SSP Serial Ports 8

This chapter describes the signal definitions and operation of the three Synchronous Serial Protocol (SSP) serial ports: SSP1, SSP2, and SSP3 included in the Intel® PXA27x Processor (PXA27x processor). All three SSP ports are mostly identical in operation, but differ as follows:

- External port connections
- Memory-map base location
- No external clock and external clock enable for SSP3

8.1 Overview

The SSP ports are a synchronous serial interfaces that connect to a variety of external analog-to-digital (A/D) converters, audio and telecommunication Codecs, and many other devices that use serial protocols for data transfer. The SSP ports provide 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 SSP ports operate as full-duplex devices for the TI Synchronous Serial Protocol, SPI, and PSP protocols and as a half-duplex device for the Microwire protocol.

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

8.2 Features

- Supports the TI Synchronous Serial Protocol, the Motorola SPI protocol, National Semiconductor Microwire, and a Programmable Serial Protocol (PSP)
- One transmit FIFO and one receive FIFO, each 16 samples deep by 32-bits wide
- Sample sizes from four to 32-bits
- Bit-rates from 6.3 Kbps (minimum) to 13 Mbps (maximum)
- Master-mode and slave-mode operation
- Receive-without-transmit operation
- Network mode with up to eight time slots and independent transmit/receive in any/all/none of the time slots—available only with TI Synchronous Serial Protocol and Programmable Serial Protocol (PSP) formats
- Audio clock control to provide a 4x output clock and support for selection of most standard audio Codec frequencies

8.3 Signal Descriptions

Table 8-2 lists the external signals between the SSP serial ports and external devices. If any port is unused, its pins are available for GPIO use. See Chapter 24, "General-Purpose I/O Controller" for details on configuring pin direction and Chapter 25, "Interrupt Controller" for interrupt capabilities.

Table 8-1. SSP Serial Port I/O Signal Descriptions (Sheet 1 of 2)

Name	Type	Description
SSPSCLK	I/O	Serial bit-clock to control the timing of a transfer. SSPSCLK is generated internally (master mode) or is supplied externally (slave mode) as indicated by SSCR1_1[SCLKDIR] as defined in Table 8-7.
SSPSYSCLK	Output	SSPSYSCLK is SSPSCLK x 4 when using the audio clock PLL. Select (SSACD_1[ACPS]) and Audio Clock Divider (SSACD_1[ACDS]) as defined in Table 8-16.
SSPSFRM	I/O	Serial frame signal 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_1[SFRMDIR] and defined in Table 8-7.
SSPTXD	Output	Transmit data (serial data out) serialized data line.†
SSPRXD	Input	Receive data (serial data in) serialized data line.†
SSPEXTCLK	Input	External clock that can be selected to replace the internal 13-MHz clock. SSPEXTCLK is used only when SSCR0_1[ECS] is set (Table 8-6) and SSCR0_1[NCS], SSCR0_1[ACS], and SSCR1_1[SCLKDIR] are cleared (Table 8-7). The maximum allowable frequency for the external clock is 13 MHz. SSPEXTCLK is multiplexed with the SSPSCLKEN alternate function (refer to Chapter 24, "General-Purpose I/O Controller").
SSPSCLKEN	Input	Asynchronous external enable for SSPSCLK. SSPSCLKEN is recognized only when SSCR0_1[ECS] is cleared and when the port is the master (SSCR1_1[SCLKDIR] is cleared). When high, SSPSCLK is enabled; when low, SSPSCLK is disabled. SSPSCLKxEN is multiplexed with the SSPEXTCLK alternate function (refer to Chapter 24, "General-Purpose I/O Controller").
SSPSCLK2	I/O	Serial bit-clock to control the timing of a transfer. SSPSCLK2 is generated internally (master mode) or is supplied externally (slave mode) as indicated by SSCR1_2[SCLKDIR] as defined in Table 8-7.
SSPSYSCLK2	Output	SSPSYSCLK2 is SSPSCLK2 x 4 when using the audio clock PLL. Select (SSACD_2[ACPS]) and Audio Clock Divider (SSACD_2[ACDS]) as defined in Table 8-16).
SSPSFRM2	I/O	Serial frame signal that indicates the beginning and the end of a serialized data word. SSPSFRM2 is generated internally (master mode) or is supplied externally (slave mode) as indicated by SSCR1_2[SFRMDIR] as defined in Table 8-7.
SSPTXD2	Output	Transmit data (serial data out) serialized data line.†
SSPRXD2	Input	Receive data (serial data in) serialized data line.†
SSPEXTCLK2	Input	External clock that can be selected to replace the internal 13-MHz clock. SSPEXTCLK2 is used only when SSCR0_2[ECS] is set (Table 8-6) and when SSCR0_2[NCS], SSCR0_2[ACS], and SSCR1_2[SCLKDIR] are cleared (Table 8-7). SSPEXTCLK2 is multiplexed with the SSPSCLK2EN alternate function (refer to Chapter 24, "General-Purpose I/O Controller").

Table 8-1. SSP Serial Port I/O Signal Descriptions (Sheet 2 of 2)

Name	Type	Description
SSPSCLK2EN	Input	Asynchronous external enable for SSPSCLK2. SSPSCLK2EN is recognized only when SSCR0_2[ECS] is cleared and when the port is the master (SSCR1_2[SCLKDIR] is cleared). When high, SSPSCLK2 is enabled; when low, SSPSCLK2 is disabled. SSPSCLK2EN is multiplexed with the SSPEXTCLK2 alternate function (refer to Chapter 24, "General-Purpose I/O Controller").
SSPSCLK3	Inout	Serial bit-clock to control the timing of a transfer. SSPSCLK3 is generated internally (master mode) or is supplied externally (slave mode) as indicated by SSPCR1_3[SCLKDIR] as defined in Table 8-7.
SSPSYSCLK3	Output	SSPSYSCLK3 is SSPSCLK3 x 4 when using the audio clock PLL. Select (SSACD_3[ACPS]) and Audio Clock Divider (SSACD_3[ACDS]) as defined in Table 8-16).
SSPSFRM3	Inout	Serial frame signal that indicates the beginning and the end of a serialized data word. SSPSFRM3 is generated internally (master mode) or is supplied externally (slave mode) as indicated by SSPCR1_3[SFRMDIR] as defined in Table 8-7.
SSPTXD3	Output	Transmit data (serial data out) serialized data line.†
SSPRXD3	Input	Receive data (serial data in) serialized data line.†
CLK_EXT	Input	The network clock (CLK_EXT) is an external clock that can replace the internal 13-MHz clock. Use CLK_EXT when SSCR0_x[NCS] is set (Table 8-6) and SSCR1_x[SCLKDIR] is cleared (Table 8-7). CLK_EXT can be used by multiple SSPs.
† Sample length is a function of the selected serial data sample size set by SSCR0_x[EDSS] and the SSCR0_x[DSS], as defined in Table 8-6.		

Note: SSP3 does not have an external clock input or an external clock-enable input.

8.4 Operation

The SSP port controller transfers serial data between the PXA27x processor and an external device through FIFOs in the SSP ports. The CPU initiates the transfers using programmed I/O (PIO), or DMA bursts are used. 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_x). DMA transfers data between memory and the SSP Data register (SSDR_x). Data written to the SSDR_x (by either the CPU or DMA) is automatically transferred to the transmit FIFO. When reading SSDR_x (by either the CPU or DMA), the "oldest" data in the receive FIFO is automatically transferred to the SSDR_x. DMA descriptor programming guidelines are located in Section 5.4.4, "Programming Tips" on page 5-16.

8.4.1 Processor and DMA FIFO Access

The CPU or DMA accesses data through the SSP port's transmit and receive FIFOs. Programmed I/O takes the form of a CPU access, transferring one FIFO entry per access. CPU accesses are normally triggered off of an SSSR_x interrupt and are always 32-bits wide. CPU right-justified writes to the FIFOs ignore bits beyond the programmed FIFO data width (SSCR0_x[EDSS] and

SSCR0_x[DSS] values); and CPU reads return zeroes in the MSBs down to the programmed data width. The FIFOs can also be accessed by DMA bursts (in multiples of one, two or four bytes) depending upon the SSCR0_x[EDSS] value. When SSCR0_x[EDSS] is set, DMA bursts must be in multiples of four bytes (the DMA must have the SSP port configured as a 32-bit peripheral).When SSCR0_x[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_x[EDSS] and SSCR0_x[DSS]. The FIFO is seen as one 32-bit location by the processor. For writes, the SSP port takes the data from the transmit FIFO, serializes it, and sends it over the serial wire (SSPTXDx) to the external device. Receive data from the external device (on SSPRXDx) is converted to parallel words and stored right-justified with zeroes packed on the left in the receive FIFO.

CPU or DMA data written to the SSP Data register (SSDR_x) is automatically transferred into the transmit FIFO. CPU or DMA reads from SSDR_x contain the "oldest" data sample that is automatically transferred from the receive FIFO. From a memory-map perspective, both reads and writes are at the same address. The FIFOs are 16 samples deep by 32 bits wide. Each FIFO location contains one SSP data sample (four to 32 bits).

When exceeded, a programmable transmit FIFO trigger threshold generates an interrupt or DMA service request that, if SSCR1_x[TIE] or SSCR1_x[TSRE] are enabled, signal the CPU or DMA, respectively, to move data to the SSDR_x. Similarly, a programmable receive FIFO trigger threshold generates an interrupt or DMA service request that, if SSCR1_x[RIE] or SSCR1_x[RSRE] are enabled, signal the CPU or DMA, respectively, to read data from SSDR_x.

Note: Do not set the value of SSCR1_x[RFT] too high for the system; otherwise, the receive FIFO can overrun because of the bus latencies caused by other internal and external peripherals. This is especially important when using interrupts and polled modes that require a longer time to service.

Note: Do not set the value of SSCR1_x[TFT] too low for the system; otherwise, the transmit FIFO can underrun because of the bus latencies caused by other internal and external peripherals. This is especially important when using interrupts and polled modes that require a longer time to service.

8.4.2 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 can be handled by either the DMA or the CPU, (see SSCR1_x[TRAIL]). Trailing bytes are identified by a time-out mechanism and the existence of data within the receive FIFO.

8.4.2.1 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_x). When a time-out occurs, the receiver time-out interrupt (SSSR_x[TINT]) is set. If the time-out interrupt is enabled (SSCR1_x[TINTE] set), a time-out interrupt signals the CPU that a time-out condition has occurred. The time-out timer is reset after a new sample enters the receive FIFO or after the CPU reads the receive FIFO. Once SSSR_x[TINT] is set, it must be cleared by writing 0b1 to it. If the time-out interrupt is enabled, clearing TINT also causes the time-out interrupt to be deasserted.

8.4.2.2 Peripheral Trailing Byte Interrupt

It is possible for the DMA to reach the end of its descriptor chain while removing trailing bytes. When this happens, the CPU is forced to take over because the DMA can no longer service the SSP port's request until a new chain is linked. When the DMA reaches the end of its descriptor chain, the SSP port does the following:

- Sets the peripheral trailing byte interrupt bit (SSSR_x[PINT]) if it is enabled when SSCR1_x[PINTE] is set
- Signals the CPU that a peripheral trailing byte interrupt condition has occurred
- Sets the SSSR_x[EOC] status bit. If more data is received after EOC is set (and EOC has not been cleared by software), SSSR_x[PINT] is set.

Once SSSR_x[PINT] is set, it must be cleared by writing 0b1 to it. Clearing SSSR_x[PINT] deasserts the peripheral trailing byte interrupt.

Always handle the possibility of trailing bytes correctly. Remove the remaining bytes using CPU I/O as described in the processor-based method below or by reprogramming a new descriptor chain and restarting the DMA. See Chapter 5, "DMA Controller"for details of descriptor programming and "end-of-chain" events.

8.4.2.3 Removing Trailing Bytes

Processor Based (SSCR1_x[TRAIL] cleared):

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

Note: The time-out interrupt must be enabled by setting SSCR1_x[TINTE].

DMA Based (SSCR1_x[TRAIL] set):

When the DMA is required to handle trailing bytes, a DMA service request is automatically issued for the remaining number of samples left in the receive buffer. The DMA empties the contents of the receive buffer unless the DMA reaches the end of its descriptor chain (refer to Section 8.4.2.2). If a time-out occurs, the processor is interrupted by a time-out interrupt. Enable the time-out interrupt by setting SSCR1_x[TINTE]. When handling trailing bytes with DMA, if a time-out occurs and the receive FIFO is empty, an EOR is sent to the DMA controller.

Note: If an EOC occurs at the time that the last sample is read from the Receive Data register (the DMA descriptor chain was exactly long enough, but the time-out counter is still running—a time-out has not occurred and the SSTO register is not zero), then when the time-out does occur, the SSP generates a DMA request, which causes a RAS interrupt from the DMA controller. When the RAS interrupt occurs, software must reprogram the DMA registers and re-enable the channel for the SSP to send its EOR to the DMA controller.

8.4.3 Frame Counter

The SSP sends a start-of-frame signal to the OS timer to increment a frame counter (see Chapter 22, “Operating System Timers” for details). In normal mode (SSCR0_x[MOD] = 0), the start-of-frame signal is asserted on every sample that is received; in network mode (SSCR0_x[MOD] = 1), the start-of-frame signal is asserted once every time that SSPSFRMx is asserted.

8.4.4 Data Formats

Four pins transfer data between the PXA27x processor and external Codecs, modems, or other compatible serial devices. Although four serial-data formats exist, each has the same basic structure and in all cases the pins are used as follows:

- SSPSCLKx—Defines the bit rate at which serial data is driven into, and sampled from, the SSP port.
- SSPSFRMx—Defines the boundaries of a basic data unit, comprised of multiple serial bits.
- SSPTXDx—The serial data path for transmitted data from the processor to the peripheral.
- SSPRXDx—The serial data path for received data from the peripheral to the processor.

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: Texas Instruments (TI) Synchronous Serial Protocol, Motorola Serial Peripheral Interface (SPI) protocol, National Semiconductor Microwire, and a Programmable Serial Protocol (PSP).

The SSPSFRMx function and use varies between each protocol.

- For the TI Synchronous Serial Protocol, SSPSFRMx 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 Motorola SPI protocol, SSPSFRMx 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 SSPSFRMx signal is held low. Master and slave modes are supported. This is a full-duplex protocol.
- For the National Semiconductor Microwire protocol, SSPSFRMx 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. SSPSFRMx is also held low during continuous transfers. This is a half-duplex protocol.
- For the PSP, SSPSFRMx 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 function and use of SSPSCLKx varies between each protocol.

- For the TI Synchronous Serial Protocol, data sources switch transmit data on the rising edge of SSPSCLKx and sample receive data on the falling edge. Master and slave modes are supported. When SSPSCLKx is provided by the SSP port, it only toggles during active transfers (not continuously) unless the SSPSCLKENx, or SSCR1_x[ECRA] or SSCR1_x[ECRB] bits, are used. When SSPSCLKx is provided by another device, it can be either continuous or driven only during active transfers.
- For the SPI protocol, programmers select which edge of SSPSCLKx to use for switching transmit data and for sampling receive data. In addition, users can move the phase of

SSPSCLKx, shifting its active state one-half cycle earlier or later at the start and end of a frame. Master and slave modes are supported, and SSPSCLKx only toggles during active transfers (not continuously).

- For Microwire, both data sources switch transmit data (change to the next bit) on the falling edge of SSPSCLKx and sample receive data on the rising edge. Slave mode is not supported.
- For the PSP protocol, the configuration of which edge of SSPSCLKx is used for switching transmit data and the edge for sampling receive data is programmable. In addition, the idle state for SSPSCLKx can be programmed and the number of active clocks that precede and follow the data transmission can be programmed. Master and slave modes are supported. When SSPSCLKx is provided by the SSP port, it only toggles during active transfers (not continuously) unless the SSPSCLKENx, or SSCR1_x[ECRA] or SSCR1_x[ECRB] bits, are used. When SSPSCLKx is provided by another device, it can be either continuous or only driven during active transfers, but some restrictions apply (see Section 8.4.4.4).

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. SSPSCLKx is active during the entire frame.

Note: SSPSCLKx, if provided by the SSP port, toggles only while an active data transfer is underway, unless receive-without-transmit mode is enabled (by setting SSCR1_x[RWOT]) and the frame format is not Microwire. If RWOT mode is enabled and the Frame format is not Microwire, SSPSCLKx toggles regardless of whether transmit data exists within the transmit FIFO. SSPSCLKx toggles continuously if the SSP port is in network mode, or if <SSPSCLKxEN> is active, or the SSCR1_x[ECRA] or SSCR1_x[ECRB] bits, are enabled. At other times, <SSPSCLKx> holds in an inactive (idle) state as defined by the protocol.

8.4.4.1 TI Synchronous Serial Protocol

When outgoing data in the SSP port controller is ready to transmit, SSPSFRMx asserts for one clock period. On the following clock, data to be transmitted is driven on SSPTXDx one bit at a time, the most significant bit first. For receive data, the peripheral similarly drives data on the SSPRXDx pin. The word length can be from four to 32-bits. All output transitions occur on the rising edge of SSPSCLKx while data sampling occurs on the falling edge. At the end of the transfer, the SSPTXDx signal either retains the value of the last bit sent (LSB) (see Figure 8-1 and Figure 8-2) or becomes high impedance (see Figure 8-13 and Figure 8-14).

