Network SSP Serial Port 16

This chapter describes the signal definitions and operation of the Intel® PXA255 Processor Network Synchronous Serial Protocol (NSSP) serial port.

The NSSP is configured differently than the SSPC.

16.1 Overview

The NSSP is a synchronous serial interface that connects to a variety of external analog-to-digital (A/D) converters, telecommunication CODECs, and many other devices that use serial protocols for data transfer. The NSSP provides support for the following protocols:

- Texas Instruments (TI) Synchronous Serial Protocol*
- Motorola Serial Peripheral Interface* (SPI) protocol
- National Semiconductor Microwire*
- Programmable Serial Protocol (PSP)

The NSSP operates as full-duplex devices for the TI Synchronous Serial Protocol*, SPI*, and PSP protocols and as half-duplex devices for the Microwire* protocol.

The FIFOs can be loaded or emptied by the CPU using programmed I/O or DMA burst transfers.

16.2 Features

- Supports the TI Synchronous Serial Protocol*, the Motorola SPI* protocol, National Semiconductor Microwire*, and a Programmable Serial Protocol (PSP)
- Two independent transmit and receive FIFOs, each 16 samples deep by 32-bits wide
- Sample sizes from four to 32-bits
- Maximum bit rate of 13 Mbps in slave of clock mode, requires using DMA
- Master-mode and slave-mode operation
- Receive-without-transmit operation

16.3 Signal Description

Table 16-1 lists the external signals between the SSP serial ports and external device. If the port is disabled, its pins are available for GPIO use. See Section 4.1, “General-Purpose I/O” for details on configuring pin direction and Section 4.2, “Interrupt Controller” for Interrupt capabilities.

Table 16-1. SSP Serial Port I/O Signals

Name	Direction	Description
NSSPSCLK	Input/Output	NSSPSCLK is the serial bit clock used to control the timing of a transfer. NSSPSCLK is generated internally (master mode) or is supplied externally (slave mode) as indicated by SSCR1[SCLKDIR] as defined in Table 16-4.
NSSPSFRM	Input/Output	NSSPSFRM is the serial frame indicator that indicates the beginning and the end of a serialized data word. SSPSFRM is generated internally (master mode) or is supplied externally (slave mode) as indicated by SSCR1[SFRMDIR] as defined in Table 16-4.
NSSPTXD	Output	NSSPTXD is the transmit data (serial data out) serialized data line. It is available on two GPIO pins, GPIO[83] or GPIO[84]. See Section 4.1, “General-Purpose I/O” for details.
NSSPRXD	Input	NSSPRXD is the receive data (serial data in) serialized data line. It is available on two GPIO pins, GPIO[83] or GPIO[84]. See Section 4.1, “General-Purpose I/O” for details.

The Network SSP can output either NSSPTXD and NSSPRXD on either GPIO[83] or GPIO[84]. This allows a system to dynamically change the direction of transfer for this port. The NSSP can change direction if enabled, but it must be idle.

16.4 Operation

The SSP controller transfers serial data between the PXA255 processor and an external device through FIFOs. The PXA255 processor CPU initiates the transfers using programmed I/O or DMA bursts to and from memory. Separate transmit and receive FIFOs and serial data paths permit simultaneous transfers in both directions to and from the external device, depending on the protocols chosen.

Programmed I/O transfers data directly between the CPU and the SSP Data Register (SSDR). DMA transfers data between memory and the SSP Data Register (SSDR). Data written to the SSP Data Register (by either the CPU or DMA) is automatically transmitted by the transmit FIFO. Data received by the receive FIFO is automatically sent to the SSP Data Register.

16.4.1 Processor and DMA FIFO Access

The CPU or DMA accesses data through the SSP transmit and receive FIFOs. A CPU access takes the form of programmed I/O, transferring one FIFO entry per access. The FIFO are seen as one 32-bit location by the processor. CPU accesses are normally triggered by an SSSR interrupt and are always 32-bits wide. CPU writes to the FIFOs ignore bits beyond the programmed FIFO data size (EDSS/DSS value); and CPU reads return zeroes in the MSBs down to the programmed data size.

The FIFOs can also be accessed by DMA bursts (in multiples of one, two or four bytes) depending upon the EDSS value. When SSCR0[EDSS] is set, DMA bursts must be in multiples of four bytes (the DMA must have the SSP configured as a 32-bit peripheral).When SSCR0[EDSS] is cleared, DMA bursts must be in multiples of one or two bytes (the DMA's DCMD[WIDTH] register must be at least the SSP data size programmed into the SSCR0[EDSS] and SSCR0[DSS]. If the DMA DCMD[WIDTH] field is configured for 1 byte width, the DMA burst size must be 8 or 16.

For writes, the SSP takes the data from the transmit FIFO, serializes it, and sends it over the serial wire (SSPTXD) to the external device. Receive data from the external device (on SSPRXD) is converted to parallel words and stored in the receive FIFO.

When exceeded, a programmable trigger threshold generates an interrupt or DMA service request that, if SSCR1[TIE] or SSCR1[TSRE] are enabled, signal the CPU or DMA, respectively, to refill the transmit FIFO. Similarly, a programmable trigger threshold generates an interrupt or DMA service request that, if SSCR1[RIE] or SSCR1[RSRE] are enabled, signal the CPU or DMA, respectively, to empty the receive FIFO.

The receive and transmit FIFOs are differentiated by whether the access is a read or a write transfer. Reads automatically target the receive FIFO, while writes write data to the transmit FIFO. From a memory-map perspective, both reads and writes are at the same address. The FIFOs are 16 samples deep by one word wide.

16.4.2 Trailing Bytes in the Receive FIFO

When the number of samples in the receive FIFO is less than the trigger threshold and no additional data is received, the remaining bytes are called trailing bytes. Trailing bytes must be handled by the processor. Trailing bytes are identified via a time-out mechanism and the existence of data within the receive FIFO.

16.4.2.1 Time-out

A time-out condition exists when the receive FIFO is idle for the period of time defined by the Time-Out Register (SSTO). When a time-out occurs, the receiver time-out interrupt (SSSR[TINT]) is set. If the time-out interrupt is enabled (SSCR1[TINTE] set) a time-out interrupt occurs to signal the processor that a time-out condition has occurred. The time-out timer is reset after receiving a new sample or after the processor reads the receive FIFO. Once SSSR[TINT] is set it must be cleared by writing a one to it. If the time-out interrupt is enabled, clearing SSCR1[TINTE] also causes the time-out interrupt to be de-asserted.

16.4.2.2 Removing Trailing Bytes

In this case, no receive DMA service request is generated. To read out the trailing bytes, have the software wait for the time-out interrupt and then read all remaining entries as indicated by SSSR[RFL] and SSSR[RNE].

Note: The time-out interrupt must be enabled by setting SSCR1[TINTE].

16.4.3 Data Formats

Four pins transfer data between the PXA255 processor and external CODECs or modems. Although four serial-data formats exist, each has the same basic structure and in all cases the pins are used as follows:

- SSPSCLK–Defines the bit rate at which serial data is driven onto and sampled from the port.
- SSPSFRM–Defines the boundaries of a basic data unit, comprised of multiple serial bits.
- SSPTXD–The serial data path for transmitted data, from system to peripheral.
- SSPRXD–The serial data path for received data, from peripheral to system.

A data frame can contain from four to 32-bits, depending on the selected protocol. Serial data is transmitted most significant bit first. Four protocols are supported: TI Synchronous Serial Protocol*, SPI, Microwire*, and a PSP.

The SSPSFRM function and use varies between each protocol.

- For the TI Synchronous Serial Protocol*, SSPSFRM is pulsed high for one (serial) data period at the start of each frame. Master and slave modes are supported. TI Synchronous Serial Protocol* is a full-duplex protocol.
- For the SPI* protocol, SSPSFRM functions as a chip select to enable the external device (target of the transfer) and is held active-low during the data transfer (during continuous transfers, the SSPSFRM signal can be either held low or pulsed depending upon the value of SSCR1_x[SPH]). Master and slave modes are supported. SPI* is a full-duplex protocol.
- For the Microwire* protocol, SSPSFRM functions as a chip select to enable the external device (target of the transfer) and is held active-low during the data transfer. Slave mode is not supported for Microwire*. SSPSFRM for Microwire* is also held low during continuous transfers. Microwire* is a half-duplex protocol.
- For the PSP, SSPSFRM is programmable in direction, delay, polarity, and width. Master and slave modes are supported. PSP can be programmed to be either full or half duplex.

The SSPSCLK function and use varies between each protocol.

- For TI Synchronous Serial Protocol*, data sources switch transmit data on the rising edge of SSPSCLK and sample receive data on the falling edge. Master and slave modes are supported.
- For SPI*, the SSP lets programmers select which edge of SSPSCLK to use for switching transmit data and for sampling receive data. In addition, users can move the phase of SSPSCLK, shifting its active state one-half cycle earlier or later at the start and end of a frame. Master and slave modes are supported.
- For Microwire*, both data sources switch (change to the next bit) transmit data on the falling edge of SSPSCLK and sample receive data on the rising edge. Slave mode is not supported for Microwire*.
- For PSP, the protocol allows for the configuration of which edge of the SSPSCLK is used for switching transmit data and the edge for sampling receive data. In addition, the idle state for SSPSCLK can be controlled and the number of active clocks that precede and follow the data transmission. Master and slave modes are supported.

Microwire* uses a half-duplex, master-slave messaging protocol. At the start of a frame, the controller transmits a one or two-byte control message to the peripheral; no data is sent by the peripheral. The peripheral interprets the message and if the message is a read request, the peripheral responds with the requested data, one clock after the last bit of the request message. Return data—part of the same frame—can be from four to 16-bits in length. The total frame length is 13 to 33 bits. The SSPSCLK is active during the entire frame.

Note: The serial clock (SSPSCLK), if driven by the SSP, toggles only while an active data transfer is underway, unless receive-without-transmit mode is enabled by setting SSCR1[RWOT] and the frame format is not Microwire*, in which case the SSPSCLK toggles regardless of whether transmit data exist within the transmit FIFO. At other times, SSPSCLK holds in an inactive or idle state as defined by the protocol.

