# Chapter 40
# Deserial Serial Peripheral Interface (DSPI)

## 40.1 DSPI and SPI configuration

This section describes the configuration of these modules specific to the MPC5748G.

### 40.1.1 DSPI configuration

The device contains four DSPI[0:3] modules. DSPI_2 and DSPI_3 are targeted for faster frequency.

**Table 40-1. Full Featured DSPI[1:0] configuration**

| Feature | | Available |
|---|---|---|
| Extended (32-bit) SPI Mode Support | | Yes |
| CSI and DSI mode support. | | Yes |
| 64 bit DSI Mode Support | | Yes |
| TX FIFO | | 4 |
| RX FIFO | | 4 |
| CTAR registers (transfer attribute configuration) | | 6 |
| TSB Mode | | Yes |
| PCS Signals | DSPI0 | DSPI0_PCS[0:5] |
| | DSPI1 | DSPI1_PCS[0:4] |

**Table 40-2. Full Featured DSPI[3:2] configuration**

| Feature | Available |
|---|---|
| Extended (32-bit) SPI Mode Support | Yes |
| CSI and DSI mode support. | Yes |
| 64 bit DSI Mode Support | No |
| TX FIFO | 4 |
| RX FIFO | 4 |

*Table continues on the next page...*

**Table 40-2. Full Featured DSPI[3:2] configuration (continued)**

| | | |
|---|---|---|
| CTAR registers (transfer attribute configuration) | | 6 |
| TSB Mode | | Yes |
| PCS Signals | DSPI2 | DSPI2_PCS[0:3] |
| | DSPI3 | DSPI3_PCS[0:1] |

## 40.1.2 SPI configuration

The device contains 6 SPI[0:5] modules. SPI_1and SPI_2 are targeted for faster frequency.

**Table 40-3. SPI[0:5] Configuration**

| Feature | | Available |
|---|---|---|
| 32-bit support | | Yes |
| Extended (32-bit) SPI Mode Support | | Yes |
| CSI and DSI mode support. | | No |
| 64 bit DSI Mode Support | | No |
| TX FIFO | | 4 |
| RX FIFO | | 4 |
| CTAR registers (transfer attribute configuration) | | 6 |
| TSB Mode | | No |
| PCS Signals | SPI0 | SPI0_PCS[0:3] |
| | SPI1 | SPI1_PCS[0:2] |
| | SPI2 | SPI2_PCS[0:3] |
| | SPI3 | SPI3_PCS[0:3] |
| | SPI4 | SPI4_PCS[0] |
| | SPI5 | SPI5_PCS[0] |

## 40.1.3 DSPI and SPI registers

The following table compares various registers available in DSPI and SPI modules in this device.

| DSPI registers | SPI registers |
|---|---|
| Module Configuration Register (MCR) | Module Configuration Register (MCR) |
| Transfer Count Register (TCR) | Transfer Count Register (TCR) |

*Table continues on the next page...*

| DSPI registers | SPI registers |
|---|---|
| Clock and Transfer Attributes Register (In Master Mode) (CTARn) | Clock and Transfer Attributes Register (In Master Mode) (CTARn) |
| Clock and Transfer Attributes Register (In Slave Mode) (CTARn_SLAVE) | Clock and Transfer Attributes Register (In Slave Mode) (CTARn_SLAVE) |
| Status Register (SR) | Status Register (SR) |
| DMA/Interrupt Request Select and Enable Register (RSER) | DMA/Interrupt Request Select and Enable Register (RSER) |
| PUSH TX FIFO Register In Master Mode | PUSH TX FIFO Register In Master Mode |
| PUSH TX FIFO Register In Slave Mode (PUSHR_SLAVE) | PUSH TX FIFO Register In Slave Mode (PUSHR_SLAVE) |
| POP RX FIFO Register (POPR) | POP RX FIFO Register (POPR) |
| Transmit FIFO Registers (TXFRn) | Transmit FIFO Registers (TXFRn) |
| Receive FIFO Registers (RXFRn) | Receive FIFO Registers (RXFRn) |
| DSI Configuration Register 0 (DSICR0) | |
| DSI Serialization Data Register 0 (SDR0) | |
| DSI Alternate Serialization Data Register 0 (ASDR0) | |
| DSI Transmit Comparison Register 0 (COMPR0) | |
| DSI Deserialization Data Register 0 (DDR0) | |
| DSI Configuration Register 1 (DSICR1) | |
| DSI Serialization Source Select Register 0 (SSR0) | |
| DSI Parallel Input Select Register 0 (PISR0) | |
| DSI Parallel Input Select Register 1 (PISR1) | |
| DSI Parallel Input Select Register 2 (PISR2) | |
| DSI Parallel Input Select Register 3 (PISR3) | |
| DSI Deserialized Data Interrupt Mask Register 0 (DIMR0) | |
| DSI Deserialized Data Polarity Interrupt Register 0 (DPIR0) | |
| DSI Serialization Data Register 1 (SDR1) | |
| DSI Alternate Serialization Data Register 1 (ASDR1) | |
| DSI Transmit Comparison Register 1 (COMPR1) | |
| DSI Deserialization Data Register 1 (DDR1) | |
| DSI Serialization Source Select Register 1 (SSR1) | |
| DSI Parallel Input Select Register 4 (PISR4) | |
| DSI Parallel Input Select Register 5 (PISR5) | |
| DSI Parallel Input Select Register 6 (PISR6) | |
| DSI Parallel Input Select Register 7 (PISR7) | |
| DSI Deserialized Data Interrupt Mask Register 1 (DIMR1) | |
| DSI Deserialized Data Polarity Interrupt Register 1 (DPIR1) | |
| Clock and Transfer Attributes Register Extended (CTAREn) | Clock and Transfer Attributes Register Extended (CTAREn) |
| Status Register Extended (SREX) | Status Register Extended (SREX) |
| Trigger Register (TRIG) | |

**NOTE**

Before re-configuring DSPI, it should be disabled by setting its MDIS bit. Changing the configuration of DSPI while the module's clock is still running should be avoided.

### 40.1.4 DSPI's stop-stop ack protocol

1. If DSPI is disabled, it acknowledges the stop request immediately.
2. In master mode, DSPI asserts the acknowledgement for the stop request on frame boundary of the frame being transmitted/received. Additionally, if no frame is being transmitted/received DSPI immediately asserts the acknowledgement.
3. In slave mode, DSPI asserts the acknowledgement for stop request when it's slave select is high (inactive value).

**NOTE**

In both 2 and 3, DSPI checks for pending Interrupt/DMA requests before asserting the acknowledgement.

## 40.2 Introduction

The deserial serial peripheral interface (DSPI) module provides a synchronous serial bus for communication between a chip and an external peripheral device. The module supports chip pin count reduction through serialization and deserialization of chip internal signals transmitted over the DSPI serial link.

### 40.2.1 Block Diagram

The block diagram of this module is as follows:

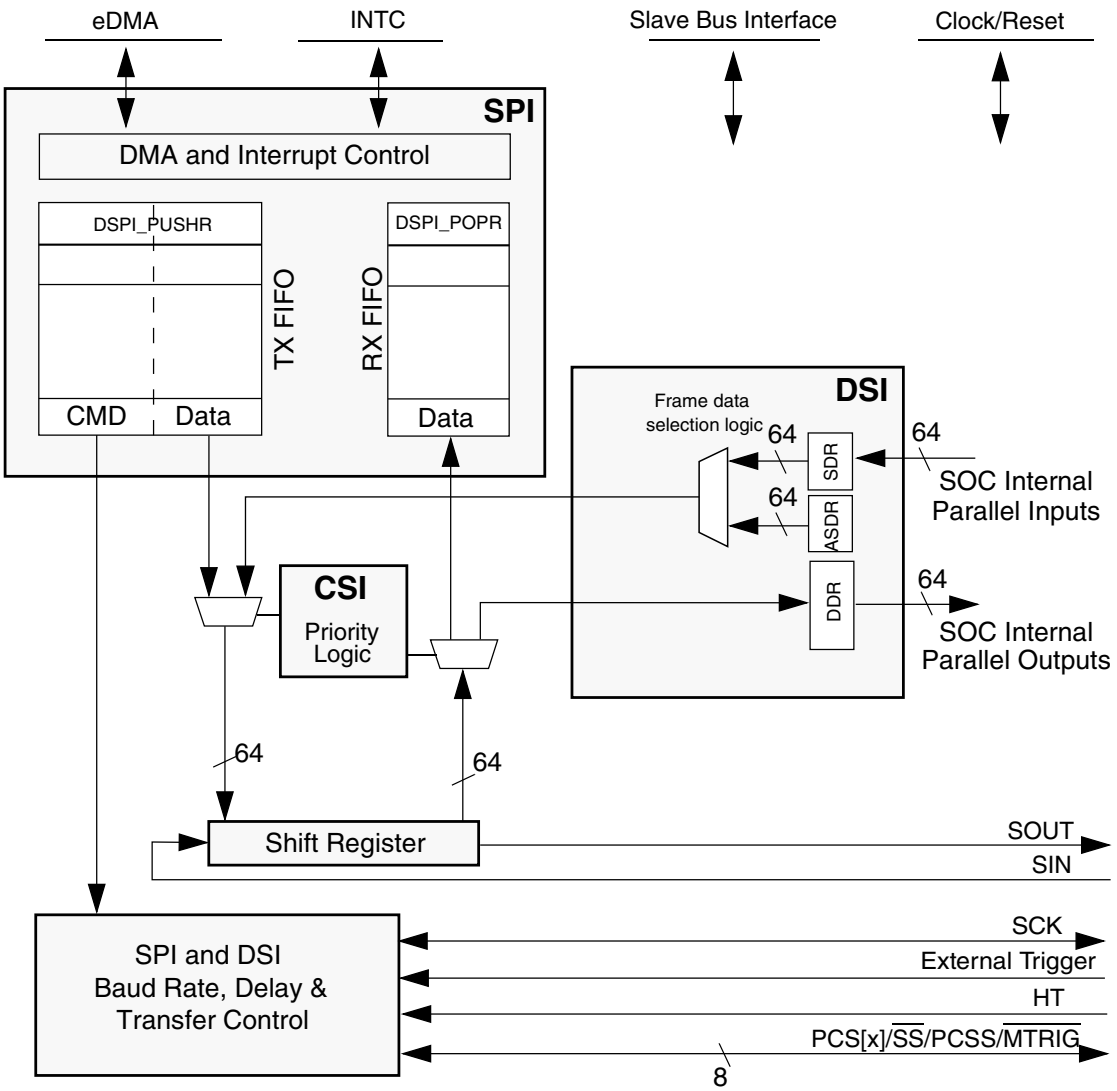



**Figure 40-1. DSPI Block Diagram**

## 40.2.2 Features

The module supports the following features:

- Full-duplex, three-wire synchronous transfers
- Master mode
- Slave mode
- Data streaming operation in Slave mode with continuous slave selection
- Buffered transmit operation using the transmit first in first out (TX FIFO) with depth of 4 entries

- Support for 8/16-bit accesses to the PUSH TX FIFO Register Data Field
- Buffered receive operation using the receive FIFO (RX FIFO) with depth of 4 entries
- Asynchronous clocking scheme for Register and Protocol Interfaces
- TX and RX FIFOs can be disabled individually for low-latency updates to SPI queues
- Visibility into TX and RX FIFOs for ease of debugging
- Programmable transfer attributes on a per-frame basis:
  - six transfer attribute registers
  - six extended transfer attribute registers
  - Serial clock (SCK) with programmable polarity and phase
  - Various programmable delays
  - Programmable serial frame size: 4 to 64 bits
    - SPI frames longer than 32 bits can be supported using the continuous selection format.
  - Continuously held chip select capability
  - Parity control
- 6 peripheral chip selects (PCSes), expandable to 64 with external demultiplexer
- Deglitching support for up to 32 peripheral chip selects (PCSes) with external demultiplexer
- DMA support for adding entries to TX FIFO and removing entries from RX FIFO:
  - TX FIFO is not full (TFFF)
  - RX FIFO is not empty (RFDF)
  - CMD FIFO is not full (CMDFFF)
- Interrupt conditions:
  - End of Queue reached (EOQF)
  - TX FIFO is not full (TFFF)
  - CMD FIFO is not full (CMDFFF)
  - Transfer of current frame complete (TCF)
  - Transfers due from current command frame complete (CMDTCF)

    - Transfer of current SPI frame complete (SPITCF)
    - Attempt to transmit with an empty Transmit FIFO (TFUF)
    - RX FIFO is not empty (RFDF)
    - Frame received while Receive FIFO is full (RFOF)
    - SPI Parity Error (SPEF)
    - Data present in TX FIFO while CMD FIFO is empty (TFIWF)
- Global interrupt request line
- Modified SPI transfer formats for communication with slower peripheral devices
- Power-saving architectural features:
    - Support for Stop mode

The DSPI also supports pin reduction through serialization and deserialization if enabled for the module.

- Two sources of serialized data:
    - The module memory-mapped register
    - Parallel Input signals
    - Programmable selection of source data on bit basis
- Deserialized data is provided as Parallel Output signals and as bits in a memory-mapped register
- Interrupt conditions:
    - Deserialized data matches pre-programmed pattern (DDIF)
    - Transfer of current DSI frame complete (DSITCF)
    - DSI Parity Error (DPEF)
- DMA Request support for following conditions:
    - Deserialized data matches pre-programmed pattern (DDIF)
- Transfer initiation conditions:
    - Continuous
    - Edge sensitive hardware trigger
    - Change in data

- Support for parallel and serial chaining of several modules inside the chip
- Pin serialization/deserialization with interleaved SPI frames for control and diagnostics
- Continuous serial communications clock

DSPI supports a combined SPI and DSI modes of operation known as the Combined Serial Interface (CSI). It also supports the downstream Micro Second Channel in the Timed Serial Bus (TSB) configuration or Interleaved Frames Configuration; configurable by software.

Both TSB and Interleaved TSB modes have the following common features:

- Transmission of frames is done at frame boundaries
- Frames from SPI and DSI are identifiable by a bit transmitted at the start of each frame
- Separate interrupts for frame completion from SPI and DSI

## 40.2.3 Interface configurations

### 40.2.3.1 SPI configuration

The Serial Peripheral Interface (SPI) configuration allows the module to send and receive serial data. This configuration allows the module to operate as a basic SPI block with internal FIFOs supporting external queue operation. Transmitted data and received data reside in separate FIFOs. The host CPU or a DMA controller read the received data from the Receive FIFO and write transmit data to the Transmit FIFO.

For queued operations, the SPI queues can reside in system RAM, external to the module. Data transfers between the queues and the module FIFOs are accomplished by a DMA controller or host CPU. The following figure shows a system example with DMA, DSPI, and external queues in system RAM.

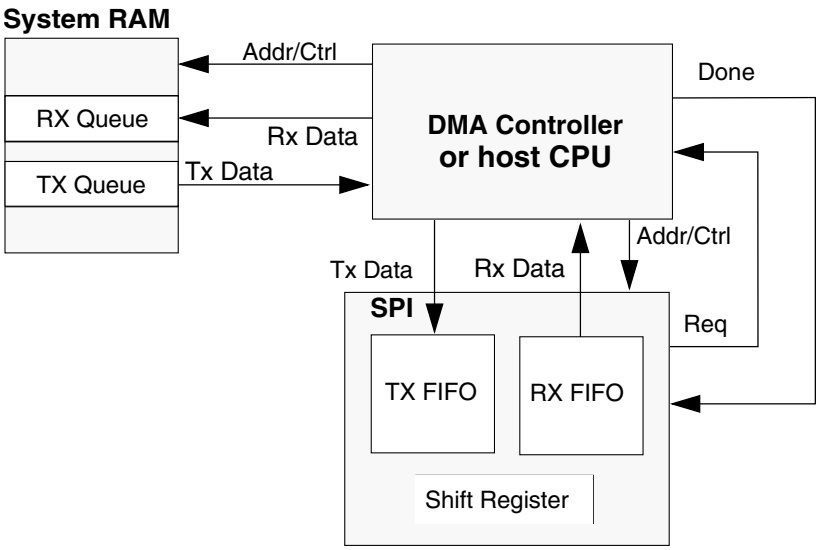



**Figure 40-2. SPI with queues and DMA**

### 40.2.3.2 DSI configuration

In the Deserial Serial Interface (DSI) configuration, DSPI serializes up to 64 parallel input signals or register bits. DSPI also deserializes the received data to parallel output signals or to a memory-mapped register. DSPI transfers the data using an SPI-like protocol.

Timed Serial Bus (TSB) mode provides the MicroSecond Channel (MSC) downstream support, serializing from 4 to 64 parallel input signals or register bits. For more information on TSB mode, see Timed Serial Bus (TSB).

### 40.2.3.3 CSI configuration

The Combined Serial Interface (CSI) configuration supports further configuration as follows:

- Normal: A combination of the SPI and DSI configurations. The DSPI interleaves DSI data frames with SPI data frames. Interleaving is done on the frame boundaries.
- TSB: Transmission of SPI data has higher priority than DSI data.
- Interleaved TSB (ITSB): The frames from SPI and DSI are interleaved without priority. On every trigger, frames from DSI are sent when there are no frames in the SPI or the previous transmission was a frame from SPI. For more information, see Interleaved TSB (ITSB) Mode.

## 40.2.4 Modes of Operation

The module supports the following modes of operation that can be divided into two categories:

- Module-specific modes:
  - Master mode
  - Slave mode
  - Module Disable mode
- Chip-specific modes:
  - External Stop mode
  - Debug mode

The module enters module-specific modes when the host writes a module register. The chip-specific modes are controlled by signals external to the module. The chip-specific modes are modes that a chip may enter in parallel to the block-specific modes.

### 40.2.4.1 Master Mode

Master mode allows the module to initiate and control serial communication. In this mode, these signals are controlled by the module and configured as outputs:

- SCK
- PCS[*x*]

### 40.2.4.2 Slave Mode

Slave mode allows the module to communicate with SPI/DSI bus masters. In this mode, the module responds to externally controlled serial transfers. The SCK signal and the PCS[0]/$\overline{SS}$ signals are configured as inputs and driven by an SPI bus master.

### 40.2.4.3 Module Disable Mode

The Module Disable mode can be used for chip power management. The clock to the non-memory mapped logic in the module can be stopped while in the Module Disable mode.

#### 40.2.4.4 External Stop Mode

External Stop mode is used for chip power management. The module supports the Peripheral Bus Stop mode mechanism. When a request is made to enter External Stop mode, it acknowledges the request and completes the transfer that is in progress. When the module reaches the frame boundary, it signals that the protocol clock to the module may be shut off.

#### 40.2.4.5 Debug Mode

Debug mode is used for system development and debugging. The MCR[FRZ] bit controls module behavior in the Debug mode:

- If the bit is set, the module stops all serial transfers, when the chip is in debug mode.
- If the bit is cleared, the chip debug mode has no effect on the module.

## 40.3 Module signal descriptions

This table describes the signals on the boundary of the module that may connect off chip (in alphabetical order).

**Table 40-4. Module signal descriptions**

| Signal | Master mode | Slave mode | I/O |
|---|---|---|---|
| HT | Hardware Trigger | Hardware Trigger | I |
| PCS0/$\overline{SS}$ | Peripheral Chip Select 0 (O) | Slave Select (I) | I/O |
| PCS[1:3] | Peripheral Chip Selects 1–3 | (Unused) | O |
| PCS4/ $\overline{MTRIG}$ | Peripheral Chip Select 4 | Master Trigger | O |
| PCS5/ $\overline{PCSS}$ | Peripheral Chip Select 5 /Peripheral Chip Select Strobe | (Unused) | O |
| SCK | Serial Clock (O) | Serial Clock (I) | I/O |
| SIN | Serial Data In | Serial Data In | I |
| SOUT | Serial Data Out | Serial Data Out | O |

### 40.3.1 HT—Hardware Trigger

Master mode: Hardware Trigger (I)—Honored only when the module is configured for the DSI or CSI configuration (MCR[DCONF]), the multiple transfer operation is enabled (DSICR0[MTOE]), hardware-trigger reception is enabled (DSICR0[TRRE] or

TRIG[TRRE]), and the module is in the Running state (SR[TXRXS]). Receives hardware triggers that initiate data transfers. DSICR0[TPOL] specifies whether the module uses rising or falling edges of HT as hardware triggers.

Slave mode: Hardware Trigger (I)—Honored only when the module is configured for the DSI or CSI configuration (MCR[DCONF]), the multiple transfer operation is enabled (DSICR0[MTOE]), hardware-trigger reception is enabled (DSICR0[TRRE] or TRIG[TRRE]), and the module is in the Running state (SR[TXRXS]). Receives hardware triggers that initiate the assertion of $\overline{\text{MTRIG}}$. DSICR0[TPOL] specifies whether the module uses rising or falling edges of HT as hardware triggers.

## 40.3.2 PCS0/$\overline{\text{SS}}$—Peripheral Chip Select/Slave Select

Master mode: Peripheral Chip Select 0 (O)—Selects an SPI slave to receive data transmitted from the module.

Slave mode: Slave Select (I)—Selects the module to receive data transmitted from an SPI master.

**NOTE**

Do not tie the DSPI slave select pin to ground. Otherwise, DSPI cannot function properly.

## 40.3.3 PCS1–PCS3—Peripheral Chip Selects 1–3

Master mode: Peripheral Chip Selects 1–3 (O)—Select an SPI slave to receive data transmitted by the module.

Slave mode: Unused

## 40.3.4 PCS4/$\overline{\text{MTRIG}}$—Peripheral Chip Select 4/Master Trigger

Master mode: Peripheral Chip Select 4 (O)—Selects an SPI slave to receive data transmitted by the module.

Slave mode: Master Trigger (O)—Used only when the multiple transfer operation is enabled (DSICR0[MTOE]). Indicates via a pulse that a change in data to be serialized has occurred. The pulse is 4 protocol clock cycles in duration.

### 40.3.5 PCS5/$\overline{\text{PCSS}}$—Peripheral Chip Select 5/Peripheral Chip Select Strobe

Master mode:

- Peripheral Chip Select 5 (O)—Used only when the peripheral-chip-select strobe is disabled (MCR[PCSSE]). Selects an SPI slave to receive data transmitted by the module.
- Peripheral Chip Select Strobe (O)—Used only when the peripheral-chip-select strobe is enabled (MCR[PCSSE]). Strobes an off-module peripheral-chip-select demultiplexer, which decodes the module's PCS signals other than PCS5, preventing glitches on the demultiplexer outputs.

Slave mode: Unused

### 40.3.6 SCK—Serial Clock

Master mode: Serial Clock (O)—Supplies a clock signal from the module to SPI slaves.

Slave mode: Serial Clock (I)—Supplies a clock signal to the module from an SPI master.

### 40.3.7 SIN—Serial Input

Master mode: Serial Input (I)—Receives serial data.

Slave mode: Serial Input (I)—Receives serial data.

### 40.3.8 SOUT—Serial Output

Master mode: Serial Output (O)—Transmits serial data.

Slave mode: Serial Output (O)—Transmits serial data.

## 40.4 Memory Map/Register Definition

Register accesses to memory addresses that are reserved or undefined result in a transfer error. Any Write access to the POPR and RXFRn also results in a transfer error.