Figure 8-1 shows the TI Synchronous Serial Protocol for a single transmitted frame. Figure 8-2 shows the TI Synchronous Serial Protocol for when back-to-back frames are transmitted. Once the transmit FIFO contains data, SSPSFRMx is pulsed high for one SSPSCLKx 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 SSPSCLKx, the most significant bit of the four to 32-bit data frame is shifted to the SSPTXDx pin. Likewise, the MSB of the received data is shifted onto the SSPRXDx pin by the off-chip serial slave device. Both the SSP port and the off-chip serial slave device then latch each data bit into the serial shifter on the falling edge of each SSPSCLKx. The received data is transferred from the serial shifter to the receive FIFO on the first rising edge of SSPSCLKx 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 port is a master to the SSPSFRMx and a slave to SSPSCLKx, 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).

Note: When configured as either master or slave to SSPSCLKx or SSPSFRMx, the SSP port continues to drive SSPTXDx until the last bit of data is sent (the LSB) or the SSPTXDx line becomes high impedance (see Figure 8-13 and Figure 8-14). If SSCR0_x[SSE] is cleared, the SSPTXDx line goes low. SSPSP_x[EDTS] has no effect when in SSP mode. SSPRXDx is undefined before the MSB is sent and after the LSB is sent. SSPRXDx must not float.

Figure 8-1. Texas Instruments Synchronous Serial Frame Protocol (Single Transfers)

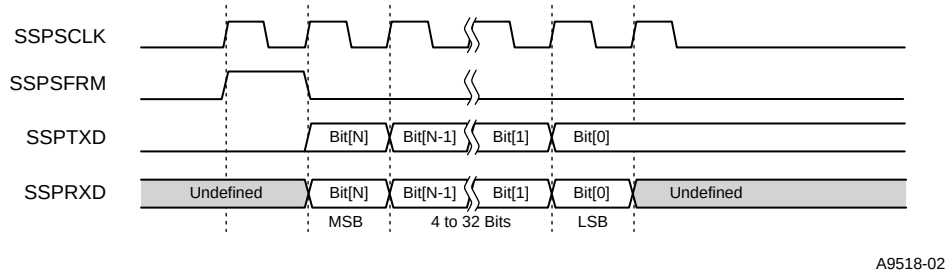
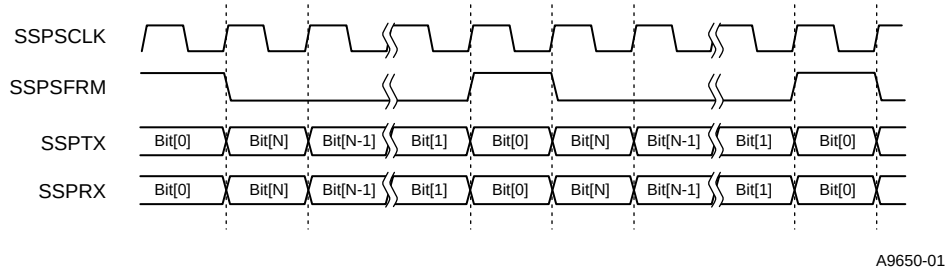
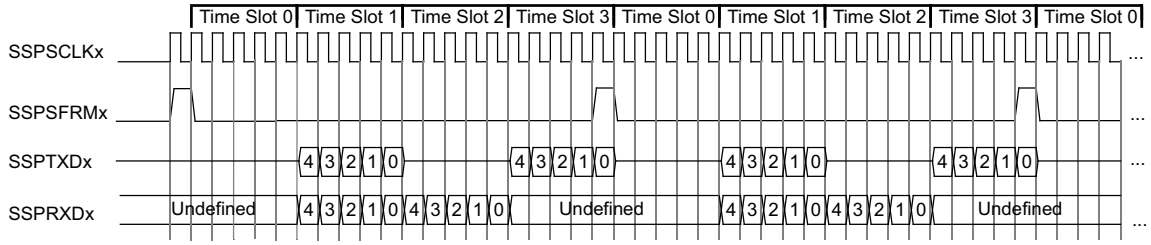


Figure 8-2. Texas Instruments Synchronous Serial Frame Protocol (Multiple Transfers)



The TI SSP supports network mode as described in Table 8-6. Figure 8-3 shows the SSP in network mode with five bits of data per sample. The TXD Tristate enable (SSCR1_x[TTE]) bit and the TXD Tristate Enable on Last Phase (SSCR1_x[TTELP] bit are both set. The SSP is the master of both SSPSCLKx and SSPSFRMx. The SSP has been programmed for four time slots (SSCR0_x[FRDC] = 0b011), the TX Time Slot Active register (SSTSA_x[TTSA]) is programmed to 0x0000 000A, and the RX Time Slot Active register (SSRSA_x[RTSA]) is programmed to 0x0000 0006.

Figure 8-3. TI SSP Network Mode Example (Four Time Slots)



8.4.4.2 Motorola SPI Protocol

The SPI protocol has four possible sub-modes, depending on the SSPSCLKx edges selected for driving data and sampling received data, and on the selection of the phase mode of SSPSCLKx (see Section 8.4.4.2.1 for complete descriptions of each mode).

Note: The following description only applies when SPH = 0 and SPO = 0. Other combinations of SPH and SPO result in different polarities and timings (see Figure 8-5 and Figure 8-6).

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

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

For back-to-back transfers, frames start and complete similar to single transfers, except SSPSFRMx does not deassert 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 port can either be a master or a slave device. However, the clock and frame direction must be the same. For example, the SSCR1_x[SCLKDIR] and SSCR0_x[SFRMDIR] must both be set or both be cleared.

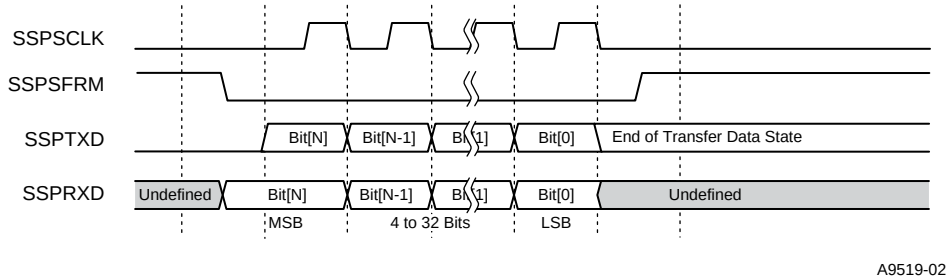
When configured as either master or slave and SSPSP_x[ETDS] is set, the SSP port continues to drive SSPTXDx with the last bit of data sent (the LSB). If SSPSP_x[ETDS] is cleared, SSPTXDx goes low after transmitting the last data bit. The state of SSPRXDx is undefined before the MSB and after the LSB is received. SSPRXDx must not float. When the SSP port is configured as a master and SSCR1_x[TTE] is set, SSPSP_x[ETDS] is ignored and SSPTXDx becomes high impedance between active transfers (see Figure 8-15).

Note: The input clock to the SSP port must not be active when SSPSFRMx is deasserted.

Note: When the SSP port is slave to clock and frame, SSCR1_x[SCFR] must be set.

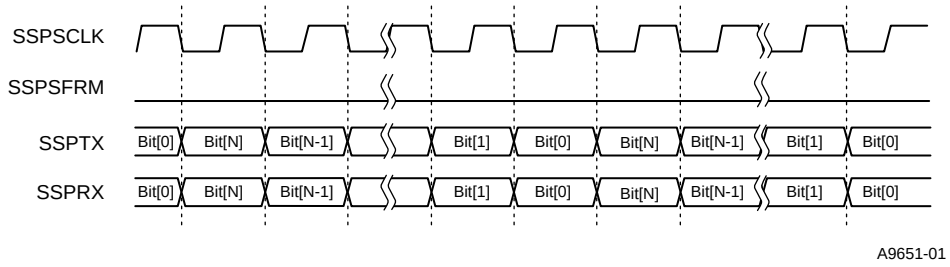
Figure 8-4 shows one of the four possible configurations for the Motorola SPI frame protocol for a single transmitted frame. Figure 8-5 shows when back-to-back frames are transmitted for the Motorola SPI frame protocol.

Figure 8-4. Motorola SPI Frame Protocol (Single Transfers)



Note: The phase and polarity of SSPSCLKx can be configured for four different modes. This example shows just one of those modes (SSCR1_x[SPO] and SSCR1_x[SPH] cleared). Other settings for SPO and SPH result in different polarities and timing.

Figure 8-5. Motorola SPI Frame Protocol (Multiple Transfers)



8.4.4.2.1 Serial Clock Phase (SPH)

The phase relationship between SSPSCLKx and SSPSFRMx when the Motorola SPI protocol is selected is controlled by SSCR1_x[SPH].

The combination of the SSCR1_x[SPO] and SSCR1_x[SPH] settings determine when SSPSCLKx is active during the assertion of SSPSFRMx and which SSPSCLKx edge transmits and receives data on SSPTXDx and SSPRXDx.

When SPH is cleared, SSPSCLKx remains in its inactive (idle) state (as determined by SSCR1_x[SPO]) for one full cycle after SSPSFRMx is asserted low at the beginning of a frame. SSPSCLKx continues to toggle for the rest of the frame. It is then held in its inactive state for one-half of an SSPSCLKx period before SSPSFRMx is deasserted high at the end of the frame.

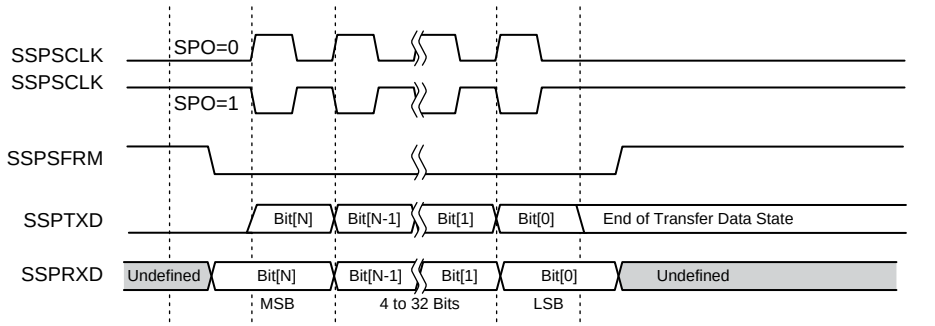
When SPH is set, SSPSCLKx remains in its inactive or idle state (as determined by SSCR1_x[SPO]) for one-half cycle after SSPSFRMx is asserted low at the beginning of a frame. SSPSCLKx continues to toggle for the remainder of the frame and is then held in its inactive state for one full SSPSCLKx period before SSPSFRMx is deasserted high at the end of the frame.

When programming SSCR1_x[SPO] and SSCR1_x[SPH] to the same value (both set or both cleared), transmit data is driven on the falling edge of SSPSCLKx and receive data is latched on the rising edge of SSPSCLKx. When programming SSCR1_x[SPO] and SSCR1_x[SPH] to opposite values (one set and the other cleared), transmit data is driven on the rising edge of SSPSCLKx and receive data is latched on the falling edge of SSPSCLKx.

Note: SSCR1_x[SPH] is ignored for all data frame formats except for the Motorola SPI protocol.

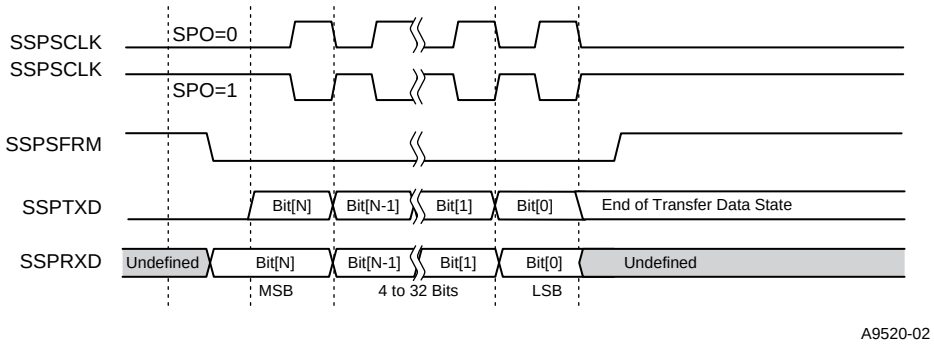
Figure 8-6 and Figure 8-7 show the timing for all four programming combinations of SSCR1_x[SPO] and SSCR1_x[SPH]. SSCR1_x[SPO] inverts the polarity of SSPSCLKx and SSCR1_x[SPH] determines the phase relationship between SSPSCLKx and SSPSFRMx, shifting SSPSCLKx one-half phase to the left or right during the assertion of SSPSFRMx.

Figure 8-6. Motorola SPI Frame Protocols for SPO and SPH Programming (SPH Set)



Note: When in Motorola SPI format, if the SSP port is the master and SSPSP_x[ETDS] is cleared, the End of Transfer Data State for SSPTXDx is low. If the SSP port is the master and SSPSP_x[ETDS] is set, the End of Transfer Data State for SSPTXDx remains at the last bit transmitted (LSB). If the SSP port is the slave, then SSPSP_x[ETDS] is undefined. SSPRXDx is undefined before the frame is active and after the LSB is received. SSPRXDx must not float. When the SSP port is configured as a master and SSCR1_x[TTE] is set, SSPSP_x[ETDS] is ignored and SSPTXDx becomes high impedance between active transfers (see Figure 8-15).

Figure 8-7. Motorola SPI Frame Protocols for SPO and SPH Programming (SPH Cleared)



Note: When in Motorola SPI format, if the SSP port is the master and SSPSP_x[ETDS] is cleared, the End of Transfer Data State for SSPTXDx is low. If the SSP port is the master and SSPSP_x[ETDS] is set, the End of Transfer Data State for SSPTXDx remains at the last bit transmitted (LSB). If the SSP port is the slave, then the SSPSP_x[ETDS] is undefined. SSPRXDx is undefined before the frame is active and after the LSB is received. SSPRXD must not float. When the SSP port is configured as a master and SSCR1_x[TTE] is set, SSPSP_x[ETDS] is ignored and SSPTXDx becomes high impedance between active transfers (see Figure 8-15).

8.4.4.3 Microwire Protocol

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 port is disabled, SSPSCLKx and SSPTXDx are low and SSPSFRMx is high.

Each serial transmission begins with SSPSFRMx asserting low, followed by an eight or 16-bit command word sent from the controller to the peripheral on SSPTXDx. The command word data size is selected by the Microwire transmit SSCR1_x[MWDS] data size bit. SSPRXDx is controlled by the peripheral and remains in a high-impedance state. SSPSCLKx asserts high (rising edge) midway into the command's most significant bit and continues toggling at the configured SSPSCLKx bit rate.

One bit period after the last command bit, the peripheral returns the serial-data requested most significant bit first on SSPRXDx. Data transitions on the falling edge of SSPSCLKx and is sampled on the rising edge. The last falling edge of SSPSCLKx coincides with the end of the last data bit on SSPRXDx and SSPSCLKx remains low after that (if it is the only word or the last word of the transfer). SSPSFRMx deasserts 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, SSPSFRMx remains asserted (low) throughout the transfer. The end of a data word on SSPRXDx is followed immediately by the start of the next command byte on SSPTXDx with no dead time. If SSCR1_x[TTE] is set (selecting SSPTXDx to be high impedance between active transfers), then a three-wire Microwire mode can be supported where SSPTXDx and SSPRXDx are externally tied together (shorted).

When using the Microwire protocol, the SSP port can function only as a master (frame and clock are outputs). Therefore, both SSCR1_x[SCLKDIR] and SSCR0_x[SFRMDIR] must both be cleared. Figure 8-8 shows the National Semiconductor Microwire frame protocol with 8- or 16-bit

command words for a single transmitted frame. Figure 8-9 shows the National Semiconductor Microwire frame protocol with eight-bit command words when back-to-back frames are transmitted.

Figure 8-8. National Semiconductor Microwire Frame Protocol (Single Transfer)

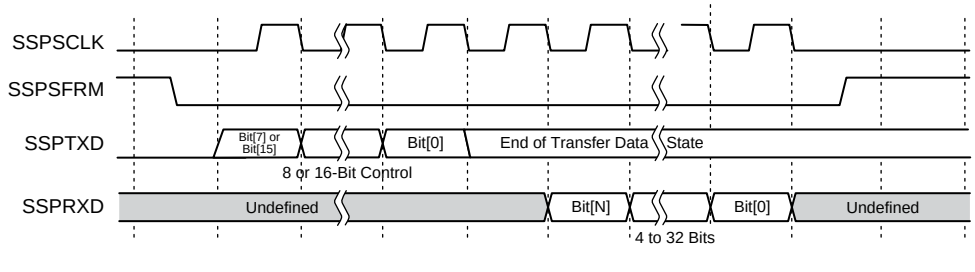
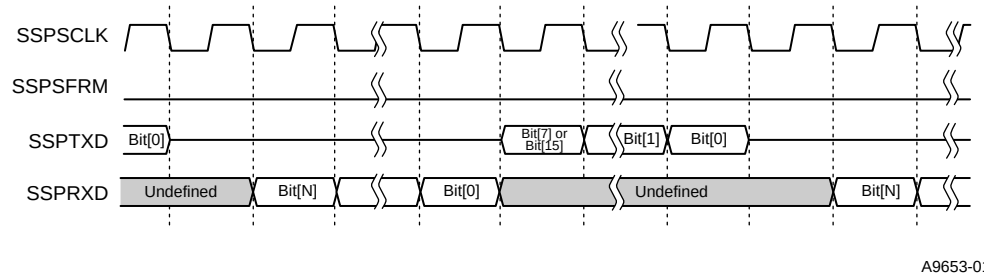


Figure 8-9. National Semiconductor Microwire Frame Protocol (Multiple Transfers)



Note: Microwire format is normally used with SSPSP_x[ETDS] cleared so that the End of Transfer Data State for SSPTXDx is low. If SSPSP_x[ETDS] is set then SSPTXDx remains at the level of the last transmitted bit (LSB). When SSCR1_x[TTE] is set, SSPSPx[ETDS] is ignored and SSPTXDx becomes high impedance between active transfers (see Figure 8-16).