16.4.3.1 TI Synchronous Serial Protocol* Details

When outgoing data in the SSP controller is ready to transmit, SSPSFRM asserts for one clock period. On the following clock, data to be transmitted is driven on SSPTXD one bit at a time, the most significant bit first. For receive data, the peripheral similarly drives data on the SSPRXD pin. The word length can be from four to 32-bits. All output transitions occur on the rising edge of SSPSCLK while data sampling occurs on the falling edge. At the end of the transfer, the SSPTXD signal either retains the value of the last bit sent (LSB) or clears depending on the serial form and the value of the SSPSP[ETDS] (See Figure 16-1 through Figure 16-8). If the SSP is disabled or reset, SSPTXD is forced low.

Figure 16-1 shows the TI Synchronous Serial Protocol for when back-to-back frames are transmitted. Figure 16-2 shows the TI Synchronous Serial Protocol for a single transmitted frame. Once the transmit FIFO contains data, SSPSFRM is pulsed high for one SSPSCLK period and the value to be transmitted is transferred from the transmit FIFO to the transmit logic serial shift register. On the next rising edge of SSPSCLK, the most significant bit of the four to 32-bit data frame is shifted to the SSPTXD pin. Likewise, the MSB of the received data is shifted onto the SSPRXD pin by the off-chip serial slave device. Both the SSP and the off-chip serial slave device then latch each data bit into the serial shifter on the falling edge of each SSPSCLK. The received data is transferred from the serial shifter to the receive FIFO on the first rising edge of SSPSCLK after the last bit has been latched.

For back-to-back transfers, the start of one frame is the completion of the previous frame. The MSB of one transfer immediately follows the LSB of the preceding with no “dead” time between them. When the SSP is a master to the frame sync (SSPSFRM) and a slave to the clock (SSPSCLK), at least three extra clocks are needed at the beginning and end of each block of transfers to synchronize internal control signals (a block of transfers is a group of back-to-back continuous transfers).

Figure 16-1. Texas Instruments Synchronous Serial Frame* Protocol (multiple transfers)

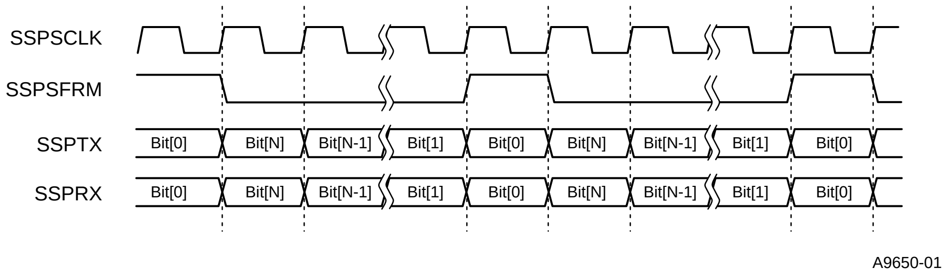
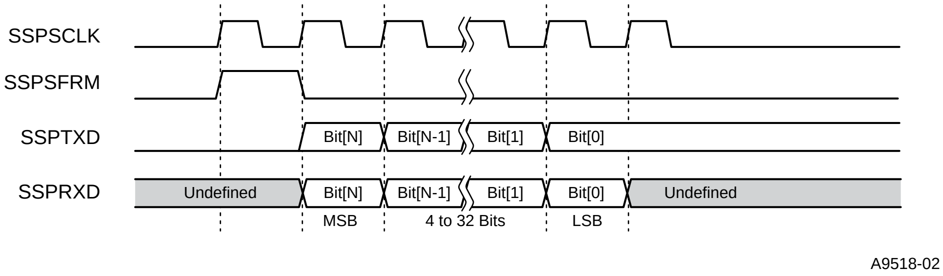


Figure 16-2. Texas Instruments Synchronous Serial Frame* Protocol (single transfers)



16.4.3.2 SPI Protocol Details

The SPI protocol has four possible sub-modes, depending on the SSPSCLK edges selected for driving data and sampling received data and on the selection of the phase mode of SSPSCLK (see Section 16.4.3.2.1 for complete descriptions of each mode).

When the SSP is disabled or in idle mode, SSPSCLK and SSPTXD are low and SSPSFRM is high. When transmit data is available to send, SSPSFRM goes low (one clock period before the first rising edge of SSPSCLK) and stays low for the remainder of the frame. The most significant bit of the serial data is driven onto SSPTXD one half-cycle later. Halfway into the first bit period, SSPSCLK asserts high and continues toggling for the remaining data bits. Data transitions on the falling edge of SSPSCLK. Four to 32 bits can be transferred per frame.

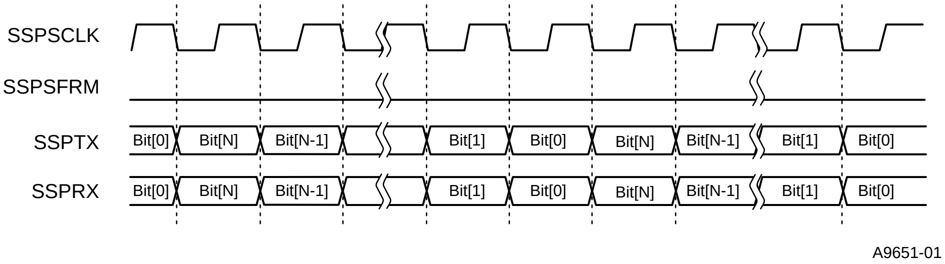
With the assertion of SSPSFRM, receive data is simultaneously driven from the peripheral on SSPRXD, MSB first. Data transitions on SSPSCLK falling edges and is sampled by the controller on rising edges. At the end of the frame, SSPSFRM is de-asserted high one clock period (one half clock cycle after the last falling edge of SSPSCLK) after the last bit latched at its destination and the completed incoming word is shifted into the incoming FIFO. The peripheral can drive SSPRXD to a high-impedance state after sending the last bit of the frame. SSPTXD retains the last value transmitted when the controller goes into idle mode, unless the SSP is disabled or reset (which forces SSPTXD low).

For back-to-back transfers, frames start and complete similar to single transfers, except SSPSFRM does not de-assert between words. Both transmitter and receiver are configured for the word length and internally track the start and end of frames. There are no dead bits; the least significant bit of one frame is followed immediately by the most significant bit of the next.

When using the SPI protocol, the SSP can either be a master or a slave device. However, the clock and frame direction must be the same. For example, the SSCR1[SCLKDIR] and SSCR0[SFRMDIR] must both be set or both be cleared.

Figure 16-3 shows when back-to-back frames are transmitted for the Motorola SPI* frame protocol. Figure 16-4 shows one of the four possible configurations for the Motorola SPI* frame protocol for a single transmitted frame.

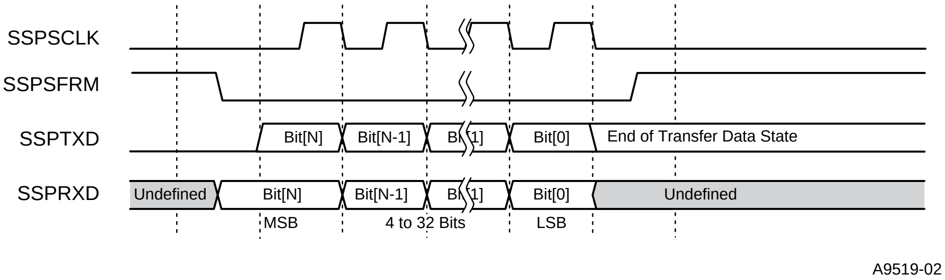
Figure 16-3. Motorola SPI* Frame Protocol (multiple transfers)



Note: When configured as either master or slave (to clock or frame) the SSP continues to drive SSPTXD with the last bit of data sent (the LSB). If SSCR0[SSE] is cleared, SSPTXD goes low. The state of SSPRXD is undefined before the MSB and after the LSB is transmitted. For minimum power consumption, this pin must not float.

Note: The phase and polarity of SSPSCLK can be configured for four different modes. This example shows just one of those modes (SSCR1[SPO] = 0, SSCR1[SPH] = 0).

Figure 16-4. Motorola SPI* Frame Protocol (single transfers)



Note: When configured as either master or slave (to clock or frame) the SSP continues to drive SSPTXD with the last bit of data sent (the LSB). If SSCR0[SSE] is cleared, SSPTXD goes low. The state of SSPRXD is undefined before the MSB and after the LSB is transmitted. For minimum power consumption, this pin must not float.

16.4.3.2.1 Serial Clock Phase (SPH)

The phase relationship between the SSPSCLK and the serial frame (SSPSFRM) pins when the Motorola SPI* protocol is selected is controlled by SSCR1[SPH].

When SPH is cleared, SSPSCLK remains in its inactive or idle state (as determined by SSCR1[SPO]) for one full cycle after SSPSFRM is asserted low at the beginning of a frame. SSPSCLK continues to transition for the rest of the frame. It is then held in its inactive state for one-half of an SSPSCLK period before SSPSFRM is de-asserted high at the end of the frame.

When SPH is set, SSPSCLK remains in its inactive or idle state (as determined by SSCR1[SPO]) for one-half cycle after SSPSFRM is asserted low at the beginning of a frame. SSPSCLK continues to transition for the remainder of the frame and is then held in its inactive state for one full SSPSCLK period before SSPSFRM is de-asserted high at the end of the frame.

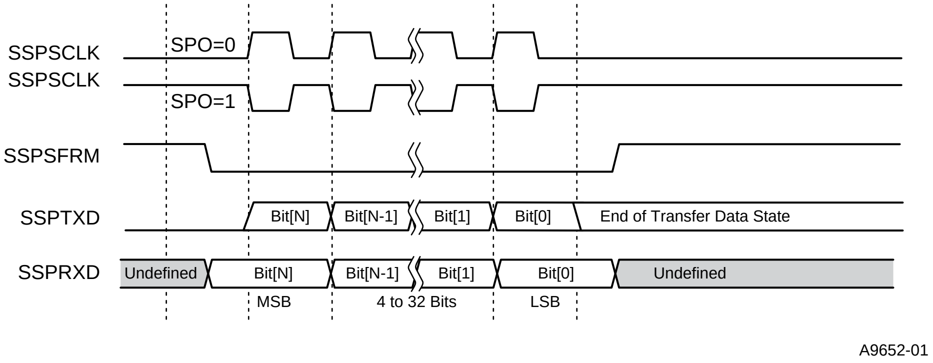
The combination of the SSCR1[SPO] and SSCR1[SPH] settings determine when SSPSCLK is active during the assertion of SSPSFRM and which SSPSCLK edge transmits and receives data on the SSPTXD and SSPRXD pins.