**NOTE**

While the module is in the running state, do not write to these registers:

- PISRn
- DIMRn, DPIRn, and SSRn
- CTAREn
- TRIG
- TSL and TS_CONF

## DSPI memory map

| Absolute address (hex) | Register name | Width (in bits) | Access | Reset value | Section/ page |
|---|---|---|---|---|---|
| FBE7_0000 | Module Configuration Register (DSPI_1_MCR) | 32 | R/W | 0000_4001h | 40.4.1/1272 |
| FBE7_0008 | Transfer Count Register (DSPI_1_TCR) | 32 | R/W | 0000_0000h | 40.4.2/1276 |
| FBE7_000C | Clock and Transfer Attributes Register (In Master Mode) (DSPI_1_CTAR0) | 32 | R/W | 7800_0000h | 40.4.3/1276 |
| FBE7_000C | Clock and Transfer Attributes Register (In Slave Mode) (DSPI_1_CTAR0_SLAVE) | 32 | R/W | 7800_0000h | 40.4.4/1281 |
| FBE7_0010 | Clock and Transfer Attributes Register (In Master Mode) (DSPI_1_CTAR1) | 32 | R/W | 7800_0000h | 40.4.3/1276 |
| FBE7_0010 | Clock and Transfer Attributes Register (In Slave Mode) (DSPI_1_CTAR1_SLAVE) | 32 | R/W | 7800_0000h | 40.4.4/1281 |
| FBE7_0014 | Clock and Transfer Attributes Register (In Master Mode) (DSPI_1_CTAR2) | 32 | R/W | 7800_0000h | 40.4.3/1276 |
| FBE7_0018 | Clock and Transfer Attributes Register (In Master Mode) (DSPI_1_CTAR3) | 32 | R/W | 7800_0000h | 40.4.3/1276 |
| FBE7_001C | Clock and Transfer Attributes Register (In Master Mode) (DSPI_1_CTAR4) | 32 | R/W | 7800_0000h | 40.4.3/1276 |
| FBE7_0020 | Clock and Transfer Attributes Register (In Master Mode) (DSPI_1_CTAR5) | 32 | R/W | 7800_0000h | 40.4.3/1276 |
| FBE7_002C | Status Register (DSPI_1_SR) | 32 | R/W | 0201_0000h | 40.4.5/1284 |
| FBE7_0030 | DMA/Interrupt Request Select and Enable Register (DSPI_1_RSER) | 32 | R/W | 0000_0000h | 40.4.6/1288 |
| FBE7_0034 | PUSH TX FIFO Register In Master Mode (DSPI_1_PUSHR) | 32 | R/W | 0000_0000h | 40.4.7/1291 |
| FBE7_0034 | PUSH TX FIFO Register In Slave Mode (DSPI_1_PUSHR_SLAVE) | 32 | R/W | 0000_0000h | 40.4.8/1293 |
| FBE7_0038 | POP RX FIFO Register (DSPI_1_POPR) | 32 | R | 0000_0000h | 40.4.9/1294 |
| FBE7_003C | Transmit FIFO Registers (DSPI_1_TXFR0) | 32 | R | 0000_0000h | 40.4.10/ 1294 |
| FBE7_0040 | Transmit FIFO Registers (DSPI_1_TXFR1) | 32 | R | 0000_0000h | 40.4.10/ 1294 |
| FBE7_0044 | Transmit FIFO Registers (DSPI_1_TXFR2) | 32 | R | 0000_0000h | 40.4.10/ 1294 |
| FBE7_0048 | Transmit FIFO Registers (DSPI_1_TXFR3) | 32 | R | 0000_0000h | 40.4.10/ 1294 |
| FBE7_007C | Receive FIFO Registers (DSPI_1_RXFR0) | 32 | R | 0000_0000h | 40.4.11/ 1295 |
| FBE7_0080 | Receive FIFO Registers (DSPI_1_RXFR1) | 32 | R | 0000_0000h | 40.4.11/ 1295 |

*Table continues on the next page...*

## DSPI memory map (continued)

| Absolute address (hex) | Register name | Width (in bits) | Access | Reset value | Section/ page |
|---|---|---|---|---|---|
| FBE7_0084 | Receive FIFO Registers (DSPI_1_RXFR2) | 32 | R | 0000_0000h | 40.4.11/ 1295 |
| FBE7_0088 | Receive FIFO Registers (DSPI_1_RXFR3) | 32 | R | 0000_0000h | 40.4.11/ 1295 |
| FBE7_00BC | DSI Configuration Register 0 (DSPI_1_DSICR0) | 32 | R/W | 0000_0000h | 40.4.12/ 1296 |
| FBE7_00C0 | DSI Serialization Data Register 0 (DSPI_1_SDR0) | 32 | R | 0000_0000h | 40.4.13/ 1299 |
| FBE7_00C4 | DSI Alternate Serialization Data Register 0 (DSPI_1_ASDR0) | 32 | R/W | 0000_0000h | 40.4.14/ 1299 |
| FBE7_00C8 | DSI Transmit Comparison Register 0 (DSPI_1_COMPR0) | 32 | R | 0000_0000h | 40.4.15/ 1300 |
| FBE7_00CC | DSI Deserialization Data Register 0 (DSPI_1_DDR0) | 32 | R | 0000_0000h | 40.4.16/ 1300 |
| FBE7_00D0 | DSI Configuration Register 1 (DSPI_1_DSICR1) | 32 | R/W | 0000_0000h | 40.4.17/ 1301 |
| FBE7_00D4 | DSI Serialization Source Select Register 0 (DSPI_1_SSR0) | 32 | R/W | 0000_0000h | 40.4.18/ 1303 |
| FBE7_00D8 | DSI Parallel Input Select Register 0 (DSPI_1_PISR0) | 32 | R/W | 0000_0000h | 40.4.19/ 1303 |
| FBE7_00DC | DSI Parallel Input Select Register 1 (DSPI_1_PISR1) | 32 | R/W | 0000_0000h | 40.4.20/ 1304 |
| FBE7_00E0 | DSI Parallel Input Select Register 2 (DSPI_1_PISR2) | 32 | R/W | 0000_0000h | 40.4.21/ 1305 |
| FBE7_00E4 | DSI Parallel Input Select Register 3 (DSPI_1_PISR3) | 32 | R/W | 0000_0000h | 40.4.22/ 1306 |
| FBE7_00E8 | DSI Deserialized Data Interrupt Mask Register 0 (DSPI_1_DIMR0) | 32 | R/W | 0000_0000h | 40.4.23/ 1307 |
| FBE7_00EC | DSI Deserialized Data Polarity Interrupt Register 0 (DSPI_1_DPIR0) | 32 | R/W | 0000_0000h | 40.4.24/ 1307 |
| FBE7_00F0 | DSI Serialization Data Register 1 (DSPI_1_SDR1) | 32 | R | 0000_0000h | 40.4.25/ 1308 |
| FBE7_00F4 | DSI Alternate Serialization Data Register 1 (DSPI_1_ASDR1) | 32 | R/W | 0000_0000h | 40.4.26/ 1309 |
| FBE7_00F8 | DSI Transmit Comparison Register 1 (DSPI_1_COMPR1) | 32 | R | 0000_0000h | 40.4.27/ 1309 |
| FBE7_00FC | DSI Deserialization Data Register 1 (DSPI_1_DDR1) | 32 | R | 0000_0000h | 40.4.28/ 1310 |
| FBE7_0100 | DSI Serialization Source Select Register 1 (DSPI_1_SSR1) | 32 | R/W | 0000_0000h | 40.4.29/ 1311 |
| FBE7_0104 | DSI Parallel Input Select Register 4 (DSPI_1_PISR4) | 32 | R/W | 0000_0000h | 40.4.30/ 1311 |
| FBE7_0108 | DSI Parallel Input Select Register 5 (DSPI_1_PISR5) | 32 | R/W | 0000_0000h | 40.4.31/ 1312 |

*Table continues on the next page...*

## DSPI memory map (continued)

| Absolute address (hex) | Register name | Width (in bits) | Access | Reset value | Section/ page |
|---|---|---|---|---|---|
| FBE7_010C | DSI Parallel Input Select Register 6 (DSPI_1_PISR6) | 32 | R/W | 0000_0000h | 40.4.32/ 1313 |
| FBE7_0110 | DSI Parallel Input Select Register 7 (DSPI_1_PISR7) | 32 | R/W | 0000_0000h | 40.4.33/ 1314 |
| FBE7_0114 | DSI Deserialized Data Interrupt Mask Register 1 (DSPI_1_DIMR1) | 32 | R/W | 0000_0000h | 40.4.34/ 1315 |
| FBE7_0118 | DSI Deserialized Data Polarity Interrupt Register 1 (DSPI_1_DPIR1) | 32 | R/W | 0000_0000h | 40.4.35/ 1315 |
| FBE7_011C | Clock and Transfer Attributes Register Extended (DSPI_1_CTARE0) | 32 | R/W | 0000_0001h | 40.4.36/ 1316 |
| FBE7_0120 | Clock and Transfer Attributes Register Extended (DSPI_1_CTARE1) | 32 | R/W | 0000_0001h | 40.4.36/ 1316 |
| FBE7_0124 | Clock and Transfer Attributes Register Extended (DSPI_1_CTARE2) | 32 | R/W | 0000_0001h | 40.4.36/ 1316 |
| FBE7_0128 | Clock and Transfer Attributes Register Extended (DSPI_1_CTARE3) | 32 | R/W | 0000_0001h | 40.4.36/ 1316 |
| FBE7_012C | Clock and Transfer Attributes Register Extended (DSPI_1_CTARE4) | 32 | R/W | 0000_0001h | 40.4.36/ 1316 |
| FBE7_0130 | Clock and Transfer Attributes Register Extended (DSPI_1_CTARE5) | 32 | R/W | 0000_0001h | 40.4.36/ 1316 |
| FBE7_013C | Status Register Extended (DSPI_1_SREX) | 32 | R | 0000_0000h | 40.4.37/ 1317 |
| FBE7_0140 | Trigger Register (DSPI_1_TRIG) | 32 | R/W | 0000_0000h | 40.4.38/ 1318 |
| FBE7_0150 | Time Slot Length Register (DSPI_1_TSL) | 32 | R/W | 0000_0000h | 40.4.39/ 1319 |
| FBE7_0154 | Time Slot Configuration Register (DSPI_1_TS_CONF) | 32 | R/W | 0000_0000h | 40.4.40/ 1320 |
| FBE7_4000 | Module Configuration Register (DSPI_3_MCR) | 32 | R/W | 0000_4001h | 40.4.1/1272 |
| FBE7_4008 | Transfer Count Register (DSPI_3_TCR) | 32 | R/W | 0000_0000h | 40.4.2/1276 |
| FBE7_400C | Clock and Transfer Attributes Register (In Master Mode) (DSPI_3_CTAR0) | 32 | R/W | 7800_0000h | 40.4.3/1276 |
| FBE7_400C | Clock and Transfer Attributes Register (In Slave Mode) (DSPI_3_CTAR0_SLAVE) | 32 | R/W | 7800_0000h | 40.4.4/1281 |
| FBE7_4010 | Clock and Transfer Attributes Register (In Master Mode) (DSPI_3_CTAR1) | 32 | R/W | 7800_0000h | 40.4.3/1276 |
| FBE7_4010 | Clock and Transfer Attributes Register (In Slave Mode) (DSPI_3_CTAR1_SLAVE) | 32 | R/W | 7800_0000h | 40.4.4/1281 |
| FBE7_4014 | Clock and Transfer Attributes Register (In Master Mode) (DSPI_3_CTAR2) | 32 | R/W | 7800_0000h | 40.4.3/1276 |
| FBE7_4018 | Clock and Transfer Attributes Register (In Master Mode) (DSPI_3_CTAR3) | 32 | R/W | 7800_0000h | 40.4.3/1276 |
| FBE7_401C | Clock and Transfer Attributes Register (In Master Mode) (DSPI_3_CTAR4) | 32 | R/W | 7800_0000h | 40.4.3/1276 |

*Table continues on the next page...*

## DSPI memory map (continued)

| Absolute address (hex) | Register name | Width (in bits) | Access | Reset value | Section/ page |
|---|---|---|---|---|---|
| FBE7_4020 | Clock and Transfer Attributes Register (In Master Mode) (DSPI_3_CTAR5) | 32 | R/W | 7800_0000h | 40.4.3/1276 |
| FBE7_402C | Status Register (DSPI_3_SR) | 32 | R/W | 0201_0000h | 40.4.5/1284 |
| FBE7_4030 | DMA/Interrupt Request Select and Enable Register (DSPI_3_RSER) | 32 | R/W | 0000_0000h | 40.4.6/1288 |
| FBE7_4034 | PUSH TX FIFO Register In Master Mode (DSPI_3_PUSHR) | 32 | R/W | 0000_0000h | 40.4.7/1291 |
| FBE7_4034 | PUSH TX FIFO Register In Slave Mode (DSPI_3_PUSHR_SLAVE) | 32 | R/W | 0000_0000h | 40.4.8/1293 |
| FBE7_4038 | POP RX FIFO Register (DSPI_3_POPR) | 32 | R | 0000_0000h | 40.4.9/1294 |
| FBE7_403C | Transmit FIFO Registers (DSPI_3_TXFR0) | 32 | R | 0000_0000h | 40.4.10/ 1294 |
| FBE7_4040 | Transmit FIFO Registers (DSPI_3_TXFR1) | 32 | R | 0000_0000h | 40.4.10/ 1294 |
| FBE7_4044 | Transmit FIFO Registers (DSPI_3_TXFR2) | 32 | R | 0000_0000h | 40.4.10/ 1294 |
| FBE7_4048 | Transmit FIFO Registers (DSPI_3_TXFR3) | 32 | R | 0000_0000h | 40.4.10/ 1294 |
| FBE7_407C | Receive FIFO Registers (DSPI_3_RXFR0) | 32 | R | 0000_0000h | 40.4.11/ 1295 |
| FBE7_4080 | Receive FIFO Registers (DSPI_3_RXFR1) | 32 | R | 0000_0000h | 40.4.11/ 1295 |
| FBE7_4084 | Receive FIFO Registers (DSPI_3_RXFR2) | 32 | R | 0000_0000h | 40.4.11/ 1295 |
| FBE7_4088 | Receive FIFO Registers (DSPI_3_RXFR3) | 32 | R | 0000_0000h | 40.4.11/ 1295 |
| FBE7_40BC | DSI Configuration Register 0 (DSPI_3_DSICR0) | 32 | R/W | 0000_0000h | 40.4.12/ 1296 |
| FBE7_40C0 | DSI Serialization Data Register 0 (DSPI_3_SDR0) | 32 | R | 0000_0000h | 40.4.13/ 1299 |
| FBE7_40C4 | DSI Alternate Serialization Data Register 0 (DSPI_3_ASDR0) | 32 | R/W | 0000_0000h | 40.4.14/ 1299 |
| FBE7_40C8 | DSI Transmit Comparison Register 0 (DSPI_3_COMPR0) | 32 | R | 0000_0000h | 40.4.15/ 1300 |
| FBE7_40CC | DSI Deserialization Data Register 0 (DSPI_3_DDR0) | 32 | R | 0000_0000h | 40.4.16/ 1300 |
| FBE7_40D0 | DSI Configuration Register 1 (DSPI_3_DSICR1) | 32 | R/W | 0000_0000h | 40.4.17/ 1301 |
| FBE7_40D4 | DSI Serialization Source Select Register 0 (DSPI_3_SSR0) | 32 | R/W | 0000_0000h | 40.4.18/ 1303 |
| FBE7_40D8 | DSI Parallel Input Select Register 0 (DSPI_3_PISR0) | 32 | R/W | 0000_0000h | 40.4.19/ 1303 |
| FBE7_40DC | DSI Parallel Input Select Register 1 (DSPI_3_PISR1) | 32 | R/W | 0000_0000h | 40.4.20/ 1304 |

*Table continues on the next page...*

## DSPI memory map (continued)

| Absolute address (hex) | Register name | Width (in bits) | Access | Reset value | Section/ page |
|---|---|---|---|---|---|
| FBE7_40E0 | DSI Parallel Input Select Register 2 (DSPI_3_PISR2) | 32 | R/W | 0000_0000h | 40.4.21/ 1305 |
| FBE7_40E4 | DSI Parallel Input Select Register 3 (DSPI_3_PISR3) | 32 | R/W | 0000_0000h | 40.4.22/ 1306 |
| FBE7_40E8 | DSI Deserialized Data Interrupt Mask Register 0 (DSPI_3_DIMR0) | 32 | R/W | 0000_0000h | 40.4.23/ 1307 |
| FBE7_40EC | DSI Deserialized Data Polarity Interrupt Register 0 (DSPI_3_DPIR0) | 32 | R/W | 0000_0000h | 40.4.24/ 1307 |
| FBE7_40F0 | DSI Serialization Data Register 1 (DSPI_3_SDR1) | 32 | R | 0000_0000h | 40.4.25/ 1308 |
| FBE7_40F4 | DSI Alternate Serialization Data Register 1 (DSPI_3_ASDR1) | 32 | R/W | 0000_0000h | 40.4.26/ 1309 |
| FBE7_40F8 | DSI Transmit Comparison Register 1 (DSPI_3_COMPR1) | 32 | R | 0000_0000h | 40.4.27/ 1309 |
| FBE7_40FC | DSI Deserialization Data Register 1 (DSPI_3_DDR1) | 32 | R | 0000_0000h | 40.4.28/ 1310 |
| FBE7_4100 | DSI Serialization Source Select Register 1 (DSPI_3_SSR1) | 32 | R/W | 0000_0000h | 40.4.29/ 1311 |
| FBE7_4104 | DSI Parallel Input Select Register 4 (DSPI_3_PISR4) | 32 | R/W | 0000_0000h | 40.4.30/ 1311 |
| FBE7_4108 | DSI Parallel Input Select Register 5 (DSPI_3_PISR5) | 32 | R/W | 0000_0000h | 40.4.31/ 1312 |
| FBE7_410C | DSI Parallel Input Select Register 6 (DSPI_3_PISR6) | 32 | R/W | 0000_0000h | 40.4.32/ 1313 |
| FBE7_4110 | DSI Parallel Input Select Register 7 (DSPI_3_PISR7) | 32 | R/W | 0000_0000h | 40.4.33/ 1314 |
| FBE7_4114 | DSI Deserialized Data Interrupt Mask Register 1 (DSPI_3_DIMR1) | 32 | R/W | 0000_0000h | 40.4.34/ 1315 |
| FBE7_4118 | DSI Deserialized Data Polarity Interrupt Register 1 (DSPI_3_DPIR1) | 32 | R/W | 0000_0000h | 40.4.35/ 1315 |
| FBE7_411C | Clock and Transfer Attributes Register Extended (DSPI_3_CTARE0) | 32 | R/W | 0000_0001h | 40.4.36/ 1316 |
| FBE7_4120 | Clock and Transfer Attributes Register Extended (DSPI_3_CTARE1) | 32 | R/W | 0000_0001h | 40.4.36/ 1316 |
| FBE7_4124 | Clock and Transfer Attributes Register Extended (DSPI_3_CTARE2) | 32 | R/W | 0000_0001h | 40.4.36/ 1316 |
| FBE7_4128 | Clock and Transfer Attributes Register Extended (DSPI_3_CTARE3) | 32 | R/W | 0000_0001h | 40.4.36/ 1316 |
| FBE7_412C | Clock and Transfer Attributes Register Extended (DSPI_3_CTARE4) | 32 | R/W | 0000_0001h | 40.4.36/ 1316 |
| FBE7_4130 | Clock and Transfer Attributes Register Extended (DSPI_3_CTARE5) | 32 | R/W | 0000_0001h | 40.4.36/ 1316 |
| FBE7_413C | Status Register Extended (DSPI_3_SREX) | 32 | R | 0000_0000h | 40.4.37/ 1317 |

*Table continues on the next page...*

## DSPI memory map (continued)

| Absolute address (hex) | Register name | Width (in bits) | Access | Reset value | Section/ page |
|---|---|---|---|---|---|
| FBE7_4140 | Trigger Register (DSPI_3_TRIG) | 32 | R/W | 0000_0000h | 40.4.38/ 1318 |
| FBE7_4150 | Time Slot Length Register (DSPI_3_TSL) | 32 | R/W | 0000_0000h | 40.4.39/ 1319 |
| FBE7_4154 | Time Slot Configuration Register (DSPI_3_TS_CONF) | 32 | R/W | 0000_0000h | 40.4.40/ 1320 |
| FFE7_0000 | Module Configuration Register (DSPI_0_MCR) | 32 | R/W | 0000_4001h | 40.4.1/1272 |
| FFE7_0008 | Transfer Count Register (DSPI_0_TCR) | 32 | R/W | 0000_0000h | 40.4.2/1276 |
| FFE7_000C | Clock and Transfer Attributes Register (In Master Mode) (DSPI_0_CTAR0) | 32 | R/W | 7800_0000h | 40.4.3/1276 |
| FFE7_000C | Clock and Transfer Attributes Register (In Slave Mode) (DSPI_0_CTAR0_SLAVE) | 32 | R/W | 7800_0000h | 40.4.4/1281 |
| FFE7_0010 | Clock and Transfer Attributes Register (In Master Mode) (DSPI_0_CTAR1) | 32 | R/W | 7800_0000h | 40.4.3/1276 |
| FFE7_0010 | Clock and Transfer Attributes Register (In Slave Mode) (DSPI_0_CTAR1_SLAVE) | 32 | R/W | 7800_0000h | 40.4.4/1281 |
| FFE7_0014 | Clock and Transfer Attributes Register (In Master Mode) (DSPI_0_CTAR2) | 32 | R/W | 7800_0000h | 40.4.3/1276 |
| FFE7_0018 | Clock and Transfer Attributes Register (In Master Mode) (DSPI_0_CTAR3) | 32 | R/W | 7800_0000h | 40.4.3/1276 |
| FFE7_001C | Clock and Transfer Attributes Register (In Master Mode) (DSPI_0_CTAR4) | 32 | R/W | 7800_0000h | 40.4.3/1276 |
| FFE7_0020 | Clock and Transfer Attributes Register (In Master Mode) (DSPI_0_CTAR5) | 32 | R/W | 7800_0000h | 40.4.3/1276 |
| FFE7_002C | Status Register (DSPI_0_SR) | 32 | R/W | 0201_0000h | 40.4.5/1284 |
| FFE7_0030 | DMA/Interrupt Request Select and Enable Register (DSPI_0_RSER) | 32 | R/W | 0000_0000h | 40.4.6/1288 |
| FFE7_0034 | PUSH TX FIFO Register In Master Mode (DSPI_0_PUSHR) | 32 | R/W | 0000_0000h | 40.4.7/1291 |
| FFE7_0034 | PUSH TX FIFO Register In Slave Mode (DSPI_0_PUSHR_SLAVE) | 32 | R/W | 0000_0000h | 40.4.8/1293 |
| FFE7_0038 | POP RX FIFO Register (DSPI_0_POPR) | 32 | R | 0000_0000h | 40.4.9/1294 |
| FFE7_003C | Transmit FIFO Registers (DSPI_0_TXFR0) | 32 | R | 0000_0000h | 40.4.10/ 1294 |
| FFE7_0040 | Transmit FIFO Registers (DSPI_0_TXFR1) | 32 | R | 0000_0000h | 40.4.10/ 1294 |
| FFE7_0044 | Transmit FIFO Registers (DSPI_0_TXFR2) | 32 | R | 0000_0000h | 40.4.10/ 1294 |
| FFE7_0048 | Transmit FIFO Registers (DSPI_0_TXFR3) | 32 | R | 0000_0000h | 40.4.10/ 1294 |
| FFE7_007C | Receive FIFO Registers (DSPI_0_RXFR0) | 32 | R | 0000_0000h | 40.4.11/ 1295 |
| FFE7_0080 | Receive FIFO Registers (DSPI_0_RXFR1) | 32 | R | 0000_0000h | 40.4.11/ 1295 |

*Table continues on the next page...*

## DSPI memory map (continued)

| Absolute address (hex) | Register name | Width (in bits) | Access | Reset value | Section/ page |
|---|---|---|---|---|---|
| FFE7_0084 | Receive FIFO Registers (DSPI_0_RXFR2) | 32 | R | 0000_0000h | 40.4.11/ 1295 |
| FFE7_0088 | Receive FIFO Registers (DSPI_0_RXFR3) | 32 | R | 0000_0000h | 40.4.11/ 1295 |
| FFE7_00BC | DSI Configuration Register 0 (DSPI_0_DSICR0) | 32 | R/W | 0000_0000h | 40.4.12/ 1296 |
| FFE7_00C0 | DSI Serialization Data Register 0 (DSPI_0_SDR0) | 32 | R | 0000_0000h | 40.4.13/ 1299 |
| FFE7_00C4 | DSI Alternate Serialization Data Register 0 (DSPI_0_ASDR0) | 32 | R/W | 0000_0000h | 40.4.14/ 1299 |
| FFE7_00C8 | DSI Transmit Comparison Register 0 (DSPI_0_COMPR0) | 32 | R | 0000_0000h | 40.4.15/ 1300 |
| FFE7_00CC | DSI Deserialization Data Register 0 (DSPI_0_DDR0) | 32 | R | 0000_0000h | 40.4.16/ 1300 |
| FFE7_00D0 | DSI Configuration Register 1 (DSPI_0_DSICR1) | 32 | R/W | 0000_0000h | 40.4.17/ 1301 |
| FFE7_00D4 | DSI Serialization Source Select Register 0 (DSPI_0_SSR0) | 32 | R/W | 0000_0000h | 40.4.18/ 1303 |
| FFE7_00D8 | DSI Parallel Input Select Register 0 (DSPI_0_PISR0) | 32 | R/W | 0000_0000h | 40.4.19/ 1303 |
| FFE7_00DC | DSI Parallel Input Select Register 1 (DSPI_0_PISR1) | 32 | R/W | 0000_0000h | 40.4.20/ 1304 |
| FFE7_00E0 | DSI Parallel Input Select Register 2 (DSPI_0_PISR2) | 32 | R/W | 0000_0000h | 40.4.21/ 1305 |
| FFE7_00E4 | DSI Parallel Input Select Register 3 (DSPI_0_PISR3) | 32 | R/W | 0000_0000h | 40.4.22/ 1306 |
| FFE7_00E8 | DSI Deserialized Data Interrupt Mask Register 0 (DSPI_0_DIMR0) | 32 | R/W | 0000_0000h | 40.4.23/ 1307 |
| FFE7_00EC | DSI Deserialized Data Polarity Interrupt Register 0 (DSPI_0_DPIR0) | 32 | R/W | 0000_0000h | 40.4.24/ 1307 |
| FFE7_00F0 | DSI Serialization Data Register 1 (DSPI_0_SDR1) | 32 | R | 0000_0000h | 40.4.25/ 1308 |
| FFE7_00F4 | DSI Alternate Serialization Data Register 1 (DSPI_0_ASDR1) | 32 | R/W | 0000_0000h | 40.4.26/ 1309 |
| FFE7_00F8 | DSI Transmit Comparison Register 1 (DSPI_0_COMPR1) | 32 | R | 0000_0000h | 40.4.27/ 1309 |
| FFE7_00FC | DSI Deserialization Data Register 1 (DSPI_0_DDR1) | 32 | R | 0000_0000h | 40.4.28/ 1310 |
| FFE7_0100 | DSI Serialization Source Select Register 1 (DSPI_0_SSR1) | 32 | R/W | 0000_0000h | 40.4.29/ 1311 |
| FFE7_0104 | DSI Parallel Input Select Register 4 (DSPI_0_PISR4) | 32 | R/W | 0000_0000h | 40.4.30/ 1311 |
| FFE7_0108 | DSI Parallel Input Select Register 5 (DSPI_0_PISR5) | 32 | R/W | 0000_0000h | 40.4.31/ 1312 |

