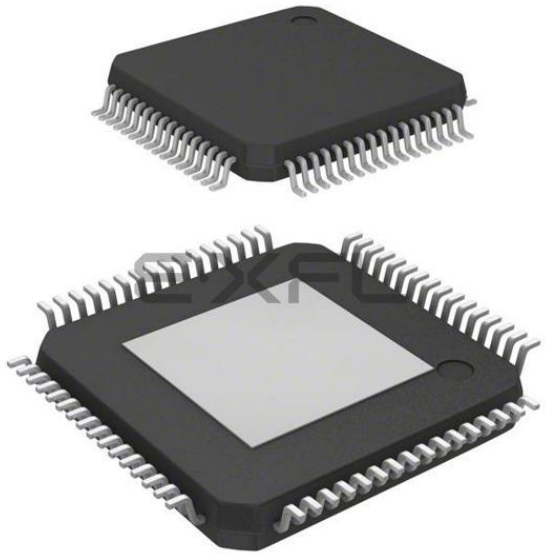


**Welcome to E-XFL.COM**

## 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 | 42 |
| Program Memory Size | 128KB (128K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | 2K x 8 |
| RAM Size | 8K x 8 |
| Voltage - Supply (Vcc/Vdd) | 3.5V ~ 40V |
| Data Converters | A/D 16x10b; D/A 1x8b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 105°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 64-LQFP Exposed Pad |
| Supplier Device Package | 64-LQFP (10x10) |
| Purchase URL | https://www.e-xfl.com/product-detail/nxp-semiconductors/s912zvc12f0vkhr |

#### 1.7.16.2 SCK[1:0] signals

This signal is associated with the serial clock SCK functionality of the serial peripheral interface SPI0 and SPI1.

#### 1.7.16.3 MISO[1:0] signals

This signal is associated with the MISO functionality of the serial peripheral interface SPI0 and SPI1. This signal acts as master input during master mode or as slave output during slave mode.

#### 1.7.16.4 MOSI[1:0] signals

This signal is associated with the MOSI functionality of the serial peripheral interface SPI0 and SPI1. This signal acts as master output during master mode or as slave input during slave mode

### 1.7.17 SCI signals

#### 1.7.17.1 RXD[1:0] signals

These signals are associated with the receive functionality of the serial communication interfaces SCI[1:0].

#### 1.7.17.2 TXD[1:0] signals

These signals are associated with the transmit functionality of the serial communication interfaces SCI[1:0].

### 1.7.18 Timer IOC[7:0] signals

The signals IOC0[7:0] and IOC1[3:0] are associated with the input capture or output compare functionality of the timer modules TIM0 and TIM1.

### 1.7.19 PWM[7:0] signals

The signals PWM0[7:0] and PWM1[7:0] are associated with the outputs of the PWM0 and PWM1 modules.

### 1.7.20 IIC signals

#### 1.7.20.1 SDA signal

This signal is associated with the serial data pin of IIC.

#### 1.7.20.2 SCL signal

This signal is associated with the serial clock pin of IIC.

The ACK handshake protocol does not support nested ACK pulses. If a BDC command is not acknowledged by an ACK pulse, the host needs to abort the pending command first in order to be able to issue a new BDC command. The host can decide to abort any possible pending ACK pulse in order to be sure a new command can be issued. Therefore, the protocol provides a mechanism in which a command, and its corresponding ACK, can be aborted.

Commands With-Status do not generate an ACK, thus if ACK is enabled and a With-Status command is issued, the host must use the 512 cycle timeout to calculate when the data is ready for retrieval.

#### 3.4.7.1 Long-ACK Hardware Handshake Protocol

If a command results in an error condition, whereby a BDCCSRL flag is set, then the target generates a “Long-ACK” low pulse of 64 BDCSI clock cycles, followed by a brief speed pulse. This indicates to the host that an error has occurred. The host can subsequently read BDCCSR to determine the type of error. Whether normal ACK or Long-ACK, the ACK pulse is not issued earlier than 32 BDCSI clock cycles after the BDC command was issued. The end of the BDC command is assumed to be the 16th BDCSI clock cycle of the last bit. The 32 cycle minimum delay differs from the 16 cycle delay time with ACK disabled.

If a BDC access request does not gain access within 512 core clock cycles, the request is aborted, the NORESP flag is set and a Long-ACK pulse is transmitted to indicate an error case.

Following a STOP or WAI instruction, if the BDC is enabled, the first ACK, following stop or wait mode entry is a long ACK to indicate an exception.

### 3.4.8 Hardware Handshake Abort Procedure

The abort procedure is based on the SYNC command. To abort a command that has not responded with an ACK pulse, the host controller generates a sync request (by driving BKGD low for at least 128 BDCSI clock cycles and then driving it high for one BDCSI clock cycle as a speedup pulse). By detecting this long low pulse in the BKGD pin, the target executes the SYNC protocol, see Section 3.4.4.1, “SYNC”, and assumes that the pending command and therefore the related ACK pulse are being aborted. After the SYNC protocol has been completed the host is free to issue new BDC commands.

The host can issue a SYNC close to the 128 clock cycles length, providing a small overhead on the pulse length to assure the sync pulse is not misinterpreted by the target. See Section 3.4.4.1, “SYNC”.

Figure 3-11 shows a SYNC command being issued after a READ_MEM, which aborts the READ_MEM command. Note that, after the command is aborted a new command is issued by the host.

## 8.2 Signal Description

This section lists and describes the signals that connect off chip as well as internal supply nodes and special signals.

### 8.2.1 $\overline{\text{RESET}}$

Pin $\overline{\text{RESET}}$ is an active-low bidirectional pin. As an input it initializes the MCU asynchronously to a known start-up state. As an open-drain output it indicates that an MCU-internal reset has been triggered.

### 8.2.2 EXTAL and XTAL

These pins provide the interface for a crystal to control the internal clock generator circuitry. EXTAL is the input to the crystal oscillator amplifier. XTAL is the output of the crystal oscillator amplifier. If XOSCLCP is enabled, the MCU internal OSCCLK_LCP is derived from the EXTAL input frequency. If OSCE=0, the EXTAL pin is pulled down by an internal resistor of approximately 200 kΩ and the XTAL pin is pulled down by an internal resistor of approximately 700 kΩ.