Note: The state of SSPRXD is undefined before the MSB and after the LSB is transmitted. This pin must not float.

8.4.4.4 Programmable Serial Protocol (PSP)

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

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

For the PSP, the idle and disable modes of SSPTXDx, SSPSCLKx, and SSPSFRMx are programmable using SSPSP_x[ETDS], SSPSP_x[SCMODE] and SSPSP_x[SFRMP]. When transmit data is ready, SSPSCLKx remains in its idle state for the number of SSPSCLKx periods programmed within the SSPSP_x[STRTDLY] start delay field. SSPSCLKx then starts toggling,

SSPTXDx remains in the idle state for the number of cycles programmed within the SSPSP_x[DMYSTRT] dummy start field. SSPSFRMx asserts after the number of clocks programmed in the SSPSP_x[SFRMDLY] frame delay field. SSPSFRMx remains asserted for the number of half-clocks programmed within the SSPSP_x[SFRMWDTH] frame width field. Four to 32-bits can be transferred per frame. Once the LSB transfers, SSPSCLKx continues toggling based on the SSPSP_x[DMYSTOP] dummy stop field. SSPTXDx either retains the last value transmitted or is forced to zero, depending on the value programmed within the SSPSP_x[ETDS] end of transfer data state when the controller goes into idle mode, unless the SSP port is disabled or reset (which forces SSPTXDx low). When SSCR1_x[TTE] is set, SSPSP_x[ETDS] is ignored and SSPTXDx becomes high impedance between active transfers (see Figure 8-17 and Figure 8-18).

With the assertion of SSPSFRMx, receive data is simultaneously driven from the peripheral on SSPRXDx, MSB first. Data transitions on SSPSCLKx are based on the serial clock mode selected and are sampled by the controller on the opposite edge. When the SSP port is a master to SSPSFRMx and a slave to SSPSCLKx, 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).

Note: When in PSP mode, if the SSP port is the master of the clock and SSPSP_x[ETDS] is cleared, the End of Transfer Data State for SSPTXD is low. If the SSP port is the master of the clock and SSPSP_x[ETDS] is set, the End of Transfer Data State for SSPTXD remains at the last bit transmitted (LSB). When SSCR1_x[TTE] is set, SSPSP_x[ETDS] is ignored and SSPTXDx becomes high impedance between active transfers (see Figure 8-17 and Figure 8-18).

If the SSP port is a slave to the clock and SSPSP_x[SCMODE] is 0b01 or 0b11, then SSPSP_x[ETDS] can only change from the LSB if more clocks are sent to the SSP port (dummy stop clocks or slave clock is free running).

Figure 8-10. Programmable Serial Protocol (Single Transfer)

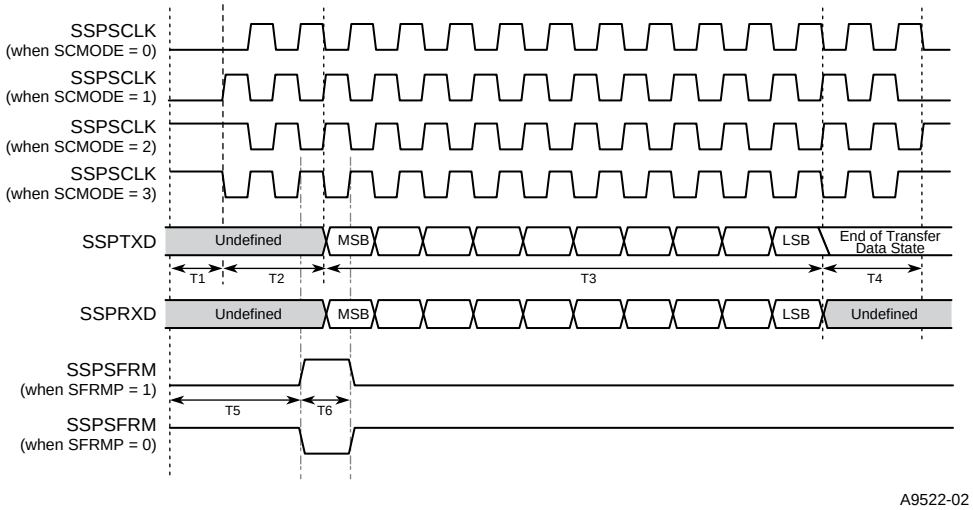


Figure 8-11. Programmable Serial Protocol (Multiple Transfers)

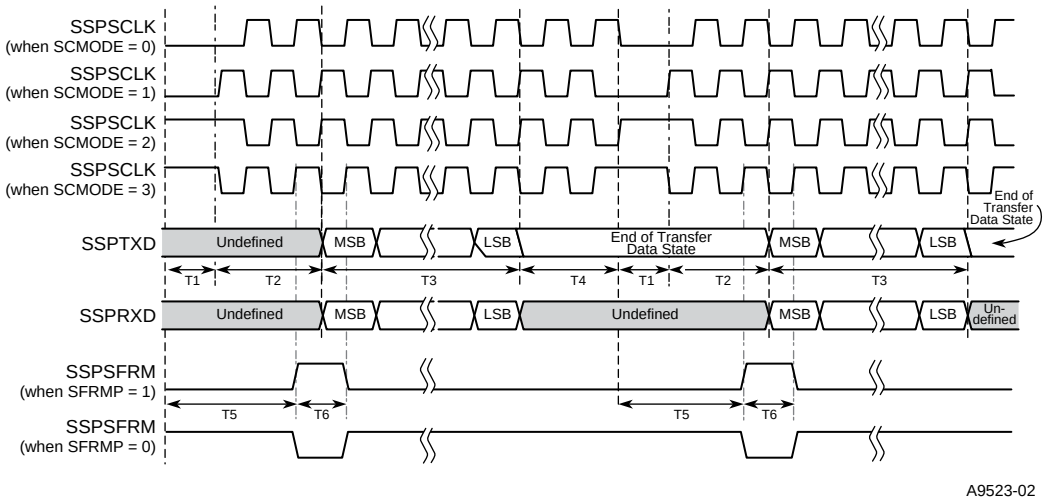
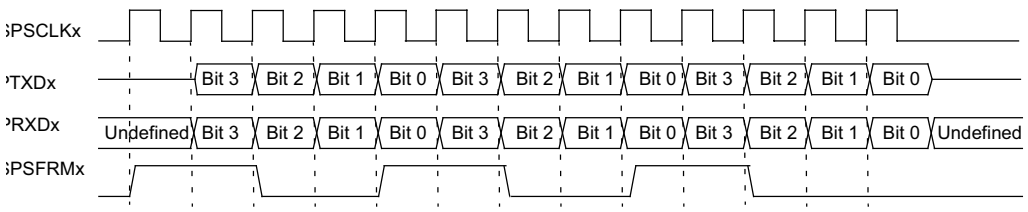


Figure 8-12. Programmable Serial Protocol Format (with Consecutive Transfers and SSPSP[FSRT] Set)



NOTE: This example uses a Frame width of 2, dummy start = 0, dummy stop = 0, start delay = 0, frame delay = 0, scmode = 1, Txd Tristate = 1, FSRT = 1, and frame polarity = 1.

Table 8-2. Programmable Serial Protocol (PSP) Parameters (Sheet 1 of 2)

Symbol	Definition	Range	Units
—	Serial Clock Mode (SSPSP_x[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_x[SFRMP])	High or Low	—
T1	Start Delay (SSPSP_x[STRTDLY])	0–7	Clock period
T2	Dummy Start (SSPSP_x[DMYSTRT])	0–3	Clock period
T3	Data Size (SSCR0_x[EDSS] and SSCR0_x[DSS])	4–32	Clock period

Table 8-2. Programmable Serial Protocol (PSP) Parameters (Sheet 2 of 2)

Symbol	Definition	Range	Units
T4	Dummy Stop (SSPSP_x[DMYSTOP])	0–3	Clock period
T5	Frame Delay (SSPSP_x[SFRMDLY]	0–88	Half-clock period
T6	Frame Width (SSPSP_x[SFRMWDTH]	1–44	Clock period
	End of Transfer Data State (SSPSP_x[ETDS])	Low or [bit 0]	—

Note: SSPSFRMx delay must not extend beyond the end of T4. SSPSFRMx must be asserted for at least one SSPSCLKx, and must be deasserted before the end of the T4 cycle (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 SSPSFRMx is asserted for at least 2 edges of the SSPSCLKx). Additionally, T1+T2 ≥ T5. The T1 start delay value must be programmed to zero when SSPSCLKx is enabled by SSPCLKxEN or either of SSCR1_x[ECRA] or SSCR1_x[ECRB]. While the SSP can be programmed to generate the assertion of SSPSFRMx during the middle of the data transfer (after the MSB was sent), the SSP port is not able to receive data in frame slave mode (SSCR1_x[SFRMDIR] is set) if the assertion of frame is not before the MSB is sent (for example, T5 ≤ T2 if SSCR1_x[SFRMDIR] is set). Transmit data transitions from the “End of Transfer Data State” to the next MSB value upon the assertion of SSPSFRMx. The start delay field must be programmed to zero whenever SSPSCLKx or SSPSFRMx is configured as an input.

8.4.5 High Impedance on SSPTXDx

The SSP port supports placing SSPTXDx into high impedance during idle times (instead of driving SSPTXDx) as controlled by SSCR1_x[TTE] and SSCR1_x[TTELP]. The TTE bit enables high impedance on SSPTXDx. The TTELP bit determines which SSPSCLKx phase that SSPTXDx becomes high impedance.

8.4.5.1 TI SSP Format

When SSCR1_x[TTE] is clear, the SSP port behaves as described in Section 8.4.4.1.

When SSCR1_x[TTE] is set, SSPTXDx behaves as shown in Figure 8-13 or Figure 8-14, depending on SSCR1_x[TTELP].

Figure 8-13. TI SSP with SSCR1_x[TTE] = 1 and SSCR1_x[TTELP]] = 0

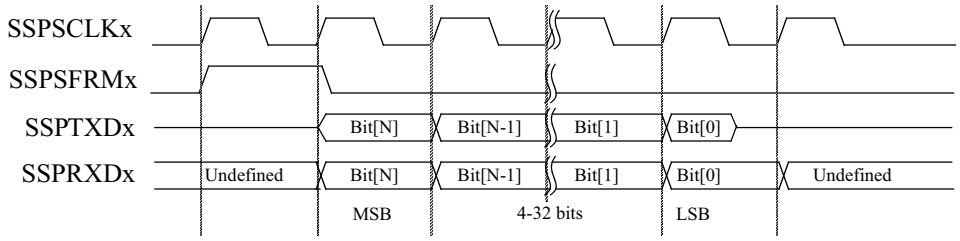
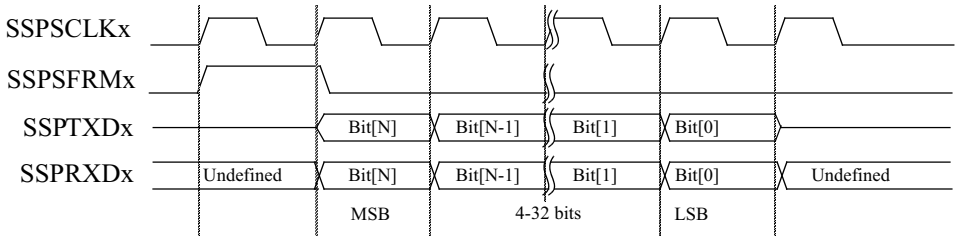


Figure 8-14. TI SSP with SSCR1_x[TTE] = 1 and SSCR1_x[TTELP] = 1

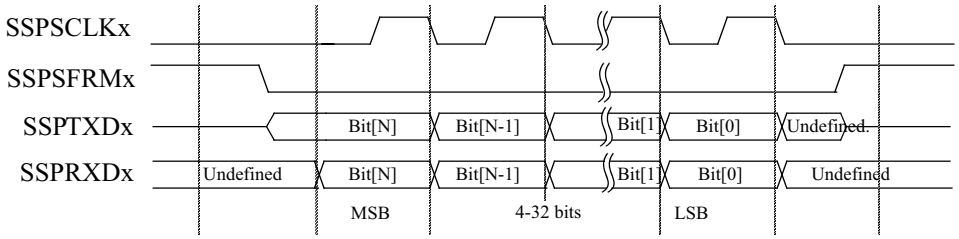


8.4.5.2 Motorola SPI Format

When SSCR1_x[TTE] is cleared, the SSP port behaves as described in Section 8.4.4.2.

When SSCR1_x[TTE] is set, SSPTXDx behaves as in Figure 8-15. SSCR1_x[TTELP] must be cleared for Motorola SPI format, and SSPTXDx becomes high impedance whenever SSPSFRMx is not active (high).

Figure 8-15. Motorola SPI with SSCR1_x[TTE] = 1 and SSCR1_x[TTELP] = 0

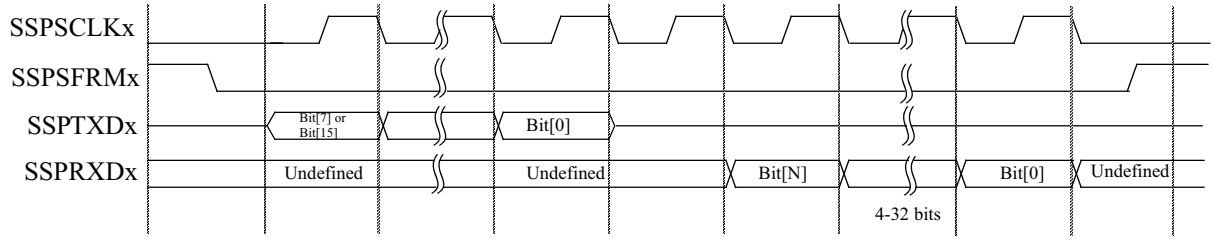


8.4.5.3 National Microwire Format

When SSCR1_x[TTE] is cleared, the SSP port behaves as described in Section 8.4.4.3.

When SSCR1_x[TTE] is set, SSPTXDx behaves as in Figure 8-16. SSCR1_x[TTELP] must be cleared for National Microwire format, and SSPTXDx becomes high impedance whenever SSPSFRMx is inactive (and after the LSB is sent on SSPTXDx).

Figure 8-16. National Microwire with SSCR1_x[TTE] = 1 and SSCR1_x[TTELP] = 0



8.4.5.4 PSP Format

When SSCR1_x[TTE] is cleared, the SSP port behaves as described in Section 8.4.4.4.

When SSCR1_x[TTE] is set, SSPTXDx behaves as shown in Figure 8-17 or Figure 8-18, depending on SSCR1_x[TTELP].

Figure 8-17. PSP Format with SSCR1_x[TTE] = 1 and SSCR1_x[TTELP] = 0

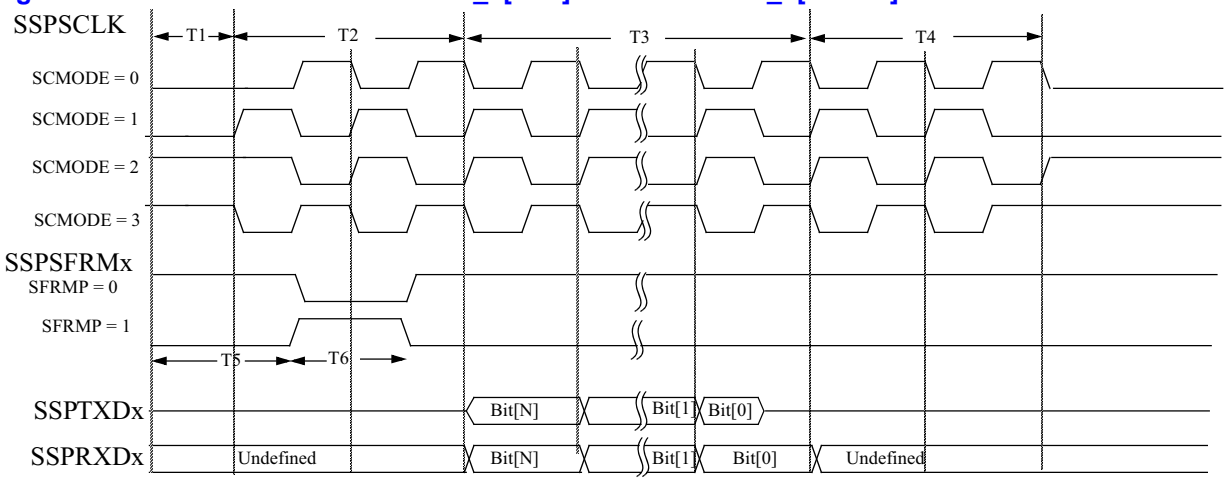
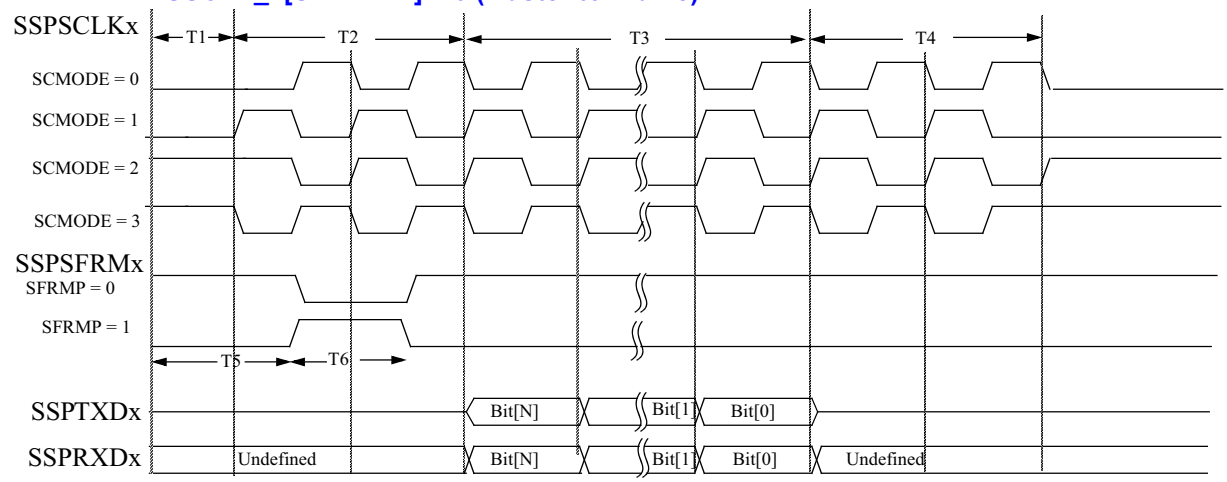


Figure 8-18. PSP Format with SSCR1_x[TTE] = 1 and Either SSCR1_x[TTELP] = 1 or SSCR1_x[SFRMDIR] = 0 (Master to Frame)



8.4.6 Network Mode

Network mode (see Figure 8-3) is typically used in systems where several devices are connected to the same SSPTXDx, SSPRXDx, SSPSFRMx, and SSPSCLKx lines. In network mode, from 1–8 time slots can be chosen. The SSP port can transmit or receive in any of the slots (see Table 8-13 and Table 8-14). Only TI SSP and PSP formats support network mode, which is enabled by setting SSCR0_x[MOD] (see Table 8-6).

