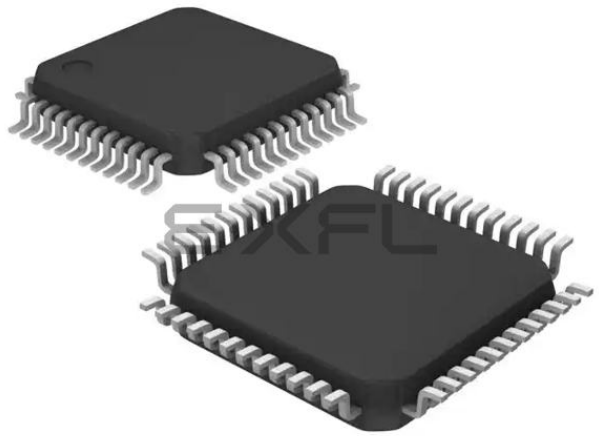


Welcome to E-XFL.COM

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, SCI, SPI
Peripherals	DMA, POR, PWM, WDT
Number of I/O	28
Program Memory Size	192KB (192K x 8)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	12K x 8
Voltage - Supply (Vcc/Vdd)	3.5V ~ 40V
Data Converters	A/D 10x12b; D/A 1x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°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/s912zvca19f0mlfr

- Flexible maskable identifier filter supports two full-size (32-bit) extended identifier filters, or four 16-bit filters, or eight 8-bit filters
- Programmable wake-up functionality with integrated low-pass filter

1.4.10 SENT Transmitter (SENT_TX)

- Features a 13-bit prescaler to derive a SENT-protocol compatible time unit of 3 to 90 µs from bus-clock.
- Programmable number of transmitted data-nibbles (1 to 6).
- Provides hardware to support SAE J2716-2010 (SENT) Fast Channel communication.
- CRC nibble generation:
  - Supports SENT legacy method CRC generation in hardware.
  - Supports SENT recommended method CRC generation in hardware.
  - Optionally, the SENT configuration- and status-nibble can be included in the automatic calculation of the CRC nibble.
  - Automatic CRC generation hardware can be bypassed to supply the CRC nibble directly from software
- Supports optional pause-pulse generation. The optional pause pulse can have a fixed length or its length can be automatically adapted to get a fixed overall message length.
- Supports both continuous and software-triggered transmission.
- Interrupt-driven operation with five flags:
  - Transmit buffer empty
  - Transmission complete
  - Calibration start
  - Transmitter underrun
  - Pause-pulse Rising-Edge

1.4.11 Serial Communication Interface Module (SCI)

- Full-duplex or single-wire operation
- Standard mark/space non-return-to-zero (NRZ) format
- Selectable IrDA 1.4 return-to-zero-inverted (RZI) format with programmable pulse widths
- Baud rate generator by a 16-bit divider from the bus clock
- Programmable character length
- Programmable polarity for transmitter and receiver
- Active-edge receive wakeup
- Break detect and transmit collision detect supporting LIN

1.11 Security

The MCU security mechanism prevents unauthorized access to the flash memory. It must be emphasized that part of the security must lie with the application code. An extreme example would be application code that dumps the contents of the internal memory. This would defeat the purpose of security. Also, if an application has the capability of downloading code through a serial port and then executing that code (e.g. an application containing bootloader code), then this capability could potentially be used to read the EEPROM and Flash memory contents even when the microcontroller is in the secure state. In this example, the security of the application could be enhanced by requiring a response authentication before any code can be downloaded.

Device security details are also described in the flash block description.

1.11.1 Features

The security features of the S12Z chip family are:

- Prevent external access of the non-volatile memories (Flash, EEPROM) content
- Restrict execution of NVM commands

1.11.2 Securing the Microcontroller

The chip can be secured by programming the security bits located in the options/security byte in the Flash memory array. These non-volatile bits keep the device secured through reset and power-down.

This byte can be erased and programmed like any other Flash location. Two bits of this byte are used for security (SEC[1:0]). The contents of this byte are copied into the Flash security register (FSEC) during a reset sequence.

The meaning of the security bits SEC[1:0] is shown in Table 1-11. For security reasons, the state of device security is controlled by two bits. To put the device in unsecured mode, these bits must be programmed to SEC[1:0] = ‘10’. All other combinations put the device in a secured mode. The recommended value to put the device in secured state is the inverse of the unsecured state, i.e. SEC[1:0] = ‘01’.

Table 1-11. Security Bits

SEC[1:0]	Security State
00	1 (secured)
01	1 (secured)
10	***0 (unsecured)***
11	1 (secured)

NOTE

Please refer to the Flash block description for more security byte details.