**NOTE**

NXP recommends an evaluation of the application board and chosen resonator or crystal by the resonator or crystal supplier.
The loop controlled circuit (XOSCLCP) is not suited for overtone resonators and crystals.

### 8.2.3 VSUP — Regulator Power Input Pin

Pin VSUP is the power input of VREGAUTO. All currents sourced into the regulator loads flow through this pin.

A suitable reverse battery protection network can be used to connect VSUP to the car battery supply network.

### 8.2.4 VDDA, VSSA — Regulator Reference Supply Pins

Pins VDDA and VSSA are used to supply the analog parts of the regulator. Internal precision reference circuits are supplied from these signals.

An off-chip decoupling capacitor (220 nF(X7R ceramic)) between VDDA and VSSA is required and can improve the quality of this supply.

VDDA has to be connected externally to VDDX.

### 8.2.5 VDDX, VSSX — Pad Supply Pins

VDDX is the supply domain for the digital Pads. VDDX has to be connected externally to VDDA.

An off-chip decoupling capacitor (10µF plus 220 nF(X7R ceramic)) between VDDX and VSSX is required.

**Table 8-6. CPMUCLKS Descriptions (continued)**

| Field | Description |
|---|---|
| 1<br>RTIOSCSEL | **RTI Clock Select**— RTIOSCSEL selects the clock source to the RTI. Either IRCCLK or OSCCLK. Changing the RTIOSCSEL bit re-starts the RTI time-out period.<br>RTIOSCSEL can only be set to 1, if UPOSC=1.<br>UPOSC= 0 clears the RTIOSCSEL bit.<br>0 RTI clock source is IRCCLK.<br>1 RTI clock source is OSCCLK. |
| 0<br>COP<br>OSCSEL0 | **COP Clock Select 0** — COPOSCSEL0 and COPOSCSEL1 combined determine the clock source to the COP (see also Table 8-7)<br>If COPOSCSEL1 = 1, COPOSCSEL0 has no effect regarding clock select and changing the COPOSCSEL0 bit does not re-start the COP time-out period.<br>When COPOSCSEL1=0,COPOSCSEL0 selects the clock source to the COP to be either IRCCLK or OSCCLK. Changing the COPOSCSEL0 bit re-starts the COP time-out period.<br>COPOSCSEL0 can only be set to 1, if UPOSC=1.<br>UPOSC= 0 clears the COPOSCSEL0 bit.<br>0 COP clock source is IRCCLK.<br>1 COP clock source is OSCCLK |

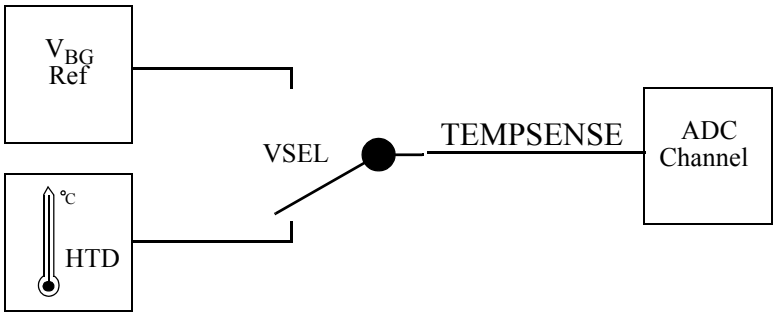
**Table 8-7. COPOSCSEL1, COPOSCSEL0 clock source select description**

| COPOSCSEL1 | COPOSCSEL0 | COP clock source |
|---|---|---|
| 0 | 0 | IRCCLK |
| 0 | 1 | OSCCLK |
| 1 | x | ACLK |

**Table 8-16. CPMUHTCTL Field Descriptions**

| Field | Description |
|---|---|
| 5<br>VSEL | **Voltage Access Select Bit** — If set, the bandgap reference voltage $V_{BG}$ can be accessed internally (i.e. multiplexed to an internal Analog to Digital Converter channel). If not set, the die temperature proportional voltage $V_{HT}$ of the temperature sensor can be accessed internally. See device level specification for connectivity. For any of these access the HTE bit must be set.<br>0 An internal temperature proportional voltage $V_{HT}$ can be accessed internally.<br>1 Bandgap reference voltage $V_{BG}$ can be accessed internally. |
| 3<br>HTE | **High Temperature Sensor/Bandgap Voltage Enable Bit** — This bit enables the high temperature sensor and bandgap voltage amplifier.<br>0 The temperature sensor and bandgap voltage amplifier is disabled.<br>1 The temperature sensor and bandgap voltage amplifier is enabled. |
| 2<br>HTDS | **High Temperature Detect Status Bit** — This read-only status bit reflects the temperature status. Writes have no effect.<br>0 Junction Temperature is below level $T_{HTID}$ or RPM.<br>1 Junction Temperature is above level $T_{HTIA}$ and FPM. |
| 1<br>HTIE | **High Temperature Interrupt Enable Bit**<br>0 Interrupt request is disabled.<br>1 Interrupt will be requested whenever HTIF is set. |
| 0<br>HTIF | **High Temperature Interrupt Flag** — HTIF is set to 1 when HTDS status bit changes. This flag can only be cleared by writing a 1.<br>Writing a 0 has no effect. If enabled (HTIE=1), HTIF causes an interrupt request.<br>0 No change in HTDS bit.<br>1 HTDS bit has changed. |

**Figure 8-18. Voltage Access Select**



### 9.4.2.4 ADC Timing Register (ADCTIM)

Module Base + 0x0003

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | PRS[6:0] | | | | | | |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |

= Unimplemented or Reserved

**Figure 9-7. ADC Timing Register (ADCTIM))**

Read: Anytime

Write: These bits are writable if bit ADC_EN is clear or bit SMOD_ACC is set

**Table 9-6. ADCTIM Field Descriptions**

| Field | Description |
|---|---|
| 6-0<br>PRS[6:0] | **ADC Clock Prescaler** — These 7bits are the binary prescaler value PRS. The ADC conversion clock frequency is calculated as follows:<br>$f_{ATDCLK} = \frac{f_{BUS}}{2 \times (PRS + 1)}$<br>Refer to Device Specification for allowed frequency range of $f_{ATDCLK}$. |

If signal Restart is asserted before signal LoadOK is set the conversion starts from top of currently active CSL at the next Trigger Event (no exchange of CSL list).
If signal Restart is asserted after or simultaneously with signal LoadOK the conversion starts from top of the other CSL at the next Trigger Event (CSL is switched) if CSL is configured for double buffer mode.

