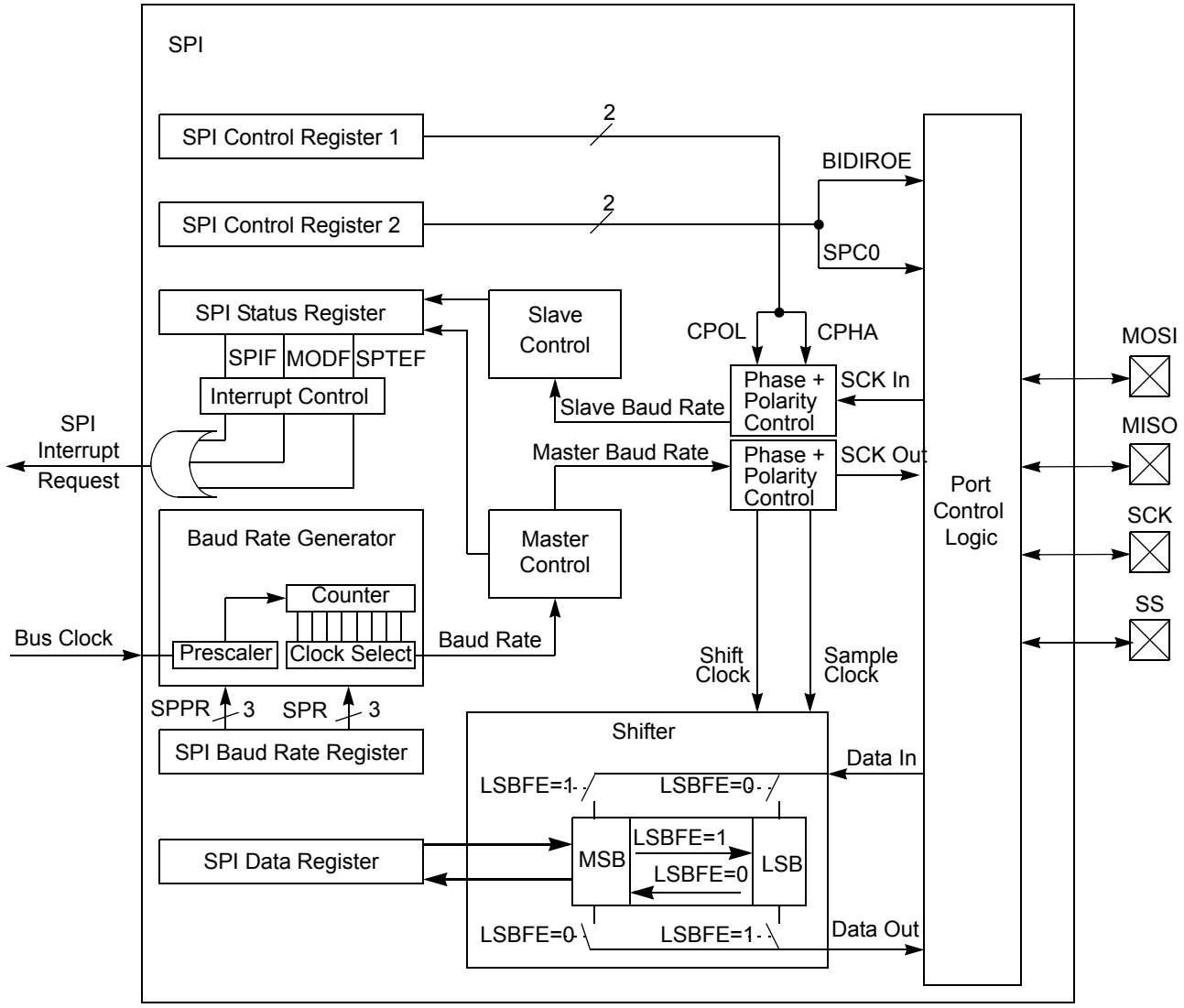

Figure 19-1. SPI Block Diagram

19.2 External Signal Description

This section lists the name and description of all ports including inputs and outputs that do, or may, connect off chip. The S12SPIV5 module has a total of four external pins.

19.2.1 MOSI — Master Out/Slave In Pin

This pin is used to transmit data out of the SPI module when it is configured as a master and receive data when it is configured as slave.

- Automated program and erase algorithm with verify and generation of ECC parity bits
- Fast sector erase and phrase program operation
- Ability to read the P-Flash memory while programming a word in the EEPROM memory
- Flexible protection scheme to prevent accidental program or erase of P-Flash memory

22.1.2.2 EEPROM Features

- The EEPROM memory is composed of one Flash block divided into sectors of 4 bytes
- Single bit fault correction and double bit fault detection within a word during read operations
- Automated program and erase algorithm with verify and generation of ECC parity bits
- Fast sector erase and word program operation
- Protection scheme to prevent accidental program or erase of EEPROM memory
- Ability to program up to four words in a burst sequence

22.1.2.3 Other Flash Module Features

- No external high-voltage power supply required for Flash memory program and erase operations
- Interrupt generation on Flash command completion and Flash error detection
- Security mechanism to prevent unauthorized access to the Flash memory

22.1.3 Block Diagram

The block diagrams of the Flash modules are shown in the following figures.