3.4.3 Clock Source

The BDC clock source can be mapped to a constant frequency clock source or a PLL based fast clock. The clock source for the BDC is selected by the CLKSW bit as shown in Figure 3-5. The BDC internal clock is named BDCSI clock. If BDCSI clock is mapped to the BDCCLK by CLKSW then the serial interface communication is not affected by bus/core clock frequency changes. If the BDC is mapped to BDCFCLK then the clock is connected to a PLL derived source at device level (typically bus clock), thus can be subject to frequency changes in application. Debugging through frequency changes requires SYNC pulses to re-synchronize. The sources of BDCCLK and BDCFCLK are specified at device level.

BDC accesses of internal device resources always use the device core clock. Thus if the ACK handshake protocol is not enabled, the clock frequency relationship must be taken into account by the host.

When changing the clock source via the CLKSW bit a minimum delay of 150 cycles at the initial clock speed must elapse before a SYNC can be sent. This guarantees that the start of the next BDC command uses the new clock for timing subsequent BDC communications.

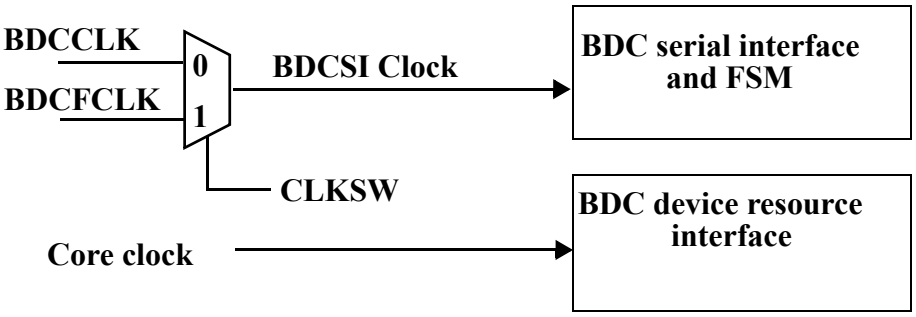


Figure 3-5. Clock Switch

3.4.4 BDC Commands

BDC commands can be classified into three types as shown in Table 3-7.

Table 3-7. BDC Command Types

Command Type	Secure Status	BDC Status	CPU Status	Command Set
Always-available	Secure or Unsecure	Enabled or Disabled	—	• Read/write access to BDCCSR • Mass erase flash memory using ERASE_FLASH • SYNC • ACK enable/disable

Table 7-9. Memory access cycles

Access type	ECC error	access cycle	Internal operation	Memory content	Error indication
1 or 3 byte write, non-aligned 2 byte write	no	2	read data from the memory write old + new data to the memory	old + new data	—
	single bit	2	read data from the memory write corrected + new data to the memory	corrected + new data	SBEEIF
	double bit	2	read data from the memory ignore write data	unchanged	initiator module is informed
read access	no	1	read from memory	unchanged	-
	single bit	1[1]	read data from the memory write corrected data back to memory	corrected data	SBEEIF
	double bit	1	read from memory	unchanged	data mark as invalid

1 The next back to back read access to the memory will be delayed by one clock cycle

The single bit ECC error generates an interrupt when enabled. The double bit ECC errors are reported by the SRAM_ECC module, but handled at MCU level. For more information, see the MMC description.

7.3.1 Aligned 2 and 4 Byte Memory Write Access

During an aligned 2 or 4 byte memory write access, no ECC check is performed. The internal ECC logic generates the new ECC value based on the write data and writes the data words together with the generated ECC values into the memory.

7.3.2 Other Memory Write Access

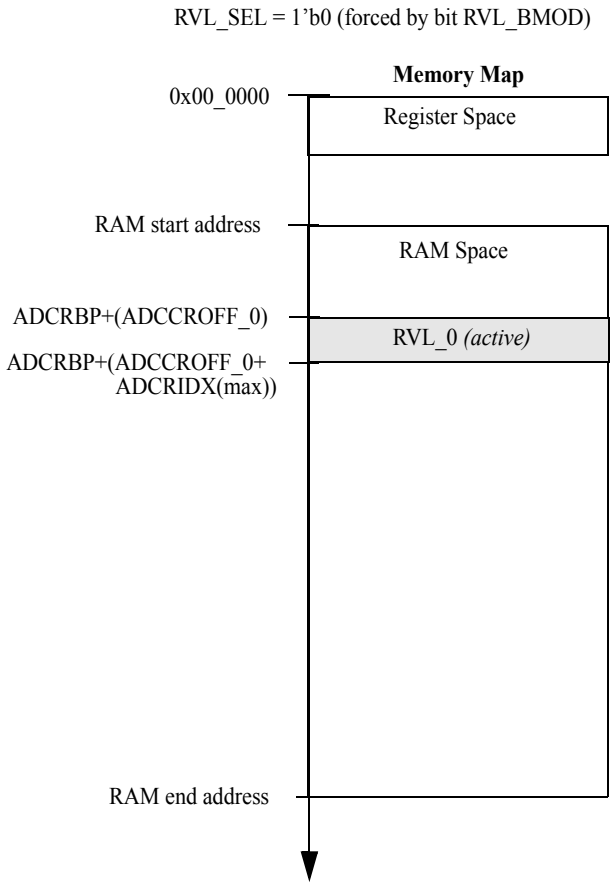
Other types of write accesses are separated into a read-modify-write operation. During the first cycle, the logic reads the data from the memory and performs an ECC check. If no ECC errors were detected then the logic generates the new ECC value based on the read and write data and writes the new data word together with the new ECC value into the memory. If required both 2 byte data words are updated.