- **Sequence Abort Event**
  Internal Interface Signal: Seq_Abort
  Corresponding Bit Name: SEQA
  - *Function:*
    Abort any possible ongoing conversion at next conversion boundary and abort current conversion sequence and active CSL
  - *Requested by:*
    - Positive edge of internal interface signal Seq_Abort
    - Write Access via data bus to set control bit SEQA
  - *When finished:*
    This bit gets cleared when an ongoing conversion is finished and the result is stored and/or an ongoing conversion sequence is aborted and current active CSL is aborted (ADC idle, RVL done)
  - *Mandatory Requirement:*
    - In all ADC conversion flow control modes bit SEQA can only be set if:
    * ADC not idle (a conversion or conversion sequence is ongoing)
    * ADC idle but RVL done condition not reached
    The RVL done condition is not reached if:
    * An “End Of List” command type has not been executed
    * A Sequence Abort Event has not been executed (bit SEQA not already set)
    - In all ADC conversion flow control modes a Sequence Abort Event can be issued at any time
    - In ADC conversion flow control mode “Restart Mode” after a conversion sequence abort request has been executed it is mandatory to set bit RSTA. If a Trigger Event occurs before a Restart Event is executed (bit RSTA set and cleared by hardware), bit TRIG is set, error flag TRIG_EIF is set, and the ADC can only be continued by a Soft-Reset. After the Restart Event the ADC accepts new Trigger Events (bit TRIG set) and begins conversion from top of the currently active CSL.
    - In ADC conversion flow control mode “Restart Mode” after a Sequence Abort Event has been executed, a Restart Event causes only the RSTA bit being set. The ADC executes a Restart Event only.
  - In both conversion flow control modes (“Restart Mode” and “Trigger Mode”) when conversion flow control bit RSTA gets set automatically bit SEQA gets set when the ADC has not reached one of the following scenarios:
    * An “End Of List” command type has been executed or is about to be executed
    * A Sequence Abort request is about to be executed or has been executed.
    In case bit SEQA is set automatically the Restart error flag RSTA_EIF is set to indicate an unexpected Restart Request.

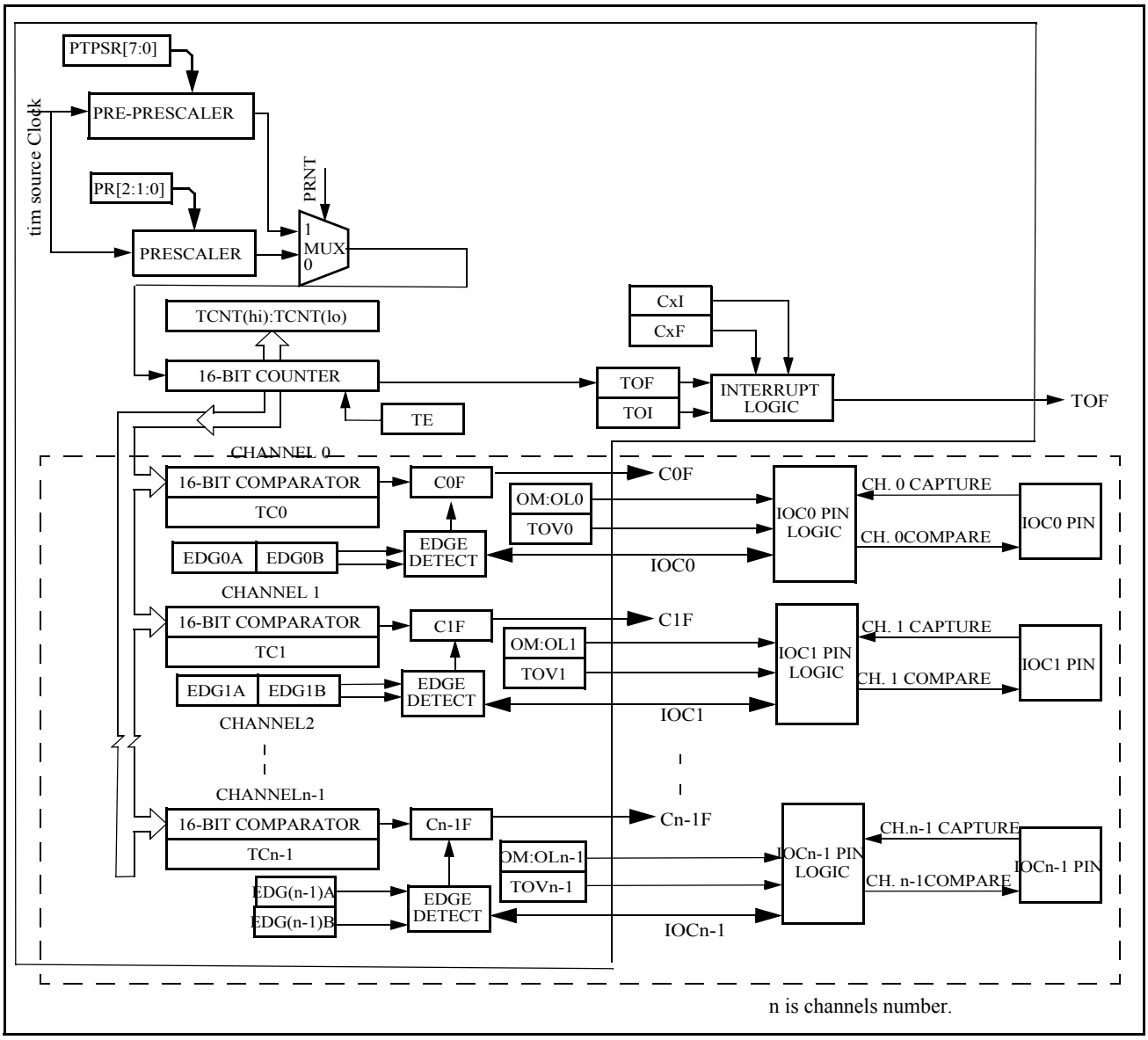



**Figure 12-22. Detailed Timer Block Diagram**

## 12.4.1 Prescaler

The prescaler divides the Core clock by 1, 2, 4, 8, 16, 32, 64 or 128. The prescaler select bits, PR[2:0], select the prescaler divisor. PR[2:0] are in timer system control register 2 (TSCR2).