*Table continues on the next page...*

## DSPI memory map (continued)

| Absolute address (hex) | Register name | Width (in bits) | Access | Reset value | Section/ page |
|---|---|---|---|---|---|
| FFE7_010C | DSI Parallel Input Select Register 6 (DSPI_0_PISR6) | 32 | R/W | 0000_0000h | 40.4.32/ 1313 |
| FFE7_0110 | DSI Parallel Input Select Register 7 (DSPI_0_PISR7) | 32 | R/W | 0000_0000h | 40.4.33/ 1314 |
| FFE7_0114 | DSI Deserialized Data Interrupt Mask Register 1 (DSPI_0_DIMR1) | 32 | R/W | 0000_0000h | 40.4.34/ 1315 |
| FFE7_0118 | DSI Deserialized Data Polarity Interrupt Register 1 (DSPI_0_DPIR1) | 32 | R/W | 0000_0000h | 40.4.35/ 1315 |
| FFE7_011C | Clock and Transfer Attributes Register Extended (DSPI_0_CTARE0) | 32 | R/W | 0000_0001h | 40.4.36/ 1316 |
| FFE7_0120 | Clock and Transfer Attributes Register Extended (DSPI_0_CTARE1) | 32 | R/W | 0000_0001h | 40.4.36/ 1316 |
| FFE7_0124 | Clock and Transfer Attributes Register Extended (DSPI_0_CTARE2) | 32 | R/W | 0000_0001h | 40.4.36/ 1316 |
| FFE7_0128 | Clock and Transfer Attributes Register Extended (DSPI_0_CTARE3) | 32 | R/W | 0000_0001h | 40.4.36/ 1316 |
| FFE7_012C | Clock and Transfer Attributes Register Extended (DSPI_0_CTARE4) | 32 | R/W | 0000_0001h | 40.4.36/ 1316 |
| FFE7_0130 | Clock and Transfer Attributes Register Extended (DSPI_0_CTARE5) | 32 | R/W | 0000_0001h | 40.4.36/ 1316 |
| FFE7_013C | Status Register Extended (DSPI_0_SREX) | 32 | R | 0000_0000h | 40.4.37/ 1317 |
| FFE7_0140 | Trigger Register (DSPI_0_TRIG) | 32 | R/W | 0000_0000h | 40.4.38/ 1318 |
| FFE7_0150 | Time Slot Length Register (DSPI_0_TSL) | 32 | R/W | 0000_0000h | 40.4.39/ 1319 |
| FFE7_0154 | Time Slot Configuration Register (DSPI_0_TS_CONF) | 32 | R/W | 0000_0000h | 40.4.40/ 1320 |
| FFE7_4000 | Module Configuration Register (DSPI_2_MCR) | 32 | R/W | 0000_4001h | 40.4.1/1272 |
| FFE7_4008 | Transfer Count Register (DSPI_2_TCR) | 32 | R/W | 0000_0000h | 40.4.2/1276 |
| FFE7_400C | Clock and Transfer Attributes Register (In Master Mode) (DSPI_2_CTAR0) | 32 | R/W | 7800_0000h | 40.4.3/1276 |
| FFE7_400C | Clock and Transfer Attributes Register (In Slave Mode) (DSPI_2_CTAR0_SLAVE) | 32 | R/W | 7800_0000h | 40.4.4/1281 |
| FFE7_4010 | Clock and Transfer Attributes Register (In Master Mode) (DSPI_2_CTAR1) | 32 | R/W | 7800_0000h | 40.4.3/1276 |
| FFE7_4010 | Clock and Transfer Attributes Register (In Slave Mode) (DSPI_2_CTAR1_SLAVE) | 32 | R/W | 7800_0000h | 40.4.4/1281 |
| FFE7_4014 | Clock and Transfer Attributes Register (In Master Mode) (DSPI_2_CTAR2) | 32 | R/W | 7800_0000h | 40.4.3/1276 |
| FFE7_4018 | Clock and Transfer Attributes Register (In Master Mode) (DSPI_2_CTAR3) | 32 | R/W | 7800_0000h | 40.4.3/1276 |
| FFE7_401C | Clock and Transfer Attributes Register (In Master Mode) (DSPI_2_CTAR4) | 32 | R/W | 7800_0000h | 40.4.3/1276 |

*Table continues on the next page...*

## DSPI memory map (continued)

| Absolute address (hex) | Register name | Width (in bits) | Access | Reset value | Section/ page |
|---|---|---|---|---|---|
| FFE7_4020 | Clock and Transfer Attributes Register (In Master Mode) (DSPI_2_CTAR5) | 32 | R/W | 7800_0000h | 40.4.3/1276 |
| FFE7_402C | Status Register (DSPI_2_SR) | 32 | R/W | 0201_0000h | 40.4.5/1284 |
| FFE7_4030 | DMA/Interrupt Request Select and Enable Register (DSPI_2_RSER) | 32 | R/W | 0000_0000h | 40.4.6/1288 |
| FFE7_4034 | PUSH TX FIFO Register In Master Mode (DSPI_2_PUSHR) | 32 | R/W | 0000_0000h | 40.4.7/1291 |
| FFE7_4034 | PUSH TX FIFO Register In Slave Mode (DSPI_2_PUSHR_SLAVE) | 32 | R/W | 0000_0000h | 40.4.8/1293 |
| FFE7_4038 | POP RX FIFO Register (DSPI_2_POPR) | 32 | R | 0000_0000h | 40.4.9/1294 |
| FFE7_403C | Transmit FIFO Registers (DSPI_2_TXFR0) | 32 | R | 0000_0000h | 40.4.10/ 1294 |
| FFE7_4040 | Transmit FIFO Registers (DSPI_2_TXFR1) | 32 | R | 0000_0000h | 40.4.10/ 1294 |
| FFE7_4044 | Transmit FIFO Registers (DSPI_2_TXFR2) | 32 | R | 0000_0000h | 40.4.10/ 1294 |
| FFE7_4048 | Transmit FIFO Registers (DSPI_2_TXFR3) | 32 | R | 0000_0000h | 40.4.10/ 1294 |
| FFE7_407C | Receive FIFO Registers (DSPI_2_RXFR0) | 32 | R | 0000_0000h | 40.4.11/ 1295 |
| FFE7_4080 | Receive FIFO Registers (DSPI_2_RXFR1) | 32 | R | 0000_0000h | 40.4.11/ 1295 |
| FFE7_4084 | Receive FIFO Registers (DSPI_2_RXFR2) | 32 | R | 0000_0000h | 40.4.11/ 1295 |
| FFE7_4088 | Receive FIFO Registers (DSPI_2_RXFR3) | 32 | R | 0000_0000h | 40.4.11/ 1295 |
| FFE7_40BC | DSI Configuration Register 0 (DSPI_2_DSICR0) | 32 | R/W | 0000_0000h | 40.4.12/ 1296 |
| FFE7_40C0 | DSI Serialization Data Register 0 (DSPI_2_SDR0) | 32 | R | 0000_0000h | 40.4.13/ 1299 |
| FFE7_40C4 | DSI Alternate Serialization Data Register 0 (DSPI_2_ASDR0) | 32 | R/W | 0000_0000h | 40.4.14/ 1299 |
| FFE7_40C8 | DSI Transmit Comparison Register 0 (DSPI_2_COMPR0) | 32 | R | 0000_0000h | 40.4.15/ 1300 |
| FFE7_40CC | DSI Deserialization Data Register 0 (DSPI_2_DDR0) | 32 | R | 0000_0000h | 40.4.16/ 1300 |
| FFE7_40D0 | DSI Configuration Register 1 (DSPI_2_DSICR1) | 32 | R/W | 0000_0000h | 40.4.17/ 1301 |
| FFE7_40D4 | DSI Serialization Source Select Register 0 (DSPI_2_SSR0) | 32 | R/W | 0000_0000h | 40.4.18/ 1303 |
| FFE7_40D8 | DSI Parallel Input Select Register 0 (DSPI_2_PISR0) | 32 | R/W | 0000_0000h | 40.4.19/ 1303 |
| FFE7_40DC | DSI Parallel Input Select Register 1 (DSPI_2_PISR1) | 32 | R/W | 0000_0000h | 40.4.20/ 1304 |

*Table continues on the next page...*

## DSPI memory map (continued)

| Absolute address (hex) | Register name | Width (in bits) | Access | Reset value | Section/ page |
|---|---|---|---|---|---|
| FFE7_40E0 | DSI Parallel Input Select Register 2 (DSPI_2_PISR2) | 32 | R/W | 0000_0000h | 40.4.21/ 1305 |
| FFE7_40E4 | DSI Parallel Input Select Register 3 (DSPI_2_PISR3) | 32 | R/W | 0000_0000h | 40.4.22/ 1306 |
| FFE7_40E8 | DSI Deserialized Data Interrupt Mask Register 0 (DSPI_2_DIMR0) | 32 | R/W | 0000_0000h | 40.4.23/ 1307 |
| FFE7_40EC | DSI Deserialized Data Polarity Interrupt Register 0 (DSPI_2_DPIR0) | 32 | R/W | 0000_0000h | 40.4.24/ 1307 |
| FFE7_40F0 | DSI Serialization Data Register 1 (DSPI_2_SDR1) | 32 | R | 0000_0000h | 40.4.25/ 1308 |
| FFE7_40F4 | DSI Alternate Serialization Data Register 1 (DSPI_2_ASDR1) | 32 | R/W | 0000_0000h | 40.4.26/ 1309 |
| FFE7_40F8 | DSI Transmit Comparison Register 1 (DSPI_2_COMPR1) | 32 | R | 0000_0000h | 40.4.27/ 1309 |
| FFE7_40FC | DSI Deserialization Data Register 1 (DSPI_2_DDR1) | 32 | R | 0000_0000h | 40.4.28/ 1310 |
| FFE7_4100 | DSI Serialization Source Select Register 1 (DSPI_2_SSR1) | 32 | R/W | 0000_0000h | 40.4.29/ 1311 |
| FFE7_4104 | DSI Parallel Input Select Register 4 (DSPI_2_PISR4) | 32 | R/W | 0000_0000h | 40.4.30/ 1311 |
| FFE7_4108 | DSI Parallel Input Select Register 5 (DSPI_2_PISR5) | 32 | R/W | 0000_0000h | 40.4.31/ 1312 |
| FFE7_410C | DSI Parallel Input Select Register 6 (DSPI_2_PISR6) | 32 | R/W | 0000_0000h | 40.4.32/ 1313 |
| FFE7_4110 | DSI Parallel Input Select Register 7 (DSPI_2_PISR7) | 32 | R/W | 0000_0000h | 40.4.33/ 1314 |
| FFE7_4114 | DSI Deserialized Data Interrupt Mask Register 1 (DSPI_2_DIMR1) | 32 | R/W | 0000_0000h | 40.4.34/ 1315 |
| FFE7_4118 | DSI Deserialized Data Polarity Interrupt Register 1 (DSPI_2_DPIR1) | 32 | R/W | 0000_0000h | 40.4.35/ 1315 |
| FFE7_411C | Clock and Transfer Attributes Register Extended (DSPI_2_CTARE0) | 32 | R/W | 0000_0001h | 40.4.36/ 1316 |
| FFE7_4120 | Clock and Transfer Attributes Register Extended (DSPI_2_CTARE1) | 32 | R/W | 0000_0001h | 40.4.36/ 1316 |
| FFE7_4124 | Clock and Transfer Attributes Register Extended (DSPI_2_CTARE2) | 32 | R/W | 0000_0001h | 40.4.36/ 1316 |
| FFE7_4128 | Clock and Transfer Attributes Register Extended (DSPI_2_CTARE3) | 32 | R/W | 0000_0001h | 40.4.36/ 1316 |
| FFE7_412C | Clock and Transfer Attributes Register Extended (DSPI_2_CTARE4) | 32 | R/W | 0000_0001h | 40.4.36/ 1316 |
| FFE7_4130 | Clock and Transfer Attributes Register Extended (DSPI_2_CTARE5) | 32 | R/W | 0000_0001h | 40.4.36/ 1316 |
| FFE7_413C | Status Register Extended (DSPI_2_SREX) | 32 | R | 0000_0000h | 40.4.37/ 1317 |

*Table continues on the next page...*

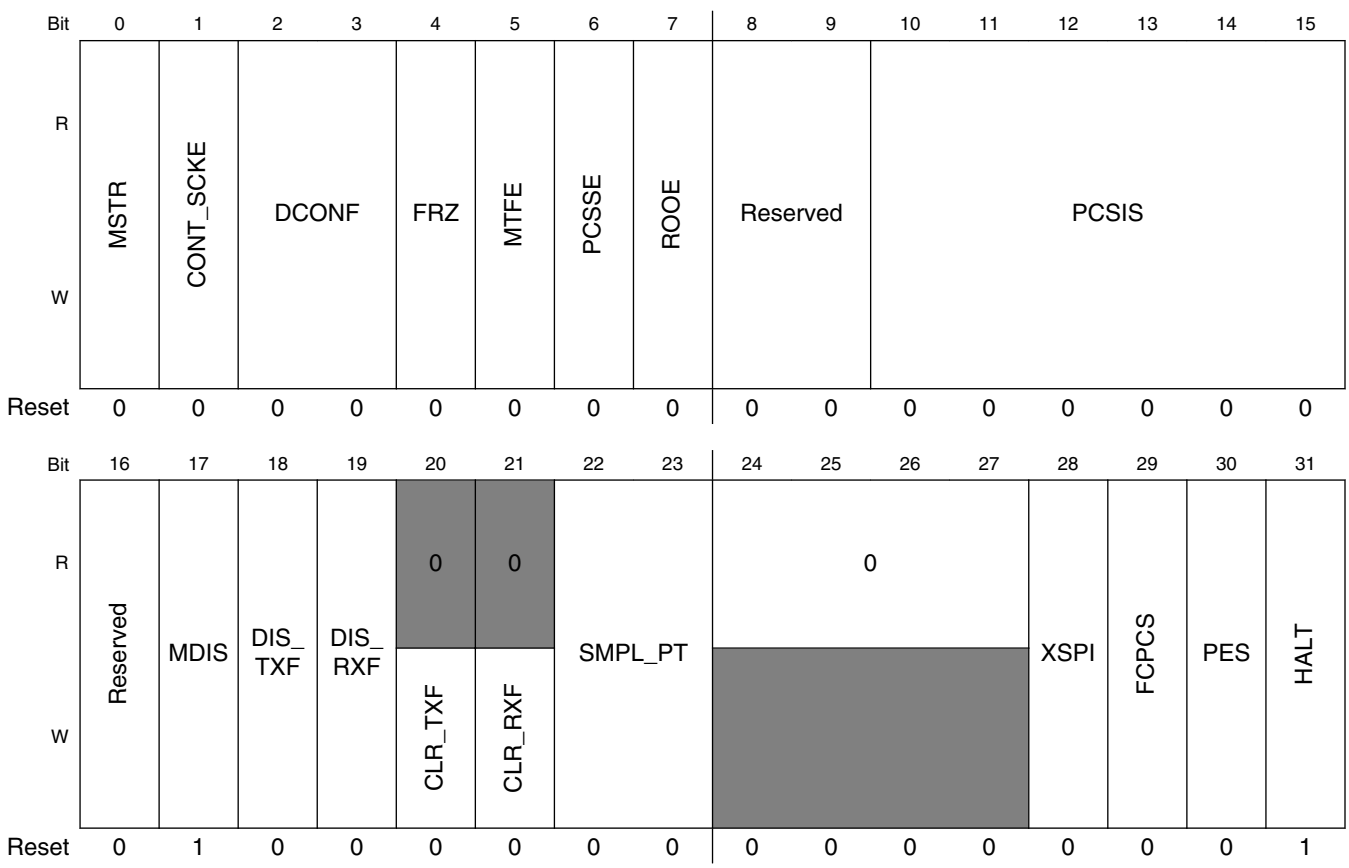
**DSPI memory map (continued)**

| Absolute address (hex) | Register name | Width (in bits) | Access | Reset value | Section/ page |
|---|---|---|---|---|---|
| FFE7_4140 | Trigger Register (DSPI_2_TRIG) | 32 | R/W | 0000_0000h | 40.4.38/ 1318 |
| FFE7_4150 | Time Slot Length Register (DSPI_2_TSL) | 32 | R/W | 0000_0000h | 40.4.39/ 1319 |
| FFE7_4154 | Time Slot Configuration Register (DSPI_2_TS_CONF) | 32 | R/W | 0000_0000h | 40.4.40/ 1320 |

## 40.4.1 Module Configuration Register (DSPIx_MCR)

Contains bits to configure various attributes associated with the module operations. The HALT and MDIS bits can be changed at any time, but the effect takes place only on the next frame boundary. Only the HALT and MDIS bits in the MCR can be changed, while the module is in the Running state.

Address: Base address + 0h offset



## DSPIx_MCR field descriptions

| Field | Description |
|---|---|
| 0<br>MSTR | Master/Slave Mode Select<br><br>Enables either Master mode (if supported) or Slave mode (if supported) operation.<br><br>0 Enables Slave mode<br>1 Enables Master mode |
| 1<br>CONT_SCKE | Continuous SCK Enable<br><br>Enables the Serial Communication Clock (SCK) to run continuously.<br><br>0 Continuous SCK disabled.<br>1 Continuous SCK enabled. |
| 2–3<br>DCONF | Interface Configuration<br><br>Selects among the different interface configurations of the module.<br><br>00 SPI<br>01 DSI<br>10 CSI<br>11 Reserved |
| 4<br>FRZ | Freeze<br><br>Enables transfers to be stopped on the next frame boundary when the device enters Debug mode.<br><br>0 Do not halt serial transfers in Debug mode.<br>1 Halt serial transfers in Debug mode. |
| 5<br>MTFE | Modified Transfer Format Enable<br><br>Enables a modified transfer format to be used.<br><br>0 Modified SPI transfer format disabled.<br>1 Modified SPI transfer format enabled. |
| 6<br>PCSSE | Peripheral Chip Select Strobe Enable<br><br>Enables the PCS5/ $\overline{\text{PCSS}}$ to operate as a PCS Strobe output signal.<br><br>0 PCS5/ $\overline{\text{PCSS}}$ is used as the Peripheral Chip Select[5] signal.<br>1 PCS5/ $\overline{\text{PCSS}}$ is used as an active-low PCS Strobe signal. |
| 7<br>ROOE | Receive FIFO Overflow Overwrite Enable<br><br>In the RX FIFO overflow condition, configures the module to ignore the incoming serial data or overwrite existing data. If the RX FIFO is full and new data is received, the data from the transfer, generating the overflow, is ignored or shifted into the shift register.<br><br>0 Incoming data is ignored.<br>1 Incoming data is shifted into the shift register. |
| 8–9<br>Reserved | Always write the reset value to this field.<br><br>This field is reserved. |
| 10–15<br>PCSIS | Peripheral Chip Select x Inactive State |

*Table continues on the next page...*

## DSPIx_MCR field descriptions (continued)

| Field | Description |
|---|---|
| | Determines the inactive state of PCSx. Refer to the chip-specific D SPI information for the number of PCS signals used in this chip.<br>**NOTE:** The effect of this bit only takes place when module is enabled. Ensure that this bit is configured correctly before enabling the DSPI interface.<br>0 The inactive state of PCSx is low.<br>1 The inactive state of PCSx is high. |
| 16<br>Reserved | This field is reserved. |
| 17<br>MDIS | Module Disable<br>Allows the clock to be stopped to the non-memory mapped logic in the module effectively putting it in a software-controlled power-saving state. The reset value of the MDIS bit is parameterized, with a default reset value of 1. When the module is used in Slave Mode, it is recommended to leave this bit 0, because a slave doesn't have control over master transactions.<br>0 Enables the module clocks.<br>1 Allows external logic to disable the module clocks. |
| 18<br>DIS_TXF | Disable Transmit FIFO<br>When the TX FIFO is disabled, the transmit part of the module operates as a simplified double-buffered SPI. This bit can be written only when the MDIS bit is cleared.<br>0 TX FIFO is enabled.<br>1 TX FIFO is disabled. |
| 19<br>DIS_RXF | Disable Receive FIFO<br>When the RX FIFO is disabled, the receive part of the module operates as a simplified double-buffered SPI. This bit can only be written when the MDIS bit is cleared.<br>0 RX FIFO is enabled.<br>1 RX FIFO is disabled. |
| 20<br>CLR_TXF | Clear TX FIFO<br>Flushes the TX FIFO. Writing a 1 to CLR_TXF clears the TX FIFO Counter. The CLR_TXF bit is always read as zero.<br>0 Do not clear the TX FIFO counter.<br>1 Clear the TX FIFO counter. |
| 21<br>CLR_RXF | CLR_RXF<br>Flushes the RX FIFO. Writing a 1 to CLR_RXF clears the RX Counter. The CLR_RXF bit is always read as zero.<br>**NOTE:** After every RX FIFO clear operation (MCR [CLR_RXF] = 0b1) following a RX FIFO overflow (SR [RFOF] = 0b1) scenario, immediately perform a single POP from the RX FIFO and discard the read data. The POP and discard operation should be completed before the reception of new incoming frame.<br>0 Do not clear the RX FIFO counter.<br>1 Clear the RX FIFO counter. |

*Table continues on the next page...*

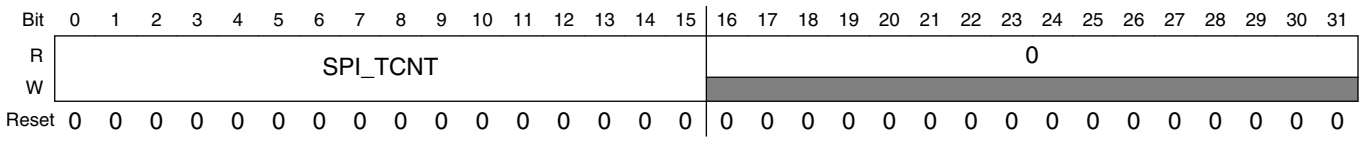
## DSPIx_MCR field descriptions (continued)

| Field | Description |
|---|---|
| 22–23<br>SMPL_PT | Sample Point<br><br>Controls when the module master samples SIN in Modified Transfer Format. This field is valid only when CPHA bit in CTARn[CPHA] is 0.<br><br>00 0 protocol clock cycles between SCK edge and SIN sample<br>01 1 protocol clock cycle between SCK edge and SIN sample<br>10 2 protocol clock cycles between SCK edge and SIN sample<br>11 Reserved |
| 24–27<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 28<br>XSPI | Extended SPI Mode<br><br>This bit enables usage of CTARE (Command and Transfer Attribute Register Extended) Registers. CTARE registers allow the user to send up to 32 bit SPI frames. Command Cycling is also enabled which allows the user to send multiple Data Frames using a single Command Frame. When MCR[DIS_TXF] is asserted, the Extended SPI Mode cannot be used to transmit SPI frames which are more than 16 bits in size.<br><br>0 Normal SPI Mode. Frame size can be up to 16 bits. Command Cycling is not available in this mode.<br>1 Extended SPI Mode. Up to 32 bit SPI Frames along with Command Cycling is Enabled. |
| 29<br>FCPCS | Fast Continuous PCS Mode.<br><br>This bit enables the masking of “After SCK ($t_{ASC}$)” and “PCS to SCK ($t_{CSC}$)” delays when operating in Continuous PCS mode. This masking is not available if Continuous SCK mode is enabled. The individual delay masks are selected via bits MASC and MCSC of the PUSHR register. The firmware should select appropriate masks when providing continuous frames via the PUSHR register.<br><br>0 Normal or Slow Continuous PCS mode. Masking of delays is disabled.<br>1 Fast Continuous PCS mode. Delays masked via control bits in PUSHR register. |
| 30<br>PES | Parity Error Stop<br><br>Controls SPI operation when a parity error is detected in a received SPI frame.<br><br>0 SPI frame transmission continues.<br>1 SPI frame transmission stops. |
| 31<br>HALT | Halt<br><br>The HALT bit starts and stops frame transfers. See Start and Stop of Module transfers<br><br>0 Start transfers.<br>1 Stop transfers. |

## 40.4.2 Transfer Count Register (DSPIx_TCR)

TCR contains a counter that indicates the number of SPI transfers made. The transfer counter is intended to assist in queue management. Do not write the TCR when the module is in the Running state.

Address: Base address + 8h offset




**DSPIx_TCR field descriptions**

| Field | Description |
|---|---|
| 0–15<br>SPI_TCNT | SPI Transfer Counter<br><br>Counts the number of SPI transfers the module makes. The SPI_TCNT field increments every time the last bit of an SPI frame is transmitted. A value written to SPI_TCNT presets the counter to that value. SPI_TCNT is reset to zero at the beginning of the frame when the CTCNT field is set in the executing SPI command. The Transfer Counter wraps around; incrementing the counter past 65535 resets the counter to zero. |
| 16–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |

## 40.4.3 Clock and Transfer Attributes Register (In Master Mode) (DSPIx_CTAR*n*)

CTAR registers are used to define different transfer attributes. Do not write to the CTAR registers while the module is in the Running state.

In Master mode, the CTAR registers define combinations of transfer attributes such as frame size, clock phase and polarity, data bit ordering, baud rate, and various delays. In slave mode, a subset of the bitfields in CTAR0 and CTAR1 are used to set the slave transfer attributes.