If the module detects a single bit ECC error during the read cycle, then the logic generates the new ECC value based on the corrected read and new write read. In the next cycle, the new data word and the new ECC value are written into the memory. If required both 2 byte data words are updated. The SBEEIF bit is set. Hence, the single bit ECC error was corrected by the write access. Figure 7-9 shows an example of a 2 byte non-aligned memory write access.

If the module detects a double bit ECC error during the read cycle, then the write access to the memory is blocked and the initiator module is informed about the error.

Table 8-15. COP Watchdog Rates if COPOSCSEL1=1.

CR2	CR1	CR0	COPCLK Cycles to time-out (COPCLK is ACLK divided by 2)
0	0	0	COP disabled
0	0	1	2^{7}
0	1	0	2^{9}
0	1	1	2^{11}
1	0	0	2^{13}
1	0	1	2^{15}
1	1	0	2^{16}
1	1	1	2^{17}



Note: Address register names in () are not absolute addresses instead they are a sample offset or sample index

Figure 9-34. Result Value List Schema in Single Buffer Mode

While ADC is enabled, one Result Value List is active (indicated by bit RVL_SEL). The conversion Result Value List can be read anytime. When the ADC is enabled the conversion result address registers (ADCRBP, ADCCROFF_0/1, ADCRIDX) are read only and register ADCRIDX is under control of the ADC.

A conversion result is always stored as 16bit entity in unsigned data representation. Left and right justification inside the entity is selected via the DJM control bit. Unused bits inside an entity are stored zero.

Table 9-32. Conversion Result Justification Overview

Conversion Resolution (SRES[1:0])	Left Justified Result (DJM = 1'b0)	Right Justified Result (DJM = 1'b1)
8 bit	{Result[7:0],8'b00000000}	{8'b00000000,Result[7:0]}
10 bit	{Result[9:0],6'b000000}	{6'b000000,Result[9:0]}
12 bit	{Result[11:0],4'b0000}	{4'b0000,Result[11:0]}

By enabling the PRNT bit of the TSCR1 register, the performance of the timer can be enhanced. In this case, it is possible to set additional prescaler settings for the main timer counter in the present timer by using PTPSR[7:0] bits of PTPSR register generating divide by 1, 2, 3, 4,....20, 21, 22, 23,.....255, or 256.

12.4.2 Input Capture

Clearing the I/O (input/output) select bit, IOSx, configures channel x as an input capture channel. The input capture function captures the time at which an external event occurs. When an active edge occurs on the pin of an input capture channel, the timer transfers the value in the timer counter into the timer channel registers, TCx.

The minimum pulse width for the input capture input is greater than two Core clocks.

An input capture on channel x sets the CxF flag. The CxI bit enables the CxF flag to generate interrupt requests. Timer module must stay enabled (TEN bit of TSCR1 register must be set to one) while clearing CxF (writing one to CxF).

12.4.3 Output Compare

Setting the I/O select bit, IOSx, configures channel x when available as an output compare channel. The output compare function can generate a periodic pulse with a programmable polarity, duration, and frequency. When the timer counter reaches the value in the channel registers of an output compare channel, the timer can set, clear, or toggle the channel pin if the corresponding OCPDx bit is set to zero. An output compare on channel x sets the CxF flag. The CxI bit enables the CxF flag to generate interrupt requests. Timer module must stay enabled (TEN bit of TSCR1 register must be set to one) while clearing CxF (writing one to CxF).

The output mode and level bits, OMx and OLx, select set, clear, toggle on output compare. Clearing both OMx and OLx results in no output compare action on the output compare channel pin.

Setting a force output compare bit, FOCx, causes an output compare on channel x. A forced output compare does not set the channel flag.

Writing to the timer port bit of an output compare pin does not affect the pin state. The value written is stored in an internal latch. When the pin becomes available for general-purpose output, the last value written to the bit appears at the pin.

12.4.3.1 OC Channel Initialization

The internal register whose output drives OCx can be programmed before the timer drives OCx. The desired state can be programmed to this internal register by writing a one to CFORCx bit with TIOSx, OCPDx and TEN bits set to one.