The prescaler divides the Core clock by a prescalar value. Prescaler select bits PR[2:0] of in timer system control register 2 (TSCR2) are set to define a prescalar value that generates a divide by 1, 2, 4, 8, 16, 32, 64 and 128 when the PRNT bit in TSCR1 is disabled.

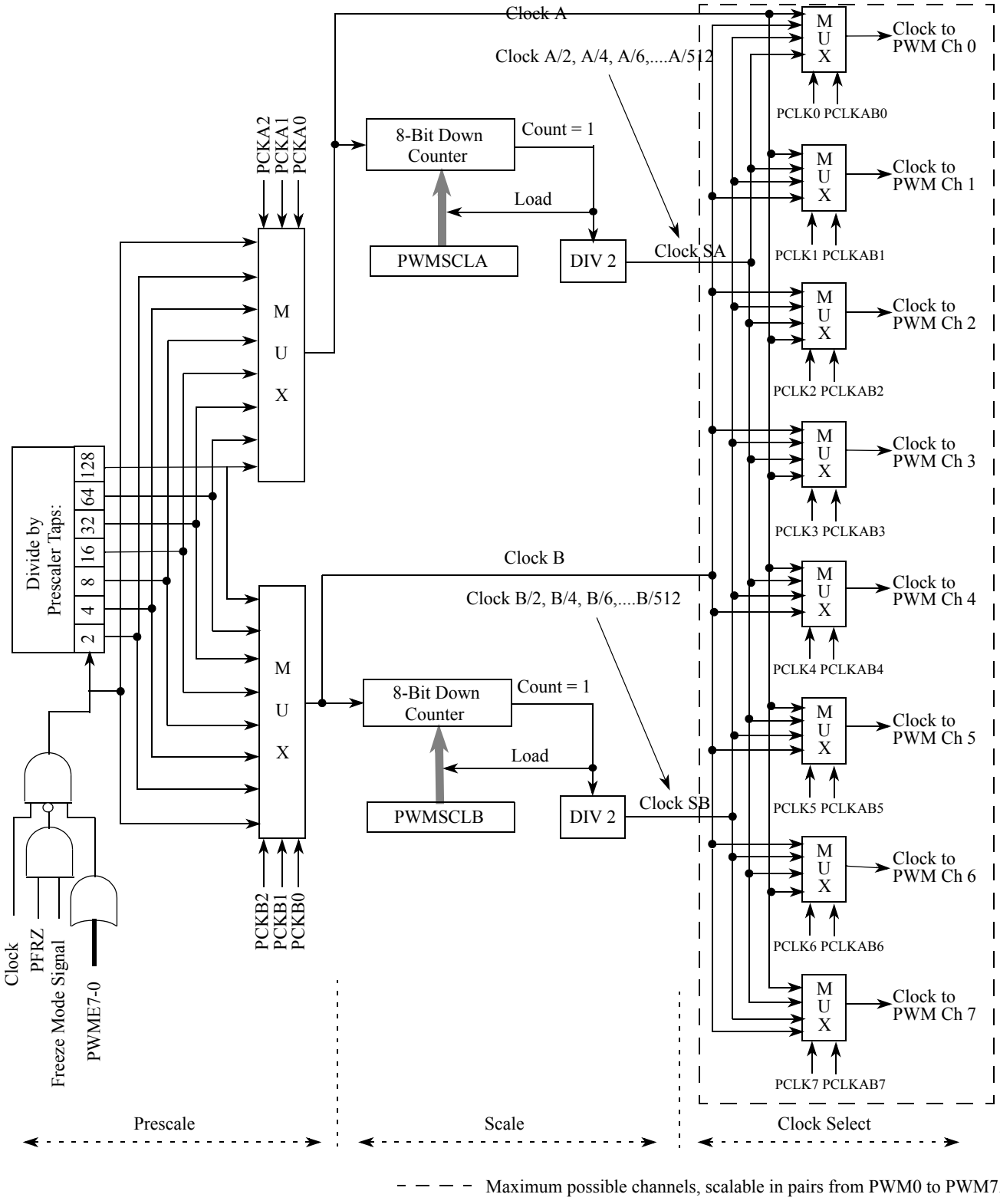



**Figure 13-15. PWM Clock Select Block Diagram**

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.

### 13.4.2.2 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 13-16 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.

### 13.4.2.3 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.

### 13.4.2.4 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 13.4.1, “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 13-16. 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 13-16 and described in Section 13.4.2.5, “Left Aligned Outputs” and Section 13.4.2.6, “Center Aligned Outputs”.

### 14.3.2.8 SCI Status Register 2 (SCISR2)

Module Base + 0x0005

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | AMAP | 0 | 0 | TXPOL | RXPOL | BRK13 | TXDIR | RAF |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