When the module is configured as a SPI master, the CTAS field in the command portion of the TX FIFO entry selects which of the CTAR registers is used. When the module is configured as an SPI bus slave, it uses the CTAR0 register.

When the module is configured as a DSI master, the DSICTAS field in the DSI Configuration Register 0 (DSICR0), selects which of the CTAR register is used. When the module is configured as a DSI bus slave, the CTAR1 register is used.

In CSI Configuration, the transfer attributes are selected based on whether the current frame is SPI data or DSI data. SPI transfers in CSI Configuration follow the protocol described for SPI Configuration, and DSI transfers in CSI Configuration follow the protocol described for DSI Configuration.CSI Configuration is valid only in conjunction with master mode.

TSB mode sets some limitations on transfer attributes:

- Clock phase is forced to be CPHA = 1 and the CPHA bit setting has no effect.
- PCS lines are driven at the driving edge of the SCK clock together with SOUT, so PCS assertion and negation delays control is unavailable and PCSSCK, PASC, CSSCK and ASC fields have no effect.
- Delay after transfer can be set from 1 to 64 serial clocks with help of PDT and DT fields

Address: Base address + Ch offset + (4d × i), where i=0d to 5d

| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R<br>W | DBR | FMSZ | | | | CPOL | CPHA | LSBFE | PCSSCK | | PASC | | PDT | | PBR | |
| Reset | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R<br>W | CSSCK | | | | ASC | | | | DT | | | | BR | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

### DSPIx_CTAR*n* field descriptions

| Field | Description |
|---|---|
| 0<br>DBR | Double Baud Rate<br><br>Doubles the effective baud rate of the Serial Communications Clock (SCK). This field is used only in master mode. It effectively halves the Baud Rate division ratio, supporting faster frequencies, and odd division ratios for the Serial Communications Clock (SCK). When the DBR bit is set, the duty cycle of the Serial Communications Clock (SCK) depends on the value in the Baud Rate Prescaler and the Clock Phase bit as listed in the following table. See the BR field description for details on how to compute the baud rate. |

**Table 40-5. DSPI SCK Duty Cycle**

| DBR | CPHA | PBR | SCK Duty Cycle |
|---|---|---|---|
| 0 | any | any | 50/50 |
| 1 | 0 | 00 | 50/50 |
| 1 | 0 | 01 | 33/66 |
| 1 | 0 | 10 | 40/60 |

*Table continues on the next page...*

## DSPIx_CTAR*n* field descriptions (continued)

| Field | Description |
|---|---|

**Table 40-5. DSPI SCK Duty Cycle (continued)**

| DBR | CPHA | PBR | SCK Duty Cycle |
|---|---|---|---|
| 1 | 0 | 11 | 43/57 |
| 1 | 1 | 00 | 50/50 |
| 1 | 1 | 01 | 66/33 |
| 1 | 1 | 10 | 60/40 |
| 1 | 1 | 11 | 57/43 |

0 The baud rate is computed normally with a 50/50 duty cycle.
1 The baud rate is doubled with the duty cycle depending on the Baud Rate Prescaler.

**1–4 FMSZ**

Frame Size

The number of bits transferred per frame is equal to the FMSZ value plus 1. Regardless of the transmission mode, the minimum valid frame size value is 4. There is a constraint on the minimum allowable Frame size, which depends on the ratio of the Register Read/Write clock to the Protocol clock. The minimum Frame size (FMSZ + 1) is determined by the following equation. Upper Ceiling must be applied for non-integer values.

Minimum Frame Size = ( (4 x $f_p$ ) + (3 x $f_r$ ) ) / (n x $f_r$ )

$f_p$ = Protocol clock frequency

$f_r$ = Register interface clock frequency

n = (PBR x BR) / (1 + DBR) = Multiple of protocol clock required to get Baud (SCK) Clock

However, the minimum frame size can never be less than 4. There is no constraint on the maximum programmable frame size. Also note that `fp' can be equal to, less than or greater than `fr' purely based on user requirement.

When the DSPI operates in TSB mode, the FMSZ field value plus 1 is equal to the data frame bit number, where control of the PCS assertion switches from the DSICR0 to the DSICR1 register. If TSB Mode is operated when DSI is in 64-bit mode, then DSICR0[FMSZ4] also affects the switching of PCS assertion control.

**5 CPOL**

Clock Polarity

Selects the inactive state of the Serial Communications Clock (SCK). This bit is used in both master and slave mode. For successful communication between serial devices, the devices must have identical clock polarities. When the Continuous Selection Format is selected, switching between clock polarities without stopping the module can cause errors in the transfer due to the peripheral device interpreting the switch of clock polarity as a valid clock edge.

**NOTE:** In case of Continuous SCK mode, when the module goes in low power mode(disabled), inactive state of SCK is not guaranteed.

0 The inactive state value of SCK is low.
1 The inactive state value of SCK is high.

**6 CPHA**

Clock Phase

Selects which edge of SCK causes data to change and which edge causes data to be captured. This bit is used in both master and slave mode. For successful communication between serial devices, the devices

*Table continues on the next page...*

## DSPIx_CTAR*n* field descriptions (continued)

| Field | Description |
|---|---|
| | must have identical clock phase settings. In Continuous SCK mode or TSB mode, the bit value is ignored and the transfers are done as if the CPHA bit is set to 1.<br><br>0 Data is captured on the leading edge of SCK and changed on the following edge.<br>1 Data is changed on the leading edge of SCK and captured on the following edge. |
| 7<br>LSBFE | LSB First<br><br>Specifies whether the LSB or MSB of the frame is transferred first. When the DSPI operates in TSB configuration, this bit should be set to be compliant to the MSC specification.<br><br>0 Data is transferred MSB first.<br>1 Data is transferred LSB first. |
| 8–9<br>PCSSCK | PCS to SCK Delay Prescaler<br><br>Selects the prescaler value for the delay between assertion of PCS and the first edge of the SCK. See the CSSCK field description for information on how to compute the PCS to SCK Delay. Refer PCS to SCK Delay ($t_{CSC}$ ) for more details. In the TSB mode, the PCSSCK field has no effect.<br><br>00 PCS to SCK Prescaler value is 1.<br>01 PCS to SCK Prescaler value is 3.<br>10 PCS to SCK Prescaler value is 5.<br>11 PCS to SCK Prescaler value is 7. |
| 10–11<br>PASC | After SCK Delay Prescaler<br><br>Selects the prescaler value for the delay between the last edge of SCK and the negation of PCS. See the ASC field description for information on how to compute the After SCK Delay. In the TSB mode, the PASC field has no effect. Refer After SCK Delay ($t_{ASC}$ ) for more details.<br><br>00 Delay after Transfer Prescaler value is 1.<br>01 Delay after Transfer Prescaler value is 3.<br>10 Delay after Transfer Prescaler value is 5.<br>11 Delay after Transfer Prescaler value is 7. |
| 12–13<br>PDT | Delay after Transfer Prescaler<br><br>Selects the prescaler value for the delay between the negation of the PCS signal at the end of a frame and the assertion of PCS at the beginning of the next frame. The PDT field is only used in master mode. In the TSB mode the PDT field defines two most-signficant bits of the Delay after Transfer. See the DT field description for details on how to compute the Delay after Transfer. Refer Delay after Transfer ($t_{DT}$ ) for more details.<br><br>00 Delay after Transfer Prescaler value is 1.<br>01 Delay after Transfer Prescaler value is 3.<br>10 Delay after Transfer Prescaler value is 5.<br>11 Delay after Transfer Prescaler value is 7. |
| 14–15<br>PBR | Baud Rate Prescaler<br><br>Selects the prescaler value for the baud rate. This field is used only in master mode. The baud rate is the frequency of the SCK. The protocol clock is divided by the prescaler value before the baud rate selection takes place. See the BR field description for details on how to compute the baud rate.<br><br>00 Baud Rate Prescaler value is 2.<br>01 Baud Rate Prescaler value is 3. |

*Table continues on the next page...*

## DSPIx_CTAR*n* field descriptions (continued)

| Field | Description |
|---|---|
| | 10 Baud Rate Prescaler value is 5.<br>11 Baud Rate Prescaler value is 7. |
| 16–19<br>CSSCK | PCS to SCK Delay Scaler<br><br>Selects the scaler value for the PCS to SCK delay. This field is used only in master mode. In the TSB mode the field has no effect. The PCS to SCK Delay is the delay between the assertion of PCS and the first edge of the SCK. The delay is a multiple of the protocol clock period, and it is computed according to the following equation:<br><br>$t_{CSC} = (1/f_P) \times PCSSCK \times CSSCK$.<br><br>The following table lists the delay scaler values.<br><br>**Table 40-6. Delay Scaler Encoding**<br><br>**Field Value** / **Delay Scaler Value**<br>0000 / 2<br>0001 / 4<br>0010 / 8<br>0011 / 16<br>0100 / 32<br>0101 / 64<br>0110 / 128<br>0111 / 256<br>1000 / 512<br>1001 / 1024<br>1010 / 2048<br>1011 / 4096<br>1100 / 8192<br>1101 / 16384<br>1110 / 32768<br>1111 / 65536<br><br>Refer PCS to SCK Delay ($t_{CSC}$) for more details. |
| 20–23<br>ASC | After SCK Delay Scaler<br><br>Selects the scaler value for the After SCK Delay. This field is used only in master mode. The After SCK Delay is the delay between the last edge of SCK and the negation of PCS. The delay is a multiple of the protocol clock period, and it is computed according to the following equation:<br><br>$t_{ASC} = (1/f_P) \times PASC \times ASC$<br><br>See Delay Scaler Encoding table in CTARn[CSSCK] bit field description for scaler values. Refer After SCK Delay ($t_{ASC}$) for more details. |
| 24–27<br>DT | Delay After Transfer Scaler<br><br>Selects the Delay after Transfer Scaler. This field is used only in master mode. The Delay after Transfer is the time between the negation of the PCS signal at the end of a frame and the assertion of PCS at the beginning of the next frame. |

*Table continues on the next page...*

**DSPIx_CTAR*n* field descriptions (continued)**

| Field | Description |
|---|---|
| | In the Continuous Serial Communications Clock operation, the DT value is fixed to one SCK clock period, The Delay after Transfer is a multiple of the protocol clock period, and it is computed according to the following equation:<br>$t_{DT} = (1/f_P) \times PDT \times DT$<br>See Delay Scaler Encoding table in CTARn[CSSCK] bit field description for scaler values.<br>In the TSB mode, the Delay after Transfer is equal to a number formed by concatenation of the PDT and DT fields plus 1 of the SCK clock periods. Refer Delay after Transfer ($t_{DT}$) for more details. |
| 28–31<br>BR | Baud Rate Scaler<br>Selects the scaler value for the baud rate. This field is used only in master mode. The prescaled protocol clock is divided by the Baud Rate Scaler to generate the frequency of the SCK. The baud rate is computed according to the following equation:<br>SCK baud rate = ($f_P$ /PBR) x [(1+DBR)/BR]<br>The following table lists the baud rate scaler values. |

**Table 40-7. Baud Rate Scaler**

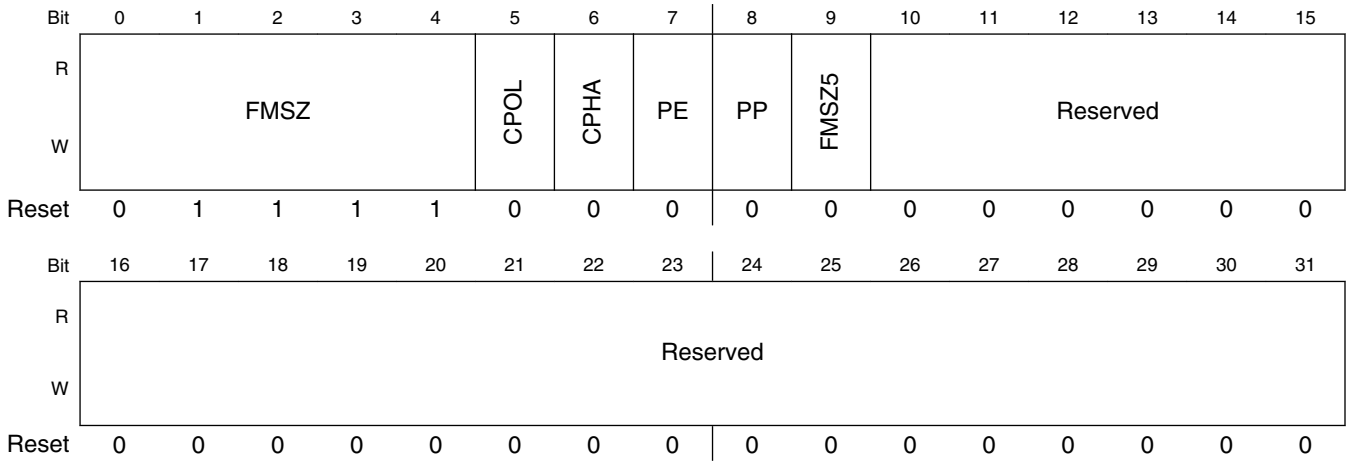
| CTARn[BR] | Baud Rate Scaler Value |
|---|---|
| 0000 | 2 |
| 0001 | 4 |
| 0010 | 6 |
| 0011 | 8 |
| 0100 | 16 |
| 0101 | 32 |
| 0110 | 64 |
| 0111 | 128 |
| 1000 | 256 |
| 1001 | 512 |
| 1010 | 1024 |
| 1011 | 2048 |
| 1100 | 4096 |
| 1101 | 8192 |
| 1110 | 16384 |
| 1111 | 32768 |

## 40.4.4 Clock and Transfer Attributes Register (In Slave Mode) (DSPIx_CTAR*n*_SLAVE)

When the module is configured as an SPI bus slave, the CTAR0 register is used.

When the module is configured as a DSI master, the DSICTAS field in the DSI Configuration Register 0 (DSICR0), selects which of the CTAR register is used. When the module is configured as a DSI bus slave, the CTAR1 register is used.

Address: Base address + Ch offset + (4d × i), where i=0d to 1d




## DSPIx_CTAR*n*_SLAVE field descriptions

| Field | Description |
|---|---|
| 0–4<br>FMSZ | Frame Size<br><br>The number of bits transferred per frame is equal to the concatenated field {CTARn_SLAVE[FMSZ5], CTARn_SLAVE[FMSZ]} value plus 1. Note that the minimum valid value of frame size is 4. |
| 5<br>CPOL | Clock Polarity<br><br>Selects the inactive state of the Serial Communications Clock (SCK).<br><br>**NOTE:** In case of Continuous SCK mode, when the module goes in low power mode(disabled), inactive state of SCK is not guaranteed.<br><br>0 The inactive state value of SCK is low.<br>1 The inactive state value of SCK is high. |
| 6<br>CPHA | Clock Phase<br><br>Selects which edge of SCK causes data to change and which edge causes data to be captured. This bit is used in both master and slave mode. For successful communication between serial devices, the devices must have identical clock phase settings. In Continuous SCK mode or TSB mode, the bit value is ignored and the transfers are done as if the CPHA bit is set to 1.<br><br>0 Data is captured on the leading edge of SCK and changed on the following edge.<br>1 Data is changed on the leading edge of SCK and captured on the following edge. |
| 7<br>PE | Parity Enable<br><br>Enables parity bit transmission and reception for the frame.<br><br>0 No parity bit included/checked.<br>1 Parity bit is transmitted instead of last data bit in frame, parity checked for received frame. |
| 8<br>PP | Parity Polarity<br><br>Controls polarity of the parity bit transmitted and checked. |

*Table continues on the next page...*

## DSPIx_CTAR*n*_SLAVE field descriptions (continued)

| Field | Description |
|---|---|
| | 0 Even Parity: the number of 1 bits in the transmitted frame is even. The SR[SPEF] bit is set if the number of 1 bits is odd in the received frame.<br>1 Odd Parity: the number of 1 bits in the transmitted frame is odd. The SR[SPEF] bit is set if the number of 1 bits is even in the received frame. |
| 9<br>FMSZ5 | MSB of Frame Size when DSI is used in 64-bit Mode<br><br>The number of bits transferred per frame is equal to the concatenated field {CTARn_SLAVE[FMSZ[5]], CTARn_SLAVE[FMSZ]} value plus 1. |
| 10–31<br>Reserved | This field is reserved. |

## 40.4.5 Status Register (DSPIx_SR)

SR contains status and flag bits. The bits reflect the status of the module and indicate the occurrence of events that can generate interrupt or DMA requests. Software can clear flag bits in the SR by writing a 1 to them. Writing a 0 to a flag bit has no effect. This register may not be writable in Module Disable mode due to the use of power saving mechanisms.

Address: Base address + 2Ch offset

| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R | TCF | TXRXS | SPITCF | EOQF | TFUF | DSITCF | TFFF | BSYF | CMDTCF | DPEF | SPEF | DDIF | RFOF | TFIWF | RFDF | CMDFFF |
| W | w1c | | w1c | w1c | w1c | w1c | w1c | | w1c | w1c | w1c | w1c | w1c | w1c | w1c | w1c |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| Bit | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R | TXCTR | | | | TXNXTPTR | | | | RXCTR | | | | POPNXTPTR | | | |
| W | | | | | | | | | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**DSPIx_SR field descriptions**

| Field | Description |
|---|---|
| 0<br>TCF | Transfer Complete Flag<br><br>Indicates that all bits in a frame have been shifted out. TCF remains set until it is cleared by writing a 1 to it.<br><br>0 Transfer not complete.<br>1 Transfer complete. |

*Table continues on the next page...*

## DSPIx_SR field descriptions (continued)

| Field | Description |
|---|---|
| 1<br>TXRXS | TX and RX Status<br><br>Reflects the run status of the module.<br><br>0 Transmit and receive operations are disabled (The module is in Stopped state).<br>1 Transmit and receive operations are enabled (The module is in Running state). |
| 2<br>SPITCF | SPI Frame Transfer Complete Flag.<br><br>This bit functions similar to TCF except that this bit is asserted only on SPI frame transmission completion in the following modes:<br>• CSI<br>• TSB in CSI<br>• ITSB<br><br>This flag will not be asserted when the module is used in SPI or DSI Modes.<br><br>0 SPI frame transfer is not complete.<br>1 SPI frame transfer is complete. |
| 3<br>EOQF | End of Queue Flag<br><br>Indicates that the last entry in a queue has been transmitted when the module is in Master mode. The EOQF bit is set when the TX FIFO entry has the EOQ bit set in the command halfword and the end of the transfer is reached. The EOQF bit remains set until cleared by writing a 1 to it. When the EOQF bit is set, the TXRXS bit is automatically cleared.<br><br>0 EOQ is not set in the executing command.<br>1 EOQ is set in the executing SPI command. |
| 4<br>TFUF | Transmit FIFO Underflow Flag<br><br>Indicates an underflow condition in the TX FIFO. The transmit underflow condition is detected only for DSPI blocks operating in Slave mode and SPI configuration. TFUF is set when the TX FIFO of the module operating in SPI Slave mode is empty and an external SPI master initiates a transfer. The TFUF bit remains set until cleared by writing 1 to it.<br><br>0 No TX FIFO underflow.<br>1 TX FIFO underflow has occurred. |
| 5<br>DSITCF | DSI Frame Transfer Complete Flag.<br><br>This bit functions similar to TCF except that this bit is asserted only on DSI frame transmission completion in the following modes:<br>• CSI<br>• TSB in CSI<br>• ITSB<br><br>This flag will not be asserted when the module is in SPI or DSI mode.<br><br>0 DSI frame transfer is not complete.<br>1 DSI frame transfer is complete. |
| 6<br>TFFF | Transmit FIFO Fill Flag<br><br>Indicates whether there is an available location to be filled in the FIFO. Either a DMA request or an interrupt indication can be used to add another entry to the FIFO. Note that this bit is set if at least one location is free in the FIFO. The TFFF bit can be cleared by writing 1 to it or by acknowledgement from the DMA controller to the TX FIFO full request. |

*Table continues on the next page...*

## DSPIx_SR field descriptions (continued)

| Field | Description |
|---|---|
| | **NOTE:** The reset value of this bit is 0 when the module is disabled,(MCR[MDIS]=1).<br><br>0 TX FIFO is full.<br>1 TX FIFO is not full. |
| 7<br>BSYF | Busy Flag.<br><br>This bit is valid only when DSPI_MCR[XSPI] is enabled. Indicates that the current Command Frame is being used for transmiting multiple data frames. This bit is not set for the last Data Frame of a Cyclic command Transfer or when DSPI_CTARE[DTCP] = 1. Refer Command First In First Out (CMD FIFO) Buffering Mechanism for more details.<br><br>0 No Cyclic Command Transfer in Progress.<br>1 Cyclic Command Transfer is in progress. Current Data Frame is not the last data frame for on-going cyclic command transfer.. |
| 8<br>CMDTCF | Command Transfer Complete Flag.<br><br>Indicates that the last Data frame for the current Cyclic Command has been transmitted. Hence this bit is set only for the last Data Frame of a Cyclic Command Transfer or when DSPI_CTARE[DTCP] = 1. The bit remains set until it is cleared by writing a ‘1’ to it.<br><br>0 Data Transfer by current Command not complete.<br>1 Data Transfer by current Command is complete. |
| 9<br>DPEF | DSI Parity Error Flag<br><br>Indicates that a DSI frame with parity error had been received. The bit remains set until it is cleared by writing a 1 to it.<br><br>0 No parity error.<br>1 Parity error has occurred. |
| 10<br>SPEF | SPI Parity Error Flag<br><br>Indicates that a SPI frame with parity error had been received. The bit remains set until it is cleared by writing a 1 to it.<br><br>0 No parity error.<br>1 Parity error has occurred. |
| 11<br>DDIF | DSI Data Received with Active Bits<br><br>Indicates that a DSI frame had been received with bits selected by DIMR with active polarity, defined by the DPIR register. The bit remains set until it is cleared by writing a 1 to it.<br><br>0 No DSI data with active bits was received.<br>1 DSI data with active bits was received. |
| 12<br>RFOF | Receive FIFO Overflow Flag<br><br>Indicates an overflow condition in the RX FIFO. The field is set when the RX FIFO and shift register are full and a transfer is initiated. The bit remains set until it is cleared by writing a 1 to it.<br><br>0 No Rx FIFO overflow.<br>1 Rx FIFO overflow has occurred. |
| 13<br>TFIWF | Tranmit FIFO Invalid Write Flag |

*Table continues on the next page...*

## DSPIx_SR field descriptions (continued)

| Field | Description |
|---|---|
| | Indicates Data Write on TX FIFO while CMD FIFO is empty. Without a Command, the Data entries present in TXFIFO are invalid. This bit remains set until it is cleared by writing a '1' to it.<br><br>0 No Invalid Data present in TX FIFO.<br>1 Invalid Data present in TX FIFO since CMD FIFO is empty. |
| 14<br>RFDF | Receive FIFO Drain Flag<br><br>Indicates whether there is an available location to be drained from the FIFO. Either a DMA request or an interrupt indication can be used to read from the FIFO. Note that this bit is set if at least one location can be read from the FIFO. The RFDF bit can be cleared by writing 1 to it or by acknowledgement from the DMA controller when the RX FIFO is empty.<br><br>0 RX FIFO is empty.<br>1 RX FIFO is not empty. |
| 15<br>CMDFFF | Command FIFO Fill Flag<br><br>Indicates whether there is an available location to be filled in the FIFO. Either a DMA request or an interrupt indication can be used to add another entry to the FIFO. Note that this bit is set if at least one location is free in the FIFO. The CMDFFF is cleared by writing a '1' to it or by acknowledgement from the DMA controller to the CMD FIFO full request.<br><br>0 CMD FIFO is full.<br>1 CMD FIFO is not full. |
| 16–19<br>TXCTR | TX FIFO Counter<br><br>Indicates the number of valid entries in the TX FIFO. The TXCTR is incremented every time the PUSHR is written. The TXCTR is decremented every time an SPI command is executed and the SPI data is transferred to the shift register. |
| 20–23<br>TXNXTPTR | Transmit Next Pointer<br><br>Indicates which TX FIFO entry is transmitted during the next transfer. The TXNXTPTR field is updated every time SPI data is transferred from the TX FIFO to the shift register. |
| 24–27<br>RXCTR | RX FIFO Counter<br><br>Indicates the number of entries in the RX FIFO. The RXCTR is decremented every time the POPR is read. The RXCTR is incremented every time data is transferred from the shift register to the RX FIFO. |
| 28–31<br>POPNXTPTR | Pop Next Pointer<br><br>Contains a pointer to the RX FIFO entry to be returned when the POPR is read. The POPNXTPTR is updated when the POPR is read. |

## 40.4.6 DMA/Interrupt Request Select and Enable Register (DSPIx_RSER)

RSER controls DMA and interrupt requests. Do not write to the RSER while the module is in the Running state.