When programming SSCR1[SPO] and SSCR1[SPH] to the same value (both set or both cleared), transmit data is driven on the falling edge of SSPSCLK and receive data is latched on the rising edge of SSPSCLK. When programming SSCR1[SPO] and SSCR1[SPH] to opposite values (one set and the other cleared), transmit data is driven on the rising edge of SSPSCLK and receive data is latched on the falling edge of SSPSCLK.

Note: SSCR1[SPH] is ignored for all data frame formats except for the Motorola SPI* protocol.

Figure 16-6 shows the pin timing for all four programming combinations of SSCR1[SPO] and SSCR1[SPH]. The SSCR1[SPO] inverts the polarity of the SSPSCLK signal and SSCR1[SPH] determines the phase relationship between SSPSCLK and SSPSFRM, shifting the SSPSCLK signal one-half phase to the left or right during the assertion of SSPSFRM.

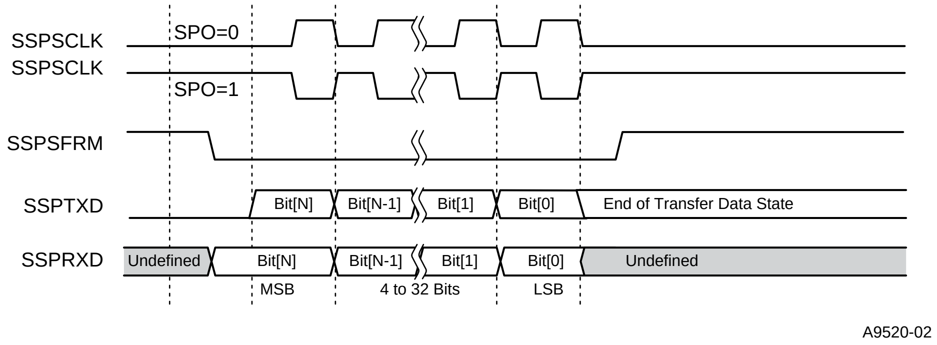
Figure 16-5. Motorola SPI* Frame Protocols for SPO and SPH Programming (multiple transfers)



Note: When configured as either master or slave (to clock or frame) the SSP continues to drive SSPTXD with the last bit of data sent (the LSB). If SSCR0[SSE] is cleared, SSPTXD goes low. The state of

SSPRXD is undefined before the MSB and after the LSB is transmitted. For minimum power consumption, this pin must not float.

Figure 16-6. Motorola SPI* Frame Protocols for SPO and SPH Programming (single transfers)



Note: When configured as either master or slave (to clock or frame) the SSP continues to drive SSPTXD with the last bit of data sent (the LSB). If SSCR0[SSE] is cleared, SSPTXD goes low. The state of SSPRXD is undefined before the MSB and after the LSB is transmitted. For minimum power consumption, this pin must not float.

16.4.3.3 Microwire* Protocol Details

The Microwire* protocol is similar to SPI, except transmission is half-duplex instead of full-duplex and it uses master-slave message passing. While in the idle state or when the SSP is disabled, SSPSCLK and SSPTXD are low and SSPSFRM is high.

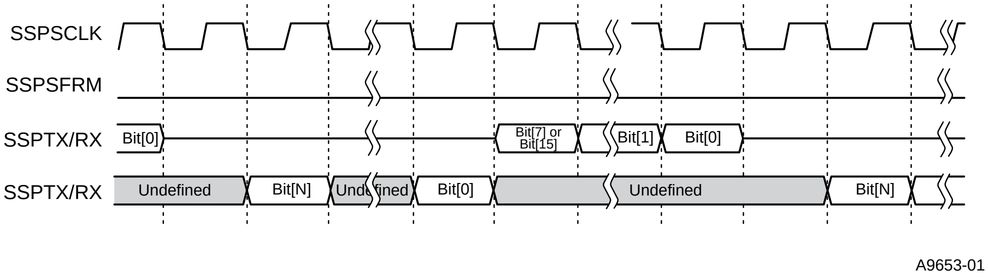
Each serial transmission begins with SSPSFRM asserting low, followed by an eight or 16-bit command word sent from the controller to the peripheral on SSPTXD. The command word data size is selected by the Microwire* transmit data size bit (SSCR1[MWDS]). SSPRXD is controlled by the peripheral and remains in a high-impedance state. SSPSCLK asserts high midway into the command's most significant bit and continues toggling at the bit rate.

One bit-period after the last command bit, the peripheral returns the serial-data requested most significant bit first on SSPRXD. Data transitions on the falling edge of SSPSCLK and is sampled on the rising edge. The last falling edge of SSPSCLK coincides with the end of the last data bit on SSPRXD and SSPSCLK remains low after that (if it is the only word or the last word of the transfer). SSPSFRM de-asserts high one-half clock period later.

The start and end of a series of back-to-back transfers are like those of a single transfer; however, SSPSFRM remains asserted (low) throughout the transfer. The end of a data word on SSPRXD is followed immediately by the start of the next command byte on SSPTXD with no dead time.

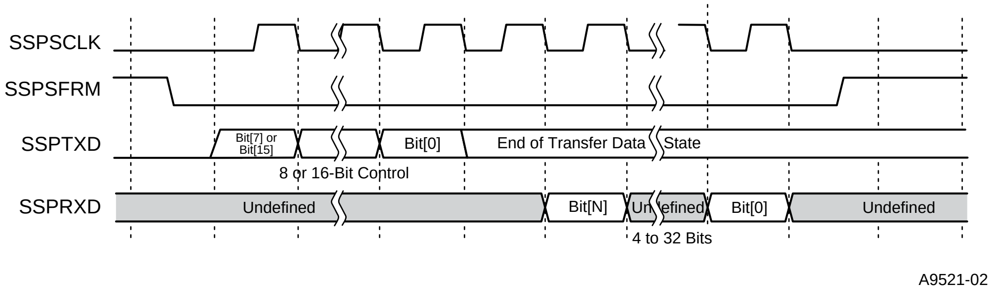
When using the Microwire* protocol, the SSP can function only as a master (frame and clock are outputs). Therefore, both SSCR1[SCLKDIR] and SSCR0[SFRMDIR] must both be cleared. Figure 16-7 shows the National Semiconductor Microwire* frame protocol with eight-bit command words when back-to-back frames are transmitted. Figure 16-8 shows the National Semiconductor Microwire* frame protocol with eight-bit command words for a single transmitted frame.

Figure 16-7. National Semiconductor Microwire* Frame Protocol (multiple transfers)



Note: When configured master the SSP continues to drive SSPTXD with the last bit of data sent (the LSB) or it drives zero, depending on the status of SSPSP[ETDS]. If SSCR0[SSE] is cleared, SSPTXD goes low. The state of SSPRXD is undefined before the MSB and after the LSB is transmitted. For minimum power consumption, this pin must not float.

Figure 16-8. National Semiconductor Microwire* Frame Protocol (single transfers)



Note: When configured master the SSP continues to drive SSPTXD with the last bit of data sent (the LSB) or it drives zero, depending on the status of SSPSP[ETDS]. If SSCR0[SSE] is cleared, SSPTXD goes low. The state of SSPRXD is undefined before the MSB and after the LSB is transmitted. For minimum power consumption, this pin must not float.

16.4.3.4 PSP Details

The PSP provides programmability for several parameters that determine the transfer timings between data.

There are four possible serial clock sub-modes, depending on the SSPSCLK edges selected for driving data and sampling received data and the selection of idle state of the clock.

For the PSP, the idle and disable modes of the SSPTXD, SSPSCLK, and SSPSFRM are programmable via SSPSP[ETDS], SSPSP[SCMODE] and SSPSP[SFRMP]. When transmit data is ready, the SSPSCLK remains in its idle state for the number of serial clock (SSPSCLK) clock periods programmed within the start delay (SSPSP[STRTDLY]) field. SSPSCLK then starts toggling, SSPTXD remains in the idle state for the number of cycles programmed within the dummy start field (SSPSP[DMYSTRT]). The SSPSFRM signal asserts after the number of half-clocks programmed in the field SSPSP[SFRMP]. The SSPSFRM remains asserted for the number of half-clocks programmed within SSPSP[SFRMWDTH]. Four to 32-bits can be transferred per frame. Once the LSB transfers, the SSPSCLK continues toggling based on the dummy stop field (SSPSP[DMYSTOP]). SSPTXD either retains the last value transmitted or is forced to zero, depending on the value programmed within the end of transfer data state field (SSPSP[ETDS]), when the controller goes into idle mode, unless the SSP is disabled or reset (which forces SSPTXD low). Refer to Table 16-2 for more information.

With the assertion of SSPSFRM, receive data is simultaneously driven from the peripheral on SSPRXD, MSB first. Data transitions on SSPSCLK based on the serial clock mode selected and are sampled by the controller on the opposite edge. When the SSP is a master to the frame sync (SSPSFRM) and a slave to the clock (SSPSCLK), at least three extra clocks are needed at the beginning and end of each block of transfers to synchronize internal control signals (a block of transfers is a group of back-to-back continuous transfers).