**Figure 14-11. SCI Status Register 2 (SCISR2)**

Read: Anytime

Write: Anytime

**Table 14-12. SCISR2 Field Descriptions**

| Field | Description |
|---|---|
| 7<br>AMAP | **Alternative Map** — This bit controls which registers sharing the same address space are accessible. In the reset condition the SCI behaves as previous versions. Setting AMAP=1 allows the access to another set of control and status registers and hides the baud rate and SCI control Register 1.<br>0 The registers labelled SCIBDH (0x0000),SCIBDL (0x0001), SCICR1 (0x0002) are accessible<br>1 The registers labelled SCIASR1 (0x0000),SCIACR1 (0x0001), SCIACR2 (0x00002) are accessible |
| 4<br>TXPOL | **Transmit Polarity** — This bit control the polarity of the transmitted data. In NRZ format, a one is represented by a mark and a zero is represented by a space for normal polarity, and the opposite for inverted polarity. In IrDA format, a zero is represented by short high pulse in the middle of a bit time remaining idle low for a one for normal polarity, and a zero is represented by short low pulse in the middle of a bit time remaining idle high for a one for inverted polarity.<br>0 Normal polarity<br>1 Inverted polarity |
| 3<br>RXPOL | **Receive Polarity** — This bit control the polarity of the received data. In NRZ format, a one is represented by a mark and a zero is represented by a space for normal polarity, and the opposite for inverted polarity. In IrDA format, a zero is represented by short high pulse in the middle of a bit time remaining idle low for a one for normal polarity, and a zero is represented by short low pulse in the middle of a bit time remaining idle high for a one for inverted polarity.<br>0 Normal polarity<br>1 Inverted polarity |
| 2<br>BRK13 | **Break Transmit Character Length** — This bit determines whether the transmit break character is 10 or 11 bit respectively 13 or 14 bits long. The detection of a framing error is not affected by this bit.<br>0 Break character is 10 or 11 bit long<br>1 Break character is 13 or 14 bit long |
| 1<br>TXDIR | **Transmitter Pin Data Direction in Single-Wire Mode** — This bit determines whether the TXD pin is going to be used as an input or output, in the single-wire mode of operation. This bit is only relevant in the single-wire mode of operation.<br>0 TXD pin to be used as an input in single-wire mode<br>1 TXD pin to be used as an output in single-wire mode |
| 0<br>RAF | **Receiver Active Flag** — RAF is set when the receiver detects a logic 0 during the RT1 time period of the start bit search. RAF is cleared when the receiver detects an idle character.<br>0 No reception in progress<br>1 Reception in progress |

SPI operation in wait mode is a configurable low power mode, controlled by the SPISWAI bit located in the SPICR2 register. In wait mode, if the SPISWAI bit is clear, the SPI operates like in run mode. If the SPISWAI bit is set, the SPI goes into a power conservative state, with the SPI clock generation turned off. If the SPI is configured as a master, any transmission in progress stops, but is resumed after CPU goes into run mode. If the SPI is configured as a slave, reception and transmission of data continues, so that the slave stays synchronized to the master.

- Stop mode

  The SPI is inactive in stop mode for reduced power consumption. If the SPI is configured as a master, any transmission in progress stops, but is resumed after CPU goes into run mode. If the SPI is configured as a slave, reception and transmission of data continues, so that the slave stays synchronized to the master.

For a detailed description of operating modes, please refer to Section 15.4.7, “Low Power Mode Options”.

### 15.1.4 Block Diagram

Figure 15-1 gives an overview on the SPI architecture. The main parts of the SPI are status, control and data registers, shifter logic, baud rate generator, master/slave control logic, and port control logic.