Address: Base address + 30h offset

| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R<br>W | TCF_RE | CMDFFF_RE | SPITCF_RE | EOQF_RE | TFUF_RE | DSITCF_RE | TFFF_RE | TFFF_DIRS | CMDTCF_RE | DPEF_RE | SPEF_RE | DDIF_RE | RFOF_RE | TFIWF_RE | RFDF_RE | RFDF_DIRS |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R | CMDFFF_DIRS | DDIF_DIRS | 0 | | | | | | | | | | | | | |
| W | | | | | | | | | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**DSPIx_RSER field descriptions**

| Field | Description |
|---|---|
| 0<br>TCF_RE | Transmission Complete Request Enable<br><br>Enables TCF flag in the SR to generate an interrupt request.<br><br>0 TCF interrupt requests are disabled.<br>1 TCF interrupt requests are enabled. |
| 1<br>CMDFFF_RE | Command FIFO Fill Flag Request Enable.<br><br>Enables the CMDFFF flag in the SR to generate a request. The CMDFFF_DIRS bit selects between generating an interrupt request or a DMA request.<br><br>0 CMDFFF interrupts or DMA requests are disabled.<br>1 CMDFFF interrupts or DMA requests are enabled. |
| 2<br>SPITCF_RE | SPI Frame Transmission Complete Request Enable.<br><br>The SPITCF_RE bit enables SPITCF flag in the SR to generate an interrupt request.<br><br>0 SPITCF interrupt requests are disabled.<br>1 SPITCF interrupt requests are enabled. |
| 3<br>EOQF_RE | Finished Request Enable<br><br>Enables the EOQF flag in the SR to generate an interrupt request. |

*Table continues on the next page...*

## DSPIx_RSER field descriptions (continued)

| Field | Description |
|---|---|
| | 0 EOQF interrupt requests are disabled.<br>1 EOQF interrupt requests are enabled. |
| 4<br>TFUF_RE | Transmit FIFO Underflow Request Enable<br><br>Enables the TFUF flag in the SR to generate an interrupt request.<br><br>0 TFUF interrupt requests are disabled.<br>1 TFUF interrupt requests are enabled. |
| 5<br>DSITCF_RE | DSI Frame Transmission Complete Request Enable.<br><br>The DSITCF_RE bit enables DSITCF flag in the SR to generate an interrupt request.<br><br>0 DSITCF interrupt requests are disabled.<br>1 DSITCF interrupt requests are enabled. |
| 6<br>TFFF_RE | Transmit FIFO Fill Request Enable<br><br>Enables the TFFF flag in the SR to generate a request. The TFFF_DIRS bit selects between generating an interrupt request or a DMA request.<br><br>0 TFFF interrupts or DMA requests are disabled.<br>1 TFFF interrupts or DMA requests are enabled. |
| 7<br>TFFF_DIRS | Transmit FIFO Fill DMA or Interrupt Request Select<br><br>Selects between generating a DMA request or an interrupt request. When SR[TFFF] and RSER[TFFF_RE] are set, this field selects between generating an interrupt request or a DMA request.<br><br>0 TFFF flag generates interrupt requests.<br>1 TFFF flag generates DMA requests. |
| 8<br>CMDTCF_RE | Command Transmission Complete Request Enable.<br><br>The CMDTCF_RE bit enables CMDTCF flag in the SR to generate an interrupt request.<br><br>0 CMDTCF interrupt requests are disabled.<br>1 CMDTCF interrupt requests are enabled. |
| 9<br>DPEF_RE | DSI Parity Error Request Enable<br><br>Enables the DPEF flag in the SR to generate an interrupt request.<br><br>0 DPEF interrupt requests are disabled.<br>1 DPEF interrupt requests are enabled. |
| 10<br>SPEF_RE | SPI Parity Error Request Enable<br><br>Enables the SPEF flag in the SR to generate an interrupt request.<br><br>0 SPEF interrupt requests are disabled.<br>1 SPEF interrupt requests are enabled. |
| 11<br>DDIF_RE | DSI data received with active bits Request Enable<br><br>Enables the SR[DDIF] flag to generate an interrupt or DMA request.<br><br>0 Disabled<br>1 Enabled |

*Table continues on the next page...*

## DSPIx_RSER field descriptions (continued)

| Field | Description |
|---|---|
| 12<br>RFOF_RE | Receive FIFO Overflow Request Enable<br><br>Enables the RFOF flag in the SR to generate an interrupt request.<br><br>0 RFOF interrupt requests are disabled.<br>1 RFOF interrupt requests are enabled. |
| 13<br>TFIWF_RE | Transmit FIFO Invalid Write Request Enable.<br><br>Enables the TFIWF flag in the SR to generate an interrupt request.<br><br>0 TFIWF interrupt requests are disabled.<br>1 TFIWF interrupt requests are enabled. |
| 14<br>RFDF_RE | Receive FIFO Drain Request Enable<br><br>Enables the RFDF flag in the SR to generate a request. The RFDF_DIRS bit selects between generating an interrupt request or a DMA request.<br><br>0 RFDF interrupt or DMA requests are disabled.<br>1 RFDF interrupt or DMA requests are enabled. |
| 15<br>RFDF_DIRS | Receive FIFO Drain DMA or Interrupt Request Select<br><br>Selects between generating a DMA request or an interrupt request. When the RFDF flag bit in the SR is set, and the RFDF_RE bit in the RSER is set, the RFDF_DIRS bit selects between generating an interrupt request or a DMA request.<br><br>0 Interrupt request.<br>1 DMA request. |
| 16<br>CMDFFF_DIRS | Command FIFO Fill DMA or Interrupt Request Select<br><br>Selects between generating a DMA request or an interrupt request. When the CMDFFF flag bit in the SR is set, and the CMDFFF_RE bit in the RSER is set, the CMDFFF_DIRS bit selects between generating an interrupt indication or a DMA request.<br><br>0 CMDFFF flag generates interrupt requests.<br>1 CMDFFF flag generates DMA requests. |
| 17<br>DDIF_DIRS | DSI data received with active bits - DMA or Interrupt Request Select.<br><br>Selects between generating a DMA request or an interrupt request. When the DDIF flag bit in the SR is set, and the DDIF_RE bit in the RSER is set, the DDIF_DIRS bit selects between generating an interrupt request or a DMA request.<br><br>0 Interrupt request.<br>1 DMA request. |
| 18–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |

## 40.4.7 PUSH TX FIFO Register In Master Mode (DSPIx_PUSHR)

Specifies data to be transferred to the TX FIFO and CMD FIFO. User must write 16-bits data into TXDATA field. An 8- or 16-bit write access to the TXDATA field transfers 16 bits of data bus to the TX FIFO. A write access to the command fields transfers the 16 bits of command information to the CMD FIFO. In Master mode, the register transfers 16 bits of data to the TX FIFO and 16 bits of command information to the CMD FIFO.

If Extended SPI Mode is disabled (MCR[XSPI]), the TX FIFO and CMD FIFO must be filled simultaneously. In other words, you must perform write accesses to both the data and command fields for every PUSHR operation. With Extended SPI Mode disabled and both the TX FIFO and CMD FIFO are written to and read from simultaneously, they behave as a single 32 bit FIFO. When Extended SPI mode is enabled (MCR[XSPI]), the TX FIFO and CMD FIFO can be written to independently.

A read access of PUSHR returns the topmost TX FIFO and CMD FIFO entries concatenated.

When the module is disabled, writing to this register does not update the FIFO. Therefore, any reads performed while the module is disabled return the last PUSHR write performed while the module was still enabled.

Address: Base address + 34h offset

| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R<br>W | CONT | CTAS | | | EOQ | CTCNT | PE_MASC | PP_MCSC | Reserved | | PCS | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R<br>W | TXDATA | | | | | | | | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**DSPIx_PUSHR field descriptions**

| Field | Description |
|---|---|
| 0<br>CONT | Continuous Peripheral Chip Select Enable<br><br>Selects a continuous selection format. The bit is used in SPI Master mode. The bit enables the selected PCS signals to remain asserted between transfers. |

*Table continues on the next page...*

## DSPIx_PUSHR field descriptions (continued)

| Field | Description |
|---|---|
| | 0 Return PCSn signals to their inactive state between transfers.<br>1 Keep PCSn signals asserted between transfers. |
| 1–3<br>CTAS | Clock and Transfer Attributes Select<br><br>Selects which CTAR to use in master mode to specify the transfer attributes for the associated SPI frame. In SPI Slave mode, CTAR0 is used. See the chip specific section for details to determine how many CTARs this device has. You should not program a value in this field for a register that is not present.<br><br>000 CTAR0<br>001 CTAR1<br>010 CTAR2<br>011 CTAR3<br>100 CTAR4<br>101 CTAR5<br>110 Reserved<br>111 Reserved |
| 4<br>EOQ | End Of Queue<br><br>Host software uses this bit to signal to the module that the current SPI transfer is the last in a queue. At the end of the transfer, the EOQF bit in the SR is set.<br><br>0 The SPI data is not the last data to transfer.<br>1 The SPI data is the last data to transfer. |
| 5<br>CTCNT | Clear Transfer Counter<br><br>Clears the TCNT field in the TCR register. The TCNT field is cleared before the module starts transmitting the current SPI frame.<br><br>0 Do not clear the TCR[TCNT] field.<br>1 Clear the TCR[TCNT] field. |
| 6<br>PE_MASC | Parity Enable or Mask $T_{ASC}$ delay in the current frame<br><br>PE – This bit enables parity bit transmission and parity reception check for the SPI frame.<br>MASC - The current frame has the “after SCK” delay masked if this bit is asserted. See Fast Continuous Selection Format for more details.<br><br>**NOTE:** This bit is used as Mask $T_{ASC}$ in the Fast Continuous PCS mode when MCR[FCPCS] is set.<br><br>0 PE - No parity bit included/checked.<br>MASC - $T_{ASC}$ delay is not masked and the current frame has the after SCK delay.<br>1 PE - Parity bit is transmitted instead of the last data bit in the frame; parity is checked for the received frame.<br>MASC - $T_{ASC}$ delay is masked in the current frame. |
| 7<br>PP_MCSC | Parity Polarity or Mask $T_{CSC}$ delay in the next frame<br><br>PP - It controls the polarity of the parity bit transmitted and checked.<br>MCSC - The next frame has the “PCS to SCK” delay masked if this bit is asserted. See Fast Continuous Selection Format for more details.<br><br>**NOTE:** This bit is used as Mask $T_{CSC}$ in the Fast Continuous PCS mode when MCR[FCPCS] is set. |

*Table continues on the next page...*

**DSPIx_PUSHR field descriptions (continued)**

| Field | Description |
|---|---|
| | 0 PP - Even Parity: the number of 1 bits in the transmitted frame is even. The SR[SPEF] bit is set if the number of 1 bits is odd in the received frame.<br>MCSC - $T_{CSC}$ delay is not masked and the next frame has the PCS to SCK delay.<br>1 PP - Odd Parity: the number of 1 bits in the transmitted frame is odd. The SR[SPEF] bit is set if the number of 1 bits is even in the received frame.<br>MCSC - $T_{CSC}$ delay is masked in the next frame. |
| 8–9<br>Reserved | Always write the reset value to this field.<br>This field is reserved. |
| 10–15<br>PCS | Select which PCS signals are to be asserted for the transfer. Refer to the chip-specific DSPI information for the number of PCS signals used in this chip.<br>0 Negate the PCS[x] signal.<br>1 Assert the PCS[x] signal. |
| 16–31<br>TXDATA | Transmit Data<br>Holds SPI data to be transferred according to the associated SPI command. |

## 40.4.8 PUSH TX FIFO Register In Slave Mode (DSPIx_PUSHR_SLAVE)

Specifies data to be transferred to the TX FIFO in slave mode. An 8- or 16-bit write access to PUSHR transfers the 16-bit TXDATA field to the TX FIFO.

When Extended SPI mode is enabled (MCR[XSPI]), up to 32-bit SPI frames may be queued for transmission or reception.

Address: Base address + 34h offset

| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R | 0 | | | | | | | | | | | | | | | | TXDATA | | | | | | | | | | | | | | | |
| W | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 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 | 0 |

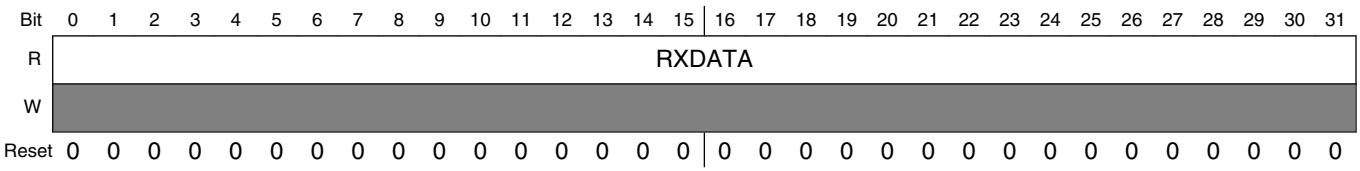
**DSPIx_PUSHR_SLAVE field descriptions**

| Field | Description |
|---|---|
| 0–15<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 16–31<br>TXDATA | Transmit Data<br>Holds SPI data to be transferred according to the associated SPI command. |

## 40.4.9 POP RX FIFO Register (DSPIx_POPR)

POPR is used to read the RX FIFO. Eight- or sixteen-bit read accesses to the POPR have the same effect on the RX FIFO as 32-bit read accesses. A write to this register will generate a Transfer Error.

Address: Base address + 38h offset



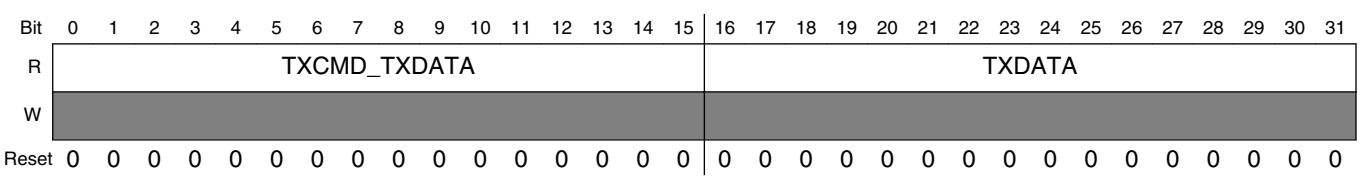

**DSPIx_POPR field descriptions**

| Field | Description |
|---|---|
| 0–31<br>RXDATA | Received Data<br><br>Contains the SPI data from the RX FIFO entry to which the Pop Next Data Pointer points. |

## 40.4.10 Transmit FIFO Registers (DSPIx_TXFR*n*)

TXFRn registers provide visibility into the TX FIFO for debugging purposes. Each register is an entry in the TX FIFO. The registers are read-only and cannot be modified. Reading the TXFRx registers does not alter the state of the TX FIFO. When the module is operating in Extended SPI mode, reading TXFRn registers is invalid.

Address: Base address + 3Ch offset + (4d × i), where i=0d to 3d




**DSPIx_TXFR*n* field descriptions**

| Field | Description |
|---|---|
| 0–15<br>TXCMD_<br>TXDATA | Transmit Command or Transmit Data<br><br>In Master mode the TXCMD field contains the command that sets the transfer attributes for the SPI data. In Slave mode, this field is reserved. |

*Table continues on the next page...*

**DSPIx_TXFR*n* field descriptions (continued)**

| Field | Description |
|---|---|
| 16–31<br>TXDATA | Transmit Data<br><br>Contains the SPI data to be shifted out. |

## 40.4.11 Receive FIFO Registers (DSPIx_RXFR*n*)

RXFRn provide visibility into the RX FIFO for debugging purposes. Each register is an entry in the RX FIFO. The RXFR registers are read-only. Reading the RXFRx registers does not alter the state of the RX FIFO. The field MCR[MDIS] must be 0 when RXFR is read.

Address: Base address + 7Ch offset + (4d × i), where i=0d to 3d

| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R | RXDATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 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 | 0 |

**DSPIx_RXFR*n* field descriptions**

| Field | Description |
|---|---|
| 0–31<br>RXDATA | Receive Data<br><br>Contains the received SPI data. |

## 40.4.12 DSI Configuration Register 0 (DSPIx_DSICR0)

Selects various attributes associated with the following configurations:

- DSI
- TSB
- ITSB

Do not write to the DSICR0 while the module is in the Running state.

Address: Base address + BCh offset

| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R | MTOE | FMSZ4 | MTOCNT | | | | | | FMSZ5 | 0 | ITSB | TSBC | TXSS | TPOL | TRRE | CID |
| W | | | | | | | | | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R | DCONT | DSICTAS | | | DMS | PES | PE | PP | DPCSx | | | | | | | |
| W | | | | | | | | | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**DSPIx_DSICR0 field descriptions**

| Field | Description |
|---|---|
| 0<br>MTOE | Multiple Transfer Operation Enable<br><br>Enables multiple DSPIs to be connected in a parallel or serial configuration. The MTOE and TSB bits should not be set simulteneously.<br><br>0 Multiple transfer operation disabled.<br>1 Multiple transfer operation enabled. |
| 1<br>FMSZ4 | MSB of the frame size in master mode.<br><br>If the bit is set, 16 is added to the frame size, as defined by the CTARn[FMSZ] field. The CTARx register is selected by the DSICTAS field. |
| 2–7<br>MTOCNT | Multiple Transfer Operation Count<br><br>Selects the number of bits to be shifted out during a transfer in Multiple Transfer Operation. The field sets the number of SCK cycles that the bus master must generate to complete the transfer. The number of SCK cycles used is one more than the value in the MTOCNT field. The number of SCK cycles defined by MTOCNT must be equal to or greater than the frame size. When TSBC is set, MTOCNT has no effect. |
| 8<br>FMSZ5 | MSB of the frame size in master mode when DSI is used in 64-bit mode. |

*Table continues on the next page...*

## DSPIx_DSICR0 field descriptions (continued)

| Field | Description |
|---|---|
| | If the bit is set, 32 is added to the frame size, as defined by the concatenation of {DSICR0[FMSZ[4]] , CTARn[FMSZ]} fields. The CTARn register is selected by the DSICTAS field. Hence the final concatenated frame size using this bit would be {DSICR0[FMSZ[5]], DSICR0[FMSZ[4]] , CTARn[FMSZ]}. This field is only valid when DSICR1[DSI64E] is enabled. |
| 9<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 10<br>ITSB | Interleaved TSB mode.<br><br>This bit enables the Interleaved TSB mode of operation. When enabled, there is no priority among frames from the SPI/DSI. DSI frames are always sent when either the previous transmission was a SPI frame or if the TX_FIFO is empty. Frames are transmitted on every trigger whose source is selected by the TRG bit setting. The ITSB mode requires the DSICR0[TSBC] bit to be set and should be written only when MCR[HALT] bit is asserted. If ITSB bit is set without setting DSICR0[TSBC], then the module operates in the Normal Mode depending on MCR[DCONF] bit.<br><br>See Interleaved TSB (ITSB) Mode for more details.<br><br>0 Interleaved mode is disabled.<br>1 Interleaved mode is enabled. |
| 11<br>TSBC | Timed Serial Bus Configuration.<br><br>Enables the Timed Serial Bus interface configuration and allows $t_{DT}$ to be programmable.<br><br>0 Timed Serial Bus Configuration disabled.<br>1 Timed Serial Bus Configuration enabled. |
| 12<br>TXSS | Transmit Data Source Select.<br><br>Selects the source of data to be serialized. The source can be either data from host software written to the DSI Alternate Serialization Data Register (ASDR0/1) or Parallel Input pin states latched into the DSI Serialization Data Register (SDR0/1).<br><br>0 Source of serialized data is the SDR.<br>1 Source of serialized data is the ASDR. |
| 13<br>TPOL | Trigger Polarity<br><br>Selects the active edge of the hardware trigger input signal (HT), initiating DSI frames transfer.<br><br>0 Falling edge initiates a transfer.<br>1 Rising edge initiates a transfer. |
| 14<br>TRRE | Trigger Reception Enable<br><br>Enables the module to initiate a DSI frames transfer with an external trigger signal.<br><br>**NOTE:** When TRIG[ENABLE] is set, the TRRE bit is given by TRIG[TRRE] which overrides DSICR0[TRRE].<br><br>0 Trigger signal reception disabled.<br>1 Trigger signal reception enabled. |
| 15<br>CID | Change In Data Transfer Enable<br><br>Enables a change in serialization data to initiate a DSI frames transfer in DSI and CSI configurations. When the CID bit is set, DSI frames are initiated when the current DSI data differs from the previous DSI data shifted out. |

*Table continues on the next page...*

## DSPIx_DSICR0 field descriptions (continued)

| Field | Description |
|---|---|
| | **NOTE:** When TRIG[ENABLE] is set, the CID bit is given by TRIG[CID] which overrides DSICR0[CID]. |
| 16<br>DCONT | DSI Continuous Peripheral Chip Select Enable<br><br>Enables the PCS signals to remain asserted between transfers. The DCONT bit affects the PCS signals only in DSI master mode. When the TSBC bit is set, DCONT has no effect.<br><br>0 Return Peripheral Chip Select signals to their inactive state after the transfer is complete.<br>1 Keep Peripheral Chip Select signals asserted after transfer is complete. |
| 17–19<br>DSICTAS | DSI Clock and Transfer Attributes Select<br><br>Selects the CTAR register to use to provide transfer attributes for DSI frames. The DSICTAS field is used in DSI master mode. In DSI slave mode, CTAR1 is always selected. |
| 20<br>DMS | Data Match Stop<br><br>If set, stops DSI frame transmissions if DDIF flag is set in the SR register.<br><br>0 DDIF flag does not have an effect on DSI frame transmissions.<br>1 DDIF flag stops DSI frame transmissions. |
| 21<br>PES | Parity Error Stop<br><br>If set, stops DSI operation if the parity error happened in the received DSI frame.<br><br>0 Parity error does not stop DSI frame transmissions.<br>1 Parity error stops all DSI frame transmissions. |
| 22<br>PE | Parity Enable<br><br>Enables parity bit transmission and parity reception check for the DSI frames.<br><br>0 No parity bit included/checked.<br>1 Parity bit is transmitted instead of the last data bit in the frame; parity is checked for the received frame. |
| 23<br>PP | Parity Polarity<br><br>Controls the polarity of the parity bit transmitted and checked.<br><br>0 Even Parity: the number of 1 bits in the transmitted frame is even. The SR[DPEF] bit is set if in the received frame number of 1 bits is odd.<br>1 Odd Parity: the number of 1 bits in the transmitted frame is odd. The SR[DPEF] bit is set if in the received frame number of 1 bits is even. |
| 24–31<br>DPCSx | DSI Peripheral Chip Select 0–7<br><br>Selects which of the PCS signals to assert during a DSI master mode transfer.<br><br>0 Negate PCS[x].<br>1 Assert PCS[x]. |

## 40.4.13 DSI Serialization Data Register 0 (DSPIx_SDR0)

SDR0 contains the states of the 32 LSB Parallel Input signals. The states of the Parallel Input signals are latched into the SDR0 on the rising edge of every bus clock. The SDR0 is read-only. When the TXSS bit in the DSICR0 is cleared, the data in the SDR0 is used as the source of the DSI frames. When DSI 64-bit mode is enabled (DSICR1[DSI64E]), the concatenation of {SDR1, SDR0} provides the 64-bit Serialization Data value.

Address: Base address + C0h offset

| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R | SER_DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 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 | 0 |

**DSPIx_SDR0 field descriptions**

| Field | Description |
|---|---|
| 0–31<br>SER_DATA | Serialized Data<br><br>Contains the signal states of the 32 LSB Parallel Input signals. |

## 40.4.14 DSI Alternate Serialization Data Register 0 (DSPIx_ASDR0)

ASDR0 is used by host software to write the 32 LSB of the data to be serialized. When the TXSS bit in the DSICR0 is set, the data in the ASDR0 is the source of the DSI frames. Writes to the ASDR0 take effect on the next frame boundary. When DSI 64-bit mode is enabled (DSICR1[DSI64E]), the concatenation of {ASDR1, ASDR0} provides the 64-bit Alternate Serialization Data value.

Address: Base address + C4h offset

| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R<br>W | ASER_DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 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 | 0 |

**DSPIx_ASDR0 field descriptions**

| Field | Description |
|---|---|
| 0–31<br>ASER_DATA | Alternate Serialized Data<br><br>Holds the alternate 32 LSB of the data to be serialized. |

## 40.4.15 DSI Transmit Comparison Register 0 (DSPIx_COMPR0)

COMPR0 holds a copy of the 32 LSB of the last transmitted DSI data. COMPR0 is read-only. DSI data is transferred to this register as it is loaded into the TX Shift Register. When DSI 64-bit mode is enabled (DSICR1[DSI64E]), the concatenation of {COMPR1, COMPR0} provides the 64-bit Transmit Comparison value.

Address: Base address + C8h offset

| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R | COMP_DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 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 | 0 |

**DSPIx_COMPR0 field descriptions**

| Field | Description |
|---|---|
| 0–31<br>COMP_DATA | Compare Data<br><br>Holds the 32 LSB of the last serialized DSI data. |

## 40.4.16 DSI Deserialization Data Register 0 (DSPIx_DDR0)

DDR0 holds the 32 LSB signal states for the Parallel Output signals. DDR0 is read-only and host software can read data from incoming DSI frames. When DSI 64-bit mode is enabled (DSICR1[DSI64E]), the concatenation of {DDR1, DDR0} provides the 64-bit Deserialization Data value.