Figure 16-9. Programmable Serial Protocol (multiple transfers)

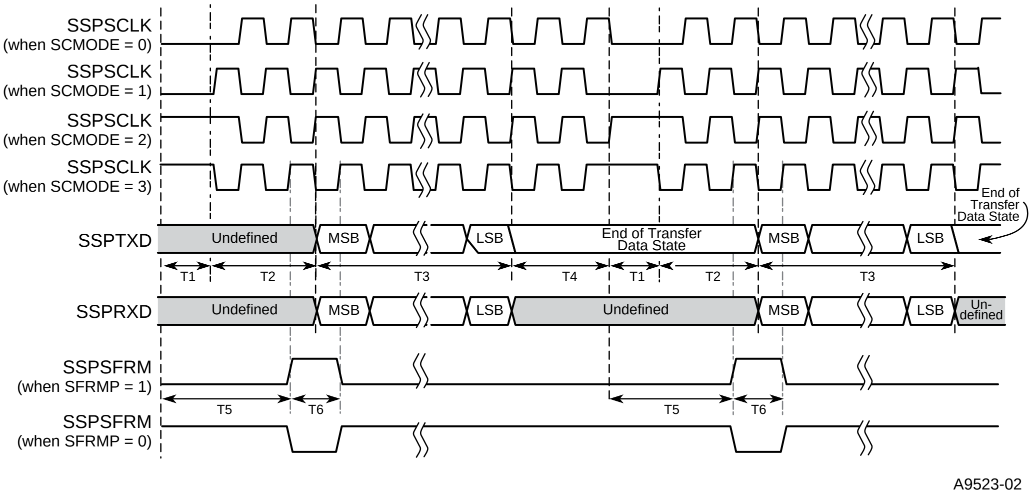


Figure 16-10. Programmable Serial Protocol (single transfers)

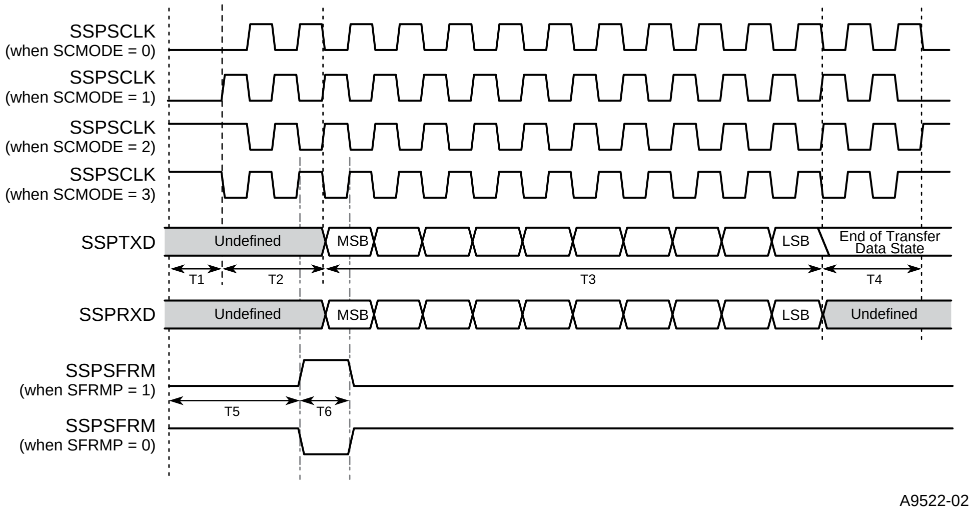


Table 16-2. Programmable Serial Protocol (PSP) Parameters

Symbol	Definition	Range	Units
—	Serial clock mode (SSPSP[SCMODE])	(Drive, Sample, SSPSCLK Idle) 0 - Fall, Rise, Low 1 - Rise, Fall, Low 2 - Rise, Fall, High 3 - Fall, Rise, High	—
—	Serial frame polarity (SSPSP[SFRMP])	High or Low	—
T1	Start delay (SSPSP[STRTDLY])	0 - 7	Clock period
T2	Dummy start (SSPSP[DMYSTRT])	0 - 3	Clock period
T3	Data size (SSCR0[EDSS] and SSCR0[DSS])	4 - 32	Clock period
T4	Dummy stop (SSPSP[DMYSTOP])	0 - 3	Clock period
T5	SSPSFRM delay (SSPSP[SFRMDLY]	0 - 88	Half clock period
T6	SSPSFRM width (SSPSP[SFRMWDTH]	1 - 44	Clock period
	End of transfer data state (SSPSP[ETDS])	Low or [bit 0]	—

Note: The SSPSFRM delay must not extend beyond the end of T4. SSPSFRM Width must be asserted for at least 1 SSPSCLK, and must be deasserted before the end of the T4 cycle (i.e. in terms of time, not bit values, (T5 + T6) <= (T1 + T2 + T3 + T4), 1<= T6 < (T2 + T3 + T4), and (T5 + T6) >= (T1 + 1) to ensure that SSPSFRM is asserted for at least 2 edges of the SSPSCLK). While the PSP can be programmed to generate the assertion of SSPSFRM during the middle of the data transfer (after the MSB was sent), the SSP is not able to receive data in frame slave mode (SSCR1[SFRMDIR] is set) if the assertion of frame is not before the MSB is sent (For example, T5 <= T2 if SSCR1[SFRMDIR] is set). Transmit Data transitions from the “End of Transfer Data State” to the next MSB value upon the assertion of frame. The start delay field should be programmed to 0 whenever SSPSCLK or SSPSFRM is configured as an input.

16.4.4 Hi-Z on SSPTXD

The PXA255 processor NSSP supports placing SSPTXD into Hi-Z during idle times instead of driving SSPTXD.

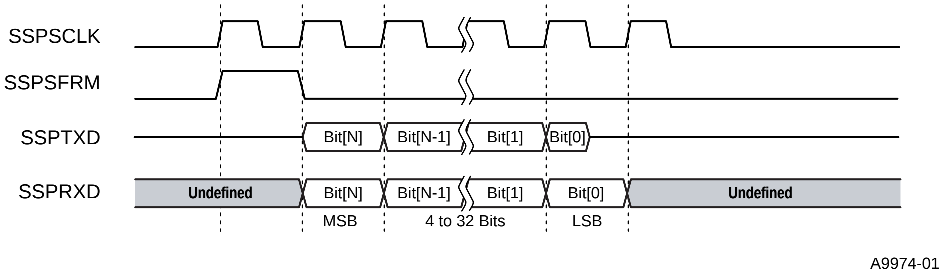
SSCR1[TTE] enables Hi-Z on SSPTXD. SSCR1[TTELP] controls when SSPTXD is placed into Hi-Z.

16.4.4.1 TI Synchronous Serial Port

When SSCR1[TTE] is 0, the SSP behaves as described in Section 16.4.3.1.

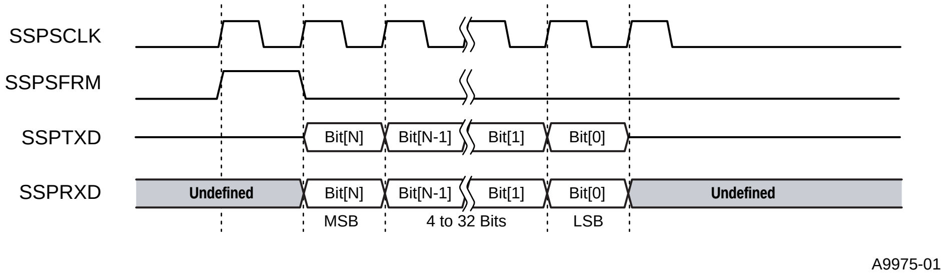
If SSCR1[TTE] is 1 and SSCR1[TTELP] is 0, SSPTXD is driven with the MSB at the first rising edge of SSPSCLK after SSPSFRM is asserted. SSPTXD is Hi-Z after the falling edge of SSPSCLK for the LSB (1 clock edge after the clock edge that starts the LSB). Figure 16-11shows the pin timing for this mode.

Figure 16-11. TI SSP with SSCR[TTE]=1 and SSCR[TTELP]=0



If SSCR1[TTE] is 1 and SSCR1[TTELP] is 1, SSPTXD is driven with the MSB at the first rising edge of SSPSCLK after SSPSFRM is asserted. SSPTXD is Hi-Z at the next rising edge of SSPSCLK after the LSB (2 clock edges after the clock edge that starts the LSB). Figure 16-12 shows the pin timing for this mode.

Figure 16-12. TI SSP with SSCR[TTE]=1 and SSCR[TTELP]=1



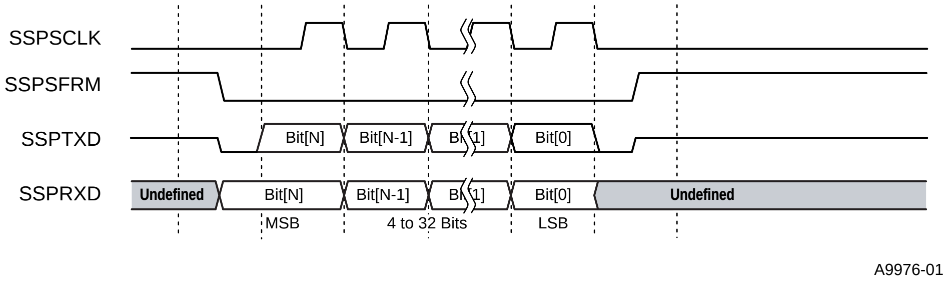
Note: If SSPSCLK is an input, the device driving SSPSCLK must provide another clock edge to cause the TXD line to go to Hi-Z.

16.4.4.2 Motorola SPI

When SSCR1[TTE] is 0, the SSP behaves as described in Section 16.4.3.2.

If SSCR1[TTE] is 1, SSPTXD is driven only when SSPSFRM is 0. When SSPSFRM is 1, SSPTXD is Hi-Z. During the time between the last falling edge and SSPSFRM rising, SSPSP[EDTS] controls the value driven on SSPTXD. Figure 16-13 shows the pin timing for this mode.

Figure 16-13. Motorola SPI with SSCR[TTE]=1



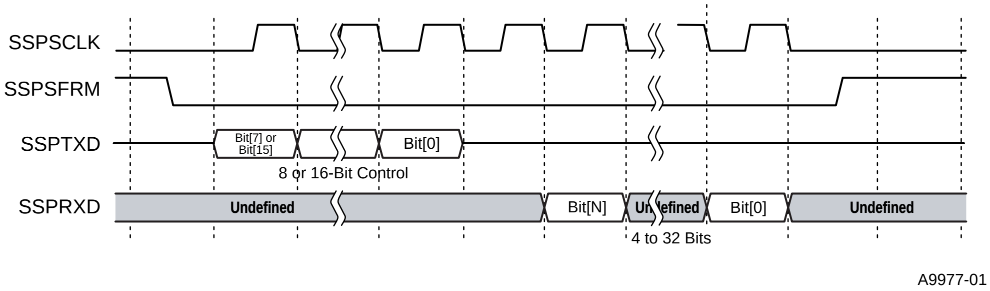
Note: SSCR1[TTELP] must be 0 for Motorola SPI.