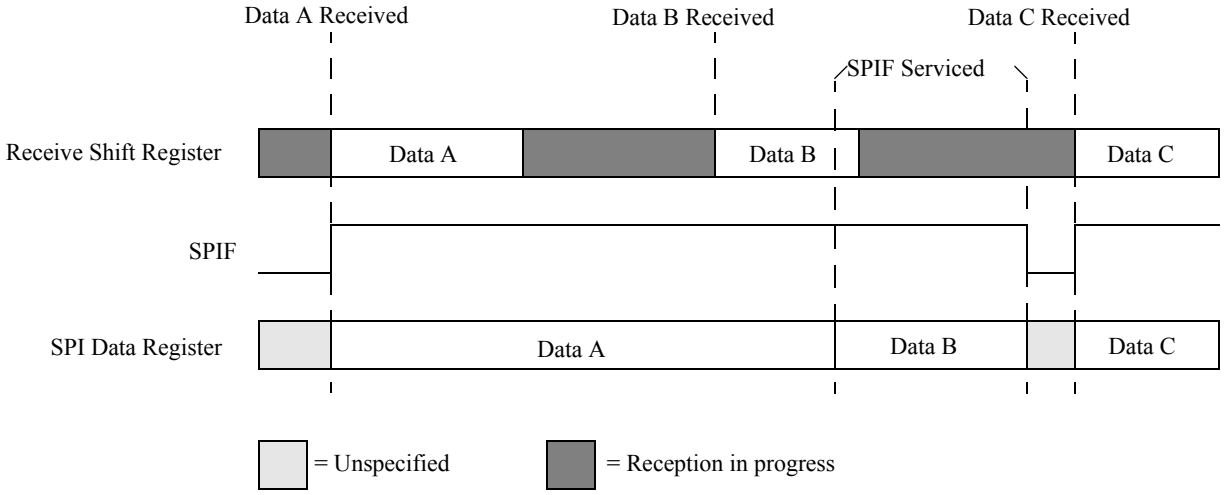



**Figure 15-9. Reception with SPIF serviced in Time**

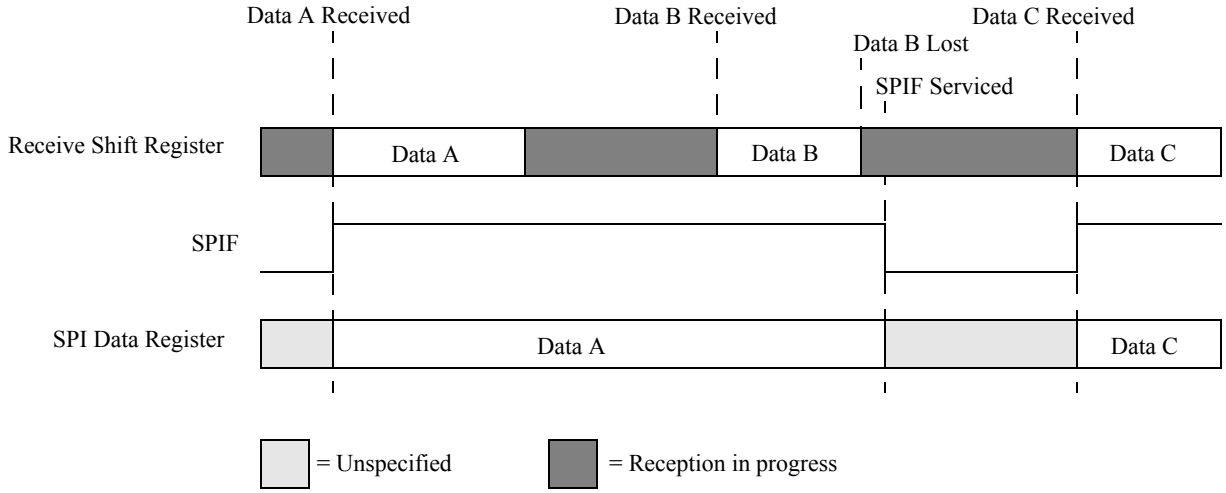



**Figure 15-10. Reception with SPIF serviced too late**

## 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)

#### 18.4.5.7 Disabled Mode

The MSCAN is in disabled mode out of reset (CANE=0). All module clocks are stopped for power saving, however the register map can still be accessed as specified.

#### 18.4.5.8 Programmable Wake-Up Function

The MSCAN can be programmed to wake up from sleep or power down mode as soon as CAN bus activity is detected (see control bit WUPE in MSCAN Control Register 0 (CANCTL0). The sensitivity to existing CAN bus action can be modified by applying a low-pass filter function to the RXCAN input line (see control bit WUPM in Section 18.3.2.2, “MSCAN Control Register 1 (CANCTL1)”).

This feature can be used to protect the MSCAN from wake-up due to short glitches on the CAN bus lines. Such glitches can result from—for example—electromagnetic interference within noisy environments.

### 18.4.6 Reset Initialization

The reset state of each individual bit is listed in Section 18.3.2, “Register Descriptions,” which details all the registers and their bit-fields.

### 18.4.7 Interrupts

This section describes all interrupts originated by the MSCAN. It documents the enable bits and generated flags. Each interrupt is listed and described separately.

#### 18.4.7.1 Description of Interrupt Operation

The MSCAN supports four interrupt vectors (see Table 18-38), any of which can be individually masked (for details see Section 18.3.2.6, “MSCAN Receiver Interrupt Enable Register (CANRIER)” to Section 18.3.2.8, “MSCAN Transmitter Interrupt Enable Register (CANTIER)”).

Refer to the device overview section to determine the dedicated interrupt vector addresses.

**Table 18-38. Interrupt Vectors**

| Interrupt Source | CCR Mask | Local Enable |
|---|---|---|
| Wake-Up Interrupt (WUPIF) | I bit | CANRIER (WUPIE) |
| Error Interrupts Interrupt (CSCIF, OVRIF) | I bit | CANRIER (CSCIE, OVRIE) |
| Receive Interrupt (RXF) | I bit | CANRIER (RXFIE) |
| Transmit Interrupts (TXE[2:0]) | I bit | CANTIER (TXEIE[2:0]) |

#### 18.4.7.2 Transmit Interrupt

At least one of the three transmit buffers is empty (not scheduled) and can be loaded to schedule a message for transmission. The TXEx flag of the empty message buffer is set.

| Field | Description |
|---|---|
| 3<br>SINGLE | **SENTTX Single Shot Operation** — If set, this bit causes the SENTTX module to stop after the current transmission is complete and wait for the Transmit-Buffer Empty bit to be cleared, before a new transmission is started. Transmitter Under-run is not flagged in this case. Otherwise, if this bit is cleared, messages are sent continuously back-to-back. This bit can only be changed if the CONFIG[TXINIT] bit is one.<br>1 - Single shot operation is enabled<br>0 - Single shot operation is disabled<br>**Note:** Meaningful single shot operation requires pause pulse generation to be enabled (PPULSE[PPEN]=1). Otherwise the last nibble to be sent as part of the message (CRC) will have the terminating falling edge missing (rendering the whole transmission incomplete). |
| 2<br>CRCSCN | **SENTTX CRC includes Status- and Communication Nibble** — If set, this bit causes the SENTTX module to include the status and communication nibble in the automatic generation of the CRC nibble. Otherwise, if this bit is cleared, the status and communication nibble is not included in the automatic CRC generation. This bit can only be changed if the CONFIG[TXINIT] bit is one.<br>1 - CRC generation includes Status- and Communication Nibble<br>0 - CRC generation excludes Status- and Communication Nibble |
| 1<br>CRCLEG | **SENTTX CRC Legacy Algorithm Enable** — If set, this bit causes the SENTTX module to generate the automatic CRC nibble using the legacy CRC algorithm. Otherwise, if this bit is cleared, the CRC nibble is calculated using the recommended CRC algorithm. This bit can only be changed if the CONFIG[TXINIT] bit is one.<br>1 - CRC generation uses the legacy algorithm<br>0 - CRC generation uses the recommended algorithm |
| 0<br>CRCBYP | **SENTTX Automatic CRC generation bypass** — If set, this bit causes the SENTTX module to bypass the automatic generation of the CRC nibble. The CRC information is taken from the Transmit Buffer (TXBUF[CRC]). This bit can only be changed if the CONFIG[TXINIT] bit is one.<br>1 - CRC generation bypass is enabled<br>0 - CRC generation bypass is disabled. Data in TXBUF[CRC] is ignored. |

## 21.7.2.4 SENT Transmitter Interrupt Enable Register (INTEN)

Module Base + 0x0006 Access: User read/write[1]

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | 0 | PPREIE | TUIE | CSIE | TCIE | TBEIE |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

[1] Read: Anytime.
Write: Anytime.

**Table 21-7. SENT Transmitter Interrupt Enable Register (INTEN) Field Descriptions**

| Field | Description |
|---|---|
| 4<br>PPREIE | **SENTTX Pause Pulse Rising-Edge Interrupt Enable** — This bit enables the generation of a Pause Pulse Rising-Edge interrupt whenever the Pause Pulse Rising-Edge Flag is set in the SENTTX Interrupt Flag Register (INTFLG[PPRE]).<br>1 - Pause Pulse Rising-Edge interrupt is enabled<br>0 - Pause Pulse Rising-Edge interrupt is disabled |
| 3<br>TUIE | **SENTTX Transmitter Under-run Interrupt Enable** — This bit enables the generation of a Transmitter Under-run Interrupt whenever the Transmitter Under-run Flag is set in the SENTTX Interrupt Flag Register (INTFLG[TU]).<br>1 - Transmitter Under-run Interrupt is enabled<br>0 - Transmitter Under-run Interrupt is disabled |

### 21.8.4.2 Single-buffered Transmission without Pause Pulse

This option offers the most up-to-date transmit data. The disadvantage for software is that the preparation of the transmit data has to occur in much less time (less than the length of the calibration pulse, meaning less than 56 unit-time ticks) compared to the double-buffered transmission option.