Set OCx: Write a 1 to FOCx while TEN=1, IOSx=1, OMx=1, OLx=1 and OCPDx=1

Clear OCx: Write a 1 to FOCx while TEN=1, IOSx=1, OMx=1, OLx=0 and OCPDx=1

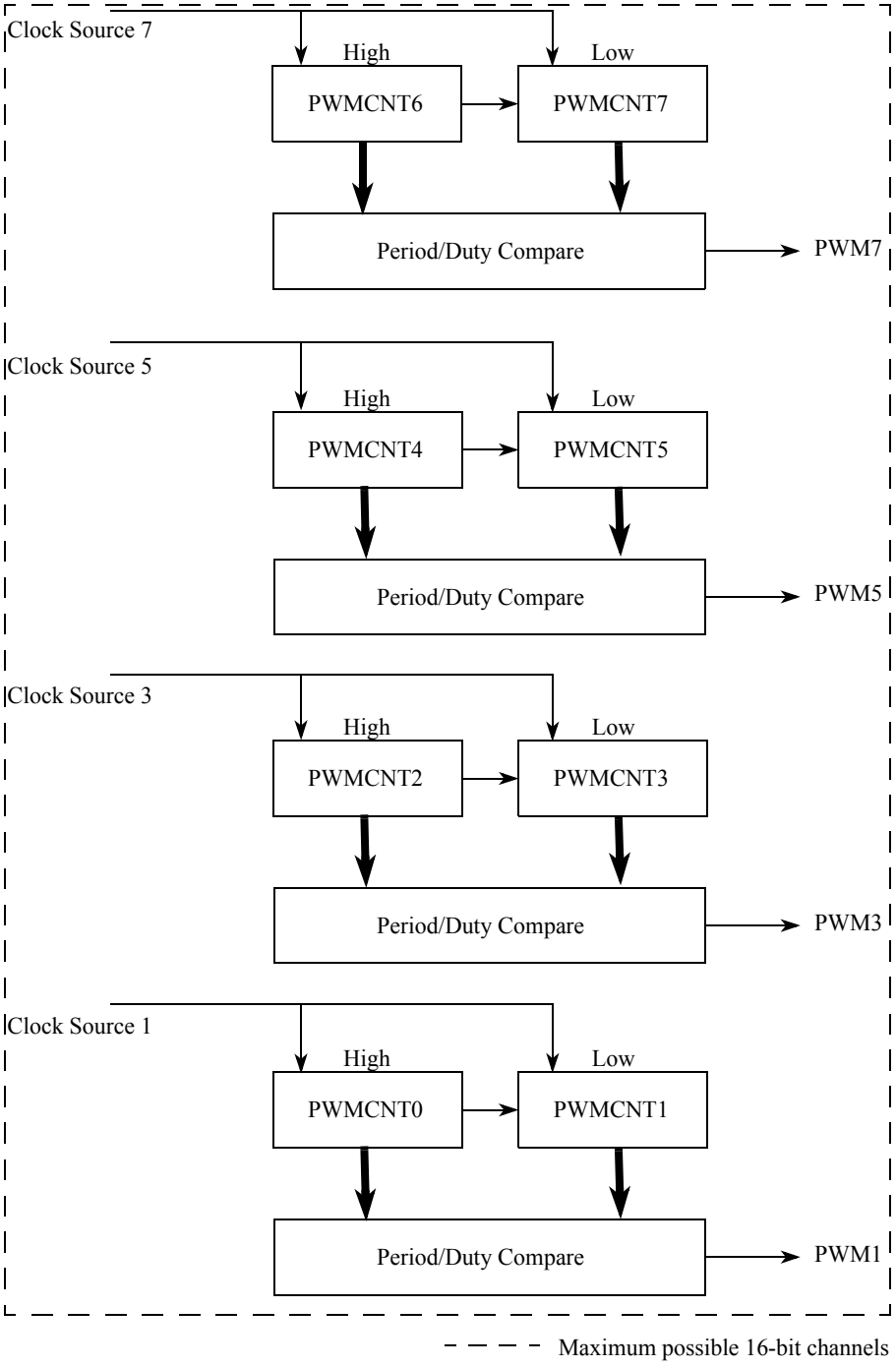


Figure 13-21. PWM 16-Bit Mode

Once concatenated mode is enabled (CONxx bits set in PWMCTL register), enabling/disabling the corresponding 16-bit PWM channel is controlled by the low order PWMEx bit. In this case, the high order bytes PWMEx bits have no effect and their corresponding PWM output is disabled.

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.