16.4.4.3 National Semiconductor Microwire

When SSCR1[TTE] is 0, the SSP behaves as described in Section 16.4.3.3.

If SSCR1[TTE] is 1, SSPTXD is driven at the same clock edge that the MSB is driven. SSPTXD is Hi-Z after the next rising edge of SSPSCLK for the LSB (1 clock edge after the clock edge that starts the LSB). Figure 16-14 shows the pin timing for this mode.

Figure 16-14. National Semiconductor Microwire with SSCR1[TTE]=1



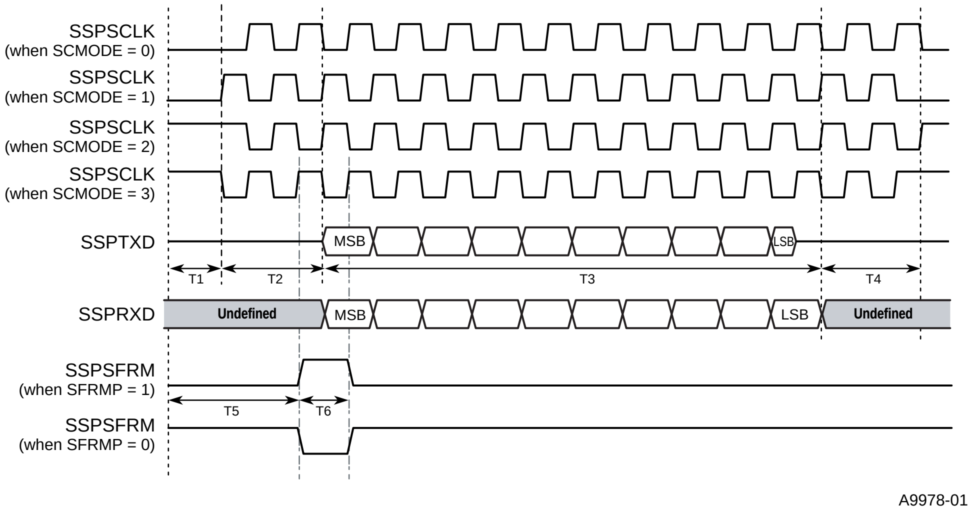
Note: SSCR1[TTELP] must be 0 for National Semiconductor Microwire.

16.4.4.4 Programmable Serial Protocol

When SSCR1[TTE] is 0, the SSP behaves as described in Section 16.4.3.4.

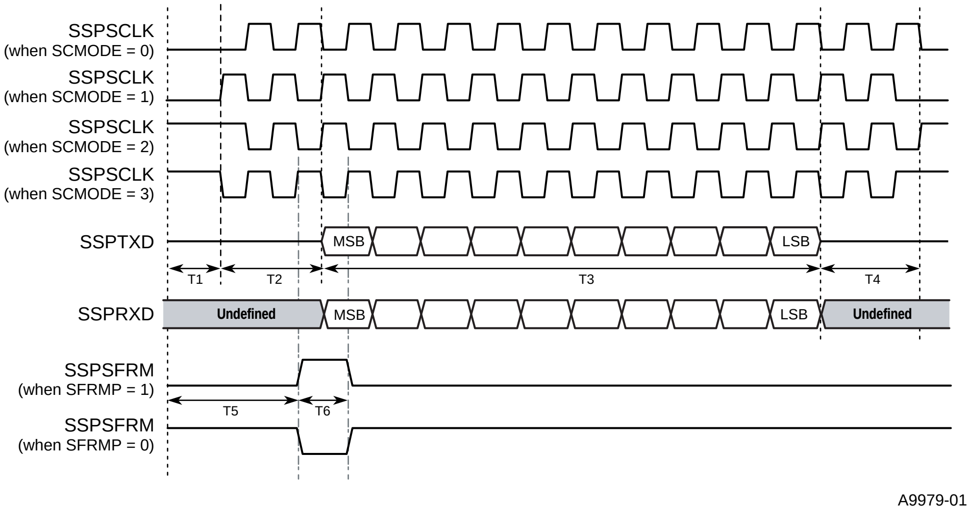
If SSCR1[TTE] is 1 and SSCR1[TTELP] is 0, SSPTXD is driven at the same clock edge that the MSB is driven. If the SSP is a slave to frame SSPTXD is Hi-Z on the clock edge after the edge that starts the LSB. Figure 16-15 shows the pin timing for this mode.

Figure 16-15. PSP mode with SSCR1[TTE]=1 and SSCR1[TTELP]=0 (slave to frame)



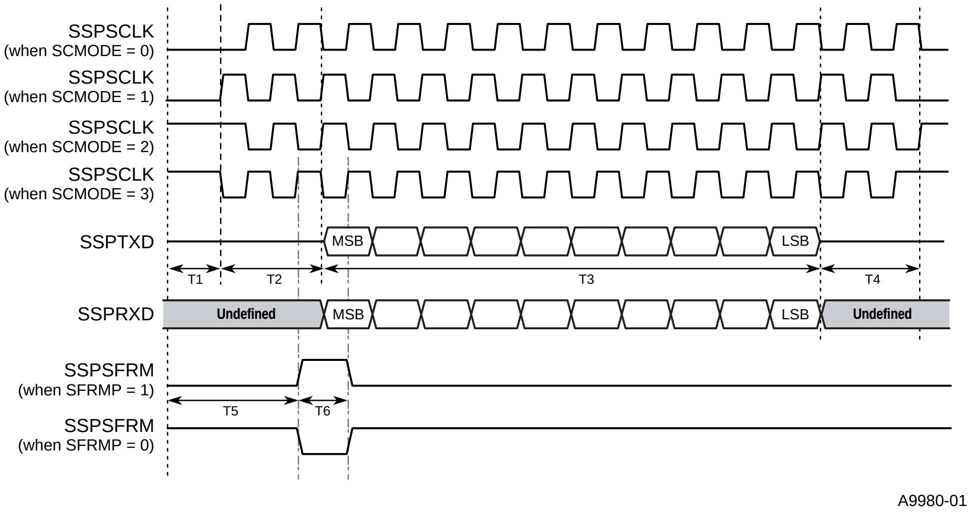
If the SSP is a master to frame, SSPTXD is Hi-Z two clock edges after the clock edge that drives the LSB. This occurs even if the SSP is a master of clock and this clock edge does not appear on the SSPSCLK. Figure 16-16 shows the pin timing for this mode.

Figure 16-16. PSP mode with SSCR1[TTE]=1 and SSCR1[TTELP]=0 (master to frame)



SSCR1[TTELP] can only be set to 1 in PSP mode if the SSP is a slave to frame. If SSCR1[TTE] is 1 and SSCR1[TTELP] is 1 and the SSP is a slave to frame, SSPTXD is driven at the same clock edge that the MSB is driven. SSPTXD is Hi-Z two clock edges after the clock edge that starts the LSB. This occurs even if the SSP is a master of clock and this clock edge does not appear on the SSPSCLK. If the SSP is a slave of clock, then the device driving SSPSCLK must provide another clock edge. Figure 16-17 shows the pin timing for this mode.

Figure 16-17. PSP mode with SSCR1[TTE]=1 and SSCR1[TTELP]=1 (must be slave to frame)



16.4.5 FIFO Operation

Two separate and independent FIFOs are present for transmit (to peripheral) and receive (from peripheral) serial data. FIFOs are filled or emptied by programmed I/O or DMA bursts.

16.4.5.1 Using Programmed I/O Data Transfers

The PXA255 processor can perform FIFO filling and emptying in response to an interrupt from the FIFO logic. Each FIFO has a programmable trigger threshold at which an interrupt is triggered. When the number of entries in the receive FIFO exceeds the value in SSCR1[RFT], an interrupt is generated (if enabled). This interrupt signals the CPU to empty the receive FIFO. When the number of entries in the transmit FIFO is less than or equal to the value of (SSCR1[TFT] + 1), an interrupt is generated (if enabled). This interrupt signals the CPU to refill the transmit FIFO.

Reading the SSP Status Register (see Section 16.5.3) shows whether the FIFO is full, empty or how many samples it contains.

16.4.5.2 Using DMA Data Transfers

The DMA controller can be programmed to transfer data to and from the SSP FIFOs. To prevent overruns of the transmit FIFO or underruns of the receive FIFO when using the DMA, take care when setting the transmit and receive trigger thresholds.

The programming model for using the DMA is as:

- Program the total number of transmit and receive byte lengths, burst sizes, and peripheral width. Program DCMD[WIDTH] to 0b01 for SSP formats of 8 bits or less; to 0b10 for SSP formats of 9 to 16 bits; to 0b11 for SSP formats of more than 16 bits. When DCMD[WIDTH] is 0b01 (1 byte), then the DMA burst size must be configured for 8 or 16 bytes per burst.
- Set the preferred values in the SSP control registers.
- Set the SSE bit in the SSP Control Register 0 to enable the SSP (see Section 16.5.1).
- Set the run bits in the DMA Command Register.
- Wait for both the DMA transmit and receive interrupt requests.
- If the transmit/receive byte length is not an even multiple of the transfer burst size, a trailing-byte condition may occur as described within Section 16.4.2.
- In full-duplex formats where the SSP always receives the same number of data samples as it transmits, the DMA channel must be set up to transmit and receive the same number of bytes.

16.4.6 Baud-Rate Generation

When the SSP is configured as the master of the SSPSCLK (as determined by SSCR1[SCLKDIR]), the baud rate (or serial bit-rate clock SSPSCLK) is generated internally by dividing the 3.6864 MHz clock by a programmable divider (SSCR0[SCR]).

This generates baud rates up to a maximum of 3.68 Mbits per second. When driven by an external clock, SSPSCLK can be driven up to 13 MHz, generating baud rates up to 13 Mbits per second. At these fast baud rates, using polled/interrupt mode is insufficient to keep the FIFO filled. You must use DMA mode.

16.5 Register Descriptions

Each SSP consists of seven registers: three control, one data, one status, one time-out, and one test.

- The SSP control registers (SSCR0, SSCR1) configure the baud rate, data length, frame format, data-transfer mechanism, and port enabling. They also permit setting the FIFO trigger threshold that triggers an interrupt.
- Access all registers using aligned words.