In network mode, SSPSCLKx runs continuously if the SSP port is a master of the clock (SSCR1_x[SCLKDIR] = 0). Only one SSPSFRMx is sent (received) for the number of time slots programmed into the SSCR0_x[FRDC] field.

When beginning in network mode (and the SSP port is a master to the frame signal), the first SSPSFRMx does not occur until after data is in the transmit FIFO. After the first SSPSFRMx, if the SSP port is a master to the frame signal, frame syncs continue to come regardless of whether there is data in the transmit FIFO or not. Therefore, the transmit underrun (SSSR_x[TUR]) bit is set if there is no data in the transmit FIFO and the SSP port is scheduled to drive SSPTXDx in the current time slot (even if the SSP port is master to frame).

When shutting down from network mode (and the SSP port is a master to frame), software must clear the MOD bit (SSCR0_x[SSE] does not need to change), then wait until SSTSS_x[NMBSY] is cleared before shutting down the SSP port (clearing SSE). The SSP port continues driving clocks and/or frame (depending upon SSCR1_x[SFMRDIR] and SSCR1_x[SCLKDIR]) until the last valid time slot is over. If the SSP port is a slave to both clock and frame, NMBSY remains asserted until the MOD bit is cleared or until one SSPSCLKx after the last valid time slot is over. When the SSP port is a master to frame, software must ensure that no data remains in the transmit FIFO after the network mode is exited (or else a non-network mode frame is sent). Due to the synchronization between the processor's clock domains, one extra frame may be sent out after the MOD bit is cleared by software.

Note: The SSP port transmits data in inactive time slots in network mode. When FRDC is set to > 0 (> 1 slots per frame) and, in SSTSA and SSRSA registers, only slot 0 is enabled and the rest of the slots are disabled, then the SSP repeats the slot 0 data on the rest of the time slots during data transmit. If SSCR1_x[TTELP] and SSCR1_x[TTE] are clear, the SSP continues to ship the last data on disabled slots. For example, if FRDC is set to 3 (four slots per frame), only slot 0 is enabled, and TTELP and TTE are clear, the SSP repeats the slot 0 data on time slot 1, 2, and 3. If TTELP and TTE are set, the SSP three-states the data lines during the disabled slots.

8.4.7 Parallel Data Formats for FIFO Storage

The CPU or the DMA transfers one FIFO entry per access. Data in the FIFOs are stored with one 32-bit value per data sample, regardless of the format data word length. Within each 32-bit field, the stored data sample is right-justified, with the LSB of the word in bit 0. In the receive FIFO, unused bits are packed as zeroes above the MSB of the data sample. In the transmit FIFO, unused bits are above the MSB of the data sample. DMA and CPU access do not have to write to the unused bit locations. Logic in the SSP port automatically formats data in the transmit FIFO so that the sample is properly transmitted on SSPTXDx in the selected frame format.

8.4.8 Continuous Serial Clock Operation

In addition to the normal operation, SSPSCLKx can be configured to run continuously even if disabled by clearing SSCR0_x[SSE]. This allows an SSP port (when it has data to transmit) to enable the serial clock of a different SSP port or an external device to enable the SSP port source clocks.

This capability supports applications where the SSP ports interface with multiple peripherals that share a common synchronized bit-clock (for instance, a cellular network clock) that is generated by one SSP port but used by other SSP ports. For example, SSP ports can interface with a Bluetooth baseband processor (to carry voice data), a cellular baseband processor, and an audio Codec—all synchronized to a network clock. This clock-gating logic ensures that the common clock is running and consuming power only when an SSP port has data to send or an external device requests the clock using SSPSCLKENx.

When the SSP controller sends a clock request to the serial clock source SSP port, the serial clock of the serial clock source SSP port (refer to Table 8-3) is driven depending on SSCR1_x[ECRA] and SSCR1_x[ECRB]. The clock request is a signal internal to the SSP port and is asserted when an SSP port has data to send (for example, the transmit FIFO, transmit FIFO holding register, and shift register are not all empty.) This condition is denoted as the tx_not_empty signal in Figure 8-19.

Note: In the following text, the term SSP_a refers to any of the three SSP ports. The term SSP_b refers to any of the three SSP ports except for those included in SSP_a. For example, if SSP_a includes SSP1, SSP_b cannot include SSP1 but can include SSP2 and SSP3.

The SSPSCLKa of SSP_a continues to run (at the baud rate SSP_a is programmed to) if at least one of the following conditions are true:

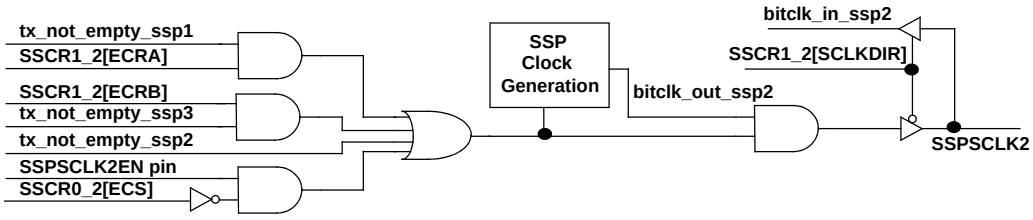
- SSP_b clock request is active (indicated by tx_not_empty in Figure 8-19) and SSCR1_x[ECRA] and SSCR1_x[ECRB] are set so that a clock request from SSP_b into SSP_a is enabled in SSP_a (refer to Table 8-3).
- SSP_a is enabled and in the middle of a transaction (normal operation).
- SSPCLKaEN of SSP_a is high and SSCR0_x[ECS] of SSP_a is cleared.

Table 8-3 shows the clock request enable selections possible. By setting either SSCR1_x[ECRA] or SSCR1_x[ECRB], the source clock of an SSP port can be controlled by another SSP port's clock request.

Table 8-3. SSP Port Clock Request Enable Selection

		Serial Clock Source SSP Port		
		SSP1	SSP2	SSP3
Clock Request SSP	SSP1		SSCR1_2[ECRA] set	SSCR1_3[ECRA] set
	SSP2	SSCR1_1[ECRA] set		SSCR1_3[ECRB] set
	SSP3	SSCR1_1[ECRB] set	SSCR1_2[ECRB] set	

Figure 8-19. Clock Enabling on SSP2



NOTES: The SSCR0_2 and SSCR1_2 control bits above are in SSP2. SSP3 does not have an external clock source; Because of this, SSPSCLKEN for SSP3 is tied to ground.

8.4.9 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 CPU programmed I/O or by DMA bursts.

8.4.9.1 Using CPU Programmed I/O Data Transfers

The CPU 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 (if enabled). When the number of entries in the receive FIFO exceeds the value in SSCR1_x[RFT] + 1, 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_x[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 8.5.3) shows whether the FIFO is full, empty, or how many samples it contains.

8.4.9.2 Using DMA Data Transfers

The DMA controller can be programmed to transfer data to and from the SSP port's FIFOs. The SSP port stores data in its FIFOs one sample per FIFO location (each FIFO has 16 locations). Therefore, the DMA burst size (DCMD_x[SIZE]) must not exceed 16 samples. For example, the maximum DMA burst size is 16 bytes when the DCMD_x[WIDTH] is set to byte wide (0b01). There must be sufficient empty room in the transmit FIFO or a sufficient number of samples in the receive FIFO before a DMA burst is enabled. The SSP port's data sample size, the DMA sample width, and the SSP port's TX/RX FIFO threshold settings together determine the allowable DMA burst size. To prevent underruns of the transmit FIFO or overruns of the receive FIFO when using DMA, take care when setting the transmit and receive trigger thresholds. Refer to Table 8-4 for the allowable DMA burst sizes for different combinations of SSP port sample size, DMA sample width and SSP port threshold levels. Refer to Chapter 5, "DMA Controller" for instructions on programming the DMA channels.

The programming model for using the DMA is as follows:

- Program the total number of transmit and receive byte lengths, burst sizes, and peripheral width. Program DCMD_x[WIDTH] to 0b01 for SSP port formats of 8 bits or less; to 0b10 for SSP port formats of 9 to 16 bits; to 0b11 for SSP port formats of more than 16 bits.
- Set the preferred values in the SSP Control registers.

- Set SSCR0_x[SSE] to enable the SSP port (see Section 8.5.1).
- Set DCSR_x[RUN].
- Wait for the DMA transmit or 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 8.4.2.

In full-duplex formats where the SSP port 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.

Table 8-4. TFT and RFT Values with Possible DMA Burst Sizes

SSP Data Size (SSCR0_x[EDSS], SSCR0_x[DSS])	DMA Width (DCMDx[WIDTH])	TX/RX Threshold (SSCR1_x[TFT], SSCR1_x[RFT]) †	Allowed DMA Burst Size (in Bytes)
4–8 bits	1 byte (0b01)	TFT = 0 RFT = 15	8 or 16 8 or 16
		0 < TFT < 8 6 < RFT < 15	8 8
		TFT > 7 RFT < 7	Do not use DMA Do not use DMA
9–16 bits	2 bytes (0b10)	TFT = 0 RFT = 15	8, 16, or 32 8, 16, or 32
		TFT < 8 6 < RFT < 15	8 or 16 8 or 16
		7 < TFT < 12 2 < RFT < 7	8 8
		TFT > 11 RFT < 3	Do not use DMA Do not use DMA
17–32 bits	4 bytes (0b11)	TFT < 8 RFT > 6	8, 16, or 32 8, 16, or 32
		7 < TFT < 12 2 < RFT < 7	8 or 16 8 or 16
		11 < TFT < 14 0 < RFT < 3	8 8
		TFT > 13 RFT = 0	Do not use DMA Do not use DMA

† Valid values for TFT and RFT are 0 through 15. For register details, see Table 8-7.

The DMA transmit burst size is limited because the smallest SSCR1_x[TFT] value is zero, which equates to one sample left in the transmit FIFO. Software must program the transmit FIFO threshold and the DMA burst size (DCMD_x[SIZE]) such that an underflow of the transmit FIFO does not occur.

The receive FIFO threshold and the DMA burst size must be configured such that an overflow of the receive FIFO does not occur.

8.4.10 Baud-Rate Generation

When the SSP port is configured as the master of SSPSCLKx as determined by SSCR1_x[SCLKDIR], the baud rate (serial bit-rate SSPSCLKx) is generated internally by dividing one of the following by a programmable divider (SSCR0_x[SCR]):

- The on-chip 13-MHz clock
- The network clock (CLK_EXT) described in Section 24.4.2, "GPIO Operation as Alternate Function" on page 24-3.
- The SSP port external clock (SSPEXTCLKx)

The division of the baud rate by one of these programmable dividers generates baud rates up to a maximum of 13 Mbits per second. If the Audio Clock Select bit is set (SSCR0_x[ACS]), the frequency of the audio clock used by the SSP port is determined by the Audio Clock Divider register (SSACD_x). The audio clock can also be routed through the baud rate divider, but its phase relationship with SSPSYSCLK will be lost.

Table 8-5 shows the clock selected to source the baud rate generator by the external clock select (SSCR0_x[ECS]), network clock select (SSCR0_x[NCS]), or audio clock select (SSCR0_x[ACS]. For register details, see Section 8.5.2).

When changing clock sources, follow these steps:

1. Turn off the SSP port's internal clock by clearing the appropriate bit in the clock unit's CKEN register—for example, CKEN[23] for SSP1. For full mapping information, see Table 3-33, "Clock Enable Mappings for CKEN Bits" on page 3-98. For CKEN register details, see Section 3.8.2.2, "Clock Enable Register (CKEN)" on page 3-97.
2. Write SSCR0_x[ECS] or SSCR0_x[NCS] or SSCR0_x[ACS].
3. Re-enable the SSP port's internal clock by setting the appropriate CKEN register bit.

Table 8-5. SSP Port Clock Selection

SSCR0_x[ACS]	SSCR0_x[ECS]	SSCR0_x[NCS]	Selected Clock
0	0	0	On-Chip Clock (internal 13-MHz PLL)
0	0	1	Network Clock (CLK_EXT)
0	1	0	SSP External Clock (SSPEXTCLK)
0	1	1	Network Clock (CLK_EXT)
1	X	X	On-Chip Audio Clock (determined from internal PLL and SSACD_x)

8.5 Register Descriptions

Each of the three SSP ports contain eleven registers: six control, one data, two status, one time-out, and one test.

- Access all registers using aligned words.
- The SSP Control registers SSCR0_x, and SSCR1_x 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. For SSCR0_x, the DSS, FRF, EDSS, ECS, NCS, and ACS bit fields must be written before the SSE bit is set. If the DSS, FRF, EDSS, ECS, NCS, or ACS bits need to be modified after the SSE bit is set; clear the SSE bit, make the modifications, and

then again set the SSE bit. For SSCR1_x, only the SPO, SPH, and SCFR bits must be written before the SSE bit is set. If the SPO, SPH, or SCFR bits need to be modified after the SSE bit is set; clear the SSE bit, make the modifications, and then again set the SSE bit.

- Any writable bits in the SSSR, SSTO, and SSITR registers can be written at any time.
- All bit fields in the SSPSP and SSACD registers must be written before the SSE bit is set. If these bits need to be modified after the SSE bit is set; clear the SSE bit, make the modifications, and then again set the SSE bit.
- Write all bits in the SSRSA and SSTSA registers before the SSE bit is set. If these bits need to be modified after the SSE bit is set; clear the SSE bit, make the modifications, and then again set the SSE bit.
- The SSP Time-Out (SSTO_x) register programs the time-out value to signal a specified period of receive FIFO inactivity.
- While in PSP mode, the SSP Programmable Serial Protocol (SSPSP_x) registers program 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 to the transmit FIFO and the other register is for reads that take data from the receive FIFO. A write cycle or burst write loads successive words into the SSP port's SSDR_x write register and then into the transmit FIFO. A read cycle or burst read takes data from the SSP port's SSDR_x read register as the receive FIFO reloads it with the "oldest" available FIFO data bits.

  Do not increment the address when using read and write DMA bursts. All accesses to the SSP Data register's memory address access either the SSDR_x read register or SSDR_x write register.

  The FIFOs are independent buffers that allow full duplex operation.
- Besides showing the state of the FIFO buffers, the status register (SSSR_x) 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 port is actively transmitting data, when the transmit FIFO is not full, and when the receive FIFO is not empty. SSSR_x[ROR] bit signals an overrun of the receive FIFO. In this case newly received data is discarded (see Section 8-11).

8.5.1 SSP Control Register 0 (SSCR0_x)

SSCR0_x, shown in Table 8-6, controls various functions within the SSP port. Before enabling the SSP port by setting SSCR0_x[SSE], the desired values for this register must be programmed.

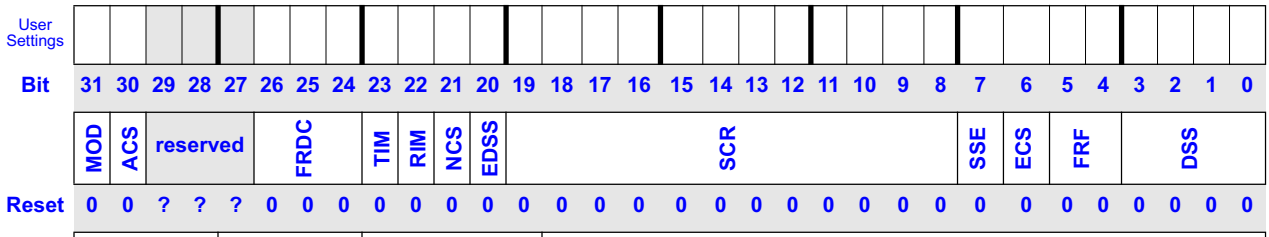
These are read/write registers. Ignore reads from reserved bits. Write 0b0 to reserved bits.