Address: Base address + CCh offset

| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R | DESER_DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 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 | 0 |

**DSPIx_DDR0 field descriptions**

| Field | Description |
|---|---|
| 0–31<br>DESER_DATA | Deserialized Data<br><br>Holds 32 LSB of deserialized data that is presented as signal states to the Parallel Output signals. |

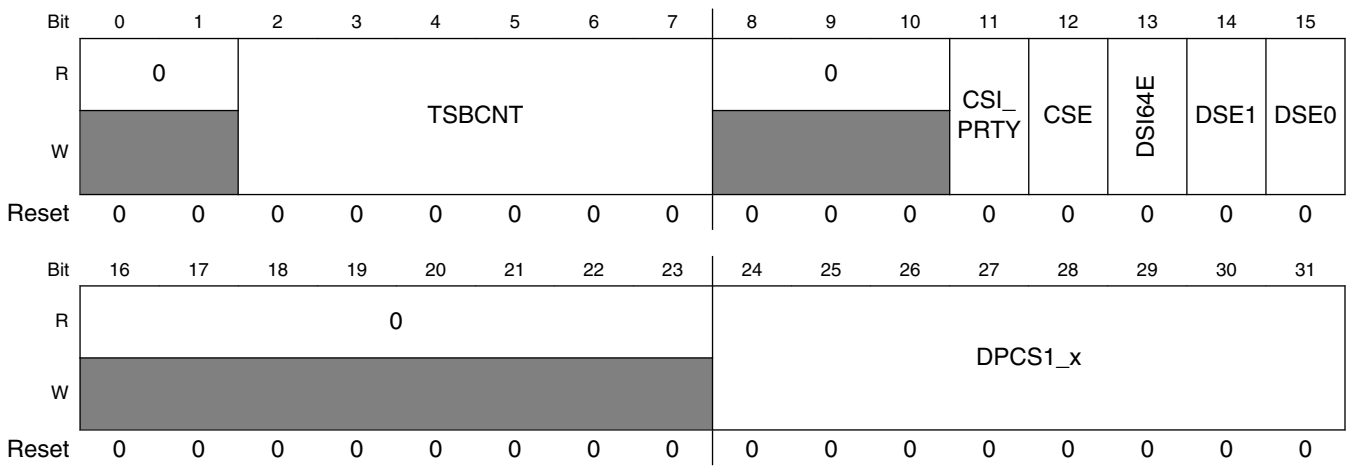
## 40.4.17 DSI Configuration Register 1 (DSPIx_DSICR1)

Selects various attributes associated with the following configurations:

- DSI
- TSB
- ITSB

The user must not write to the DSICR1 while the module is in the Running state.

Address: Base address + D0h offset




**DSPIx_DSICR1 field descriptions**

| Field | Description |
|---|---|
| 0–1<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 2–7<br>TSBCNT | Timed Serial Bus Operation Count<br><br>When TSB is enabled (DSICR0[TSBC]), TSBCNT defines the length of the data frame. When DSI 64-bit mode is enabled (DSI64E), the length value can vary from 3 to 63 (000011b to 111111b); otherwise, it can vary from 3 to 31 (000011b to 011111b). The length value specifies the number of data bits to be shifted out during a transfer in TSB mode. The number of data bits in the data frame is one more than the value in the TSBCNT field. |
| 8–10<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 11<br>CSI_PRTY | CSI Priority<br><br>This field allows the user to configure the priority of SPI and DSI frames when module is operating in CSI Mode. This is not applicable when DSICR0[ITSB] is enabled.<br><br>0 SPI frames has priority over DSI frames.<br>1 DSI frames have priority over SPI frames. |
| 12<br>CSE | Command Select Enable |

*Table continues on the next page...*

## DSPIx_DSICR1 field descriptions (continued)

| Field | Description |
|---|---|
| | When the DSICR0[ITSB] is set, the CSE bit controls insertion of command selection bit (Logic 1) in the beginning of the SPI (command) frame.<br><br>0 No selection bit is inserted in the beginning of the command frame.<br>1 Command selection bit is inserted at the beginning of the command frame. Total number of bits in the command frame is increased by 1. |
| 13<br>DSI64E | DSI 64-bit Mode Enable<br><br>Enables DSI 64-bit mode, which allows the serialization and deserialization of data frames up to 64 bits in size. The following extended registers are valid only when this mode is enabled: SDR1, ASDR1, COMPR1, DDR1, SSR1, DIMR1, DPIR1, PISR4-7.<br><br>0 DSI 64-bit Mode Disabled. DSI Mode operates in the default 32 bit configuration.<br>1 DSI 64-bit Mode Enabled. DSI Mode operates in 64-bit configuration. |
| 14<br>DSE1 | Data Select Enable 1<br><br>When the TBSC bit is set, the DSE1 bit controls insertion of the zero bit (Data Select) in the middle of the data frame. The insertion bit position is defined by the FMSZ field of CTARn register, selected by the DSICR0[DSICTAS] field. If DSI 64-bit mode is enabled (DSICR1[DSI64]), then the insertion bit position is defined by the concatenation of {DSICR0[FMSZ4], CTAR[FMSZ]}. The TSBCNT field value must be greater than the position of insertion for proper operation of DSE1.<br><br>0 No Zero bit is inserted in the middle of the data frame.<br>1 Zero bit is inserted at the middle of the data frame. Total number of bits in the data frame is increased by 1. |
| 15<br>DSE0 | Data Select Enable 0<br><br>When the TBSC bit is set, the DSE1 bit controls insertion of the zero bit (Data Select) in the beginning of the DSI frame.<br><br>0 No Zero bit is inserted in the beginning of the frame.<br>1 Zero bit is inserted at the beginning of the data frame. Total number of bits in the data frame is increased by 1. |
| 16–23<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 24–31<br>DPCS1_x | DSI Peripheral Chip Select 0–7<br><br>Specifies the PCSs to assert for the second part of the DSI frame when operating in TSB or ITSB configuration with dual receiver. DPCS1 selects the PCS signals to assert during the second part of the DSI frame. DPCS1 controls the assertions of PCS signals only in TSB and ITSB mode.<br><br>0 Negate PCS[x].<br>1 Assert PCS[x]. |

## 40.4.18 DSI Serialization Source Select Register 0 (DSPIx_SSR0)

SSR0 is used to create a combined frame for transmission that contains bits from ASDR0 and SDR0. Each bit in the SSR0 register selects a corresponding bit to be serialized. When DSICR0[TXSS] is set, the SSR0 register value has no effect. When DSI 64-bit mode is enabled (DSICR1[DSI64E]), the concatenation of {SSR1, SSR0} provides the 64-bit Serialization Source Select Value.

Address: Base address + D4h offset

| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R<br>W | SS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 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 | 0 |

**DSPIx_SSR0 field descriptions**

| Field | Description |
|---|---|
| 0–31<br>SS | Source Select<br><br>Select the serialization source for the 32 LSB of the DSI frame. Each SS bit selects data for a corresponded bit in the transmitted frame.<br><br>0 The bit in the transmitted frame is taken from the Parallel Input pin.<br>1 The bit in the transmitted frame is taken from the ASDR0 register |

## 40.4.19 DSI Parallel Input Select Register 0 (DSPIx_PISR0)

PISR0–3 select each data bit for the transmitted frame from 32 parallel input pins. Each Input Pin Select (IPS) field controls one bit in the transmitted frame. Each register contains control fields for 8 bits in the frame. The select field value is defined as a 4-bit signed integer number. The selected parallel input pin number is defined as a difference between the sum of the field number and the field value.

For example, if IPS16 is equal to the binary number 1111 (minus 1 decimal), then bit 16 in the frame is taken from parallel input pin number 15. When the IPS0 is equal to -1, then bit 0 in the frame is taken from parallel input 31. If DSI 64-bit is enabled (DSICR1[DSI64E]), bit 0 is taken from parallel input 63. When the IPS0 is equal to +1, then bit 0 in the frame is taken from Parallel Input 1 and so on.

Note that PISR0–3 only preselect the Parallel Input pins. The final selection to the transmitted frame is made with the SSR0-1 register bits or the DSICR0[TXSS] bit.

When DSI 64-bit mode is enabled (DSICR1[DSI64E]), PISR4-7 handle the 32 most-significant pins of the 64 parallel input pins.

Address: Base address + D8h offset

| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R<br>W | IPS7 | | | | IPS6 | | | | IPS5 | | | | IPS4 | | | | IPS3 | | | | IPS2 | | | | IPS1 | | | | IPS0 | | | |
| 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 | 0 |

**DSPIx_PISR0 field descriptions**

| Field | Description |
|---|---|
| 0–3<br>IPS7 | Input Pin Select 7<br><br>Selects the Parallel Input pin for transmitted frame bit 7. |
| 4–7<br>IPS6 | Input Pin Select 6<br><br>Selects the Parallel Input pin for transmitted frame bit 6. |
| 8–11<br>IPS5 | Input Pin Select 5<br><br>Selects the Parallel Input pin for transmitted frame bit 5. |
| 12–15<br>IPS4 | Input Pin Select 5<br><br>Selects the Parallel Input pin for transmitted frame bit 4. |
| 16–19<br>IPS3 | Input Pin Select 3<br><br>Selects the Parallel Input pin for transmitted frame bit 3. |
| 20–23<br>IPS2 | Input Pin Select 2<br><br>Selects the Parallel Input pin for transmitted frame bit 2. |
| 24–27<br>IPS1 | Input Pin Select 1<br><br>Selects the Parallel Input pin for transmitted frame bit 1. |
| 28–31<br>IPS0 | Input Pin Select 0<br><br>Selects the Parallel Input pin for transmitted frame bit 0. |

## 40.4.20 DSI Parallel Input Select Register 1 (DSPIx_PISR1)

Address: Base address + DCh offset

| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R<br>W | IPS15 | | | | IPS14 | | | | IPS13 | | | | IPS12 | | | | IPS11 | | | | IPS10 | | | | IPS9 | | | | IPS8 | | | |
| 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 | 0 |

**DSPIx_PISR1 field descriptions**

| Field | Description |
|---|---|
| 0–3<br>IPS15 | Input Pin Select 15 |

*Table continues on the next page...*

**DSPIx_PISR1 field descriptions (continued)**

| Field | Description |
|---|---|
| | Selects the Parallel Input pin for transmitted frame bit 15. |
| 4–7<br>IPS14 | Input Pin Select 14<br><br>Selects the Parallel Input pin for transmitted frame bit 14. |
| 8–11<br>IPS13 | Input Pin Select 13<br><br>Selects the Parallel Input pin for transmitted frame bit 13. |
| 12–15<br>IPS12 | Input Pin Select 12<br><br>Selects the Parallel Input pin for transmitted frame bit 12. |
| 16–19<br>IPS11 | Input Pin Select 11<br><br>Selects the Parallel Input pin for transmitted frame bit 11. |
| 20–23<br>IPS10 | Input Pin Select 10<br><br>Selects the Parallel Input pin for transmitted frame bit 10. |
| 24–27<br>IPS9 | Input Pin Select 9<br><br>Selects the Parallel Input pin for transmitted frame bit 9. |
| 28–31<br>IPS8 | Input Pin Select 8<br><br>Selects the Parallel Input pin for transmitted frame bit 8. |

## 40.4.21 DSI Parallel Input Select Register 2 (DSPIx_PISR2)

Address: Base address + E0h offset

| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R<br>W | IPS23 | | | | IPS22 | | | | IPS21 | | | | IPS20 | | | | IPS19 | | | | IPS18 | | | | IPS17 | | | | IPS16 | | | |
| 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 | 0 |

**DSPIx_PISR2 field descriptions**

| Field | Description |
|---|---|
| 0–3<br>IPS23 | Input Pin Select 23<br><br>Selects the Parallel Input pin for transmitted frame bit 23. |
| 4–7<br>IPS22 | Input Pin Select 22<br><br>Selects the Parallel Input pin for transmitted frame bit 22. |
| 8–11<br>IPS21 | Input Pin Select 21<br><br>Selects the Parallel Input pin for transmitted frame bit 21. |
| 12–15<br>IPS20 | Input Pin Select 20<br><br>Selects the Parallel Input pin for transmitted frame bit 20. |

*Table continues on the next page...*

**DSPIx_PISR2 field descriptions (continued)**

| Field | Description |
|---|---|
| 16–19<br>IPS19 | Input Pin Select 19<br><br>Selects the Parallel Input pin for transmitted frame bit 19. |
| 20–23<br>IPS18 | Input Pin Select 18<br><br>Selects the Parallel Input pin for transmitted frame bit 18. |
| 24–27<br>IPS17 | Input Pin Select 17<br><br>Selects the Parallel Input pin for transmitted frame bit 17. |
| 28–31<br>IPS16 | Input Pin Select 16<br><br>Selects the Parallel Input pin for transmitted frame bit 16. |

## 40.4.22 DSI Parallel Input Select Register 3 (DSPIx_PISR3)

Address: Base address + E4h offset

| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R<br>W | IPS31 | | | | IPS30 | | | | IPS29 | | | | IPS28 | | | | IPS27 | | | | IPS26 | | | | IPS25 | | | | IPS24 | | | |
| 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 | 0 |

**DSPIx_PISR3 field descriptions**

| Field | Description |
|---|---|
| 0–3<br>IPS31 | Input Pin Select 31<br><br>Selects the Parallel Input pin for transmitted frame bit 31. |
| 4–7<br>IPS30 | Input Pin Select 30<br><br>Selects the Parallel Input pin for transmitted frame bit 30. |
| 8–11<br>IPS29 | Input Pin Select 29<br><br>Selects the Parallel Input pin for transmitted frame bit 29. |
| 12–15<br>IPS28 | Input Pin Select 28<br><br>Selects the Parallel Input pin for transmitted frame bit 28. |
| 16–19<br>IPS27 | Input Pin Select 27<br><br>Selects the Parallel Input pin for transmitted frame bit 27. |
| 20–23<br>IPS26 | Input Pin Select 26<br><br>Selects the Parallel Input pin for transmitted frame bit 26. |
| 24–27<br>IPS25 | Input Pin Select 25<br><br>Selects the Parallel Input pin for transmitted frame bit 25. |
| 28–31<br>IPS24 | Input Pin Select 24<br><br>Selects the Parallel Input pin for transmitted frame bit 24. |

## 40.4.23 DSI Deserialized Data Interrupt Mask Register 0 (DSPIx_DIMR0)

DIMR0 selects bits in the 32 least-significant bits of the received DSI frame to be checked to generate the DDI interrupt. When DSI 64-bit mode is enabled (DSICR1[DSI64E]), the concatenation of {DIMR1, DIMR0} provides the 64-bit Deserialized Data Interrupt Mask value.

Address: Base address + E8h offset

| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R<br>W | MASK | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 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 | 0 |

**DSPIx_DIMR0 field descriptions**

| Field | Description |
|---|---|
| 0–31<br>MASK | Mask<br><br>Defines which bits in the 32 LSB of received deserialization data should be checked to produce the Deserialized Data Interrupt (DDI).<br><br>0 The bit in the received DSI frame does not produce a DDI interrupt.<br>1 The bit in the received DSI frame can produce a DDI interrupt if the data bit matches the configured polarity. |

## 40.4.24 DSI Deserialized Data Polarity Interrupt Register 0 (DSPIx_DPIR0)

DPIR0 defines which data bit value in the 32 least-significant bits of the received DSI frame generates the DDI interrupt. When DSI 64-bit mode is enabled (DSICR1[DSI64E]), the concatenation of {DPIR1, DPIR0} provides the 64-bit Deserialized Data Polarity Interrupt value.

Address: Base address + ECh offset

| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R<br>W | DP | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 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 | 0 |

**DSPIx_DPIR0 field descriptions**

| Field | Description |
|---|---|
| 0–31<br>DP | Data Polarity<br><br>Defines which value of the 32 LSB of received deserialization data sets the SR[DDIF] bit.<br><br>0 If the received bit is 0, the SR[DDIF] bit is set.<br>1 If the received bit is 1, the SR[DDIF] bit is set. |

## 40.4.25 DSI Serialization Data Register 1 (DSPIx_SDR1)

SDR1 contains the states of the 32 MSB parallel input signals. The states of these parallel input signals are latched into the SDR1 on the rising edge of every bus clock. The SDR1 is read-only. When the TXSS bit in the DSICR0 is cleared, the data in the SDR1 and SDR0 is used as the source of the DSI frames. The concatenation of {SDR1, SDR0} provides the 64-bit data to be serialized. This register is valid only when DSI 64-bit mode is enabled (DSICR1[DSI64E]).

Address: Base address + F0h offset

| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R | SER_DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 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 | 0 |

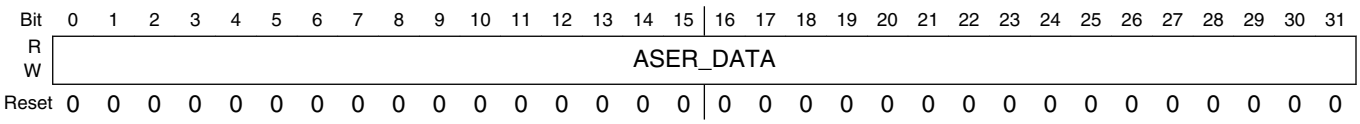
**DSPIx_SDR1 field descriptions**

| Field | Description |
|---|---|
| 0–31<br>SER_DATA | Serialized Data<br><br>Contains the 32 MSB signal states of the 64 parallel input signals. This field is valid only when DSICR1[DSI64E] is set. The concatenation of {SDR1, SDR0} provides the 64-bit data to be serialized. |

## 40.4.26 DSI Alternate Serialization Data Register 1 (DSPIx_ASDR1)

ASDR1 is used by host software to write the 32 MSB of the 64-bit data to be serialized. When the TXSS bit in the DSICR0 is set, the data in the ASDR1 and ASDR0 is the source of the DSI frames. Writes to the ASDR1 take effect on the next frame boundary. The concatenation of {ASDR1, ASDR0} provides the 64-bit alternate serialization data. This register is valid only when DSI 64-bit mode is enabled (DSICR1[DSI64E]).

Address: Base address + F4h offset



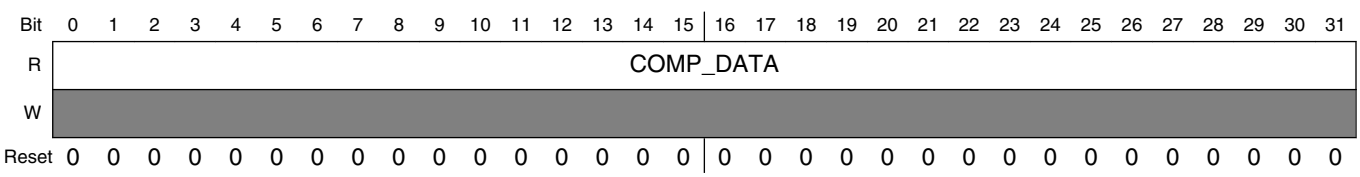

**DSPIx_ASDR1 field descriptions**

| Field | Description |
|---|---|
| 0–31<br>ASER_DATA | Alternate Serialized Data<br><br>Holds the 32 MSB of the 64-bit alternate data to be serialized. This field is valid only when DSICR1[DSI64E] is set. The concatenation of {ASDR1, ASDR0} provides the 64-bit alternate serialization data. |

## 40.4.27 DSI Transmit Comparison Register 1 (DSPIx_COMPR1)

COMPR1 holds a copy of the 32 MSB of the last transmitted 64-bit DSI data. COMPR1 is read-only. Upper 32 bits of the 64-bit DSI data is transferred to this register as it is loaded into the TX Shift Register. The concatenation of {COMPR1, COMPR0} provides the 64-bit transmit comparison data. This register is valid only when DSI 64-bit mode is enabled (DSICR1[DSI64E]).

Address: Base address + F8h offset



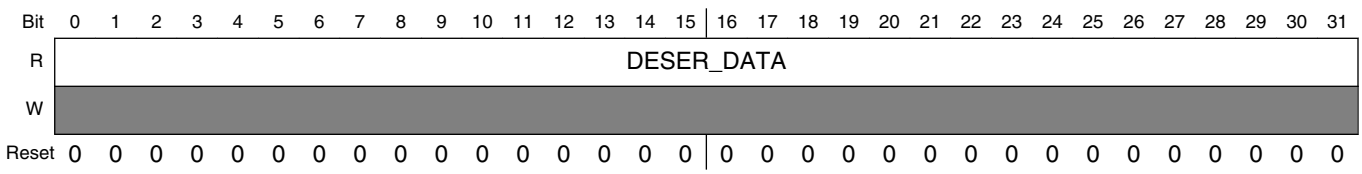
**DSPIx_COMPR1 field descriptions**

| Field | Description |
|---|---|
| 0–31<br>COMP_DATA | Compare Data<br><br>Holds the 32 MSB of the last serialized 64-bit DSI data. This field is valid only when DSICR1[DSI64E] is set. The concatenation of {COMPR1, COMPR0} provides the last serialized 64-bit DSI data. |

## 40.4.28 DSI Deserialization Data Register 1 (DSPIx_DDR1)

DDR1 holds the 32 MSB signal states for the 64 Parallel Output signals. The DDR1 is read-only and host software can read data from incoming DSI frames. The concatenation of {DDR1, DDR0} provides the 64-bit Deserialized data. This register is valid only when DSI 64-bit mode is enabled (DSICR1[DSI64E]).

Address: Base address + FCh offset




**DSPIx_DDR1 field descriptions**

| Field | Description |
|---|---|
| 0–31<br>DESER_DATA | Deserialized Data<br><br>Holds 32 MSB of the deserialized data that is presented as signal states to the 64 Parallel Output signals.This field is valid only when DSICR1[DSI64E] is set. The concatenation of {DDR1, DDR0} provides the 64-bit deserialized data. |

## 40.4.29 DSI Serialization Source Select Register 1 (DSPIx_SSR1)

SSR1 is used to create a combined frame for transmission that contains bits from ASDR1 and SDR1. Each bit in the SSR1 register selects a corresponding bit to be serialized. When DSICR0[TXSS] is set, the SSR1 register value has no effect. The concatenation of {SSR1, SSR0} provides the 64-bit Serialization Source Select Value. This register is valid only when DSI 64-bit mode is enabled (DSICR1[DSI64E]).

Address: Base address + 100h offset

| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R<br>W | SS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 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 | 0 |

**DSPIx_SSR1 field descriptions**

| Field | Description |
|---|---|
| 0–31<br>SS | Source Select<br><br>Select the serialization source for the 32 MSB of the 64-bit DSI frame. Each SS bit selects data for a corresponded bit in the transmitted frame. This field is valid only when DSICR1[DSI64E] is set. The concatenation of {SSR1, SSR0} provides the 64-bit serialization source select value.<br><br>0 The bit in the transmitted frame is taken from the Parallel Input pin.<br>1 The bit in the transmitted frame is taken from the ASDR0/1 register |

## 40.4.30 DSI Parallel Input Select Register 4 (DSPIx_PISR4)

PISR4–7 are used to select each data bit for the transmitted frame from 32 parallel input pins. Each Input Pin Select (IPS) field controls one bit in the transmitted frame. Each register contains control fields for 8 bits in the frame. The select field value is defined as a 4-bit signed integer number. The selected parallel input pin number is defined as a difference between the sum of the field number and the field value.

For example, if IPS16 is equal to the binary number 1111 (minus 1 decimal), then bit 16 in the frame is taken from parallel input pin number 15. When the IPS0 is equal to -1, then bit 0 in the frame is taken from parallel input 31 (bit 0 is taken from parallel input 63 when DSICR1[DSI64E] is enabled). When the IPS0 is equal to +1, then bit 0 in the frame is taken from parallel input 1 and so on.

Note that PISR0–7 only preselect the parallel input pins. The final selection to the transmitted frame is made with the SSR0 and SSR1 register bits or the DSICR0[TXSS] bit.

When DSI 64-bit mode is enabled (DSICR1[DSI64E]), PISR4-7 are used to handle the extra 32 bits (32 MSB pins of the 64 parallel input pins). If DSI 64-bit mode is disabled, then PISR4-7 are not used.