Note: Write the SSP registers after a reset but before the SSP is enabled.

- The SSP Time-Out (SSTO) register programs the time-out value used to signal a specified period of receive FIFO inactivity.
- While in PSP mode, the SSP Programmable Serial Protocol (SSPSP) register programs the parameters used in defining the data transfer.
- The data register is mapped as one 32-bit location, which physically points to either of two 32-bit registers: one register is for writes of data transfers to the transmit FIFO and the other register is for reads that take data from the receive FIFO. A write cycle or burst write puts successive words into the SSP write register and then into the transmit FIFO. A read cycle or burst read takes data from the SSP read register and the receive FIFO reloads it with available data bits it has stored.

  Do not increment the address using read and write DMA bursts.
- Besides showing the state of the FIFO buffers, the status register shows whether the programmable trigger threshold has been passed and whether a transmit or receive FIFO service request is active. The status register also shows how full the FIFO is. Flag bits indicate when the SSP is actively transmitting data, when the transmit FIFO is not full, and when the receive FIFO is not empty. The SSSR[ROR] bit signals an overrun of the receive FIFO In this case newly received data is discarded.

When programming registers, reserved bits must be written as zeroes and read as undefined.

16.5.1 SSP Control Register 0 (SSCR0)

SSCR0, shown in Table 16-3, contains bit fields that control various functions within the SSP. Before enabling the SSP (via SSE) the desired values for this register must be set.

These are read/write registers. Ignore reads from reserved bits. Write zeros to reserved bits.

Table 16-3. SSCR0 Bit Definitions (Sheet 1 of 2)

0x4140_0000 **SSCR0** **Network SSP Serial Port**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reserved											EDSS	SCR												SSE	reserved	FRF		DSS			
?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?	0	0	0	0	0	0

Bits	Name	Description
31:21	—	reserved
20	EDSS	EXTENDED DATA SIZE SELECT: Used in conjunction with DSS to select the size of the data transmitted and received by the SSP. 0 – Pre-appended to the DSS value. Sets the DSS range from 4-16- bits. 1 – Pre-appended to the DSS value. Sets the DSS range from 17-32-bits.
19:8	SCR	THE SERIAL CLOCK RATE: Selects the bit rate of the SSP when in master mode with respect to the SSPSCLK (as defined by SSCR1[SCLKDIR]). The maximum bit rate is 3.6864 Mbps. The clock is divided by the value of SCR plus 1 (a range of 1 to 4096) to generate the serial clock (SSPSCLK). This field is ignored when the SSP is a slave with respect to SSPSCLK (defined by SSCR1[SCLKDIR]) and transmission data rates are determined by the external device (Maximum of 13 MHz). At these fast baud rates, using polled/interrupt mode is insufficient to keep the FIFO filled. You must use DMA mode. **NOTE:** Software must not change SCR when the SSPSCLK is enabled because doing so causes the SSPSCLK frequency to immediately change. Serial bit rate = SSP Clock / (SCR + 1)
7	SSE	SYNCHRONOUS SERIAL PORT ENABLE/DISABLE: Enables and disables all SSP operations. When the port is disabled, all of its clocks can be stopped by programmers to minimize power consumption. When cleared during active operation, the SSP is disabled immediately, terminating the current frame being transmitted or received. Clearing SSE resets the port FIFOs and the status bits; however, the SSP control registers are not reset. **NOTE:** After reset or after clearing the SSE, software must ensure that the SSCR1, SSITR, SSTO, and SSPSP control registers are properly re-configured and that the SSSR register is reset before re-enabling the SSP by setting SSE. Also, SSE must be cleared before re-configuring the SSCR0, SSCR1, or SSPSP registers; any or all control bits in SSCR0 can be written at the same time as the SSE. 0 – SSP operation disabled 1 – SSP operation enabled
6	—	reserved
5:4	FRF	FRAME FORMAT: SELECTS which frame format to use. 0b00 – Serial Peripheral Interface* 0b01 – TI Synchronous Serial Protocol* 0b10 – Microwire* 0b11 – Programmable Serial Protocol

Table 16-3. SSCR0 Bit Definitions (Sheet 2 of 2)

0x4140_0000																SSCR0									Network SSP Serial Port						
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reserved											EDSS	SCR												SSE	reserved	FRF		DSS			
?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?	0	0	0	0	0	0

<table>
<tr><th>Bits</th><th>Name</th><th colspan="6">Description</th></tr>
<tr><td rowspan="18">3:0</td><td rowspan="18">DSS</td><td colspan="6">DATA SIZE SELECT:
Used in conjunction with EDSS to select the size of the data transmitted and received by the SSP. The concatenated 5-bit value of EDSS and DSS provides a data range from four to 32-bits in length.
For the Microwire* protocol, DSS and EDSS are used to determine the receive data size. The size of the transmitted data is either eight or 16-bits (determined by SSCR1[MWDS]) and the EDSS bit is ignored. The EDSS and DSS fields are ignored for Microwire* transmit data size - MWDS (alone) configures this. However, for all modes (including Microwire*) EDSS and DSS are used to determine the receive data size.
When data is programmed to be less than 32 bits, the FIFO must be programmed right-justified.</td></tr>
<tr><td>EDSS</td><td>DSS</td><td>Data Size</td><td>EDSS</td><td>DSS</td><td>Data Size</td></tr>
<tr><td>1</td><td>0b0000</td><td>17-bit data</td><td>0</td><td>0b0000</td><td>reserved, undefined</td></tr>
<tr><td>1</td><td>0b0001</td><td>18-bit data</td><td>0</td><td>0b0001</td><td>reserved, undefined</td></tr>
<tr><td>1</td><td>0b0010</td><td>19-bit data</td><td>0</td><td>0b0010</td><td>reserved, undefined</td></tr>
<tr><td>1</td><td>0b0011</td><td>20-bit data</td><td>0</td><td>0b0011</td><td>4-bit data</td></tr>
<tr><td>1</td><td>0b0100</td><td>21-bit data</td><td>0</td><td>0b0100</td><td>5-bit data</td></tr>
<tr><td>1</td><td>0b0101</td><td>22-bit data</td><td>0</td><td>0b0101</td><td>6-bit data</td></tr>
<tr><td>1</td><td>0b0110</td><td>23-bit data</td><td>0</td><td>0b0110</td><td>7-bit data</td></tr>
<tr><td>1</td><td>0b0111</td><td>24-bit data</td><td>0</td><td>0b0111</td><td>8-bit data</td></tr>
<tr><td>1</td><td>0b1000</td><td>25-bit data</td><td>0</td><td>0b1000</td><td>9-bit data</td></tr>
<tr><td>1</td><td>0b1001</td><td>26-bit data</td><td>0</td><td>0b1001</td><td>10-bit data</td></tr>
<tr><td>1</td><td>0b1010</td><td>27-bit data</td><td>0</td><td>0b1010</td><td>11-bit data</td></tr>
<tr><td>1</td><td>0b1011</td><td>28-bit data</td><td>0</td><td>0b1011</td><td>12-bit data</td></tr>
<tr><td>1</td><td>0b1100</td><td>29-bit data</td><td>0</td><td>0b1100</td><td>13-bit data</td></tr>
<tr><td>1</td><td>0b1101</td><td>30-bit data</td><td>0</td><td>0b1101</td><td>14-bit data</td></tr>
<tr><td>1</td><td>0b1110</td><td>31-bit data</td><td>0</td><td>0b1110</td><td>15-bit data</td></tr>
<tr><td>1</td><td>0b1111</td><td>32-bit data</td><td>0</td><td>0b1111</td><td>16-bit data</td></tr>
</table>

16.5.2 SSP Control Register 1 (SSCR1)

SSCR1, shown in Table 16-4**,** contains bit fields that control various SSP functions. Before enabling the port (using SSCR0[SSE]), the desired values for this register must be set.

These are read/write registers. Ignore reads from reserved bits. Write zeros to reserved bits.

Table 16-4. SSCR1 Bit Definitions (Sheet 1 of 2)

0x04140_0004 SSCR1 Network SSP Serial Port

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TTELP	TTE	EBCEI	SCFR	reserved		SCLKDIR	SFRMDIR	RWOT	reserved	TSRE	RSRE	TINTE	reserved			STRF	EFWR	RFT				TFT				MWDS	SPH	SPO	LBM	TIE	RIE
Reset	0	0	0	0	?	?	0	0	0	0	0	0	0	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description
31	TTELP	TRANSMIT HI-Z LATER PHASE: This bit modifies the behavior of TTE. It causes SSPTXD to become Hi-Z 1/2 phase (or one clock edge) later than normal. This only occurs with the TI SSP format, and the PSP format if the SSP is a slave to frame. For TI SSP format, this means the SSPTXD is Hi-Z after the rising edge after the LSB (The LSB is present a full clock). For PSP format if the SSP is a slave to frame, this means the SSPTXD is Hi-Z two clock edges after the LSB (the LSB is present a full clock). If SSPSCLK is an input, the device driving SSPSCLK must provide another clock edge. 0 – SSPTXD Hi-Z timing is as described below for TTE. 1 – SSPTXD Hi-Z timing is extended by 1/2 phase. Only valid for TI SSP, and PSP if the SSP is a slave to frame.
30	TTE	TRANSMIT HI-Z ENABLE: This bit controls whether or not SSPTXD is driven or Hi-Z when the SSP is idle. For Microwire* SSPTXD is driven at the same clock edge that the MSB is driven, and SSPTXD is Hi-Z after the next rising edge of SSPSCLK for the LSB (1 clock edge after the clock edge that starts the LSB). For SPI, SSPTXD is Hi-Z whenever SSPFRM is deasserted. For TI SSP format, SSPTXD is driven with the MSB at the first rising edge of SSPSCLK after SSPSFRM is asserted and is Hi-Z after the falling edge of SSPSCLK for the LSB (1 clock edge after the clock edge that starts the LSB). For PSP format, if the SSP is a slave to frame SSPTXD is Hi-Z on the same clock edge that starts the LSB. For PSP format if the SSP is a master to frame, SSPTXD is Hi-Z on the clock edge after the clock edge for the LSB. This occurs even if the SSP is a master of clock and this clock edge does not appear on SSPSCLK. 0 – SSPTXD line is driven when SSP is idle 1 – SSPTXD line is Hi-Z when SSP is idle
29	EBCEI	BIT COUNT ERROR INTERRUPT MASK: Disables bit count error interrupts. SSSR will still indicate an error. A bit count error occurs when the SSP is a slave to clock or frame and the SSP detects a new frame before the internal bit counter has reached 0. 0 – Bit count error events will generate an interrupt. 1 – Bit count error events will not generate an interrupt.
28	SCFR	SLAVE CLOCK FREE RUNNING: SCFR in slave mode (SCLKDIR set) must be cleared if the input clock from the external source is running continuously. In master mode (SCLKDIR cleared) this bit is ignored. Master mode only: 0 – Clock input to SSPSCLK is continuously running 1 – Clock input to SSPSCLK is active only during transfers.
27:26	—	reserved