14.4.5.3 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

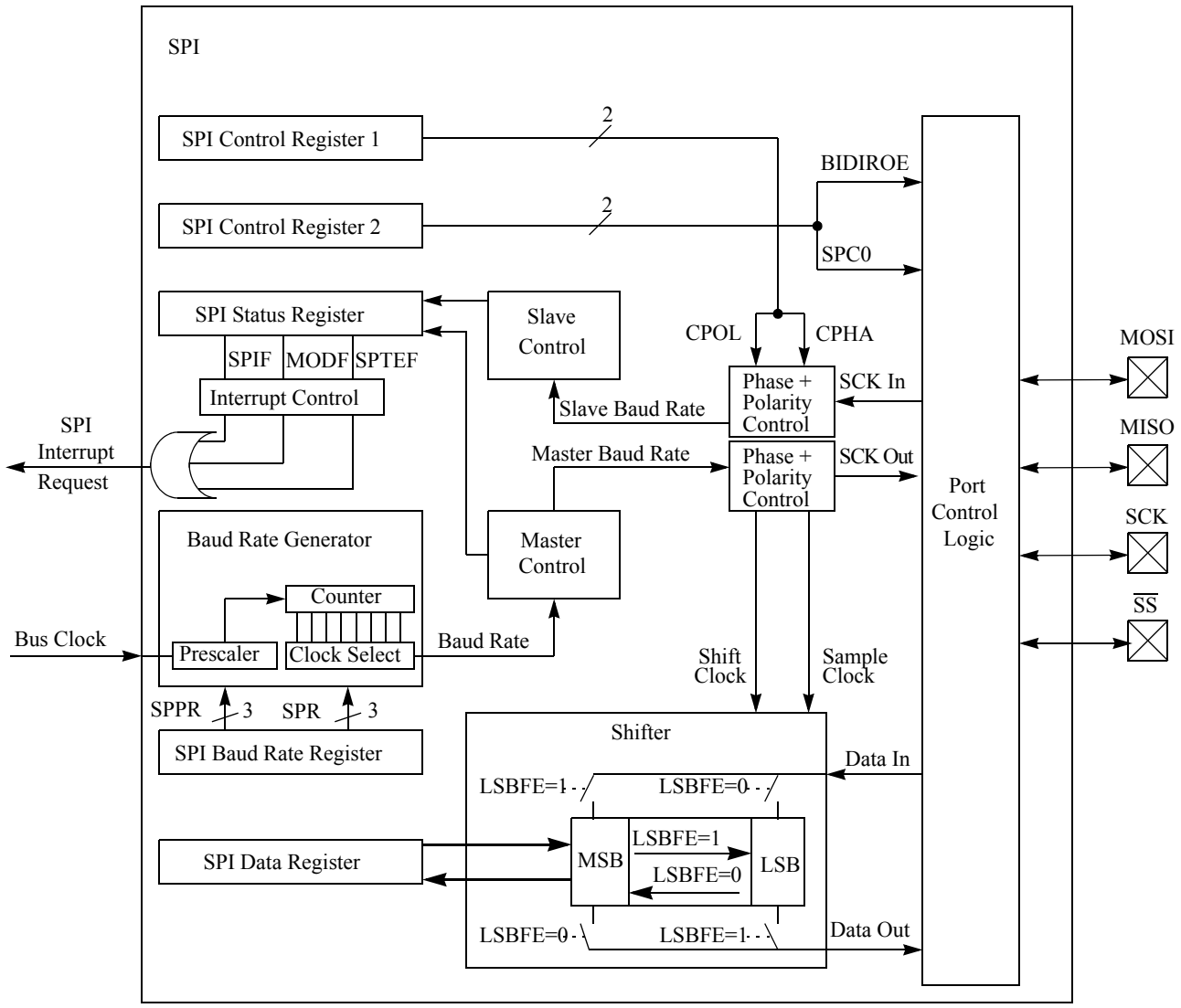


Figure 15-1. SPI Block Diagram

15.2 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 SPI module has a total of four external pins.

15.2.1 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.