Table 8-6. SSCR0_1/2/3 Bit Definitions (Sheet 1 of 4)

Physical Address
0x4100_0000
0x4170_0000
0x4190_0000

SSCR0_1
SSCR0_2
SSCR0_3

SSP Controller



Bits	Access	Name	Description
31	R/W	MOD	Mode 0 = Normal SSP port mode 1 = Network mode **NOTE:** When network mode is selected, only use DMA to move data to/from the SSP Data register (SSDR_x). Do not use interrupts or CPU polling. Set this bit only when using PSP and TI SSP formats. Setting this bit causes SSPSCLKx to run continuously (if the SSP port is a master of the clock—SSCR1_x[SCLKDIR] = 0). During network mode, only one SSPSFRMx is sent (received) for the number of time slots programmed into the SSCR0_x[FRDC] field. See Section 8.4.6 for additional information. When using the PSP format in network mode, the parameters SFRMDLY, STRTDLY, DMYSTP, DMYSTRT must be cleared. Other parameters (such as FRMPOL, SCMODE, FSRT, SFRMDWDTH) are programmable.
30	R/W	ACS	Audio Clock Select 0 = SSPSCLKx selection is determined by the NCS and ECS bits. 1 = Audio Clock (and Audio Clock Divider) creates SSPSCLKx. If the ACS bit is set (and the GPIO are properly configured), SSPSYSCLKx is continually output (even if the SSP port is disabled). SSPSCLKx is output as previously defined (determined by format, SSPSCLKENx, ECRA/ECRB functions).
29:27	—	—	reserved
26:24	R/W	FRDC	Frame Rate Divider Control Value 0-7 indicates the number of time slots per frame when in network mode (the actual number of time slots is FRDC + 1 for 1–8 time slots)
23	R/W	TIM	Transmit FIFO Underrun Interrupt Mask 0 = TUR events generate an SSP port interrupt. 1 = TUR events do not generate an SSP port interrupt. When set, this bit masks the TX FIFO Underrun (TUR) event from generating an SSP port interrupt. SSSR_x will still indicate that a TUR event has occurred. This bit can be written to at any time (before or after SSP port is enabled).
22	R/W	RIM	Receive FIFO Overrun Interrupt Mask 0 = ROR events generate an SSP port interrupt. 1 = ROR events do not generate an SSP port interrupt. When set, this bit masks the RX FIFO Overrun (ROR) event from generating an SSP port interrupt. SSSR_x will still indicate that an ROR event has occurred.This bit can be written to at any time (before or after SSP port is enabled).

Table 8-6. SSCR0_1/2/3 Bit Definitions (Sheet 2 of 4)

Physical Address
0x4100_0000
0x4170_0000
0x4190_0000

SSCR0_1
SSCR0_2
SSCR0_3

SSP Controller

	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
User Settings																																
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
	MOD	ACS	reserved			FRDC			TIM	RIM	NCS	EDSS	SCR												SSE	ECS	FRF		DSS			
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	0	0

Bits	Access	Name	Description
21	R/W	NCS	Network Clock Select Used with ECS to select the network clock. 0 = The NCS bit determines clock selection. 1 = Network Clock creates the SSP port's SSPSCLKx. Before setting the NCS bit, first disable the port. The NCS (and ECS) bits must be configured before or at the same time that the SSE bit is set. For more information, see Table 8-5.
20	R/W	EDSS	Extended Data Size Select Used with DSS to select the size of the data transmitted and received by the SSP port. 0 = Zero is pre-appended to the DSS value that sets the DSS range from 4-16- bits. 1 = One is pre-appended to the DSS value that sets the DSS range from 17-32-bits.
19:8	R/W	SCR	Serial Clock Rate Selects the bit rate of the SSP port when in master mode with respect to SSPSCLKx (as defined by SSCR1_x[SCLKDIR]). The maximum bit rate is 13 Mbps. The serial-clock generator uses clocks selected by ECS and NCS. The selected clock is divided by the value of SCR plus 1 (a range of 1 to 4096) to generate SSPSCLKx. **NOTE:** This field is ignored when the SSP port is a slave with respect to SSPSCLKx (defined by SSCR1_x[SCLKDIR]) and transmission data rates are determined by an external device. **NOTE:** Software must not change SCR when SSPSCLKx is enabled (through use of the SSPSCLKEN pin or SSPCR1_x[ECRA] or SPCR1_x[ECRB]) because doing so causes the SSPSCLKx frequency to immediately change. Values (0 to 4095) generate the clock rate of the SSP port. Serial bit rate = SSP Port Clock / (SCR + 1), where SCR is a decimal integer

Table 8-6. SSCR0_1/2/3 Bit Definitions (Sheet 3 of 4)

Physical Address	SSCR0_1 / SSCR0_2 / SSCR0_3	SSP Controller
0x4100_0000 0x4170_0000 0x4190_0000		

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
User Settings																																
	MOD	ACS	reserved			FRDC			TIM	RIM	NCS	EDSS	SCR												SSE	ECS	FRF		DSS				
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	0	0

Bits	Access	Name	Description
7	R/W	SSE	Synchronous Serial Enable Enables or disables all SSP port 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 port is disabled immediately, terminating the current frame being transmitted or received. Clearing SSE resets the port FIFOs and the status bits; however, the SSP port control registers and the receive FIFO overrun status bit are not reset. **NOTE:** After reset or after clearing the SSE, ensure that the SSCR1_x, SSITR_x, SSTO_x, and SSPSP_x control registers are properly re-configured and that the SSSR register is reset before re-enabling the SSP port by setting SSE. Also, SSE must be cleared before re-configuring the SSCR0_x, SSCR1_x, or SSPSP_x registers; any or all control bits in SSCR0_x can be written at the same time as the SSE. 0 = SSP port operation disabled. 1 = SSP port operation enabled.
6	R/W	ECS	External Clock Select Used with NCS to select the clock source for the SSP port. ECS in conjunction with NCS selects whether the SSP port uses the on-chip 13-MHz clock or one of two off-chip clocks supplied by GPIO: the network clock (CLK_EXT, described in Chapter 24, “General-Purpose I/O Controller”) or the SSP port’s external clock (SSPEXTCLK) produces serial transmission rates ranging from 6.3 Kbps (minimum recommended bit rate) to a maximum of 13 Mbps. When NCS is cleared, ECS selects between the on-chip 13-MHz clock and external clock (SSPEXTCLK). When NCS is set, the network clock (CLK_EXT) is selected. The frequency of the off-chip clock can be any value up to 13 MHz. When the SSP port is a slave with respect to SSPSCLKx (defined by the SSCR1_x[SCLKDIR]), this field is ignored and transmission data rates are determined by the external device. Before setting the ECS bit, first disable the port. The ECS (and NCS) bit must be configured before or at the same time that SSE is set. For more information see Table 8-5. When ECS is cleared, SSPEXTCLKx is treated as SSPSCLKENx, a clock enable that gates the SSPSCLKx output. See Figure 8-19 for details. When the SSPSCLKENx changes, there is a 1–2 clock lag before SSPSCLKx is started or stopped because of internal synchronization delays. 0 = On-chip clock produces the SSP port’s SSPSCLKx. 1 = SSPEXTCLK/GPIO pin creates the SSP port’s SSPSCLKx. **NOTE:** The ECS bit for SSP3, SSCR0_3[ECS], should never be set to one because the SSP3 does not have an associated SSPEXTCLK.

Table 8-6. SSCR0_1/2/3 Bit Definitions (Sheet 4 of 4)

Physical Address
0x4100_0000
0x4170_0000
0x4190_0000

SSCR0_1
SSCR0_2
SSCR0_3

SSP Controller

	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
User Settings																																
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
	MOD	ACS	reserved			FRDC			TIM	RIM	NCS	EDSS	SCR												SSE	ECS	FRF		DSS			
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	0	0

Bits	Access	Name	Description
5:4	R/W	FRF	Frame Format Selects which frame format to use. 0b00 = Motorola Serial Peripheral Interface 0b01 = TI Synchronous Serial Protocol 0b10 = Microwire 0b11 = Programmable Serial Protocol
3:0	R/W	DSS	Data Size Select Used in conjunction with EDSS to select the size of the data transmitted and received by the SSP port. 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 determine the receive data size. The size of the transmitted data is either eight or 16-bits (determined by SSCR1_x[MWDS]) and the EDSS bit is ignored. For all modes (including Microwire), EDSS and DSS determine the receive data size. When data is programmed to be less than 32 bits, data written to the TX FIFO must be right-justified.

EDSS	DSS	Data Size	EDSS	DSS	Data Size
1	0b0000	17-bit data	0	0b0000	reserved, undefined
1	0b0001	18-bit data	0	0b0001	reserved, undefined
1	0b0010	19-bit data	0	0b0010	reserved, undefined
1	0b0011	20-bit data	0	0b0011	4-bit data
1	0b0100	21-bit data	0	0b0100	5-bit data
1	0b0101	22-bit data	0	0b0101	6-bit data
1	0b0110	23-bit data	0	0b0110	7-bit data
1	0b0111	24-bit data	0	0b0111	8-bit data
1	0b1000	25-bit data	0	0b1000	9-bit data
1	0b1001	26-bit data	0	0b1001	10-bit data
1	0b1010	27-bit data	0	0b1010	11-bit data
1	0b1011	28-bit data	0	0b1011	12-bit data
1	0b1100	29-bit data	0	0b1100	13-bit data
1	0b1101	30-bit data	0	0b1101	14-bit data
1	0b1110	31-bit data	0	0b1110	15-bit data
1	0b1111	32-bit data	0	0b1111	16-bit data

8.5.2 SSP Control Register 1 (SSCR1_x)

SSCR1_x, shown in Table 8-7, controls various SSP port functions. Before enabling the port (using SSCR0_x[SSE]), the desired values for this register must be set.

These are read/write registers. Ignore reads from reserved bits. Write 0b0 to reserved bits.

Table 8-7. SSCR1_1/2/3 Bit Definitions (Sheet 1 of 10)

Physical Address
0x4100_0004
0x4170_0004
0x4190_0004

SSCR1_1
SSCR1_2
SSCR1_3

SSP Controller

	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
User Settings																																
Bit	TTELP	TTE	EBCEI	SCFR	ECRA	ECRB	SCLKDIR	SFRMDIR	RWOT	TRAIL	TSRE	RSRE	TINTE	PINTE	Reserved	IFS	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	0	0	0	0

Bits	Access	Name	Description
31	R/W	TTELP	TXD Tristate Enable on Last Phase TTELP is used in conjunction with TTE. When set, TTELP causes SSPTXDx to become high impedance 1/2 phase later than specified in the TTE bit description under either of the following 2 conditions: 1) the format is SSP; or 2) the format is PSP and the SSP port is a slave to frame (SSCR1_x[SFRMDIR] is set). For example, when in SSP format, with TTELP and TTE set, SSPTXDx goes to high impedance one full clock after the clock edge that starts the LSB. Ensure if SCLKDIR is set (SSP port is a slave to SSPSCLKx) that the device driving SSPSCLKx provides another clock edge. **NOTE:** TTELP must not be set when using Microwire or SPI formats 1 = SSPTXDx becomes high impedance one full clock after the clock edge that starts the LSB. 0 = SSPTXDx becomes high impedance 1/2 clock after the clock edge that starts the LSB.

Table 8-7. SSCR1_1/2/3 Bit Definitions (Sheet 2 of 10)

Physical Address			SSP Controller
0x4100_0004 0x4170_0004 0x4190_0004	SSCR1_1 SSCR1_2 SSCR1_3		

	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
User Settings																																
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	ECRA	ECRB	SCLKDIR	SFRMDIR	RWOT	TRAIL	TSRE	RSRE	TINTE	PINTE	Reserved	IFS	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	0	0	0	0

Bits	Access	Name	Description
30	R/W	TTE	TXD Tristate Enable If TTE is cleared, SSPTXDx is always driven. When set, TTE causes SSPTXDx to become high impedance when the SSP port is not transmitting data. The timing for the high impedance enable/disable varies according to the different serial formats and frame direction. For Microwire format, SSPTXDx is driven on the same clock edge that the MSB is driven, and SSPTXDx becomes high impedance one full clock after the clock edge that starts the LSB. See Section 8.4.5.3. For SPI protocol, SSPTXDx becomes high impedance whenever SPSFRMx is deasserted (driven high). See Section 8.4.5.2. For TI Synchronous Serial Protocol, SSPTXDx is driven at the first rising edge of the SSPSCLKx after SSPSFRMx is asserted (the same time the MSB is driven) and continues to be driven until 1/2 clock (TTELP = 1) or 1 full clock (TTELP = 0) after the clock edge that starts the LSB. See Section 8.4.5.1. For PSP format when the SSP port is a slave to frame, SSPTXDx becomes high impedance 1/2 clock after the clock edge that starts the LSB. For PSP format when the SSP port is a master to frame, SSPTXDx becomes high impedance one full clock after the clock edge that starts the LSB (even if the SSP port is a master of clock and this clock edge does not appear on SSPSCLKx). See Section 8.4.5.4. **NOTE:** TTE must be set when in network mode. 0 = SSPTXDx does not become high impedance. 1 = SSPTXDx becomes high impedance when not transmitting data.
29	R/W	EBCEI	Enable Bit Count Error Interrupt When set, EBCEI enables a bit count error interrupt. A bit count error occurs when the SSP port is a slave to SSPSCLKx and/or SSPSFRMx and the SSP port detects a new frame before the internal bit counter has reached 0 (before the LSB was driven). 0 = Interrupt due to a bit count error is disabled. 1 = Interrupt due to a bit count error is enabled.
28	R/W	SCFR	Slave Clock Free Running In slave mode (SSCR1_x[SCLKDIR] is set), SCFR must be cleared if the input clock from the external source is running continuously. In master mode (SSCR1_x[SCLKDIR] is cleared), SCFR is ignored. Slave mode only: 0 = SSPSCLKx is continuously running. 1 = SSPSCLKx is active only during transfers.

Table 8-7. SSCR1_1/2/3 Bit Definitions (Sheet 3 of 10)

Physical Address
0x4100_0004
0x4170_0004
0x4190_0004

SSCR1_1
SSCR1_2
SSCR1_3

SSP Controller

	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
User Settings																																
Bit	TTELP	TTE	EBCEI	SCFR	ECRA	ECRB	SCLKDIR	SFRMDIR	RWOT	TRAIL	TSRE	RSRE	TINTE	PINTE	Reserved	IFS	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	0	0	0	0

Bits	Access	Name	Description
27	R/W	ECRA	Enable Clock Request A Refer to Section 8.4.8 for details on the use of this bit. 0 = Clock request from another SSP port is disabled. 1 = Clock request from another SSP port is enabled.
26	R/W	ECRB	Enable Clock Request B Refer to Section 8.4.8 for details on the use of this bit. 0 = Clock request from another SSP port is disabled. 1 = Clock request from another SSP port is enabled.
25	R/W	SCLKDIR	SSPSCLKx Direction SCLKDIR determines whether the port is the master or slave (with respect to driving SSPSCLKx). Depending on the frame format selected, each transmitted bit is driven on either the rising or falling edge of SSPSCLKx, and is sampled on the opposite clock edge. When the GPIO alternate function is selected for the SSP port, this bit has precedence over the GPIO direction bit. SCLKDIR must be written before the GPIO direction bit (to prevent any possible contention on SSPSCLKx). 0 = Master mode, the port generates SSPSCLKx internally, acts as the master, and drives SSPSCLKx. 1 = Slave mode, the port acts as a slave, receives SSPSCLKx from an external device and uses it to determine when to drive transmit data on SSPTXDx and when to sample receive data on SSPRXDx. **NOTE:** When SCLKDIR is set, the SSCR0_x[NCS] and SSCR0_x[ESC] bits must be cleared.
24	R/W	SFRMDIR	SSP Frame Direction SFRMDIR determines whether the SSP port is the master or slave (with respect to driving SSPSFRMx). **NOTE:** When the port is configured as a slave to SSPSFRMx, the external device driving SSPSFRMx must wait until SSSR_x[CSS] is cleared after enabling the port and before asserting SSPSFRMx (no external clock cycles are needed). When the GPIO alternate function is selected for the port, SFRMDIR has precedence over the GPIO direction bit. SFRMDIR must be written before the GPIO direction bit (to prevent any possible contention on SSPSFRMx). 0 = Master mode, the port generates SSPSFRMx internally, acts as the master and drives SSPSFRMx. 1 = Slave mode, the port acts as a slave, receives SSPSFRMx from an external device.

Table 8-7. SSCR1_1/2/3 Bit Definitions (Sheet 4 of 10)

Physical Address
0x4100_0004
0x4170_0004
0x4190_0004

SSCR1_1
SSCR1_2
SSCR1_3

SSP Controller

	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
User Settings																																
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	ECRA	ECRB	SCLKDIR	SFRMDIR	RWOT	TRAIL	TSRE	RSRE	TINTE	PINTE	Reserved	IFS	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	0	0	0	0

Bits	Access	Name	Description
23	R/W	RWOT	Receive Without Transmit RWOT puts the SSP port into a mode similar to half duplex. This allows the port to receive data without transmitting data (half-duplex only). When the port is in master mode (SCLKDIR cleared) and 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_x[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 both SSCR1_x[TSRE,TIE]). If the transmit FIFO is empty, SSPTXDx is driven low. The transmit FIFO underrun condition does not occur when RWOT is set. When RWOT is set, SSSR_x[BSY] remains set until software clears the RWOT bit. After RWOT is cleared, and extra frame cycle may occur due to synchronization delays between the processor's clock domains. RWOT must not be used when SSCR0_x[MOD] is set. 0 = Transmit/receive mode. 1 = Receive without transmit mode.
22	R/W	TRAIL	Trailing Byte TRAIL configures how trailing bytes are handled (see Section 8.4.2 for more detail). 0 = Processor based, Trailing bytes are handled by the CPU. 1 = DMA based, Trailing bytes are handled by DMA.
21	R/W	TSRE	Transmit Service Request Enable TSRE enables the transmit FIFO DMA Service Request. **NOTE:** Clearing TSRE does not affect the current state of SSSR_x[TFS] or the ability of the transmit FIFO logic to set and clear SSSR_x[TFS]; it prevents the generation of the DMA Service Request. The state of TSRE does not effect the generation of the interrupt, which is asserted whenever the SSSR_x[TFS] is set. 0 = DMA service request is disabled and the state of the transmit FIFO DMA service request is ignored. 1 = DMA service request is enabled.