Table 16-4. SSCR1 Bit Definitions (Sheet 2 of 2)

0x04140_0004 SSCR1 Network SSP Serial Port

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TTELP	TTE	EBCEI	SCFR	reserved		SCLKDIR	SFRMDIR	RWOT	reserved	TSRE	RSRE	TINTE	reserved			STRF	EFWR	RFT				TFT				MWDS	SPH	SPO	LBM	TIE	RIE
Reset	0	0	0	0	?	?	0	0	0	0	0	0	0	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description
25	SCLKDIR	SSP SERIAL BIT RATE CLOCK DIRECTION: Determines whether the port is the master or slave (with respect to driving SSPSCLK). 0 – Master mode, the port generates SSPSCLK internally, acts as the master, and drives SSPSCLK. 1 – Slave mode, the port acts as a slave, receives SSPSCLK from an external device and uses it to determine when to drive transmit data on SSPTXD and when to sample Receive data on SSPRXD.
24	SFRMDIR	SSP FRAME DIRECTION: Determines whether the SSP is the master or slave (with respect to driving SSPSFRM.) When SFRMDIR is set, the port acts as the slave and receives the SSPSFRM signal from an external device. When the port is configured as a slave to the frame, the external device driving frame must wait at least the equivalent of 10 SSPSCLKS after enabling the port before asserting frame. (No external clock cycles are needed, the external device just needs to wait a certain amount of time before asserting frame). **NOTE:** When the GPIO alternate function is selected for the port, this bit has precedence over the GPIO direction bit. For example, the GPIO pin is an input if SFRMDIR=1. Alternately, the GPIO pin is an output if SFRMDIR=0. 0 – Master mode, the port generates SSPSFRM internally, acts as the master and drives SSPSFRM. 1 – Slave mode, the port acts as a slave, receives SSPSFRM from an external device.
23	RWOT	RECEIVE WITH OUT TRANSMIT: Puts the SSP into a mode similar to half duplex. This allows the port to receive data without transmitting data (half-duplex only). When RWOT is set, the port continues to clock in receive data, regardless of data existing in the transmit FIFO. Data is sent/received immediately after the port enable bit (SSCR0[SSE]) is set. In this mode, if there is no data to send, the DMA service requests and interrupts for the transmit FIFO must be disabled (clear TSRE and TIE). If the transmit FIFO is empty, all zeroes are transmitted which must be discarded by the external peripheral. The transmit FIFO underrun condition does not occur when RWOT is set. When RWOT is enabled, SSSR[BUSY] remains active (set to 1) until software clears the RWOT bit. 0 – Transmit/Receive mode. 1 – Receive With Out Transmit mode.
22	—	reserved

16.5.3 SSP Programmable Serial Protocol Register (SSPSP)

SSPSPx, shown in Table 16-5, contains bit fields used to program the various programmable serial-protocol parameters. The contents of these registers are ignored if the PSP is not selected.

These are read/write registers. Ignore reads from reserved bits. Write zeros to reserved bits.

Table 16-5. SSPSP Bit Definitions (Sheet 1 of 2)

0x4140_002C																SSPSP										Network SSP Serial Port						
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved							DMYSTOP		reserved	SFRMWDTH						SFRMDLY							DMYSTRT		STRTDLY			ETDS	SFRMP	SCMODE	
Reset	?	?	?	?	?	?	?	0	0	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description
31:25	—	reserved
24:23	DMYSTOP	DUMMY STOP Determines the number of serial clock (SSPSCLK) cycles that SSPSCLK is active following the last bit (bit 0) of transmitted (SSPTXD) or received data (SSPRXD).
22:16	SFRMWDTH	SERIAL FRAME WIDTH: Determines the number of serial clock periods of the frame width (SSPSFRM active). The programmed value must not be greater than: (Start Delay)$_{max}$ + (Dummy start)$_{max}$ + (Data size)$_{max}$ + (Dummy Stop)$_{max}$. In slave mode (SSCR1[SFRMDIR] set), this field is ignored, however the incoming frame signal must be asserted for at least 1 SSPSCLK duration. In PSP mode, the incoming frame signal must be deasserted for at least 1 SSPSCLK after assertion (before the next sample is transferred).
15:9	SFRMDLY	SERIAL FRAME DELAY: Determines the number of half serial clock periods that SSPSFRM is delayed from the start of the transfer. The programed value sets the number of half SSPSCLK cycles from the time TXD/RXD starts being driven to the time SSPSFRM is asserted, from 0 to 74.
8:7	DMYSTRT	DUMMY START: Determines the number of SSPSCLK cycles after STRTDLY that precede the transmitted (SSPTXD) or received data (SSPRXD).
6:4	STRTDLY	THREE-BIT START DELAY FIELD: Determines the number of SSPSCLK cycles that SSPSCLK remains in its Idle state between data transfers. The start delay field must be programmed to 0 whenever SSPSCLK or SSPSFRM is configured as an input (SSCR1[SCLKDIR] = 1 or SSCR1[SFRMDIR] = 1).
3	ETDS	END OF TRANSFER DATA STATE: Determines the state of SSPTXD at the end of a transfer. When cleared, the state of SSPTXD is forced to 0 after the last bit (bit 0) of the frame is sent and remains 0 through the next idle period. When set, the state of SSPTXD retains the value of the last bit sent (bit 0) through the next idle period. 0 – Low 1 – Last Value <Bit 0>
2	SFRMP	SERIAL FRAME POLARITY: Determines the active state of the Serial Frame signal (SSPSFRM). In Idle mode or when the SSP is disabled, SSPSFRM is in its inactive state. In slave mode (SSCR1[SFRMDIR] set), this bit indicates the polarity of the incoming frame signal. 0 – SSPSFRM is active low. 1 – SSPSFRM is active high.

Table 16-5. SSPSP Bit Definitions (Sheet 2 of 2)

0x4140_002C SSPSP Network SSP Serial Port

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved							DMYSTOP		reserved	SFRMWDTH						SFRMDLY							DMYSTRT		STRTDLY			ETDS	SFRMP	SCMODE	
Reset	?	?	?	?	?	?	?	0	0	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description
1:0	SCMODE	SERIAL BIT-RATE CLOCK MODE: Selects one of four serial clock modes when the PSP is selected (SSCR0[FRF]=0b11). Its operation is similar to how SSCR1[SPO] and SSCR1[SPH] together determine the idle state of SSPSCLK and on which edges data is driven and sampled. 0b00 - Data Driven (Falling), Data Sampled (Rising), Idle State (Low) 0b01 - Data Driven (Rising), Data Sampled (Falling), Idle State (Low) 0b10 - Data Driven (Rising), Data Sampled (Falling), Idle State (High) 0b11- Data Driven (Falling), Data Sampled (Rising), Idle State (High)

16.5.4 SSP Time Out Register (SSTO)

The SSTO register, shown in Table 16-6,specifies the time-out value used to signal a period of inactivity within the receive FIFO.

This is a read/write registers. Ignore reads from reserved bits. Write zeros to reserved bits.

Table 16-6. SSTO Bit Definitions

0X4140_0028 SSTO Network SSP Serial Port

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved								TIMEOUT																							
Reset	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description
31:24	—	reserved
23:0	TIMEOUT	TIMEOUT: Value used to set the time-out interval. When the TIMEOUT value is cleared, no time-out occurs and SSSR[TINT] is not set. The time-out interval is given by the equation: Time-out Interval = (TIMEOUT) / Peripheral Clock Frequency

16.5.5 SSP Interrupt Test Register (SSITR)

SSITR, shown in Table 16-7 on page 16-25, contains bit fields used for testing purposes only.

Setting bits in this register causes the SSP controller to generate interrupts and DMA requests if enabled. This is useful in testing the port’s functionality.

Setting any of these bits also causes the corresponding status bit(s) to be set in the SSP Status Register (SSSR). The interrupt or service request caused by the setting of one of these bits remains active until the bit is cleared.

This is a read/write register. Ignore reads from reserved bits. Write zeros to reserved bits.

Table 16-7. SSITR Bit Definitions

0x4140_000C SSITR Network SSP Serial Port

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved																								TROR	TRFS	TTFS	reserved				
Reset	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?

Bits	Access	Name	Description
31:8	—	—	reserved
7	R/W	TROR	TEST RECEIVE FIFO OVERRUN: 0 – No receive FIFO overrun service request is generated. 1 – Generates a non-maskable Interrupt to the CPU. No DMA request is generated.
6	R/W	TRFS	TEST RECEIVE FIFO SERVICE REQUEST: 0 – No receive FIFO service request is generated. 1 – Generates a non-maskable Interrupt to the CPU and a DMA request for the receive FIFO.
5	R/W	TTFS	TEST TRANSMIT FIFO SERVICE REQUEST: 0 – No transmit FIFO service request is generated. 1 – Generates a non-maskable Interrupt to the CPU and a DMA request for the transmit FIFO.
4:0	—	—	reserved

16.5.6 SSP Status Register (SSSR)

SSSR, shown in Table 16-8 contains bit fields that signal overrun errors and the transmit and receive FIFO service requests. Each of these hardware-detected events signals an interrupt request to the interrupt controller. The status register also contains flags that indicate:

- When the SSP is actively transmitting data
- When the transmit FIFO is not full
- When the receive FIFO is not empty

One interrupt signal is sent to the interrupt controller for each SSP. These events can cause an interrupt:

- Receiver time-out,
- Receive FIFO overrun,
- Receive FIFO request
- Transmit FIFO request.