15.2.2 MISO — Master In/Slave Out Pin

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

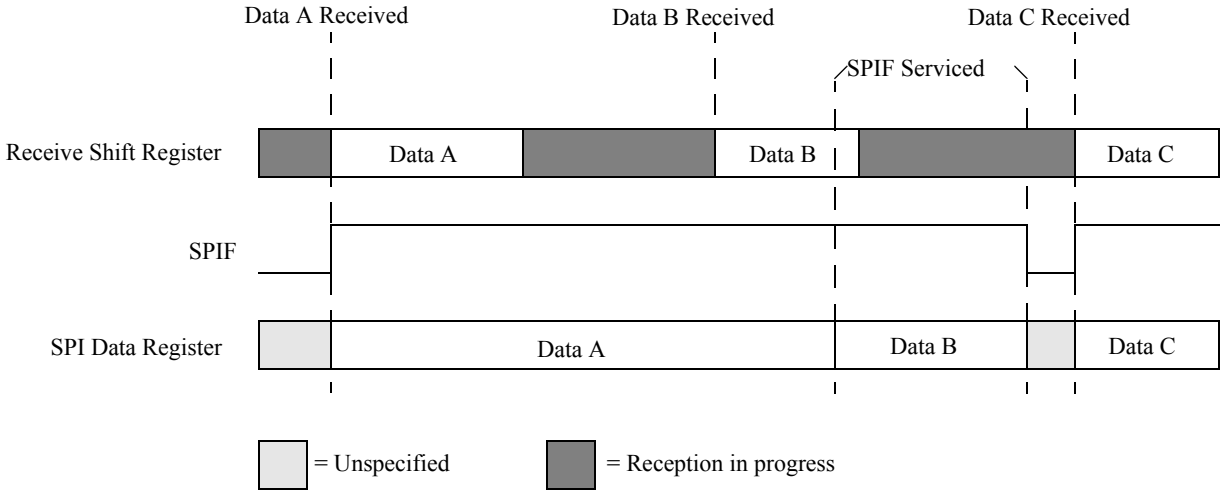


Figure 15-9. Reception with SPIF serviced in Time

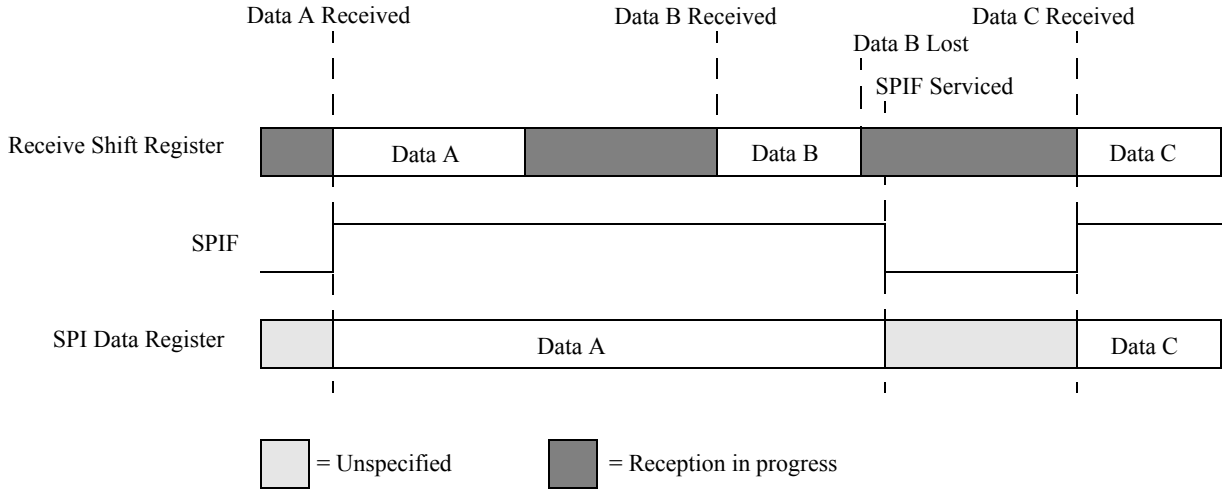


Figure 15-10. Reception with SPIF serviced too late

15.4 Functional Description

The SPI module allows a duplex, synchronous, serial communication between the MCU and peripheral devices. Software can poll the SPI status flags or SPI operation can be interrupt driven.

The SPI system is enabled by setting the SPI enable (SPE) bit in SPI control register 1. While SPE is set, the four associated SPI port pins are dedicated to the SPI function as:

- Slave select ($\overline{SS}$)
- Serial clock (SCK)
- Master out/slave in (MOSI)
- Master in/slave out (MISO)

2. Configure CAN Physical Layer slew rate
3. Enable CAN Physical Layer interrupts
4. Optionally enable SPLIT pin
5. Configure wake-up filter or disable wake-up receiver in case of other wake-up sources
6. Enable CAN Physical Layer to enter normal mode
7. Start CAN communication

17.6.2 Wake-up Mechanism

In stop mode the CAN Physical Layer passes CAN bus states to CPRXD if the wake-up function is enabled (CPCR[WUPE1:WUPE0]≠0). In order to wake up the device from stop mode, the wake-up interrupt of the connected MSCAN module is used.

If CPCR[WUPE1:WUPE0]=b10 the CAN Physical Layer is transparent in stop mode and the MSCAN can be used with or without its integrated low-pass filter for wake-up. Refer to the MSCAN chapter for details on configuring and enabling the wake-up function.

For increased robustness against false wake-up, a CAN Physical Layer pulse filter can optionally be enabled to mask the first (CPCR[WUPE1:WUPE0]=b01) or first two (CPCR[WUPE1:WUPE0]=b11) wake-up events after entering stop mode. The appropriate number of masked pulses depends on the individual CAN bus network topology.

Note that the MSCAN can generate a wake-up interrupt immediately after it acknowledges sleep mode (CANCTL1[SPLAK]=1) whereas the CAN Physical Layer pulse filter takes effect only after entering stop mode. To avoid a false wake-up in between these two events, the MSCAN low-pass filter should also be activated (CANCTL1[WUPM]=1). After sleep mode acknowledge the CPU STOP instruction should be executed before the expiration of t_{WUP}(min) to enable the CAN Physical Layer pulse filter in time.