The software uses the Transmission Complete interrupt (TC) to prepare new data for the next transmission. An example for this case is provided in Figure 21-4 below.

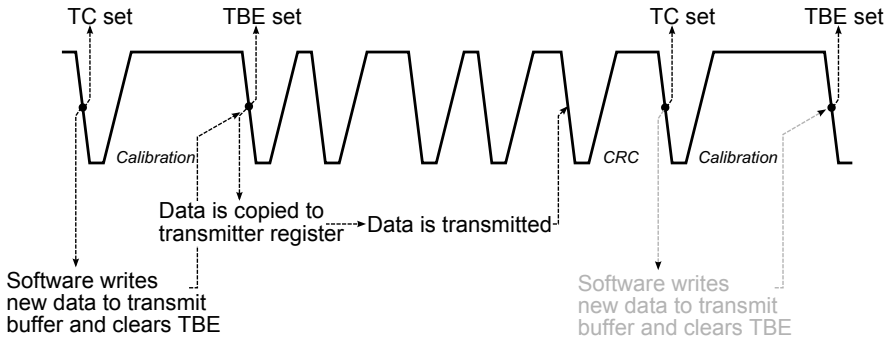



**Figure 21-4. Transmission Complete driven SENT transfer without Pause Pulse**

### 21.8.4.3 Double-buffered Transmission with Pause Pulse

This option is similar to the double-buffered transmission without pause-pulse (for details please refer to Section 21.8.4.1, “Double-buffered Transmission without Pause Pulse). The only difference is that due to the pause pulse the message periods become longer offering even more time for the software to prepare new data.

The software uses the Transmit Buffer Empty interrupt (TBE) to prepare new data for the next transmission. An example for this case is provided in Figure 21-5 below.

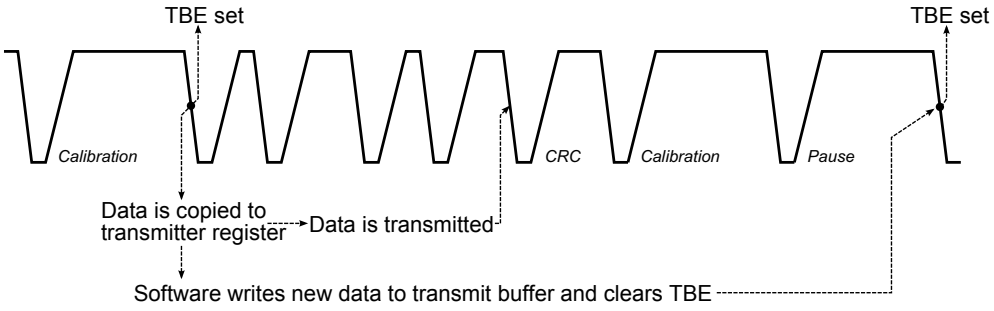



**Figure 21-5. Transmit-Buffer Empty driven SENT transfer with Pause Pulse**

### 22.4.5.4 P-Flash Commands

Table 22-29 summarizes the valid P-Flash commands along with the effects of the commands on the P-Flash block and other resources within the Flash module.

**Table 22-29. P-Flash Commands**

| FCMD | Command | Function on P-Flash Memory |
|---|---|---|
| 0x01 | Erase Verify All Blocks | Verify that all P-Flash (and EEPROM) blocks are erased. |
| 0x02 | Erase Verify Block | Verify that a P-Flash block is erased. |
| 0x03 | Erase Verify P-Flash Section | Verify that a given number of words starting at the address provided are erased. |
| 0x04 | Read Once | Read a dedicated 64 byte field in the nonvolatile information register in P-Flash block that was previously programmed using the Program Once command. |
| 0x06 | Program P-Flash | Program a phrase in a P-Flash block. |
| 0x07 | Program Once | Program a dedicated 64 byte field in the nonvolatile information register in P-Flash block that is allowed to be programmed only once. |
| 0x08 | Erase All Blocks | Erase all P-Flash (and EEPROM) blocks.<br>An erase of all Flash blocks is only possible when the FPLDIS, FPHDIS, and FPOPEN bits in the FPROT register and the DPOPEN bit in the DFPROT register are set prior to launching the command. |
| 0x09 | Erase Flash Block | Erase a P-Flash (or EEPROM) block.<br>An erase of the full P-Flash block is only possible when FPLDIS, FPHDIS and FPOPEN bits in the FPROT register are set prior to launching the command. |
| 0x0A | Erase P-Flash Sector | Erase all bytes in a P-Flash sector. |
| 0x0B | Unsecure Flash | Supports a method of releasing MCU security by erasing all P-Flash (and EEPROM) blocks and verifying that all P-Flash (and EEPROM) blocks are erased. |
| 0x0C | Verify Backdoor Access Key | Supports a method of releasing MCU security by verifying a set of security keys. |
| 0x0D | Set User Margin Level | Specifies a user margin read level for all P-Flash blocks. |
| 0x0E | Set Field Margin Level | Specifies a field margin read level for all P-Flash blocks (special modes only). |
| 0x13 | Protection Override | Supports a mode to temporarily override Protection configuration (for P-Flash and/or EEPROM) by verifying a key. |

### 22.4.5.5 EEPROM Commands

Table 22-30 summarizes the valid EEPROM commands along with the effects of the commands on the EEPROM block.