Bits that cause an interrupt signal the request as long as the bit is set. The interrupt clears when the bits clear. Read and write bits are called status bits (status bits are referred to as sticky and once set by hardware, they must be cleared by software); Read-only bits are called flags. Writing a 1 to a status bit clears it; writing a 0 has no effect. Read-only flags are set and cleared by hardware; writes have no effect. The reset state of read-write bits is zero and all bits return to their reset state when SSCR0[SSE] is cleared. Additionally, some bits that cause interrupts have corresponding mask bits in the control registers and are indicated in the section headings that follow.

Set the desired values for this register before enabling the SSP (via SSCR0[SSE]).

These are read/write registers. Ignore reads from reserved bits. Write zeros to reserved bits.

Table 16-8. SSSR Bit Definitions (Sheet 1 of 3)

0x4140_0008 SSSR Network SSP Serial Port

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved								BCE	CSS	TUR	reserved	TINT	reserved			RFL				TFL				ROR	RFS	TFS	BSY	RNE	TNF	reserved	
Reset	?	?	?	?	?	?	?	?	0	0	0	?	0	?	?	?	1	1	1	1	0	0	0	0	0	0	0	0	0	1	?	?

Bits	Name	Description
31:24	—	reserved
23	BCE	BIT COUNT ERROR: Indicates that the SSP has detected the SSPSFRM signal asserted at an incorrect time. This bit will cause an interrupt if SSCR1[BCE] is set. The SSP will ignore the current sample and the next sample in order to re-synchronize with the master. Write one to clear this bit. 0 – SSPSFRM has not been asserted out of synchronization. 1 – SSPSFRM has been asserted out of synchronization.
22	CSS	CLOCK SYNCHRONIZATION STATUS: A read-only bit that indicates the SSP is busy synchronizing the control signals. This bit is only valid when the SSP is a slave to frame. Software must wait until this bit is a 0 before allowing an external device to assert the SSPSFRM signal. 0 – The SSP is ready for slave operations. 1 – The SSP is busy synchronizing slave mode signals.
21	TUR	TRANSMIT FIFO UNDER RUN: Indicates that the transmitter tried to send data from the transmit FIFO when the transmit FIFO was empty. When set, an interrupt is generated to the CPU that cannot be locally masked by any SSP register bit. Setting TUR does not generate any DMA service request. To clear TUR, software sets it. TUR remains set until cleared by software writing a one to it which also reset its Interrupt request. Writing a zero to this bit does not affect TUR. TUR can be set only when the port is a slave to the FRAME signal (SSCR1[SFRMDIR] set) and is not set if the port is in receive-without-transmit mode (SSCR1[RWOT] set). Write one to clear this bit. 0 – Transmit FIFO has not experienced an under run 1 – Attempted read from the transmit FIFO when the FIFO was empty, request interrupt.
20	—	reserved

Table 16-8. SSSR Bit Definitions (Sheet 2 of 3)

	0x4140_0008																SSSR										Network SSP Serial Port					
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved								BCE	CSS	TUR	reserved	TINT	reserved			RFL				TFL				ROR	RFS	TFS	BSY	RNE	TNF	reserved	
Reset	?	?	?	?	?	?	?	?	0	0	0	?	0	?	?	?	1	1	1	1	0	0	0	0	0	0	0	0	0	1	?	?

Bits	Name	Description
19	TINT	RECEIVER TIME-OUT INTERRUPT: Indicates that the receive FIFO has been idle (no samples received) for the period of time defined by the value programmed within SSTO. This interrupt can be masked by SSCR1[TINTE]. Write one to clear this bit. 0 – No Receiver Time-out pending 1 – Receiver Time-out pending
18:16	—	reserved
15:12	RFL	RECEIVE FIFO LEVEL: The number of valid entries (minus 1) currently in the receive FIFO. When the value of 0xF is read, the FIFO is either empty or full and programmers must refer to RNE.
11:8	TFL	TRANSMIT FIFO LEVEL: Number of valid entries (minus 1) currently in the transmit FIFO. When the value of 0x0 is read, the FIFO is either empty or full and programmers must refer to TNF.
7	ROR	RECEIVE FIFO OVERRUN: Indicates that the Receive logic attempted to place data into the receive FIFO after it had been completely filled. When new data is received, ROR is asserted and the newly received data is discarded. This process is repeated for all new data received until at least one empty FIFO entry exists. When set, an interrupt is generated to the CPU that cannot be locally masked by any SSP register bit. The setting of ROR does not generate any DMA service request. Clearing this bit resets its interrupt request. Write one to clear this bit. 0 – Receive FIFO has not experienced an overrun 1 – Attempted data write to full receive FIFO, request Interrupt
6	RFS	RECEIVE FIFO SERVICE REQUEST: Indicates that the receive FIFO requires service to prevent an overrun. RFS is set when the number of valid entries in the receive FIFO is equal to or greater than the receive FIFO trigger threshold. It is cleared when it has fewer entries than the trigger threshold value. When RFS is set, an Interrupt is generated when SSCR1[RIE] is set. Setting RFS signals a DMA service request if SSCR1[RSRE] is set. After the CPU or DMA reads the FIFO such that it has fewer entries than the value of SSCR1[RFT], RFS (and the service request or interrupt) is automatically cleared. SSCR1[RSRE] and SSCR1[RIE] must not both be set. 0 – Receive FIFO level exceeds RFT trigger threshold or the SSP is disabled 1 – Receive FIFO level is at or above RFT trigger threshold, request Interrupt

Table 16-8. SSSR Bit Definitions (Sheet 3 of 3)

	0x4140_0008															SSSR										Network SSP Serial Port						
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved								BCE	CSS	TUR	reserved	TINT	reserved			RFL				TFL				ROR	RFS	TFS	BSY	RNE	TNF	reserved	
Reset	?	?	?	?	?	?	?	?	0	0	0	?	0	?	?	?	1	1	1	1	0	0	0	0	0	0	0	0	0	1	?	?

Bits	Name	Description
5	TFS	TRANSMIT FIFO SERVICE REQUEST: Indicates that the transmit FIFO requires service to prevent an underrun. TFS is set when the number of valid entries in the transmit FIFO is equal to or lesser than the transmit FIFO trigger threshold. It is cleared when it has fewer entries than the trigger threshold value. When TFS is set, an Interrupt is generated when SSCR1[TIE] is set. Setting TFS signals a DMA service request if SSCR1[TSRE] is set. After the CPU or DMA fills the FIFO such that it has at least as many entries as the value of SSCR1[TFT], TFS (and the service request or interrupt) is automatically cleared. SSCR1[TSRE] and SSCR1[TIE] must not both be set. 0 – Transmit FIFO level exceeds TFT trigger threshold or the SSP is disabled 1 – Transmit FIFO level is at or below TFT trigger threshold, request Interrupt
4	BSY	SSP BUSY: Indicates that the port is actively transmitting or receiving data and is cleared when the port is idle or disabled. This bit does not generate an Interrupt. Software must wait for the Tx Fifo to empty first and then wait for the BSY bit to be cleared at the end of a data transfer. 0 – SSP is idle or disabled 1 – SSP currently transmitting or receiving a frame
3	RNE	RECEIVE FIFO NOT EMPTY: Indicates that the receive FIFO contains one or more entries of valid data. It is cleared when it no longer contains any valid data. This bit does not generate an Interrupt. When using programmed I/O, this bit can be polled to remove remaining bytes of data from the receive FIFO since CPU Interrupt requests are made only when the receive FIFO trigger threshold has been met or exceeded. 0 – Receive FIFO is empty. 1 – Receive FIFO is not empty.
2	TNF	TRANSMIT FIFO NOT FULL: Indicates that the transmit FIFO contains one or more entries that do not contain valid data. TNF is cleared when the FIFO is completely full. This bit does not generate an Interrupt. When using programmed I/O, this bit can be polled to fill the transmit FIFO over its trigger threshold. 0 – Transmit FIFO is full 1 – Transmit FIFO is not full
1:0	—	reserved

16.5.7 SSP Data Register (SSDR)

SSDR, shown in Table 16-9, is a single address location that read and write data transfers access. SSDR represents two physical registers: the first is temporary storage for data on its way out through the transmit FIFO. The other register is temporary storage for data coming in through the receive FIFO.

As the system accesses the register, FIFO control logic transfers data automatically between the registers and FIFOs as fast as the system moves it. Unless attempting a write to a full transmit FIFO, data in the FIFO shifts up or down to accommodate the new word(s). Status bits show users whether the FIFO is full, above the programmable trigger threshold, below the programmable trigger threshold or empty.

For transmit data transfers, the register can be written by the system processor anytime it falls below its trigger threshold when using programmed I/O.

When a data size of less than 32-bits is selected, do not left-justify data written to the transmit FIFO. Transmit logic left-justifies the data and ignores any unused bits. Received data of less than 32-bits is automatically right-justified in the receive FIFO.

When the SSP is programmed for the Microwire* protocol and the size of the Transmit data is eight bits (SSCR1[MWDS] cleared), the most significant bits are ignored. Similarly, if the size for the Transmit data is 16 bits (SSCR1[MWDS] set), the most significant 16 bits are ignored. SSCR0[DSS] controls the Receive data size.

Both FIFOs are cleared when the port is reset, or by clearing SSCR0[SSE].

This is a read/write register. Ignore reads from reserved bits. Write zeros to reserved bits.

Table 16-9. SSDR Bit Definitions

	0x4140_0010											SSDR											Network SSP Serial Port									
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA																															
Reset	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description
31:0	DATA	TRANSMIT/RECEIVE DATA: Data word to be written to/read from Transmit/receive FIFO

16.6 Network SSP Serial Port Register Summary

Table 16-10 shows the registers associated with the NSSP and their physical addresses.

Table 16-10. NSSP Register Address Map

Physical Address	Name	Description
0x4140_0000	NSSCR0	NSSP Control register 0
0x4140_0004	NSSCR1	NSSP Control register 1
0x4140_0008	NSSSR	NSSP Status register
0x4140_000C	NSSITR	NSSP Interrupt Test register
0x4140_0010	NSSDR	NSSP Data Write Register / Data Read register
0x4140_0028	NSSTO	NSSP Time Out register
0x4140_002C	NSSPSP	NSSP Programmable Serial Protocol