Table 8-7. SSCR1_1/2/3 Bit Definitions (Sheet 5 of 10)

Physical Address	SSCR1_1 / SSCR1_2 / SSCR1_3	SSP Controller
0x4100_0004 0x4170_0004 0x4190_0004		

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
User Settings																																
	TTELP	TTE	EBCEI	SCFR	ECRA	ECRB	SCLKDIR	SFRMDIR	RWOT	TRAIL	TSRE	RSRE	TINTE	PINTE	Reserved	IFS	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	0	0	0	0

Bits	Access	Name	Description
20	R/W	RSRE	Receive Service Request Enable RSRE enables the Receive FIFO DMA Service Request. **NOTE:** Clearing RSRE does not affect the current state of SSSR_x[RFS] or the ability of the receive FIFO logic to set and clear SSSR_x[RFS]; it blocks only the generation of the DMA Service Request. The state of RFRS does not affect the generation of the interrupt, which is asserted whenever SSSR_x[RFS] is set. 0 = DMA service request is disabled and the state of the SSSR_x[RFS] is ignored. 1 = DMA service request is enabled.
19	R/W	TINTE	Receiver Time-Out Interrupt Enable TINTE enables the receiver time-out interrupt. **NOTE:** Clearing TINTE does not affect the current state of SSSR_x[TINT] or the ability of logic to set and clear SSSR_x[TINT]; it prevents the generation of the interrupt request. 0 = Receiver time-out interrupts are disabled. The interrupt is masked and the state of SSSR_x[TINT] is ignored by the interrupt controller. 1 = Receiver time-out interrupts are enabled.
18	R/W	PINTE	Peripheral Trailing Byte Interrupt Enable PINTE enables the peripheral trailing byte interrupt. **NOTE:** Clearing PINTE does not affect the current state of SSSR_x[PINT] or the ability of logic to set and clear SSSR_x[PINT]; it prevents the generation of the interrupt request. 0 = Peripheral trailing byte interrupts are disabled. The interrupt is masked and the state of SSSRx[PINT] is ignored by the interrupt controller. 1 = Peripheral trailing byte interrupt are enabled.
17	—	—	reserved
16	R/W	IFS	Invert Frame Signal 0 = SSPSFRMx polarity is determined by SSP format and PSP polarity bits. 1 = SSPSFRMx is inverted from the normal SSP frame signal (as defined by the SSP format and PSP polarity bits).

Table 8-7. SSCR1_1/2/3 Bit Definitions (Sheet 6 of 10)

Physical Address
0x4100_0004
0x4170_0004
0x4190_0004

SSCR1_1
SSCR1_2
SSCR1_3

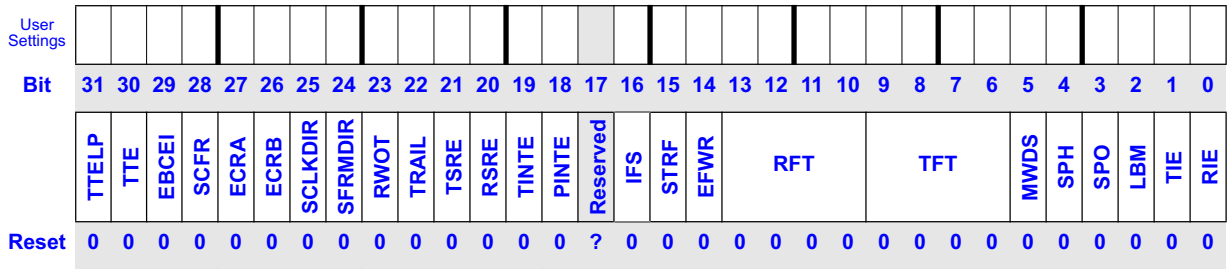
SSP Controller

	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
User Settings																																
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	ECRA	ECRB	SCLKDIR	SFRMDIR	RWOT	TRAIL	TSRE	RSRE	TINTE	PINTE	Reserved	IFS	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	0	0	0	0

Bits	Access	Name	Description
15	R/W	STRF	Select FIFO for EFWR (test mode bit only) Only when SSCR1_x[EFWR] is set, STRF selects whether the transmit or the receive FIFO is enabled for writes and reads. 0 = Transmit FIFO is selected for both writes and reads through SSDR_x 1 = Receive FIFO is selected for both writes and reads through SSDR_x
14	R/W	EFWR	Enable FIFO Write/Read (test mode bit) Enables test mode for the SSP port. When set, the SSP port enters a mode where whenever the CPU reads or writes to the SSP Data register, it reads and writes directly to either the transmit FIFO or the receive FIFO, depending on the programmed state of SSCR1_x[STRF]. In EFWR test mode, data is not transmitted on SSPTXDx, data input on SSPRXDx is not stored, and the Busy and ROR bits have no effect. However, the Interrupt Test register is still functional. Using software, this mode can test whether or not the TX FIFO or the RX FIFO operates properly as a FIFO memory stack. Verify that the SSSR_x[CSS] bit has gone from set to clear before reading the TX FIFO. This bit must be cleared for normal operation. When SSCR1_x[STRF] is clear, writes to SSDR_x are performed on the Transmit FIFO, and reads from SSDR_x read back the data written to the TX FIFO in first-in-first-out order. When the STRF is set, writes to SSDR_x are performed on the RX FIFO, and reads from SSDR_x read back the data written to the RX FIFO in first-in-first-out order. 0 = FIFO write/read special function is disabled (normal SSP port operational mode) 1 = FIFO write/read special function is enabled.
13:10	R/W	RFT	Receive FIFO Threshold RFT sets the level at or above which the FIFO controller triggers a DMA service request (if enabled) and a CPU interrupt request (if enabled). This level must be set to the desired trigger threshold value minus 1. **NOTE:** Do not to set the value of RFT too high for the system; otherwise, the receive FIFO can overrun because of the bus latencies caused by other internal and external peripherals. This is especially important when using interrupts and polled modes that require a longer time to service.

Table 8-7. SSCR1_1/2/3 Bit Definitions (Sheet 7 of 10)

Physical Address	SSCR1_1 / SSCR1_2 / SSCR1_3	SSP Controller
0x4100_0004 0x4170_0004 0x4190_0004		



Bits	Access	Name	Description
9:6	R/W	TFT	Transmit FIFO Threshold TFT sets the level at or below which the FIFO controller triggers a DMA service request (if enabled) and a CPU interrupt request (if enabled). This level must be set to the desired trigger threshold value minus 1. **NOTE:** Do not set the value of TFT too low for the system; otherwise, the transmit FIFO can underrun because of the bus latencies caused by other internal and external peripherals. This is especially important when using interrupts and polled modes that require a longer time to service.
5	R/W	MWDS	Microwire Transmit Data Size MWDS selects between an eight bit or 16-bit size for the command word transmitted using the Microwire protocol. MWDS is ignored for all other frame formats. 0 = 8-bit command word is transmitted. 1 = 16-bit command word is transmitted.

Table 8-7. SSCR1_1/2/3 Bit Definitions (Sheet 8 of 10)

Physical Address
0x4100_0004
0x4170_0004
0x4190_0004

SSCR1_1
SSCR1_2
SSCR1_3

SSP Controller

	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
User Settings																																
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	ECRA	ECRB	SCLKDIR	SFRMDIR	RWOT	TRAIL	TSRE	RSRE	TINTE	PINTE	Reserved	IFS	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	0	0	0	0

Bits	Access	Name	Description
4	R/W	SPH	Motorola SPI SSPSCLKx Phase SPH determines the phase relationship between SSPSCLKx and the SSPSFRMx when the Motorola SPI format is selected. When SPH is clear, SSPSCLKx remains in its Inactive/Idle state (as determined by the SSCR1_x[SPO] setting) for one full cycle after SSPSFRMx is asserted low at the beginning of a frame. SSPSCLKx continues to toggle for the rest of the frame and is then held in its Inactive state for one-half of an SSPSCLKx period before SSPSFRMx is deasserted high at the end of the frame. When SPH is set, SSPSCLKx remains in its Inactive/Idle state (as determined by the SSCR1_x[SPO] bit's value) for one-half cycle after SSPSFRMx is asserted low at the beginning of a frame. SSPSCLKx continues to toggle for the remainder of the frame, and is then held in its Inactive state for one full SSPSCLKx period before SSPSFRMx is deasserted high at the end of the frame. The combination of the SSCR1_x[SPO] bit and SSCR1_x[SPH] bit settings determines when SSPSCLKx is active during the assertion of SSPSFRMx, and which SSPSCLKx edge transmits and receives data on the SSPTXDx and SSPRXDx pins. When SSCR1_x[SPO] and SSCR1_x[SPH] are programmed to the same value (both clear or both set), SSPTXDx is driven on the falling edge of SSPSCLKx, and SSPRXDx is latched on the rising edge of SSPSCLKx. When SSCR1_x[SPO] and SSCR1_x[SPH] are programmed to opposite values (one clear and the other set), SSPTXDx is driven on the rising edge of SSPSCLKx and SSPRXDx is latched on the falling edge of SSPSCLKx. **NOTE:** SPH is ignored for all data frame formats except for the Motorola SPI format. Figure 8-6 and Figure 8-7 show the timing for all four programming combinations of SSCR1_x[SPO] and SSCR1_x[SPH]. SSCR1_x[SPO] inverts the polarity of SSPSCLKx, and SSCR1_x[SPH] determines the phase relationship between SSPSCLKx and SSPSFRMx, shifting the SSPSCLKx one-half phase to the left or right during the assertion of SSPSFRMx. 0 = SSPSCLKx is inactive one cycle at the start of a frame and 1/2 cycle at the end of a frame. 1 = SSPSCLKx is inactive 1/2 cycle at the start of a frame and one cycle at the end of a frame.

Table 8-7. SSCR1_1/2/3 Bit Definitions (Sheet 9 of 10)

Physical Address	SSCR1_1 / SSCR1_2 / SSCR1_3	SSP Controller
0x4100_0004 0x4170_0004 0x4190_0004		

User Settings																															
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	ECRA	ECRB	SCLKDIR	SFRMDIR	RWOT	TRAIL	TSRE	RSRE	TINTE	PINTE	Reserved	IFS	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	0	0	0	0

Bits	Access	Name	Description
3	R/W	SPO	Motorola SPI SSPSCLKx Polarity SPO selects the polarity of the inactive state of SSPSCLKx when the SPI protocol is selected. The programmed setting of SPO alone does not determine which SSPSCLKx edge transmits or receives data; SPO in combination with SSCR1_x[SPH] does. **NOTE:** SSCR1_x[SPO] is ignored for all data frame formats except for SPI protocol (SSCR0_x[FRF] = 0b00). 0 = SSPSCLKx is held low in the inactive or idle state when the SSP port is not transmitting/receiving data. 1 = SSPSCLKx is held high during the inactive or idle state.
2	R/W	LBM	Loop-Back Mode (test-mode bit only) LBM is a test mode bit that enables and disables the ability of the SSP port's transmit and receive logic to communicate. **NOTE:** The loop-back mode cannot be used with the Microwire protocol since this protocol uses half-duplex master-slave message passing. 0 = Normal SSP port operation is enabled. 1 = Output of transmit serial shifter is internally connected to the input of the receive serial shifter. SSPTXDx continues to function normally.
1	R/W	TIE	Transmit FIFO Interrupt Enable TIE enables the TX FIFO service request interrupt. **NOTE:** Clearing TIE does not affect the current state of SSSR_x[TFS] or the ability of the transmit FIFO logic to set and clear SSSR_x[TFS]—it blocks only the generation of the interrupt request. Also, the state of TIE does not effect the generation of the transmit FIFO DMA service request, which is asserted whenever SSSR_x[TFS] is set. 0 = TX FIFO level interrupt is disabled. The interrupt is masked and the state of SSSR_x[TFS] is ignored. 1 = TX FIFO level interrupt is enabled. Whenever SSSR_x[TFS] is set, an interrupt request is made to the interrupt controller.

Table 8-7. SSCR1_1/2/3 Bit Definitions (Sheet 10 of 10)

Physical Address: 0x4100_0004, 0x4170_0004, 0x4190_0004 — SSCR1_1, SSCR1_2, SSCR1_3 — SSP Controller

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
User Settings																																
	TTELP	TTE	EBCEI	SCFR	ECRA	ECRB	SCLKDIR	SFRMDIR	RWOT	TRAIL	TSRE	RSRE	TINTE	PINTE	Reserved	IFS	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	0	0	0	0

Bits	Access	Name	Description
0	R/W	RIE	Receive FIFO Interrupt Enable RIE enables the RX FIFO service request interrupt. **NOTE:** Clearing RIE does not affect the current state of SSSR_x[RFS] or the ability of the RX FIFO logic to set and clear SSSR_x[RFS]—it blocks only the generation of the interrupt request. The state of RIE does not affect the generation of the RX FIFO DMA service request, which is asserted whenever SSSR_x[RFS] is set. 0 = RX FIFO level interrupt is disabled. The interrupt is masked and SSSR_x[RFS] is ignored. 1 = RX FIFO level interrupt is enabled. Whenever SSSR_x[RFS] is set, an interrupt request is made to the interrupt controller.

8.5.3 SSP Programmable Serial Protocol Register (SSPSP_x)

SSPSP_x, shown in Table 8-8, contains bit fields to program the various programmable serial-protocol parameters.

The contents of SSPSP_x are ignored if PSP mode is not selected.

These are read/write registers. Ignore reads from reserved bits. Write 0b0 to reserved bits.

Table 8-8. SSPSP1/2/3 Bit Definitions (Sheet 1 of 2)

Physical Address	SSPSP_1 / SSPSP_2 / SSPSP_3	SSP Controller
0x4100_002C 0x4170_002C 0x4190_002C		

User Settings																															
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						FSRT	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	0

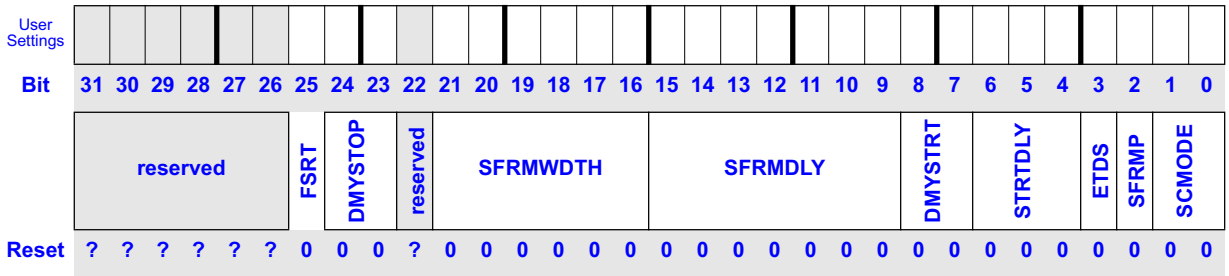
Bits	Access	Name	Description
31:26	—	—	reserved
25	R/W	FSRT	Frame Sync Relative Timing 0 = Next frame is asserted after the end of the T4 timing 1 = Next frame is asserted with the LSB of the previous frame **NOTE:** When FSRT is set, SSPSFRMx corresponding to the next sample is asserted during the transmission of the LSB from the current sample (see Figure 8-12).
24:23	R/W	DMYSTOP	Dummy Stop DMYSTOP determines the number of cycles that SSPSCLKx is active following the last bit (bit 0) of transmitted data (SSPTXDx) or received data (SSPRXDx). The value must be from 0 to 3. DMYSTOP must be cleared when PSP format is used in network mode and/or when FSRT is set.
22	—	—	reserved
21:16	R/W	SFRMWDTH	Serial Frame Width SFRMWDTH determines the number of SSPSCLKx cycles that SSPSFRMx is active. The programmed value must not be asserted past the end of DMYSTOP (T4 in Figure 8-10). The value must be from 1 to 44. In PSP slave mode (SSCR1_x[SFRMDIR] is set), SFRMWDTH is ignored. The incoming SSPSFRMx must be asserted for a duration of at least one SSPSCLKx cycle for each sample. The incoming SSPSFRMx and first data bit of the sample can be asserted at the same time. Between samples, the incoming SSPSFRMx must be deasserted for a duration of at least one SSPSCLKx cycle.
15:9	R/W	SFRMDLY	Serial Frame Delay SFRMDLY determines the number of half SSPSCLKx cycles that SSPSFRMx is delayed from the start of the transfer to the time SSPSFRMx is asserted. The value must be from 0 to 88.
8:7	R/W	DMYSTRT	Dummy Start DMYSTRT determines the number of SSPSCLKx cycles after STRTDLY and before transmitted data (SSPTXDx) or received data (SSPRXDx).