**Table 22-30. EEPROM Commands**

| FCMD | Command | Function on EEPROM Memory |
|---|---|---|
| 0x01 | Erase Verify All Blocks | Verify that all EEPROM (and P-Flash) blocks are erased. |

**Table 22-35. Erase Verify Block Command Error Handling**

| Register | Error Bit | Error Condition |
|---|---|---|
| FSTAT | ACCERR | Set if CCOBIX[2:0] != 001 at command launch |
| | | Set if an invalid global address [23:0] is supplied see Table 22-2) |
| | FPVIOL | None |
| | MGSTAT1 | Set if any errors have been encountered during the read or if blank check failed. |
| | MGSTAT0 | Set if any non-correctable errors have been encountered during the read or if blank check failed. |

### 22.4.7.3 Erase Verify P-Flash Section Command

The Erase Verify P-Flash Section command will verify that a section of code in the P-Flash memory is erased. The Erase Verify P-Flash Section command defines the starting point of the code to be verified and the number of phrases.

**Table 22-36. Erase Verify P-Flash Section Command FCCOB Requirements**

| Register | FCCOB Parameters | |
|---|---|---|
| FCCOB0 | 0x03 | Global address [23:16] of a P-Flash block |
| FCCOB1 | Global address [15:0] of the first phrase to be verified | |
| FCCOB2 | Number of phrases to be verified | |

Upon clearing CCIF to launch the Erase Verify P-Flash Section command, the Memory Controller will verify the selected section of Flash memory is erased. The CCIF flag will set after the Erase Verify P-Flash Section operation has completed. If the section is not erased, it means blank check failed, both MGSTAT bits will be set.

**Table 22-37. Erase Verify P-Flash Section Command Error Handling**

| Register | Error Bit | Error Condition |
|---|---|---|
| FSTAT | ACCERR | Set if CCOBIX[2:0] != 010 at command launch |
| | | Set if command not available in current mode (see Table 22-28) |
| | | Set if an invalid global address [23:0] is supplied see Table 22-2) |
| | | Set if a misaligned phrase address is supplied (global address [2:0] != 000) |
| | | Set if the requested section crosses a the P-Flash address boundary |
| | FPVIOL | None |
| | MGSTAT1 | Set if any errors have been encountered during the read or if blank check failed. |
| | MGSTAT0 | Set if any non-correctable errors have been encountered during the read or if blank check failed. |

In **Table D-2.** the timing characteristics for slave mode are listed.

**Table D-2. SPI Slave Mode Timing Characteristics (Junction Temperature From –40°C To +175°C)**

| Num | Characteristic | Symbol | | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min | Typ | Max | |
| 1 | SCK Frequency | $f_{sck}$ | DC | — | 1/4 | $f_{bus}$ |
| 1 | SCK Period | $t_{sck}$ | 4 | — | ∞ | $t_{bus}$ |
| 2 | Enable Lead Time | $t_{lead}$ | 4 | — | — | $t_{bus}$ |
| 3 | Enable Lag Time | $t_{lag}$ | 4 | — | — | $t_{bus}$ |
| 4 | Clock (SCK) High or Low Time | $t_{wsck}$ | 4 | — | — | $t_{bus}$ |
| 5 | Data Setup Time (Inputs) | $t_{su}$ | 8 | — | — | ns |
| 6 | Data Hold Time (Inputs) | $t_{hi}$ | 8 | — | — | ns |
| 7 | Slave Access Time (time to data active) | $t_a$ | — | — | 20 | ns |
| 8 | Slave MISO Disable Time | $t_{dis}$ | — | — | 22 | ns |
| 9 | Data Valid after SCK Edge | $t_{vsck}$ | — | — | $30 + t_{bus}$ [1] | ns |
| 10 | Data Valid after $\overline{SS}$ fall | $t_{vss}$ | — | — | $30 + t_{bus}$ [1] | ns |
| 11 | Data Hold Time (Outputs) | $t_{ho}$ | 20 | — | — | ns |
| 12 | Rise and Fall Time Inputs | $t_{rfi}$ | — | — | 8 | ns |
| 13 | Rise and Fall Time Outputs | $t_{rfo}$ | — | — | 8 | ns |

[1] $t_{bus}$ added due to internal synchronization delay

# N.4 0x0100-0x017F S12ZDBG (continued)

| Address | Name | | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0103 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0104 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0105 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0106 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0107 | DBGSCR1 | R | C3SC1 | C3SC0 | 0 | 0 | C1SC1 | C1SC0 | C0SC1 | C0SC0 |
| | | W | | | | | | | | |
| 0x0108 | DBGSCR2 | R | C3SC1 | C3SC0 | 0 | 0 | C1SC1 | C1SC0 | C0SC1 | C0SC0 |
| | | W | | | | | | | | |
| 0x0109 | DBGSCR3 | R | C3SC1 | C3SC0 | 0 | 0 | C1SC1 | C1SC0 | C0SC1 | C0SC0 |
| | | W | | | | | | | | |
| 0x010A | DBGEFR | R | 0 | TRIGF | 0 | EEVF | ME3 | 0 | ME1 | ME0 |
| | | W | | | | | | | | |
| 0x010B | DBGSR | R | 0 | 0 | 0 | 0 | 0 | SSF2 | SSF1 | SSF0 |
| | | W | | | | | | | | |
| 0x010C-0x010F | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0110 | DBGACTL | R | 0 | NDB | INST | 0 | RW | RWE | reserved | COMPE |
| | | W | | | | | | | | |
| 0x0111-0x0114 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0115 | DBGAAH | R | DBGAA[23:16] | | | | | | | |
| | | W | | | | | | | | |
| 0x0116 | DBGAAM | R | DBGAA[15:8] | | | | | | | |
| | | W | | | | | | | | |
| 0x0117 | DBGAAL | R | DBGAA[7:0] | | | | | | | |
| | | W | | | | | | | | |
| 0x0118 | DBGAD0 | R | Bit 31 | 30 | 29 | 28 | 27 | 26 | 25 | Bit 24 |
| | | W | | | | | | | | |
| 0x0119 | DBGAD1 | R | Bit 23 | 22 | 21 | 20 | 19 | 18 | 17 | Bit 16 |
| | | W | | | | | | | | |
| 0x011A | DBGAD2 | R | Bit 15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit 8 |
| | | W | | | | | | | | |
| 0x011B | DBGAD3 | R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | | W | | | | | | | | |