Address & Name		7	6	5	4	3	2	1	0
0x0003 FPSTAT	R	FPOVRD	0	0	0	0	0	0	WSTATACK
	W								
0x0004 FCNFG	R	CCIE	0	ERSAREQ	IGNSF	WSTAT[1:0]		FDFD	FSFD
	W								
0x0005 FERCNFG	R	0	0	0	0	0	0	0	SFDIE
	W								
0x0006 FSTAT	R	CCIF	0	ACCERR	FPVIOL	MGBUSY	RSVD	MGSTAT1	MGSTAT0
	W								
0x0007 FERSTAT	R	0	0	0	0	0	0	DFDF	SFDIF
	W								
0x0008 FPROT	R	FPOPEN	RNV6	FPHDIS	FPHS1	FPHS0	FPLDIS	FPLS1	FPLS0
	W								
0x0009 DFPROT[1]	R	DPOPEN	DPS6	DPS5	DPS4	DPS3	DPS2	DPS1	DPS0
	W								
0x000A FOPT	R	NV7	NV6	NV5	NV4	NV3	NV2	NV1	NV0
	W								
0x000B FRSV1	R	0	0	0	0	0	0	0	0
	W								
0x000C FCCOB0HI	R	CCOB15	CCOB14	CCOB13	CCOB12	CCOB11	CCOB10	CCOB9	CCOB8
	W								
0x000D FCCOB0LO	R	CCOB7	CCOB6	CCOB5	CCOB4	CCOB3	CCOB2	CCOB1	CCOB0
	W								
0x000E FCCOB1HI	R	CCOB15	CCOB14	CCOB13	CCOB12	CCOB11	CCOB10	CCOB9	CCOB8
	W								
0x000F FCCOB1LO	R	CCOB7	CCOB6	CCOB5	CCOB4	CCOB3	CCOB2	CCOB1	CCOB0
	W								
0x0010 FCCOB2HI	R	CCOB15	CCOB14	CCOB13	CCOB12	CCOB11	CCOB10	CCOB9	CCOB8
	W								

Figure 22-4. FTMRZ Register Summary (continued)

Table 22-31. Allowed P-Flash and EEPROM Simultaneous Operations on a single hardblock

	EEPROM				
Program Flash	**Read**	**Margin Read[2]**	**Program**	**Sector Erase**	**Mass Erase[2]**
Read	OK[1]	OK	OK	OK	
Margin Read[2]					
Program					
Sector Erase					
Mass Erase[3]					OK

1 Strictly speaking, only one read of either the P-Flash or EEPROM can occur at any given instant, but the memory controller will transparently arbitrate P-Flash and EEPROM accesses giving uninterrupted read access whenever possible.

2 A 'Margin Read' is any read after executing the margin setting commands 'Set User Margin Level' or 'Set Field Margin Level' with anything but the 'normal' level specified. See the Note on margin settings in Section 22.4.7.12 and Section 22.4.7.13.

3 The 'Mass Erase' operations are commands 'Erase All Blocks' and 'Erase Flash Block'

22.4.7 Flash Command Description

This section provides details of all available Flash commands launched by a command write sequence. The ACCERR bit in the FSTAT register will be set during the command write sequence if any of the following illegal steps are performed, causing the command not to be processed by the Memory Controller:

- Starting any command write sequence that programs or erases Flash memory before initializing the FCLKDIV register
- Writing an invalid command as part of the command write sequence
- For additional possible errors, refer to the error handling table provided for each command

If a Flash block is read during execution of an algorithm (CCIF = 0) on that same block, the read operation may return invalid data resulting in an illegal access (as described on Section 22.4.6).

If the ACCERR or FPVIOL bits are set in the FSTAT register, the user must clear these bits before starting any command write sequence (see Section 22.3.2.7).

CAUTION

A Flash word or phrase must be in the erased state before being programmed. Cumulative programming of bits within a Flash word or phrase is not allowed.

22.4.7.1 Erase Verify All Blocks Command

The Erase Verify All Blocks command will verify that all P-Flash and EEPROM blocks have been erased.

Table F-1. Static Electrical Characteristics - Supply Voltage Sense - (BATS)

Characteristics noted under conditions 5.5V ≤ V_{SUP} ≤ 18V, unless otherwise noted. Typical values noted reflect the approximate parameter mean at T_A = 25°C[1] under nominal conditions unless otherwise noted.						
Num	**Ratings**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
4	Low Voltage Warning (LBI 4)					
	Ratio = 9					
	Assert (Measured on selected pin, falling edge)	$V_{LBI4\_A\_9}$	8	9	10	V
	Deassert (Measured on selected pin, rising edge)	$V_{LBI4\_D\_9}$	–	–	10.5	V
	Hysteresis (measured on selected pin)	$V_{LBI4\_H\_9}$	–	0.4	–	V
	Ratio = 17[2]					
	Assert (Measured on selected pin, falling edge)	$V_{LBI4\_A\_17}$	8	9.5	11.0	V
	Deassert (Measured on selected pin, rising edge)	$V_{LBI4\_D\_17}$	–	–	12.0	V
	Hysteresis (measured on selected pin)	$V_{LBI4\_H\_17}$	–	1.0	–	V
5	High Voltage Warning (HBI 1)					
	Ratio = 9 (VDDX > 4.5V)					
	Assert (Measured on selected pin, rising edge)	$V_{HBI1\_A\_9}$	14.5	16.5	18	V
	Deassert (Measured on selected pin, falling edge)	$V_{HBI1\_D\_9}$	14	–	–	V
	Hysteresis (measured on selected pin)	$V_{HBI1\_H\_9}$	–	1.0	–	V
	Ratio = 17[2]					
	Assert (Measured on selected pin, rising edge)	$V_{HBI1\_A\_17}$	14.5	16.5	18	V
	Deassert (Measured on selected pin, falling edge)	$V_{HBI1\_D\_17}$	12.5	–	–	V
	Hysteresis (measured on selected pin)	$V_{HBI1\_H\_17}$	–	2.0	–	V