Table 8-8. SSPSP1/2/3 Bit Definitions (Sheet 2 of 2)

Physical Address
0x4100_002C
0x4170_002C
0x4190_002C

SSPSP_1
SSPSP_2
SSPSP_3

SSP Controller



Bits	Access	Name	Description
6:4	R/W	STRTDLY	Start Delay STRTDLY determines the number of cycles that SSPSCLKx remains in its Idle state between data transfers. The STRTDLY field must be cleared if the SSPSCLKENx, SSCR1_x[ECRA], or SSCR1_x[ECRB] clock enables are used. The STRTDLY field must be cleared whenever SSPSCLKx or SSPSFRMx is configured as an input (SSCR1_x[SCLKDIR] or SSCR1_x[SFRMDIR] are set). The value must be from 0 to 7.
3	R/W	ETDS	End-of-Transfer Data State ETDS determines the state of SSPTXDx at the end of a transfer. When cleared, the state of SSPTXDx is forced low after the LSB of the frame is sent and remains low through the next idle period. When set, the state of SSPTXDx retains the value of the LSB through the next idle period. **NOTE:** ETDS has no effect if SSCR1_x[TTE] is set. **NOTE:** ETDS bit has no effect when configured in TI Synchronous Serial Protocol. 0 = Low 1 = Last Value <Bit 0>
2	R/W	SFRMP	Serial Frame Polarity SFRMP determines the active state of SSPSFRMx. In idle mode or when the SSP port is disabled, SSPSFRMx is in its inactive state. In slave mode (SSCR1_x[SFRMDIR] is set), SFRMP indicates the polarity of the incoming SSPSFRMx. 0 = SSPSFRMx is active low. 1 = SSPSFRMx is active high.
1:0	R/W	SCMODE	Serial Bit-Rate Clock Mode SCMODE selects one of four serial clock modes when PSP format is used (SSCR0_x[FRF] = 0b11). Its operation is similar to how SSCR1_x[SPO] and SSCR1_x[SPH] together determine the idle state of SSPSCLKx 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) **NOTE:** For all selections of SCMODE, the Idle State is high impedance when SSCR1_x[TIE] is set.

8.5.4 SSP Time-Out Register (SSTO_x)

SSTO_x, shown in Table 8-9, specifies the time-out value to signal a period of inactivity within the receive FIFO.

SSTO_x are read/write registers. Ignore reads from reserved bits. Write 0b0 to reserved bits.

Table 8-9. SSTO_1/2/3 Bit Definitions

Physical Address		SSTO_1 SSTO_2 SSTO_3	SSP Controller
0x4100_0028 0x4170_0028 0x4190_0028			

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

8.5.5 SSP Interrupt Test Register (SSITR_x)

SSITR_x, shown in Table 8-10, contains bit fields used for testing purposes only.

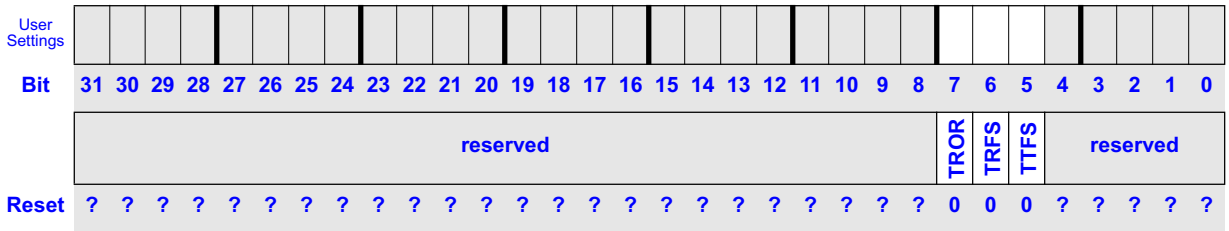
Setting bits in this register causes the SSP port controller to generate interrupts and DMA requests if they are enabled. This is useful in testing the port's functionality.

Setting any of these bits also causes corresponding status bits to be set in SSSR_x. The interrupt or DMA service request, caused by the setting of one of these bits remains active until the bit is cleared. This register must be 0 for normal operation.

These are read/write registers. Ignore reads from reserved bits. Write 0b0 to reserved bits.

Table 8-10. SSITR1/2/3 Bit Definitions

Physical Address		SSITR_1		SSP Controller
0x4100_000C		SSITR_2		
0x4170_000C		SSITR_3		



Bits	Access	Name	Description
31:8	—	—	reserved
7	R/W	TROR	Test RX FIFO Overrun 0 = No receive FIFO overrun DMA service request is generated. 1 = Generates a non-maskable interrupt to the CPU. No DMA request is generated. Write 0b0 to clear.
6	R/W	TRFS	Test RX FIFO Service Request 0 = No receive FIFO DMA service request is generated. 1 = Generates a non-maskable interrupt to the CPU and a DMA request for the RX FIFO. Write 0b0 to clear.
5	R/W	TTFS	Test TX FIFO Service Request 0 = No transmit FIFO DMA service request is generated. 1 = Generates a non-maskable interrupt to the CPU and a DMA request for the TX FIFO. Write 0b0 to clear.
4:0	—	—	reserved

8.5.6 SSP Status Register (SSSR_x)

SSSR_x, shown in Table 8-11 contains bit fields that signal overrun errors and the transmit and receive FIFO DMA service requests. Each of these hardware-detected events signal an interrupt request to the interrupt controller. The status register also contains flags that indicate:

- When the SSP port is actively transmitting data
- When the TX FIFO is not full
- When the RX FIFO is not empty

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

- End-of-chain
- Receiver time-out
- Peripheral trailing byte
- RX FIFO overrun
- RX FIFO request
- TX FIFO request

An interrupt is signaled as long as the bits are set. The interrupt clears when the bits are cleared. 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 0b1 to a sticky status bit clears it; writing 0b0 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_x[SSE] is cleared. Additionally, some bits that cause interrupts have corresponding mask bits in the control registers.

These are read/write registers. Ignore reads from reserved bits. Write 0b0 to reserved bits.

Table 8-11. SSSR1/2/3 Bit Definitions (Sheet 1 of 5)

Physical Address
0x4100_0008
0x4170_0008
0x4190_0008

SSSR_1
SSSR_2
SSSR_3

SSP Controller

User Settings

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	EOC	TINT	PINT	reserved		RFL				TFL				ROR	RFS	TFS	BSY	RNE	TNF	reserved	
Reset	?	?	?	?	?	?	?	?	0	0	0	0	0	0	?	?	1	1	1	1	0	0	0	0	0	0	0	0	0	1	?	?

Bits	Access	Name	Description
31:24	—	—	reserved
23	R/W†	BCE	Bit Count Error BCE is a read-write status bit that indicates that the SSP port has detected that SSPSFRMx has been asserted at an incorrect time. This causes an interrupt (if enabled by SSCR1_1[BCEI]). When set, BCE indicates that an error occurred and that to get re-synchronized to the master device, the SSP port may disregard both the sample that had SSPDFRMx re-asserted in the middle of it AND the next sample. **NOTE:** BCE does not operate in Motorola SPI mode. **NOTE:** To clear BCE, write 0b1 to it. 0 = The SSP port has not experienced a bit count error 1 = SSPSFRMx has been asserted when the bit counter was not 0
22	R	CSS	Clock Synchronization Status CSS indicates that the SSP port is busy synchronizing the control signals into the SSPSCLKx domain. Software only needs to check CSS when the SSP port is a slave to SSPSFRMx. Software must wait until CSS is cleared before allowing an external device to assert SSPSFRMx. 0 = The SSP port is ready for slave clock operations 1 = The SSP port is currently busy synchronizing slave mode signals

Table 8-11. SSSR1/2/3 Bit Definitions (Sheet 2 of 5)

Physical Address	SSSR_1 / SSSR_2 / SSSR_3	SSP Controller
0x4100_0008 0x4170_0008 0x4190_0008		

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
User Settings																																
	reserved								BCE	CSS	TUR	EOC	TINT	PINT	reserved		RFL				TFL				ROR	RFS	TFS	BSY	RNE	TNF	reserved	
Reset	?	?	?	?	?	?	?	?	0	0	0	0	0	0	?	?	1	1	1	1	0	0	0	0	0	0	0	0	0	1	?	?

Bits	Access	Name	Description
21	R/W†	TUR	Transmit FIFO Underrun TUR indicates that the transmitter tried to send data from the TX FIFO when the TX FIFO was empty. When set, an interrupt is generated to the CPU (that can be locally masked by the SSCR0_x[TIM] bit). Setting TUR does not generate any DMA service request. TUR remains set until cleared by software writing 0b1 to it (which also resets its interrupt request). Writing 0b0 to TUR has no effect. TUR can be set when the SSP port is a slave to SSPSFRMx (SSCR1_x[SFRMDIR] set), or if the SSP port is a master to SSPSFRMx and the SSP port is in network mode. TUR is not set if the SSP port is in receive-without-transmit mode (SSCR1_x[RWOT] set). 0 = TX FIFO has not experienced an underrun 1 = Transmitter tried to send data from the TX FIFO when the FIFO was empty, an interrupt is signaled
20	R/W†	EOC	End Of Chain EOC indicates that the DMA has signaled an end of chain. The end- of-chain event indicates that the DMA descriptor for the RX FIFO is ending. This event requires software intervention if data remains in the RX FIFO. **NOTE:** To clear EOC, write 0b1 to it. 0 = DMA has not signaled an end of chain condition. 1 = DMA has signaled an end of chain condition. An EOC interrupt is only generated when either there are trailing bytes left (the PINT bit is set) or there are no trailing bytes (the TINT bit is set). EOC bit is always set, but does not generate an interrupt if neither of the these conditions are met.
19	R/W†	TINT	Time-Out Interrupt TINT indicates that the RX FIFO has been idle (no samples received) for the period of time defined by the value programmed within SSTO_x. This interrupt can be masked by SSCR1_x[TINTE]. **NOTE:** To clear TINT, write 0b1 to it. 0 = No Receiver Time-out has occurred 1 = Receiver Time-out has occurred
18	R/W†	PINT	Peripheral Trailing Byte Interrupt PINT indicates that a DMA end of chain event has occurred and there is data within the RX FIFO. This event requires the CPU or DMA to transfer the remaining bytes from the RX FIFO (see Section 8.4.2.3). This interrupt can be masked by SSCR1_x[PINTE]. **NOTE:** To clear PINT, write 0b1 to it. 0 = No peripheral trailing byte interrupt is pending. 1 = Peripheral trailing byte interrupt is pending

Table 8-11. SSSR1/2/3 Bit Definitions (Sheet 3 of 5)

Physical Address
0x4100_0008
0x4170_0008
0x4190_0008

SSSR_1
SSSR_2
SSSR_3

SSP Controller

User Settings

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	EOC	TINT	PINT	reserved		RFL				TFL				ROR	RFS	TFS	BSY	RNE	TNF	reserved	
Reset	?	?	?	?	?	?	?	?	0	0	0	0	0	0	?	?	1	1	1	1	0	0	0	0	0	0	0	0	0	1	?	?

Bits	Access	Name	Description
17:16	—	—	reserved
15:12	R	RFL	RX FIFO Level RFL is the number of valid entries (minus 1) currently in the RX FIFO. **NOTE:** When the value of 0xF is read, the RX FIFO is either empty or full and programmers must refer to the RNE bit.
11:8	R	TFL	TX FIFO Level TFL is the number of valid entries currently in the TX FIFO. **NOTE:** When the value of 0x0 is read, the TX FIFO is either empty or full and programmers must refer to the TNF bit.
7	R/W†	ROR	RX FIFO Overrun ROR indicates that the Receive logic attempted to place data into the RX 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 RX FIFO location exists. When set, an interrupt is generated to the CPU that can be locally masked by the SSCR0_x[RIM} bit. Setting ROR does not generate any DMA service request. Clearing ROR resets its interrupt request. **NOTE:** To clear ROR, write 0b1 to it. 0 = RX FIFO has not experienced an overrun 1 = Attempted data write to a full RX FIFO, request an interrupt
6	R	RFS	Receive FIFO Service A RFS request indicates that the RX FIFO requires service to prevent an overrun. RFS is set when the number of valid entries in the RX FIFO is equal to or greater than the RX FIFO trigger threshold. RFS is cleared when the RX FIFO has fewer entries than the trigger threshold. When RFS is set, an interrupt is generated if SSCR1_x[RIE] is set. When RFS is set, a DMA service request is generated if SSCR1_x[RSRE] is set. After the CPU or DMA reads the RX FIFO such that it has fewer entries than the value of SSCR1_x[RFT], RFS (and the service request and/or interrupt) is automatically cleared. SSCR1_x[RSRE] and SSCR1_x[RIE] must not both be set. 0 = RX FIFO level is less than its trigger threshold or the SSP port is disabled 1 = RX FIFO level is equal to or above its trigger threshold, an interrupt or DMA service request is generated.

Table 8-11. SSSR1/2/3 Bit Definitions (Sheet 4 of 5)

Physical Address		SSSR_1 SSSR_2 SSSR_3	SSP Controller
0x4100_0008 0x4170_0008 0x4190_0008			

User Settings																															
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	EOC	TINT	PINT	reserved		RFL				TFL				ROR	RFS	TFS	BSY	RNE	TNF	reserved	
Reset	?	?	?	?	?	?	?	?	0	0	0	0	0	0	?	?	1	1	1	1	0	0	0	0	0	0	0	0	0	1	?	?

Bits	Access	Name	Description
5	R	TFS	Transmit FIFO Service A TFS request indicates that the TX FIFO requires service to prevent an underrun. TFS is set when the number of valid entries in the TX FIFO is equal to or less than the TX FIFO trigger threshold. TFS is cleared when the TX FIFO has more entries than the trigger threshold. When TFS is set, an interrupt is generated if SSCR1_x[TIE] is set. When TFS is set, a DMA service request is generated if SSCR1_x[TSRE] is set. After the CPU or DMA fills the TX FIFO such that it has more entries than the value of SSCR1_x[TFT], TFS (and the service request and/or interrupt) is automatically cleared. SSCR1_x[TSRE] and SSCR1_x[TIE] must not both be set. 0 = TX FIFO level exceeds its threshold (TFT + 1) or the SSP port is disabled 1 = TX FIFO level is at or below its trigger threshold (TFT + 1), an interrupt or DMA service request is generated.
4	R	BSY	Busy BSY is automatically set when the SSP port is actively transmitting and/or receiving data and BSY is automatically cleared when the SSP port is idle or disabled. BSY does not generate an interrupt. **NOTE:** When the SSP port is a master of a clock, software determines if the SSP port is active by monitoring SSSR_x[TFL] and SSSR_x[BSY]. If the SSP port is a slave to a clock, software determines if the SSP port is active by monitoring SSSR_x[TFL], SSSRx[RFL] and SSSR_x[BSY] along with the SSTO_x register. Also, using the time-out feature (the SSTO_x register and SSCR1_x[TRAIL]) to handle trailing bytes provides an indication of when the master has completed sending data. 0 = SSP port is idle or disabled 1 = SSP port is actively transmitting or receiving data
3	R	RNE	RX FIFO Not Empty RNE indicates that the RX FIFO contains one or more entries of valid data. RNE is automatically cleared when the RX FIFO no longer contains any valid data. This bit does not generate an interrupt. When using programmed I/O, RNE can be polled to remove remaining bytes of data from the RX FIFO since CPU interrupt requests are made only when the RX FIFO trigger threshold has been met or exceeded. 0 = RX FIFO is empty. 1 = RX FIFO is not empty.

Table 8-11. SSSR1/2/3 Bit Definitions (Sheet 5 of 5)

Physical Address	SSSR_1 / SSSR_2 / SSSR_3	SSP Controller
0x4100_0008 0x4170_0008 0x4190_0008		

User Settings																															
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**	**EOC**	**TINT**	**PINT**	**reserved**		**RFL**				**TFL**				**ROR**	**RFS**	**TFS**	**BSY**	**RNE**	**TNF**	**reserved**	
Reset	?	?	?	?	?	?	?	?	0	0	0	0	0	0	?	?	1	1	1	1	0	0	0	0	0	0	0	0	0	1	?	?

Bits	Access	Name	Description
2	R	TNF	TX FIFO Not Full TNF indicates that the TX FIFO contains one or more entries that do not contain valid data. TNF is automatically cleared when the TX FIFO is completely full. TNF does not generate an interrupt. When using programmed I/O, TNF can be polled to fill the TX FIFO beyond its trigger threshold. 0 = TX FIFO is full 1 = TX FIFO is not full
1:0	—	—	reserved

† Write 0b1 to clear this bit.

8.5.7 SSP Data Register (SSDR_x)

SSDR_x, shown in Table 8-12, is a single address location that is accessed by both read and write data transfers. Each SSDR register represents two physical registers: the first register provides temporary storage for data on its way to the TX FIFO and the 2nd register provides temporary storage for data coming from the RX FIFO.

As the CPU or DMA accesses the SSDR_x registers, FIFO control logic transfers data automatically between the registers and FIFOs as fast as the CPU or DMA moves it. Data in the FIFOs shift up or down to accommodate new word(s), unless attempting a write to a full transmit FIFO. Status bits (such as SSSR_x[TFL, RFL, TNF, RNE]) show if the FIFO is full, above the programmable trigger threshold, below the programmable trigger threshold, or empty.

For transmit data, SSDR_x can be loaded (written) by the processor (using programmed I/O or DMA) anytime the TX FIFO falls below its trigger threshold.

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

When the SSP port is programmed for the Microwire protocol and the size of the transmit data is eight bits (SSCR1_x[MWDS] cleared), the most significant 24-bits are ignored. Similarly, if the size for the Transmit data is 16-bits (SSCR1_x[MWDS] set), the most significant 16-bits are ignored. SSCR0_x[DSS] controls the Receive data size.

Both the TX and RX FIFOs are cleared when the SSP port is reset, or by clearing SSCR0_x[SSE].

Table 8-12. SSDR1/2/3 Bit Definitions

Physical Address		SSDR_1 SSDR_2 SSDR_3	SSP Controller
0x4100_0010 0x4170_0010 0x4190_0010			

User Settings

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	Access	Name	Description
31:0	R/W	DATA	Data to be written to/read from the TX/RX FIFO, respectively.