17.6.3 Bus Error Handling

Upon CAN bus error voltage high failures and over-current events listen-only is entered immediately and the transmitter is turned off. This mode is maintained as long as voltage failure conditions persist or, in case of over-current events, application software re-enables the transmit driver by clearing the related flags.

All high and low voltage levels for both CAN bus lines are continuously reflected in their related voltage failure status bits. A change in a status bit sets the corresponding flag and generates an interrupt if enabled. As long as any of the voltage failure high status bits is set, the transmit driver remains off. It will be turned on again automatically as soon as all voltage failure conditions have disappeared. In case of a voltage failure low condition on CANL only the CANL driver is disabled. A voltage failure low condition on CANH has no effect on the transmitter.

Voltage failure errors have informational purpose. If the application detects frequent CAN protocol errors it is advisable to take the appropriate action. No software action is need to re-enable the transmit driver.

An over-current event on either CAN bus line sets the related flag and turns off the transmit driver. This error can only be detected while driving the bus dominant. In contrast to the voltage failure the over-current

Table 18-17. CANIDAC Register Field Descriptions

Field	Description
5-4 IDAM[1:0]	**Identifier Acceptance Mode** — The CPU sets these flags to define the identifier acceptance filter organization (see Section 18.4.3, “Identifier Acceptance Filter”). Table 18-18 summarizes the different settings. In filter closed mode, no message is accepted such that the foreground buffer is never reloaded.
2-0 IDHIT[2:0]	Identifier Acceptance Hit Indicator — The MSCAN sets these flags to indicate an identifier acceptance hit (see Section 18.4.3, “Identifier Acceptance Filter”). Table 18-19 summarizes the different settings.

Table 18-18. Identifier Acceptance Mode Settings

IDAM1	IDAM0	Identifier Acceptance Mode
0	0	Two 32-bit acceptance filters
0	1	Four 16-bit acceptance filters
1	0	Eight 8-bit acceptance filters
1	1	Filter closed

Table 18-19. Identifier Acceptance Hit Indication

IDHIT2	IDHIT1	IDHIT0	Identifier Acceptance Hit
0	0	0	Filter 0 hit
0	0	1	Filter 1 hit
0	1	0	Filter 2 hit
0	1	1	Filter 3 hit
1	0	0	Filter 4 hit
1	0	1	Filter 5 hit
1	1	0	Filter 6 hit
1	1	1	Filter 7 hit

The IDHITx indicators are always related to the message in the foreground buffer (RxFG). When a message gets shifted into the foreground buffer of the receiver FIFO the indicators are updated as well.

18.3.2.13 MSCAN Reserved Register

This register is reserved for factory testing of the MSCAN module and is not available in normal system operating modes.

19.5 Functional Description

19.5.1 Functional Overview

The DAC resistor network and the operational amplifier can be used together or stand alone. Following modes are supported:

Table 19-5. DAC Modes of Operation

DACM[2:0]		Description			
		Submodules		Output	
		DAC resistor network	Operational Amplifier	DACU	AMP
Off	000	disabled	disabled	disconnected	disconnected
Operational amplifier	001	disabled	enabled	disabled	depend on AMPP and AMPM input
Internal DAC only	010	enabled	disabled	disconnected	disconnected
Unbuffered DAC	100	enabled	disabled	unbuffered resistor output voltage	disconnected
Unbuffered DAC with Operational amplifier	101	enabled	enabled	unbuffered resistor output voltage	depend on AMPP and AMPM input
Buffered DAC	111	enabled	enabled	disconnected	buffered resistor output voltage

The DAC resistor network itself can work on two different voltage ranges:

Table 19-6. DAC Resistor Network Voltage ranges

DAC Mode	Description
Full Voltage Range (FVR)	DAC resistor network provides a output voltage over the complete input voltage range, default after reset
Reduced Voltage Range	DAC resistor network provides a output voltage over a reduced input voltage range

Table 19-7 shows the control signal decoding for each mode. For more detailed mode description see the sections below.