Address: Base address + 104h offset

| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R<br>W | IPS39 | | | | IPS38 | | | | IPS37 | | | | IPS36 | | | | IPS35 | | | | IPS34 | | | | IPS33 | | | | IPS32 | | | |
| 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 | 0 |

**DSPIx_PISR4 field descriptions**

| Field | Description |
|---|---|
| 0–3<br>IPS39 | Input Pin Select 39<br><br>Selects the Parallel Input pin for transmitted frame bit 39. |
| 4–7<br>IPS38 | Input Pin Select 38<br><br>Selects the Parallel Input pin for transmitted frame bit 38. |
| 8–11<br>IPS37 | Input Pin Select 37<br><br>Selects the Parallel Input pin for transmitted frame bit 37. |
| 12–15<br>IPS36 | Input Pin Select 36<br><br>Selects the Parallel Input pin for transmitted frame bit 36. |
| 16–19<br>IPS35 | Input Pin Select 35<br><br>Selects the Parallel Input pin for transmitted frame bit 35. |
| 20–23<br>IPS34 | Input Pin Select 34<br><br>Selects the Parallel Input pin for transmitted frame bit 34. |
| 24–27<br>IPS33 | Input Pin Select 33<br><br>Selects the Parallel Input pin for transmitted frame bit 33. |
| 28–31<br>IPS32 | Input Pin Select 32<br><br>Selects the Parallel Input pin for transmitted frame bit 32. |

## 40.4.31 DSI Parallel Input Select Register 5 (DSPIx_PISR5)

Address: Base address + 108h offset

| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R<br>W | IPS47 | | | | IPS46 | | | | IPS45 | | | | IPS44 | | | | IPS43 | | | | IPS42 | | | | IPS41 | | | | IPS40 | | | |
| 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 | 0 |

**DSPIx_PISR5 field descriptions**

| Field | Description |
|---|---|
| 0–3<br>IPS47 | Input Pin Select 47 |

*Table continues on the next page...*

**DSPIx_PISR5 field descriptions (continued)**

| Field | Description |
|---|---|
| | Selects the Parallel Input pin for transmitted frame bit 47. |
| 4–7<br>IPS46 | Input Pin Select 46<br><br>Selects the Parallel Input pin for transmitted frame bit 46. |
| 8–11<br>IPS45 | Input Pin Select 45<br><br>Selects the Parallel Input pin for transmitted frame bit 45. |
| 12–15<br>IPS44 | Input Pin Select 44<br><br>Selects the Parallel Input pin for transmitted frame bit 44. |
| 16–19<br>IPS43 | Input Pin Select 43<br><br>Selects the Parallel Input pin for transmitted frame bit 43. |
| 20–23<br>IPS42 | Input Pin Select 42<br><br>Selects the Parallel Input pin for transmitted frame bit 42. |
| 24–27<br>IPS41 | Input Pin Select 41<br><br>Selects the Parallel Input pin for transmitted frame bit 41. |
| 28–31<br>IPS40 | Input Pin Select 40<br><br>Selects the Parallel Input pin for transmitted frame bit 40. |

## 40.4.32 DSI Parallel Input Select Register 6 (DSPIx_PISR6)

Address: Base address + 10Ch offset

| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R<br>W | IPS55 | | | | IPS54 | | | | IPS53 | | | | IPS52 | | | | IPS51 | | | | IPS50 | | | | IPS49 | | | | IPS48 | | | |
| 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 | 0 |

**DSPIx_PISR6 field descriptions**

| Field | Description |
|---|---|
| 0–3<br>IPS55 | Input Pin Select 55<br><br>Selects the Parallel Input pin for transmitted frame bit 55. |
| 4–7<br>IPS54 | Input Pin Select 54<br><br>Selects the Parallel Input pin for transmitted frame bit 54. |
| 8–11<br>IPS53 | Input Pin Select 53<br><br>Selects the Parallel Input pin for transmitted frame bit 53. |
| 12–15<br>IPS52 | Input Pin Select 52<br><br>Selects the Parallel Input pin for transmitted frame bit 52. |

*Table continues on the next page...*

**DSPIx_PISR6 field descriptions (continued)**

| Field | Description |
|---|---|
| 16–19<br>IPS51 | Input Pin Select 51<br><br>Selects the Parallel Input pin for transmitted frame bit 51. |
| 20–23<br>IPS50 | Input Pin Select 50<br><br>Selects the Parallel Input pin for transmitted frame bit 50. |
| 24–27<br>IPS49 | Input Pin Select 49<br><br>Selects the Parallel Input pin for transmitted frame bit 49. |
| 28–31<br>IPS48 | Input Pin Select 48<br><br>Selects the Parallel Input pin for transmitted frame bit 48. |

## 40.4.33 DSI Parallel Input Select Register 7 (DSPIx_PISR7)

Address: Base address + 110h offset

| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R<br>W | IPS63 | | | | IPS62 | | | | IPS61 | | | | IPS60 | | | | IPS59 | | | | IPS58 | | | | IPS57 | | | | IPS56 | | | |
| 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 | 0 |

**DSPIx_PISR7 field descriptions**

| Field | Description |
|---|---|
| 0–3<br>IPS63 | Input Pin Select 63<br><br>Selects the Parallel Input pin for transmitted frame bit 63. |
| 4–7<br>IPS62 | Input Pin Select 62<br><br>Selects the Parallel Input pin for transmitted frame bit 62. |
| 8–11<br>IPS61 | Input Pin Select 61<br><br>Selects the Parallel Input pin for transmitted frame bit 61. |
| 12–15<br>IPS60 | Input Pin Select 60<br><br>Selects the Parallel Input pin for transmitted frame bit 60. |
| 16–19<br>IPS59 | Input Pin Select 59<br><br>Selects the Parallel Input pin for transmitted frame bit 59. |
| 20–23<br>IPS58 | Input Pin Select 58<br><br>Selects the Parallel Input pin for transmitted frame bit 58. |
| 24–27<br>IPS57 | Input Pin Select 57<br><br>Selects the Parallel Input pin for transmitted frame bit 57. |
| 28–31<br>IPS56 | Input Pin Select 56<br><br>Selects the Parallel Input pin for transmitted frame bit 56. |

## 40.4.34 DSI Deserialized Data Interrupt Mask Register 1 (DSPIx_DIMR1)

DIMR1 selects bits from the 32 MSB in the received 64-bit DSI frame to be checked to generate the DDI interrupt. The concatenation of {DIMR1, DIMR0} provides the 64-bit Deserialized Data Interrupt Mask Value. This register is valid only when DSI 64-bit mode is enabled (DSICR1[DSI64E]).

Address: Base address + 114h offset

| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R<br>W | MASK | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 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 | 0 |

**DSPIx_DIMR1 field descriptions**

| Field | Description |
|---|---|
| 0–31<br>MASK | Mask<br><br>Defines which of the 32 MSB in the received 64-bit deserialization data should be checked to produce the Deserialized Data Interrupt (DDI).<br><br>0 The bit in the received DSI frame does not produce a DDI interrupt.<br>1 The bit in the received DSI frame can produce a DDI interrupt if the data bit matches the configured polarity. |

## 40.4.35 DSI Deserialized Data Polarity Interrupt Register 1 (DSPIx_DPIR1)

DPIR1 defines which of the 32 MSB value in the received 64-bit DSI frame generates the DDI interrupt. The concatenation of {DPIR1, DPIR0} provides the 64-bit Deserialized Data Polarity Value. This register is valid only when DSI 64-bit mode is enabled (DSICR1[DSI64E]).

Address: Base address + 118h offset

| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R<br>W | DP | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 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 | 0 |

**DSPIx_DPIR1 field descriptions**

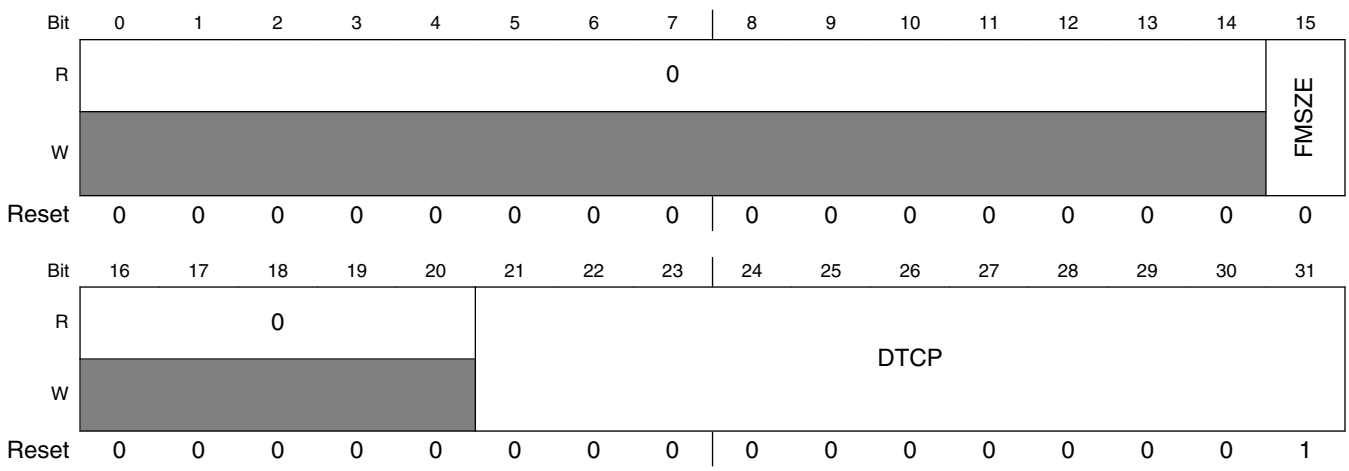
| Field | Description |
|---|---|
| 0–31<br>DP | Data Polarity<br><br>Defines which of the 32 MSB of the received 64-bit deserialization data sets the SR[DDIF] bit.<br><br>0 If the received bit is 0, the SR[DDIF] bit is set.<br>1 If the received bit is 1, the SR[DDIF] bit is set. |

## 40.4.36 Clock and Transfer Attributes Register Extended (DSPIx_CTARE*n*)

CTARE registers are used to define the extended transfer attributes for an SPI frame. These registers are valid only when Extended SPI mode is enabled (MCR[XSPI]).

When the module is configured as a SPI master, the CTAS field in CMD FIFO entry selects which of the CTARE registers is used.

Address: Base address + 11Ch offset + (4d × i), where i=0d to 5d




**DSPIx_CTARE*n* field descriptions**

| Field | Description |
|---|---|
| 0–14<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 15<br>FMSZE | Frame Size Extended<br><br>This field is valid only when MCR[XSPI] is set. This field concatenated with CTAR[FMSZ] defines the Frame size of the SPI frames to be transmitted. Effective frame size would be the concatenation of {CTARE[FMSZE], CTAR[FMSZ]} plus 1.<br><br>0 Default Mode. Up to 16 bit SPI frames can be transferred.<br>1 Up to 32 bit SPI frames can be transferred. Each Frame transfer will be a result of 2 simulataneous TX FIFO Pop Operation. |

*Table continues on the next page...*

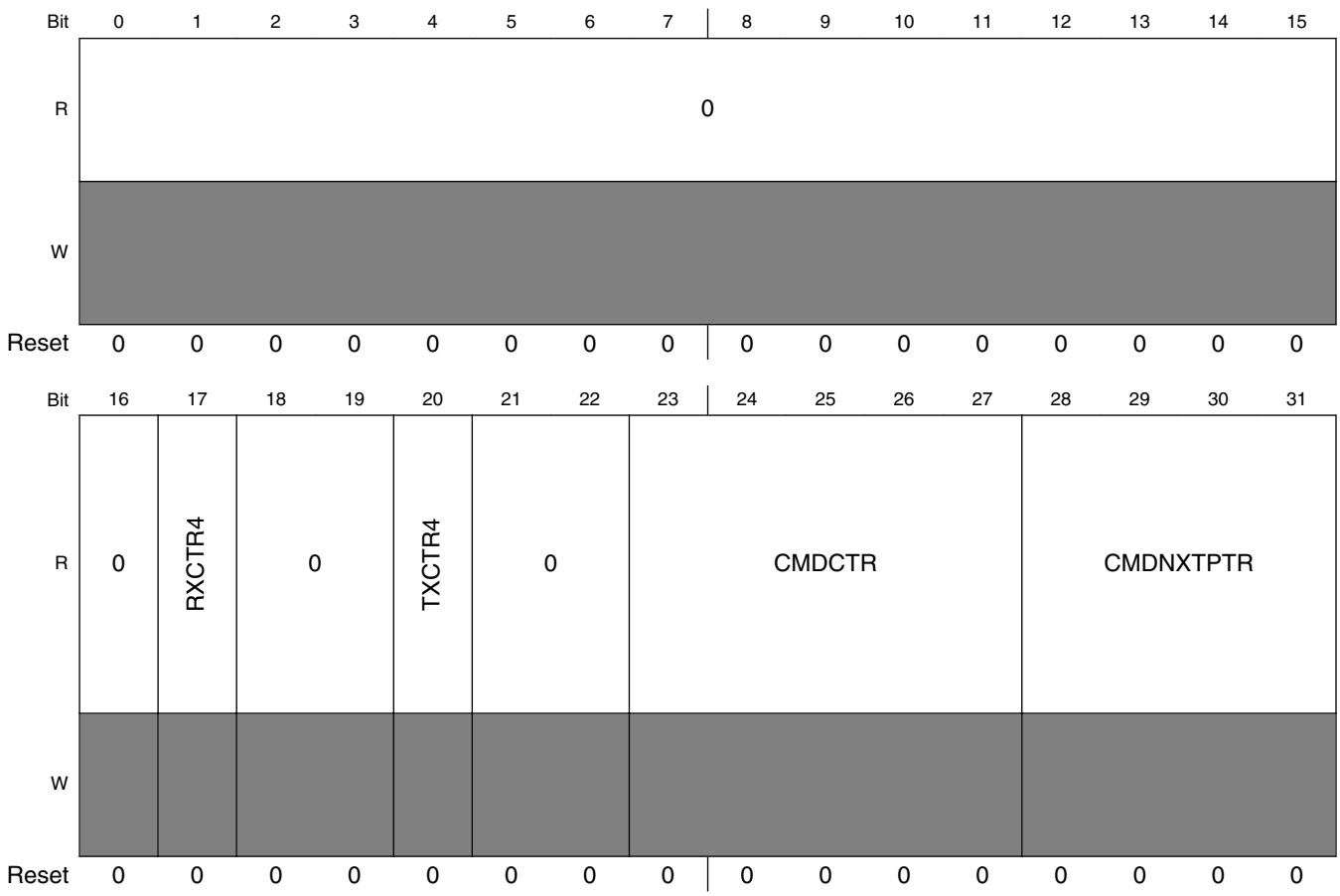
**DSPIx_CTARE*n* field descriptions (continued)**

| Field | Description |
|---|---|
| 16–20<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 21–31<br>DTCP | Data Transfer Count Preload<br><br>This field is valid only when DSPIx_MCR[XSPI] is set. This field defines the number of data frames (whose size is defined by CTARE[FMSZE] and CTAR[FMSZ]) to be transmitted using the Command frame which selected this DSPIx_CTARE register. The value 0 is reserved and should not be written in this field. The default value of this field is 1. |

## 40.4.37 Status Register Extended (DSPIx_SREX)

The register contains status fields. The fields reflect the status of the module and indicate the occurrence of events. This register is not writable.

Address: Base address + 13Ch offset



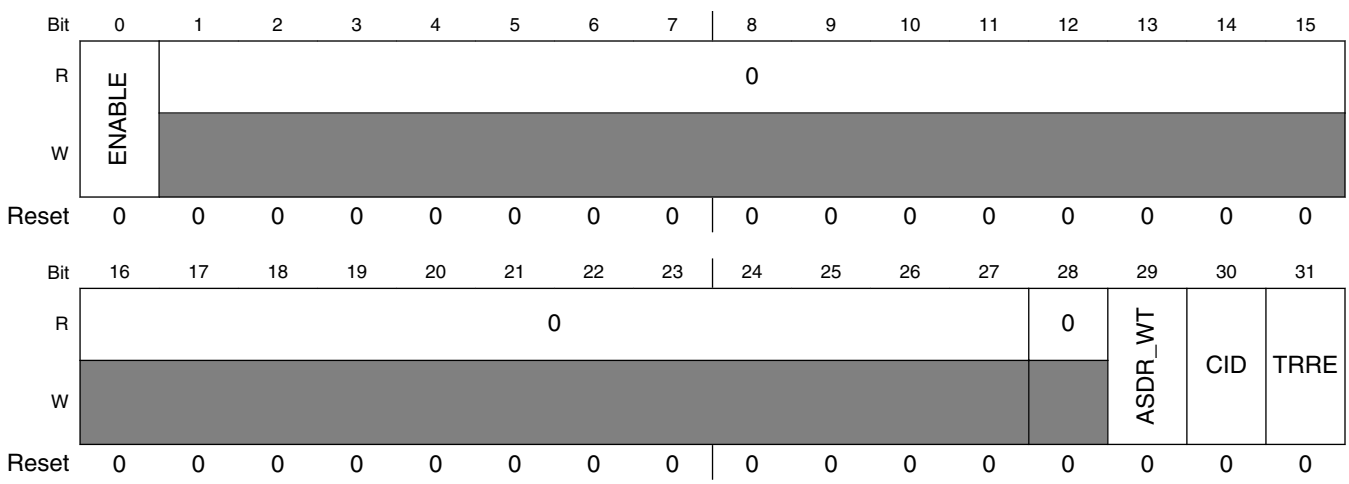
**DSPIx_SREX field descriptions**

| Field | Description |
|---|---|
| 0–16<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 17<br>RXCTR4 | RX FIFO Counter[4]<br><br>This bit is an extension of SR[RXCTR]. The concatenated field {RXCTR4, RXCTR} indicates the number of entries in the RX FIFO. This field is decremented every time the POPR is read. And this field is incremented every time data is transferred from the shift register to the RX FIFO. |
| 18–19<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 20<br>TXCTR4 | TX FIFO Counter[4]<br><br>This bit is an extension of SR[TXCTR]. The concatenated field {TXCTR4, TXCTR} indicates the number of valid entries in the TX FIFO. This field is incremented every time the PUSHR is written. And this field is decremented every time an SPI command is executed and the SPI data is transferred to the shift register. |
| 21–22<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 23–27<br>CMDCTR | CMD FIFO Counter<br><br>Indicates the number of entries in the CMD FIFO. The CMDCTR is incremented every time the command part of PUSHR is written. The CMDCTR is decremented every time a SPI command is executed (all data frames due to current command frame have been transmitted). |
| 28–31<br>CMDNXTPTR | Command Next Pointer<br><br>Indicates which CMD FIFO Entry is used during the next transfer. The CMDNXTPTR field is updated every time SPI data due to current command have been transmitted. |

## 40.4.38 Trigger Register (DSPIx_TRIG)

This register provides the user to configure trigger settings for various modes. The trigger settings in this register are activated only if the bit TRIG[ENABLE] is set.

Address: Base address + 140h offset



**DSPIx_TRIG field descriptions**

| Field | Description |
|---|---|
| 0<br>ENABLE | Trigger Register Enable<br><br>This bit is used to enable the TRIG register.<br><br>0 TRIG register is disabled. Trigger settings configured in this register have no effect on the module.<br>1 TRIG register is enabled. Trigger settings configured in this register are valid. |
| 1–27<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 28<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 29<br>ASDR_WT | ASDR Write Trigger Enable<br><br>Enables the module to initiate a DSI frames transfer on a write operation performed on ASDR0 register.<br><br>0 ASDR0 write trigger disabled.<br>1 ASDR0 write trigger enabled. |
| 30<br>CID | Change In Data Transfer Enable<br><br>Enables a change in serialization data to initiate a DSI frames transfer in DSI and CSI configurations. When the CID bit is set, DSI frames are initiated when the current DSI data differs from the previous DSI data shifted out.<br><br>**NOTE:** When TRIG[ENABLE] is set, the CID bit is given by TRIG[CID] which overrides DSICR0[CID]. |
| 31<br>TRRE | Trigger Reception Enable<br><br>Enables the module to initiate a DSI frames transfer with an external trigger signal.<br><br>**NOTE:** When TRIG[ENABLE] is set, the TRRE bit is given by TRIG[TRRE] which overrides DSICR0[TRRE].<br><br>0 Trigger signal reception disabled.<br>1 Trigger signal reception enabled. |

## 40.4.39 Time Slot Length Register (DSPIx_TSL)

The TSL register is only valid when DSICR0[ITSB] is enabled and internal triggering mechanism is used. It provides the length of every time slot in terms of baud cycles.

Address: Base address + 150h offset

| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R | 0 | TS3_LEN | | | | | | | 0 | TS2_LEN | | | | | | |
| W | | | | | | | | | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R | 0 | TS1_LEN | | | | | | | 0 | TS0_LEN | | | | | | |
| W | | | | | | | | | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**DSPIx_TSL field descriptions**

| Field | Description |
|---|---|
| 0<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 1–7<br>TS3_LEN | Time Slot 3 Length<br><br>This field provides the length of TS3 in terms of baud cycles. It must be ensured that the programmed value is greater than or equal to the length of the frame to be transmitted during TS3.Time Slot3 Length = TS3_LEN+1.This time slot shall be skipped if programmed to 0. |
| 8<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 9–15<br>TS2_LEN | Time Slot 2 Length<br><br>This field provides the length of TS2 in terms of baud cycles. It must be ensured that the programmed value is greater than or equal to the length of the frame to be transmitted during TS2.Time Slot 2 Length = TS2_LEN+1.This time slot shall be skipped if programmed to 0. |
| 16<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 17–23<br>TS1_LEN | Time Slot 1 Length<br><br>This field provides the length of TS1 in terms of baud cycles. It must be ensured that the programmed value is greater than or equal to the length of the frame to be transmitted during TS1.Time Slot 1 Length = TS1_LEN+1.This time slot shall be skipped if programmed to 0. |
| 24<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 25–31<br>TS0_LEN | Time Slot 0 Length<br><br>This field provides the length of TS0 in terms of baud cycles. It must be ensured that the programmed value is greater than or equal to the length of the frame to be transmitted during TS0.Time Slot 0 Length = TS0_LEN+1.This time slot shall be skipped if programmed to 0. |

## 40.4.40 Time Slot Configuration Register (DSPIx_TS_CONF)

This register provides the basic configuration options available for the triggered implementation of MSC protocol. This register is only valid when DSICR0[ITSB] is enabled. Some configurations in this register check for the validity of command frame. The following conditions must be true for a command frame to be valid : a) The previous transmitted frame should not be a command frame. b) Data required to transmit the command frame should be available.

Address: Base address + 154h offset

| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R | 0 | | | | | | | | | | | | | | | | TS3 | | | | TS2 | | | | TS1 | | | | TS0 | | | |
| W | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 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 | 0 |

## DSPIx_TS_CONF field descriptions

| Field | Description |
|---|---|
| 0–15<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 16–19<br>TS3 | Time Slot 3<br><br>This field is used to program the frame rule to be followed during Time Slot 3. This field is only valid when DSICR0[ITSB] mode is enabled.<br><br>0000 PF - Passive Frame<br>0001 SRDF - Single Receiver Data Frame<br>0010 DRF1 - First part of Dual Receiver Frame<br>0011 DRF2 - Second part of Dual Receiver Frame<br>0100 CMD_PF - Command Frame if valid, else Passive Frame<br>0101 CMD_SRDF - Command Frame if valid, else Single Receiver Data Frame<br>0110 CMD_DRF1 - Command Frame if valid, else first half of Dual Receiver Data Frame<br>0111 CMD_DRF2 - Command Frame if valid, else second half of Dual Receiver Data Frame<br>1xxx Reserved |
| 20–23<br>TS2 | Time Slot 2<br><br>This field is used to program the frame rule to be followed during Time Slot 2. This field is only valid when DSICR0[ITSB] mode is enabled.<br><br>0000 PF - Passive Frame<br>0001 SRDF - Single Receiver Data Frame<br>0010 DRF1 - First part of Dual Receiver Frame<br>0011 DRF2 - Second part of Dual Receiver Frame<br>0100 CMD_PF - Command Frame if valid, else Passive Frame<br>0101 CMD_SRDF - Command Frame if valid, else Single Receiver Data Frame<br>0110 CMD_DRF1 - Command Frame if valid, else first half of Dual Receiver Data Frame<br>0111 CMD_DRF2 - Command Frame if valid, else second half of Dual Receiver Data Frame<br>1xxx Reserved |
| 24–27<br>TS1 | Time Slot 1<br><br>This field is used to program the frame rule to be followed during Time Slot 1. This field is only valid when DSICR0[ITSB] mode is enabled.<br><br>0000 PF - Passive Frame<br>0001 SRDF - Single Receiver Data Frame<br>0010 DRF1 - First part of Dual Receiver Frame<br>0011 DRF2 - Second part of Dual Receiver Frame<br>0100 CMD_PF - Command Frame if valid, else Passive Frame<br>0101 CMD_SRDF - Command Frame if valid, else Single Receiver Data Frame<br>0110 CMD_DRF1 - Command Frame if valid, else first half of Dual Receiver Data Frame<br>0111 CMD_DRF2 - Command Frame if valid, else second half of Dual Receiver Data Frame<br>1xxx Reserved |
| 28–31<br>TS0 | Time Slot 0<br><br>This field is used to program the frame rule to be followed during Time Slot 0. This field is only valid when DSICR0[ITSB] mode is enabled.<br><br>0000 PF - Passive Frame |

*Table continues on the next page...*

**DSPIx_TS_CONF field descriptions (continued)**

| Field | Description |
|---|---|
| | 0001 SRDF - Single Receiver Data Frame<br>0010 DRF1 - First part of Dual Receiver Frame<br>0011 DRF2 - Second part of Dual Receiver Frame<br>0100 CMD_PF - Command Frame if valid, else Passive Frame<br>0101 CMD_SRDF - Command Frame if valid, else Single Receiver Data Frame<br>0110 CMD_DRF1 - Command Frame if valid, else first half of Dual Receiver Data Frame<br>0111 CMD_DRF2 - Command Frame if valid, else second half of Dual Receiver Data Frame<br>1xxx Reserved |

## 40.5 Functional description

The module supports full-duplex, synchronous serial communications between chips and peripheral devices. The module can also be used to reduce the number of pins required for I/O by serializing and deserializing Parallel Input/Output signals.All communications are done with SPI-like protocol.

The module has the following configurations

- The SPI Configuration in which the module operates as a basic SPI or a queued SPI.
- DSI Configuration in which the module serializes and deserializes Parallel Input/ Output signals or bits from memory-mapped registers
- When Master mode is enabled, the module also supports CSI Configuration, which combines the functionality of the SPI and DSI configurations.

The DCONF field in the Module Configuration Register (MCR) determines the module Configuration. See Module Configuration Register (DSPI_MCR) for the DSPI configuration values.

The CTARn registers hold clock and transfer attributes. The SPI configuration allows to select which CTAR to use on a frame by frame basis by setting a field in the SPI command. The Extended SPI Mode (DSPIx_MCR[XSPI]) further allows the usage CTAREn (CTARn Extended) registers which allows the user to send multiple data frames using a single command frame.