8.5.8 SSP TX Time Slot Active Register (SSTSA_x)

SSTSA_x are read-write registers that indicate in which time slot the SSP port transmits data. SSTSA_x are ignored if the SSP port is not in network mode (SSCR0_x[MOD] = 1). See Figure 8-3 for an example of using time slots only when in network mode.

These are read/write registers. Ignore reads from reserved bits. Write 0b0 to reserved bits.

Table 8-13. SSTSA1/2/3 Bit Definitions

Physical Address		SSTSA_1 SSTSA_2 SSTSA_3	SSP Controller
0x4100 0030 0x4170 0030 0x4190 0030			

User Settings

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																								TTSA							
Reset	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0

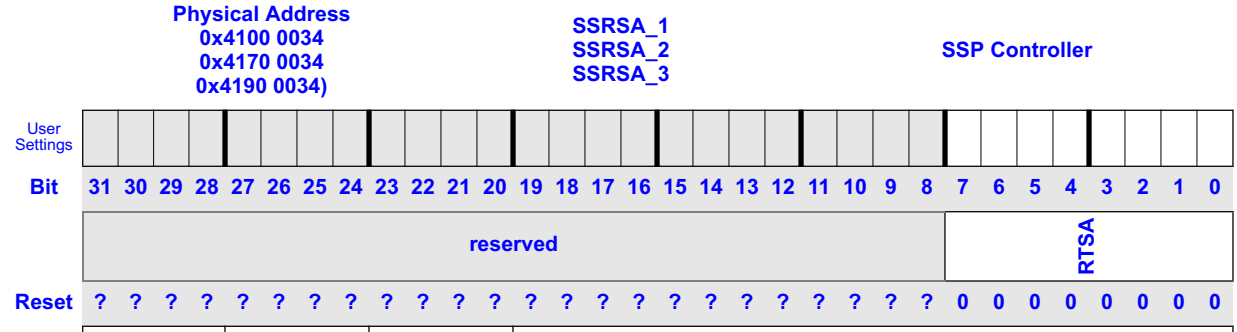
Bits	Access	Name	Description
31:8	—	—	Reserved
7:0	R/W	TTSA	TX Time Slot Active Only if the SSP port is in network mode, the 8 TTSA bits indicate in which of 8 associated time slots the SSP port transmits. Each TTSA bit selects one time slot, respectively. Time slot bits beyond the SSCR0_x[FRDC] value are ignored (if SSCR0_x[FRDC] = 0b011 to select 4 time slots, then TTSA bits 7:4 are ignored). If SSCR1_x[TTE] is set, then the SSP port causes SSPTXD to be high impedance during time slots where associated TTSA bits are programmed to 0. 0 = SSP port does not transmit data in this time slot 1 = SSP port transmits data in this time slot

8.5.9 SSP RX Time Slot Active Register (SSRSA_x)

SSRSA_x are read-write registers that indicate in which time slots the SSP port receives data. SSRSA_x are ignored if the SSP port is not in network mode. See Figure 8-3 for an example of using time slots only when in network mode.

These are read/write registers. Ignore reads from reserved bits. Write 0b0 to reserved bits.

Table 8-14. SSRSA1/2/3 Bit Definitions



Bits	Access	Name	Description
31:8	—	—	reserved
7:0	R/W	RTSA	RX Time Slot Active Only if the SSP port is in network mode, the 8 RTSA bits indicate in which of 8 associated time slots the SSP port receives data. Each RTSA bit selects one time slot, respectively. Time slot bits beyond the SSCR0_x[FRDC] value is ignored (if SSCR0_x[FRDC] = 0b011 to select 4 time slots, then RTSA bits 7:4 are ignored). 0 = SP port does not receive data in this time slot 1 = SSP port receives data in this time slot

8.5.10 SSP Time Slot Status Register (SSTSS_x)

These registers indicate which time slot the SSP port is currently in. SSTSS_x are ignored when the SSP port is not in network mode.

These are read-only registers. Ignore reads from reserved bits. Write 0b0 to reserved bits.

Table 8-15. SSTS1/2/3 Bit Definitions

Physical Address	SSTSS_1 SSTSS_2 SSTSS_3	SSP Controller
0x4100 0038) 0x4170 0038 0x4190 0038		

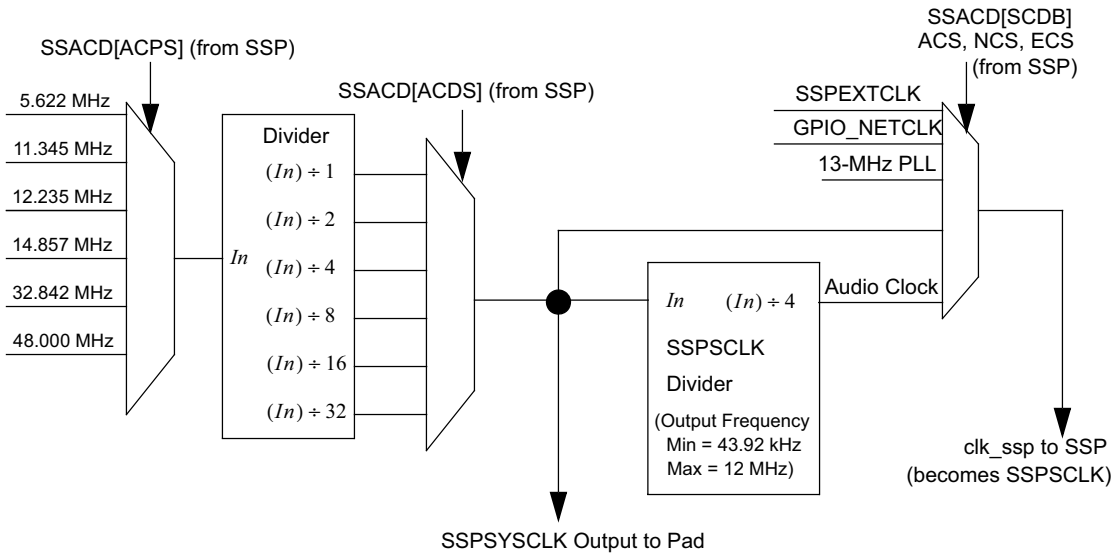
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NMBSY	reserved																												TSS		
Reset	0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0

Bits	Access	Name	Description
31	R	NMBSY	Network Mode Busy Only if the SSP port is in network mode, NMBSY indicates when the SSP port is in the middle of a frame. NMBSY can be used by software when a clean shutdown of the SSP port is needed. Software must ensure that the TX FIFO is either empty (or will be empty at the end of the next frame), deactivate the TX DMA requests, clear the SSCR0_x[MOD] bit; then software must poll NMBSY until it is 0 before disabling the SSP port (by clearing the SSCR0_x[SSE] bit). When the SSP port is a master of SSPSFRMx, NMBSY is set. If the SSP port is a slave to SSPSFRMx, NMBSY is set only if the current frame (number of bits per sample number of time slots per frame) has not expired since SSPSFRMx was asserted. 0 = No SSPSFRMx is currently asserted (network mode only) 1 = SSPSFRMx is currently asserted (network mode only)
30:3	—	—	reserved
2:0	R	TSS	Time Slot Status Only if the SSP port is in network mode, the 3-bit TSS value indicates which time slot the SSP port is in. Due to synchronization delays between clock domains, the TSS value is a delayed version of the actual time slot.

8.5.11 SSP Audio Clock Divider Register (SSACD_x)

SSACD_x select which clock frequency is sent to the SSP port and then to SSPSYSCLKx and SSPSCLKx. If SSCR0_x[SCR] is not 0, there is no guaranteed phase relationship between SSPSYSCLKx and SSPSCLKx. SSPSYSCLKx's frequency (see Figure 8-20) is calculated by dividing the chosen PLL output clock frequency (SSACD_x[ACPS]) by the chosen divider (SSACD_x[ACDS]). SSPSYSCLKx is then divided by 4 (or by 1) to get SSPSCLKx (see Figure 8-20). SSPSFRMx's frequency is calculated by dividing SSPSCLKx by the multiply-product of data size (SSCR0_x[EDSS, DSS] values) times the number of time slots being used (SSCR0_x[FRDC] value).

Figure 8-20. Audio Clock Selection



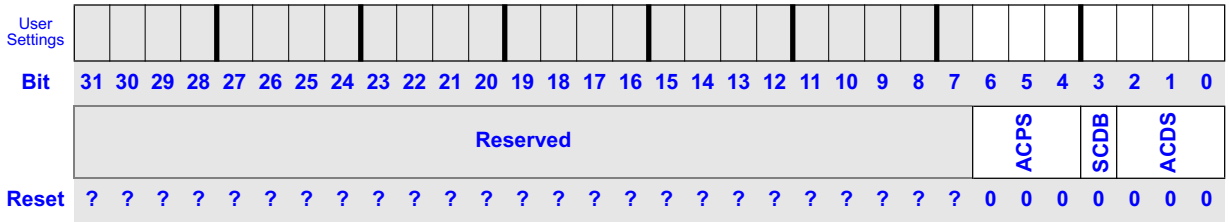
These are read/write registers. Ignore reads from reserved bits. Write 0b0 to reserved bits.

Table 8-16. SSACD1/2/3 Bit Definitions

Physical Address
0x4100 003C
0x4170 003C)
0x4190 003C

SSACD_1
SSACD_2
SSACD_3

SSP Controller



Bits	Access	Name	Description
31:7	—	—	Reserved
6:4	R/W	ACPS	Audio Clock PLL Select The ACPS value indicates which PLL output clock is sent to the clock divider. See Figure 8-20, Table 8-17, and Table 8-18. Some combinations of ACPS and ACDS are not valid (see Table 8-17). ACPS Value: PLL Output Frequency 0b000: 5.622 MHz 0b001: 11.345 MHz 0b010: 12.235 MHz 0b011: 14.857 MHz 0b100: 32.842 MHz 0b101: 48.000 MHz 0b110: Reserved 0b111: Reserved
3	R/W	SCDB	SSPSYSCLK Divider Bypass If SCDB is set and SSCR0_x[ACS] is set, SSPSYSCLKx is divided by 1 to become SSPSCLKx. If SCDB is cleared and SSCR0_x[ACS] is set), SSPSYSCLKx is divided by 4 to become SSPSCLKx. If SSCR0_x[ACS] is cleared, SCDB has no effect. 0 = SSPSYSCLKx is divided by 4 to become SSPSCLKx 1 = SSPSYSCLKx is divided by 1 to become SSPSCLKx
2:0	R/W	ACDS	Audio Clock Divider Select The ACDS value indicates which divider creates SSPSYSCLKx. See Figure 8-20, Table 8-17, and Table 8-18. Some combinations of ACPS and ACDS are not valid (see Table 8-17). ACDS Value: Clock Divider Value 0b000: 1 0b001: 2 0b010: 4 0b011: 8 0b100: 16 0b101: 32 0b110: Reserved 0b111: Reserved

Table 8-17. SSPSYSCLKx Frequency Selection

PLL Output Frequency	ACPS Value	ACDS Value					
		1	2	4	8	16	32
5.622MHz	0b000	5.622 MHz	2.811 MHz	1.405 MHz	702.7 kHz	3.514 kHz	175.7 kHz
11.345MHz	0b001	11.345 MHz	5.673 MHz	2.836 MHz	1.418 MHz	709.1 kHz	354.5 kHz
12.235MHz	0b010	12.235 MHz	6.118 MHz	3.059 MHz	1.529 MHz	764.7 kHz	382.4 kHz
14.857MHz	0b011	14.857MHz	7.429 MHz	3.714 MHz	1.857 MHz	928.6 kHz	464.3 kHz
32.842MHz	0b100	not valid	16.421MHz	8.211 MHz	4.105 MHz	2.053 MHz	1.026 MHz
48.00MHz	0b101	not valid	not valid	12.00 MHz	6.000 MHz	3.000 MHz	1.500 MHz

Table 8-18. PLL Output Frequency and Divider Selection (Selected Time Slots and Data Sizes)

PLL Output Frequency	ACPS Value	SCDB	Divider Value (ACDS choice)												Actual SSPSFRMx Frequency	Closest Standard for SSPSFRMx Frequency
			# of Time Slots for 8 bits/sample				# of Time Slots for 16 bits/sample				# of Time Slots for 32 bits/sample					
			1	2	4	8	1	2	4	8	1	2	4	8		
12.235 MHz	0b010	0	8	4	2	1	4	2	1	-	2	1	-	-	47.79 kHz	48.00 kHz
11.345 MHz	0b001	0	8	4	2	1	4	2	1	-	2	1	-	-	44.32 kHz	44.10 kHz
5.622 MHz	0b000	0	8	4	2	1	4	2	1	-	2	1	-	-	21.96 kHz	22.05 kHz
32.842 MHz	0b100	0	-	32	16	8	32	16	8	4	16	8	4	2	16.04 kHz	16.00 kHz
5.622 MHz	0b000	0	16	8	4	2	8	4	2	1	42	2	1	-	10.98 kHz	11.025 kHz
11.345 MHz	0b001	0	-	-	-	-	-	-	-	-	-	-	-	1	11.08 kHz	11.025 kHz
32.842 MHz	0b100	0	-	-	32	16	-	32	16	8	32	16	8	4	8.02 kHz	8.00 kHz
12.235 MHz	0b010	1	-	-	-	-	-	-	-	2	-	-	2	1	47.79 kHz	48.00 kHz
11.345 MHz	0b001	1	-	-	-	-	-	-	-	2	-	-	2	1	44.32 kHz	44.10 kHz
5.622 MHz	0b000	1	-	-	-	-	-	-	-	2	-	-	2	1	21.96 kHz	22.05 kHz

Note: Table 8-18 shows the recommended divider and PLL clock selection to approximate standard frame frequencies for a selected combination of bit sizes and time slots. Use the following formulas to calculate other combinations (use the second equation if SSPSYSCLKx is to be 4x SSPSCLKx):

$$TimeSlots \times BitsPerSample \times StandardFrequency = SSPSCLKFrequency$$
$$SSPSCLKFrequency \times 4 = SSPSYSCLKFrequency$$

Choose divider and PLL clock output to approximate SSPSYSCLK frequency

8.6 Register Summary

Table 8-19 summarizes the SSP port registers and their physical addresses.

Table 8-19. SSP Register Summary (Sheet 1 of 2)

Physical Address	Name	Description	Page
0x4100_0000	SSCR0_1	SSP 1 Control register 0	8-25
0x4100_0004	SSCR1_1	SSP 1 Control register 1	8-29
0x4100_0008	SSSR_1	SSP 1 Status register	8-43

Table 8-19. SSP Register Summary (Sheet 2 of 2)

Physical Address	Name	Description	Page
0x4100_000C	SSITR_1	SSP 1 Interrupt Test register	8-42
0x4100_0010	SSDR_1	SSP 1 Data Write register/Data Read register	8-48
0x4100_0014–0x4100_0024	—	reserved	
0x4100_0028	SSTO_1	SSP 1 Time-Out register	8-41
0x4100_002C	SSPSP_1	SSP 1 Programmable Serial Protocol	8-39
0x4100_0030	SSTSA_1	SSP1 TX Timeslot Active register	8-48
0x4100_0034	SSRSA_1	SSP1 RX Timeslot Active register	8-49
0x4100_0038	SSTSS_1	SSP1 Timeslot Status register	8-50
0x4100_003C	SSACD_1	SSP1 Audio Clock Divider register	8-51
0x4100_0040–0x416F_FFFC	—	reserved	
0x4170_0000	SSCR0_2	SSP2 Control register 0	8-25
0x4170_0004	SSCR1_2	SSP 2 Control register 1	8-29
0x4170_0008	SSSR_2	SSP 2 Status register	8-43
0x4170_000C	SSITR_2	SSP 2 Interrupt Test register	8-42
0x4170_0010	SSDR_2	SSP 2 Data Write register/Data Read register	8-48
0x4170_0014–0x4170_0024	—	reserved	
0x4170_0028	SSTO_2	SSP 2 Time-Out register	8-41
0x4170_002C	SSPSP_2	SSP 2 Programmable Serial Protocol	8-39
0x4170_0030	SSTSA_2	SSP2 TX Timeslot Active register	8-48
0x4170_0034	SSRSA_2	SSP2 RX Timeslot Active register	8-49
0x4170_0038	SSTSS_2	SSP2 Timeslot Status register	8-50
0x4170_003C	SSACD_2	SSP2 Audio Clock Divider register	8-51
0x4170_0040–0x418F_FFFC	—	reserved	
0x4190_0000	SSCR0_3	SSP 3 Control register 0	8-25
0x4190_0004	SSCR1_3	SSP 3 Control register 1	8-29
0x4190_0008	SSSR_3	SSP 3 Status register	8-43
0x4190_000C	SSITR_3	SSP 3 Interrupt Test register	8-42
0x4190_0010	SSDR_3	SSP 3 Data Write register/Data Read register	8-48
0x4190_0014–0x4190_0024	—	reserved	
0x4190_0028	SSTO_3	SSP 3 Time-Out register	8-41
0x4190_002C	SSPSP_3	SSP 3 Programmable Serial Protocol	8-39
0x4190_0030	SSTSA_3	SSP TX Timeslot Active register	8-48
0x4190_0034	SSRSA_3	SSP RX Timeslot Active register	8-49
0x4190_0038	SSTSS_3	SSP Timeslot Status register	8-50
0x4190_003C	SSACD_3	SSP Audio Clock Divider register	8-51
0x4190_0040–0x419f_FFFC	—	reserved	