Table 19-7. DAC Control Signals

DACM		DAC resistor network	Operational Amplifier	Switch S1	Switch S2	Switch S3
Off	000	disabled	disabled	open	open	open
Operational amplifier	001	disabled	enabled	closed	open	open
Internal DAC only	010	enabled	disabled	open	open	open
Unbuffered DAC	100	enabled	disabled	open	open	closed

21.3 Features

- Features a 14 bit pre-scaler to derive a SENT-protocol compatible time unit of 3 to 90 µs from bus clock.
- Programmable number of transmitted data-nibbles (1 to 6).
- Provides hardware to support SAE J2716 2010 (SENT) Fast Channel communication[1].
- CRC nibble generation:
  - Supports SENT legacy method CRC generation in hardware.
  - Supports SENT recommended method CRC generation in hardware.
  - Optionally, the SENT status and communication nibble can be included in the automatic calculation of the CRC nibble.
  - Automatic CRC generation hardware can be bypassed to supply the CRC nibble directly from software.
- Supports optional pause-pulse generation. The optional pause pulse can have a fixed length or its length can be automatically adapted to get a fixed overall message period.
- Supports both continuous and software-triggered transmission.
- Interrupt-driven operation with five flags:
  - Transmit buffer empty
  - Transmission complete
  - Calibration pulse start
  - Transmitter under-run
  - Pause pulse rising-edge

21.4 Block Diagram

Figure 21-1 shows a block-diagram of the SENTTX module.

1. Slow Channel communication can be implemented in software.

22.4.5 Flash Command Operations

Flash command operations are used to modify Flash memory contents.

The next sections describe:

- How to write the FCLKDIV register that is used to generate a time base (FCLK) derived from BUSCLK for Flash program and erase command operations
- The command write sequence used to set Flash command parameters and launch execution
- Valid Flash commands available for execution, according to MCU functional mode and MCU security state.

22.4.5.1 Writing the FCLKDIV Register

Prior to issuing any Flash program or erase command after a reset, the user is required to write the FCLKDIV register to divide BUSCLK down to a target FCLK of 1 MHz. Table 22-7. shows recommended values for the FDIV field based on BUSCLK frequency.

NOTE

Programming or erasing the Flash memory cannot be performed if the bus clock runs at less than 0.8 MHz. Setting FDIV too high can destroy the Flash memory due to overstress. Setting FDIV too low can result in incomplete programming or erasure of the Flash memory cells.

When the FCLKDIV register is written, the FDIVLD bit is set automatically. If the FDIVLD bit is 0, the FCLKDIV register has not been written since the last reset. If the FCLKDIV register has not been written, any Flash program or erase command loaded during a command write sequence will not execute and the ACCERR bit in the FSTAT register will set.

22.4.5.2 Command Write Sequence

The Memory Controller will launch all valid Flash commands entered using a command write sequence.

Before launching a command, the ACCERR and FPVIOL bits in the FSTAT register must be clear (see <st-blue>Section 22.3.2.7 Flash Status Register (FSTAT)) and the CCIF flag should be tested to determine the status of the current command write sequence. If CCIF is 0, the previous command write sequence is still active, a new command write sequence cannot be started, and all writes to the FCCOB register are ignored.

22.4.5.2.1 Define FCCOB Contents

The FCCOB parameter fields must be loaded with all required parameters for the Flash command being executed. The CCOBIX bits in the FCCOBIX register must reflect the amount of words loaded into the FCCOB registers (see <st-blue>Section 22.3.2.3 Flash CCOB Index Register (FCCOBIX)).

The contents of the FCCOB parameter fields are transferred to the Memory Controller when the user clears the CCIF command completion flag in the FSTAT register (writing 1 clears the CCIF to 0). The CCIF flag will remain clear until the Flash command has completed. Upon completion, the Memory Controller will

return CCIF to 1 and the FCCOB register will be used to communicate any results. The flow for a generic command write sequence is shown in Figure 22-30.

Table 22-31. Allowed P-Flash and EEPROM Simultaneous Operations

	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 <st-blue>Section 22.4.7.12 Set User Margin Level Command and <st-blue>Section 22.4.7.13 Set Field Margin Level Command.

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 <st-blue>Section 22.4.6 Allowed Simultaneous P-Flash and EEPROM Operations).

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 <st-blue>Section 22.3.2.7 Flash Status Register (FSTAT)).

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.

Table I-2. Static Electrical Characteristics

Characteristics noted under conditions 5.5V <= VSUP <=18 V >, -40°C <= Tj <= 150°C > unless otherwise noted. Typical values noted reflect the approximate parameter mean at T_A = 25°C under nominal conditions unless otherwise noted.						
Num	**Ratings**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
20	CANH to 0 V Threshold	V_{H0}	-0.75	-0.15	0	V
21	CANL to 5.0 V Threshold	V_{L5}	VDDC	VDDC +0.15	VDDC + 0.75	V
22	CANH to 5.0 V Threshold	V_{H5}	VDDC	VDDC +0.15	VDDC +0.75	V
SPLIT						
23	Output voltage Loaded condition I_{SPLIT}= +/- 500 uA Unloaded condition Rmeasure > 1 MΩ	V_{SPLIT}	0.3 0.45	0.5 0.5	0.7 0.55	VDDC VDDC
24	Leakage current -12 V < V_{SPLIT} < +12 V -22 V < V_{SPLIT} < +35 V	I_{LSPLIT}	- -	0 -	5 25	µA µA