The DSI configuration statically selects which CTAR to use. In CSI, Configuration priority logic determines if SPI data or DSI data is transferred and dictates what CTAR register is used for the data transfer. See Clock and Transfer Attributes Register (In

Master Mode) (DSPI_CTAR*n*) for information on the fields of CTAR registers. See Clock and Transfer Attributes Register Extended (DSPI_CTARE*n*) for information on the fields of CTARE registers.

Typical master to slave connections are shown in the following figure. When a data transfer operation is performed, data is serially shifted a predetermined number of bit positions. Because the modules are linked, data is exchanged between the master and the slave. The data that was in the master shift register is now in the shift register of the slave, and vice versa. At the end of a transfer, the Transfer Control Flag(TCF) bit in the Shift Register(SR) is set to indicate a completed frame transfer.

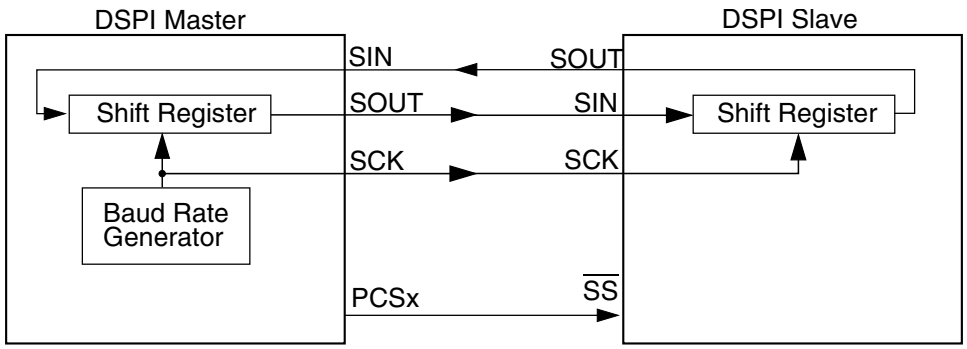



**Figure 40-3. Serial protocol overview**

Generally, more than one slave device can be connected to the module master. 6 Peripheral Chip Select (PCS) signals of the module masters can be used to select which of the slaves to communicate with. Refer to the chip specific section for details on the number of PCS signals used in this chip.

The three DSPI configurations share transfer protocol and timing properties which are described independently of the configuration in Transfer formats. The transfer rate and delay settings are described in Module baud rate and clock delay generation.

## 40.5.1 Start and Stop of module transfers

The module has two operating states: Stopped and Running. Both the states are independent of it's configuration. The default state of the module is Stopped. In the Stopped state, no serial transfers are initiated in Master mode and no transfers are responded to in Slave mode. The Stopped state is also a safe state for writing the various configuration registers of the module without causing undetermined results. In the Running state serial transfers take place.

The TXRXS bit in the SR indicates the state of module. The bit is set if the module is in Running state.

The module starts or transitions to Running when all of the following conditions are true:

- SR[EOQF] bit is clear
- Chip is not in the Debug mode or the MCR[FRZ] bit is clear
- MCR[HALT] bit is clear

The module stops or transitions from Running to Stopped after the current frame when any one of the following conditions exist:

- SR[EOQF] bit is set
- Chip in the Debug mode and the MCR[FRZ] bit is set
- MCR[HALT] bit is set

State transitions from Running to Stopped occur on the next frame boundary if a transfer is in progress, or immediately if no transfers are in progress.

### 40.5.2 Serial Peripheral Interface (SPI) configuration

The SPI configuration transfers data serially using a shift register and a selection of programmable transfer attributes. The module is in SPI configuration when the DCONF field in the MCR is 0b00. The SPI frames can be 32 bits long. The host CPU or a DMA controller transfers the SPI data from the external to the module RAM queues to a TX FIFO buffer. The received data is stored in entries in the RX FIFO buffer. The host CPU or the DMA controller transfers the received data from the RX FIFO to memory external to the module. The operation of FIFO buffers is described in the following sections:

- Transmit First In First Out (TX FIFO) buffering mechanism
- Command First In First Out (CMD FIFO) Buffering Mechanism
- Receive First In First Out (RX FIFO) buffering mechanism

The interrupt and DMA request conditions are described in Interrupts/DMA requests.

The SPI configuration supports two block-specific modes—Master mode and Slave mode. In Master mode the module initiates and controls the transfer according to the fields of the executing SPI Command. In Slave mode, the module responds only to transfers initiated by a bus master external to it and the SPI command field space is reserved.

#### 40.5.2.1 Master mode

In SPI Master mode, the module initiates the serial transfers by controlling the SCK and the PCS signals. The executing SPI Command determines which CTARs will be used to set the transfer attributes and which PCS signals to assert. The command field also contains various bits that help with queue management and transfer protocol. See PUSH TX FIFO Register In Master Mode (DSPI_PUSHR) for details on the SPI command fields. The data in the executing TX FIFO entry is loaded into the shift register and shifted out on the Serial Out (SOUT) pin. In SPI Master mode, each SPI frame to be transmitted has a command associated with it, allowing for transfer attribute control on a frame by frame basis. In Extended SPI Master Mode, multiple SPI frames can have a single command associated with them allowing for efficient SPI frame transfers requiring common transfer attributes. In addition, the Extended SPI Mode allows for larger frame sizes of up to 32 bits.

#### 40.5.2.2 Slave mode

In SPI Slave mode the module responds to transfers initiated by an SPI bus master. It does not initiate transfers. Certain transfer attributes such as clock polarity, clock phase, and frame size must be set for successful communication with an SPI master. The SPI Slave mode transfer attributes are set in the CTAR0 and CTARE0. The data is shifted out with MSB first. Shifting out of LSB is not supported in this mode.

#### 40.5.2.3 FIFO disable operation

The FIFO disable mechanisms allow SPI transfers without using the TX FIFO, CMD FIFO or RX FIFO. The module operates as a double-buffered simplified SPI when the FIFOs are disabled. The Transmit and Receive side of the FIFOs are disabled separately. Setting the MCR[DIS_TXF] bit disables the TX FIFO and CMD FIFO, and setting the MCR[DIS_RXF] bit disables the RX FIFO.

The FIFO disable mechanisms are transparent to the user and to host software. Transmit data and commands are written to the PUSHR and received data is read from the POPR.

When the TX FIFO and CMD FIFO are disabled:

- SR[TFFF], SR[TFUF], SR[CMDFFF], SREX[CMDCTR] and SR[TXCTR] behave as if there is a one-entry FIFO
- The contents of TXFRs, SR[TXNXTPTR] and SREX[CMDNXTPTR] are undefined

Similarly, when the RX FIFO is disabled, the RFDF, RFOF, and RXCTR fields in the SR behave as if there is a one-entry FIFO, but the contents of the RXFR registers and POPNXTPTR are undefined.

#### 40.5.2.4 Transmit First In First Out (TX FIFO) buffering mechanism

The TX FIFO functions as a buffer of SPI data for transmission. The TX FIFO holds 4 words, each consisting of SPI data. The number of entries in the TX FIFO is device-specific. SPI data is added to the TX FIFO by writing to the Data Field of module PUSH FIFO Register (PUSHR). TX FIFO entries can only be removed from the TX FIFO by being shifted out or by flushing the TX FIFO.

The TX FIFO Counter field (TXCTR) in the module Status Register (SR) indicates the number of valid entries in the TX FIFO. The TXCTR is updated every time a 8- or 16-bit write takes place to PUSHR[TXDATA] or SPI data is transferred into the shift register from the TX FIFO.

The TXNXTPTR field indicates the TX FIFO Entry that will be transmitted during the next transfer. The TXFRn Registers are invalid in the Extended SPI Mode, since the TX FIFO and CMD FIFO can be used independently. The TXNXTPTR field is incremented every time SPI data is transferred from the TX FIFO to the shift register. The maximum value of the field is equal to the maximum implemented TXFR number and it rolls over after reaching the maximum.

##### 40.5.2.4.1 Filling the TX FIFO

Host software or other intelligent blocks can add (push) entries to the TX FIFO and CMD FIFO by writing to the PUSHR. When the TX FIFO is not full, the TX FIFO Fill Flag (TFFF) in the SR is set. The TFFF bit is cleared when TX FIFO is full and the DMA controller indicates that a write to PUSHR is complete. Writing a '1' to the TFFF bit also clears it. The TFFF can generate a DMA request or an interrupt request. See Transmit FIFO Fill Interrupt or DMA Request for details.

The module ignores attempts to push data to a full TX FIFO, and the state of the TX FIFO does not change and no error condition is indicated.

##### 40.5.2.4.2 Draining the TX FIFO

The TX FIFO entries are removed (drained) by shifting SPI data out through the shift register. Entries are transferred from the TX FIFO to the shift register and shifted out as long as there are valid entries in the TX FIFO. Every time an entry is transferred from the TX FIFO to the shift register, the TX FIFO Counter decrements by one. When Extended SPI Mode (DSPIx_MCR[XSPI]) is enabled, if the frame size of SPI Data to be transmitted is more than 16 bits, then it causes two Data entries to be popped from TX FIFO simultaneously which are transferred to the shift register. The first of the two popped entries forms the 16 least significant bits of the SPI frame to be transmitted. Such

an operation also causes TX FIFO Counter to decrement by two. At the end of a transfer, the TCF bit in the SR is set to indicate the completion of a transfer. The TX FIFO is flushed by writing a '1' to the CLR_TXF bit in MCR.

If an external bus master initiates a transfer with a module slave while the slave's TX FIFO is empty, the Transmit FIFO Underflow Flag (TFUF) in the slave's SR is set. See Transmit FIFO Underflow Interrupt Request for details.

#### 40.5.2.5 Command First In First Out (CMD FIFO) Buffering Mechanism

The CMD FIFO functions as a buffer of SPI command used for SPI data transmission. When Extended SPI Mode (MCR[XSPI]) is disabled, the TX FIFO and CMD FIFO must be filled together, i.e. write enables should be given for both the Data and Command fields while performing a PUSHR operation.When Extended SPI Mode (MCR[XSPI]) is enabled, the TX FIFO and CMD FIFO can be filled independently.

The CMD FIFO holds 4 words, each representing SPI command fields. The number of entries in the CMD FIFO is device-specific. SPI Command is added to the CMD FIFO by writing to the command field of DSPI PUSH FIFO Register (PUSHR). CMD FIFO entries can only be removed from the CMD FIFO by being shifted out (to help transmit SPI data) or by flushing the CMD FIFO.

When Extended SPI Mode (MCR[XSPI]) is disabled, every CMD FIFO entry has a corresponding single TX FIFO entry attached to it because both these FIFO's are filled simultaneously.

When Extended SPI Mode (MCR[XSPI]) is enabled, every CMD FIFO entry can have multiple TX FIFO entries attached to it. Thus a single CMD FIFO entry can be used to transmit multiple TX FIFO entries. The CTARE[DTCP] field decides the number of SPI Data Frames having frame size as {FMSZE, FMSZ} to be transmitted using the current Command Entry. The CTAR/CTARE registers pointed by the CTAS field in the Command frame gives the FMSZ and FMSZE fields respectively. The time for which a command entry is in use is known as a Command Cycle. The Busy Flag SR[BSYF] is asserted for the duration of the Command Cycle except for the last SPI frame in the Command Cycle.

The CMD FIFO Counter field (CMDCTR) in the DSPI Status Register (SR) indicates the number of valid entries in the CMD FIFO. The CMDCTR field is updated every time a 8- or 16-bit write takes place on the lower half of DSPI_PUSHR or SPI data is transferred into the shift register from the TX FIFO.

The TXFRn Registers are invalid in the Extended SPI Mode, since the TX FIFO and CMD FIFO can be used independently. The CMDNXTPTR field indicates which CMD FIFO Entry will be used during the next command cycle. The CMDNXTPTR field is incremented every time the last SPI data in the command cycle is transferred from the TX FIFO to the shift register and it rolls over after reaching the maximum.

### 40.5.2.6 Receive First In First Out (RX FIFO) buffering mechanism

The RX FIFO functions as a buffer for data received on the SIN pin. The RX FIFO holds 4 received SPI data frames. The number of entries in the RX FIFO is device-specific. SPI data is added to the RX FIFO at the completion of a transfer when the received data in the shift register is transferred into the RX FIFO. SPI data are removed (popped) from the RX FIFO by reading the module POP RX FIFO Register (POPR). RX FIFO entries can only be removed from the RX FIFO by reading the POPR or by flushing the RX FIFO.

The RX FIFO Counter field (RXCTR) in the module's Status Register (SR) indicates the number of valid entries in the RX FIFO. The RXCTR is updated every time the POPR is read or SPI data is copied from the shift register to the RX FIFO.

The POPNXTPTR field in the SR points to the RX FIFO entry that is returned when the POPR is read. The POPNXTPTR contains the positive offset from RXFR0 in a number of 32-bit registers. For example, POPNXTPTR equal to two means that the RXFR2 contains the received SPI data that will be returned when the POPR is read. The POPNXTPTR field is incremented every time the POPR is read. The maximum value of the field is equal to the maximum implemented RXFR number and it rolls over after reaching the maximum.

#### 40.5.2.6.1 Filling the RX FIFO

The RX FIFO is filled with the received SPI data from the shift register. While the RX FIFO is not full, SPI frames from the shift register are transferred to the RX FIFO. Every time an SPI frame is transferred to the RX FIFO, the RX FIFO Counter is incremented by one.

If the RX FIFO and shift register are full and a transfer is initiated, the RFOF bit in the SR is set indicating an overflow condition. Depending on the state of the ROOE bit in the MCR, the data from the transfer that generated the overflow is either ignored or shifted in to the shift register. If the ROOE bit is set, the incoming data is shifted in to the shift register. If the ROOE bit is cleared, the incoming data is ignored.

#### 40.5.2.6.2 Draining the RX FIFO

Host CPU or a DMA can remove (pop) entries from the RX FIFO by reading the module POP RX FIFO Register (POPR). A read of the POPR decrements the RX FIFO Counter by one. Attempts to pop data from an empty RX FIFO are ignored and the RX FIFO Counter remains unchanged. The data, read from the empty RX FIFO, is undetermined.

When the RX FIFO is not empty, the RX FIFO Drain Flag (RFDF) in the SR is set. The RFDF bit is cleared when the RX_FIFO is empty and the DMA controller indicates that a read from POPR is complete or by writing a 1 to it.

### 40.5.3 Deserial Serial Interface (DSI) Configuration

The DSI Configuration supports pin count reduction by serializing Parallel Input signals or register bits and shifting them out in a SPI-like protocol. The timing and transfer protocol is described in Transfer formats. The received serial frames are converted to a parallel form (deserialized) and placed on the Parallel Output signals or in the DSPI_DDR register. The various features of the DSI Configuration are set in DSPI DSI Configuration Registers (DSPI_DSICR0/1).

The DSI frames can be from 4 to 32 bits. This can go up to 64 bits when DSICR1[DSI64E] is enabled.

With Multiple Transfer Operation (MTO) the DSPI supports serial chaining of DSPI blocks within the device to create DSI frames up to 64 bits, consisting of concatenated bits from multiple DSPIs. The DSPI also supports parallel chaining allowing several DSPIs and off-chip SPI devices to share the same Serial Communications Clock (SCK) and Peripheral Chip Select (PCS) signals. See Multiple Transfer Operation (MTO) for details on the serial and parallel chaining support.

Multiple Transfer Operation (MTO) and DSICR1[DSI64E] are not supported together and cannot be enabled simultaneously.

#### 40.5.3.1 DSI Master Mode

In DSI master mode the DSPI initiates and controls the DSI transfers. The DSI master has different conditions that can initiate a transfer:

- Continuous
- Change in data
- Trigger signal

- ASDR write
- Trigger signal combined with a change in data
- Trigger signal combined with ASDR write
- Change in data combined with ASDR write
- Trigger signal combined with change in data combined with ASDR write

These transfer initiation conditions are described in DSI Transfer Initiation Control . Transfer attributes are set during initialization. The DSICTAS field in the DSPI_DSICR0 determines which of the DSPI_CTAR registers will control the transfer attributes.

### 40.5.3.2 Slave Mode

In DSI slave mode the DSPI responds to transfers initiated by a SPI or DSI bus master. In this mode the DSPI does not initiate DSI transfers. Certain transfer attributes such as clock polarity and phase must be set for successful communication with a DSI master. The DSI slave mode transfer attributes are set in the DSPI_CTAR1. The data is shifted out with MSB first.

If the CID bit in the DSPI_DSICR0 is set and the data in DSPI_COMPR0/1 differs from the selected source of the serialized data, the slave DSPI will assert the $\overline{\text{MTRIG}}$ signal. If the slave's HT signal is asserted and the TRRE is set, the slave DSPI asserts $\overline{\text{MTRIG}}$. Also, if TRIG[ASDR_WT] is set and an ASDR0 write is detected, the slave DSPI will assert the $\overline{\text{MTRIG}}$ signal. These features are included to support chaining of several DSPI. Details about the $\overline{\text{MTRIG}}$ signal is found in Multiple Transfer Operation (MTO).

### 40.5.3.3 DSI Serialization

In the DSI Configuration, up to 64 bits can be serialized using two different sources. The TXSS bit in the DSPI_DSICR0 selects between the DSPI DSI Serialization Data Register (DSPI_SDR0–1) and the DSPI DSI Alternate Serialization Data Register (DSPI_ASDR0–1) as the source of the serialized data. The DSPI_SDR0 holds the latest 32 Parallel Input signal values which is sampled at every rising edge of the bus clock. When DSICR1[DSI64E] is enabled, 64-bit DSI Frames are supported and hence DSPI_SDR1 holds up to 32 most significant parallel input signal values which is sampled at every rising edge of the bus clock.The DSPI_ASDR0–1 register is written by host software and used as an alternate source of serialized data. (DSPI_ASDR1 is only usable when DSICR1[DSI64E] is enabled).

The DSPI has to sample the inputs (from SDR0–1 or ASDR0–1) before it transmits a DSI frame. Due to the asynchronous clocking structure within DSPI, the sampled data (from SDR0–1 or ASDR0–1) must be synchronized into the Protocol Domain before being loaded into the Shift Register for transmission.

To prevent Clock Domain Crossing issues, inputs from SDR0–1 or ASDR0–1 should not change while they are being sampled into the Protocol Domain. This is ensured in the following manner for the ASDR0 inputs. The initial write performed on ASDR0 register will cause the 32-bit ASDR0 input to be latched into a temporary register which is then locked (further latching is prevented) until this data gets synchronized into the Protocol Domain. Once synchronized, the lock is removed and the next ASDR0 write will again trigger this process. If there was a write performed on ASDR0 while the lock was asserted, this will be taken into consideration as soon as the previous synchronization process completes and lock gets de-asserted. In case,multiple writes were performed on ASDR0 when lock was asserted, only the latest value will be considered for synchronization once the lock gets de-asserted. This process is true for ASDR1 register as well.

The entire process explained above is also true for the SDR0 and SDR1 inputs, the only difference being that SDR0–1 inputs are delayed internally by a single Bus clock (register interface clock) to detect a change in the parallel input pins.

Once the synchronization process for a particular De-serial input begins, it takes up to 5 Bus clock (register interface clock) periods and 5 Protocol clock periods before the synchronization process for the next De-serial input can begin. Hence the writes performed on ASDR0–1 and SDR0–1 are constrained with this timing.

The DSPI_PISR0-3 registers (DSPI_PISR0-7 registers if DSICR1[DSI64E] is enabled) allow to change relative position of the Parallel input pins in the transmitted frame. Each transmitted frame bit can be selected from 16 adjacent Parallel Inputs by writing IPSn fields. The IPSn field is treated as 4 bit integer number, representing numbers from -8 to 7. The Parallel Input pin number, selected by IPSn field is defined by the difference between the IPSn field number (n) and the IPSn field value. If the operation result is negative the number 32 should be added. If the result is higher than 32, 32 should be subtracted from the result. (When DSICR1[DSI64E] is enabled, if the above operation result is negative, the number 64 should be added. If the result is higher than 64, 64 should be subtracted from the result).

For example, IPS0, set to minus 1 (binary 1111), preselects Parallel Input 1 to 0 position in the transmitted frame. (This preselects parallel input to 0 position in the transmitted frame when DSICR1[DSI64E] is set). Following examples are true when DSICR1[DSI64E] is disabled.

IPS6, set to 3 (binary 0011), preselects Parallel Input 3 to be bit number 6 in the transmitted frame. The value minus 2 (1110) preselects Parallel Input 8.

IPS31, set to minus 8 (binary 1000), preselects Parallel Input 7 to be bit number 31 in the transmitted frame.

### Note

The Parallel Input pin state, to be transmitted, should be selected by TXSS or DSPI_SSR0–1 register and the frame size should be higher than the bit position in the preselected frame.

DSPI_SSR0–1 register provides additional way to create the frame for transmission. Each bit from this register is OR'd with the TXSS bit and controls individual transmitted bit source. This way, the transmitted frame can have any combination of the DSPI_SDR0–1 and DSPI_ASDR0–1 bits. This feature allows control SPI based devices, requiring control and data fields in the frame. Control field may come from DSPI_ASDR0–1 register, set by the device's CPU, while data field can be generated by device peripheral modules, like PWM timers.

A copy of the last DSI frame shifted out of the Shift Register is stored in the DSPI DSI Transmit Comparison Registers (DSPI_COMPR0–1). This register provides added visibility for debugging and it serves as a reference for transfer initiation control. The following figure shows the DSI Serialization logic.

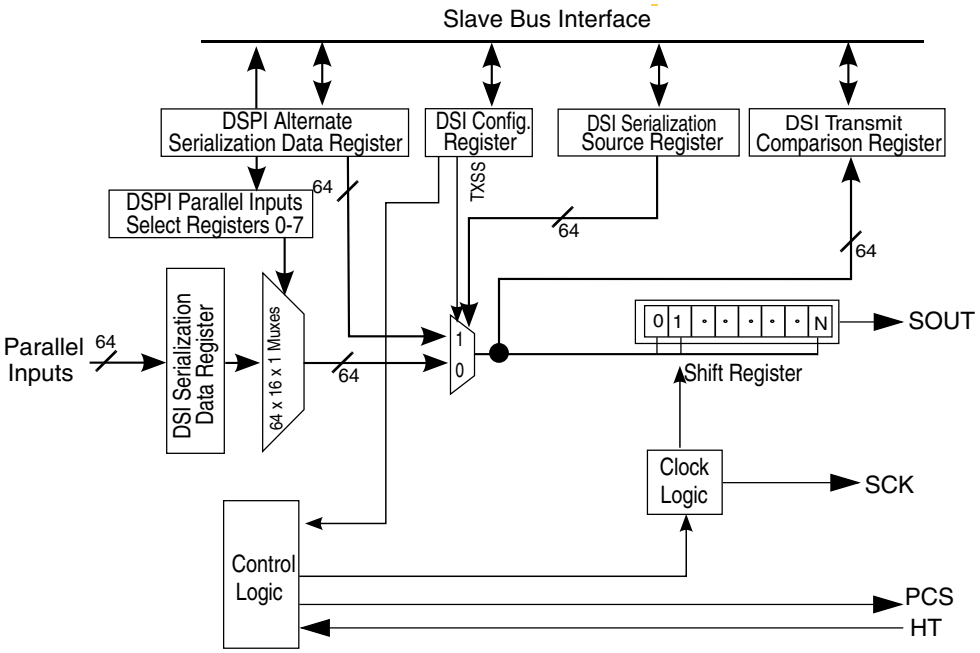



**Figure 40-4. DSI Serialization Diagram**

## 40.5.3.4 DSI Deserialization

When all bits in a DSI frame have been shifted in, the frame is copied to the DSPI DSI Deserialization Data Register (DSPI_DDR0/1). This register presents the deserialized data as Parallel Output signal values. DSPI_DDR0/1 is memory mapped to allow host software to read the deserialized data directly.

The received data is bit-wise compared to the value of the DSI Deserialized Data Polarity Interrupt Register (DSPI_DPIR0/1), bit-wise AND'ed with DSI Deserialized Interrupt Mask Register (DSPI_DIMR0/1) and the results OR'ed to produce DDIF flag in the DSPI_SR register; which in turn can cause DDI interrupt request or DMA request based on RSER[DDIF_RE] and RSER[DDIF_DIRS] bits and/or stop DSI frame transmissions if DSICR0[DMS] bit is set.

The following figure shows the DSI Deserialization logic.

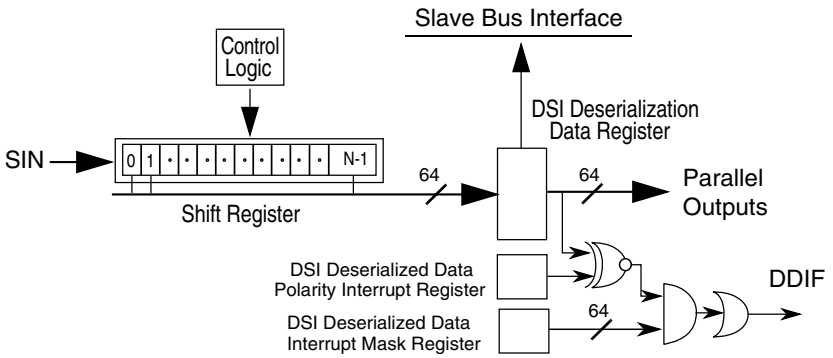



**Figure 40-5. DSI Deserialization Diagram**

## 40.5.3.5 DSI Transfer Initiation Control

Data transfers for a DSPI in DSI configuration are initiated by a condition. The transfer initiation conditions are selected by the TRRE and CID bits in the DSPI_DSICR0.

**NOTE**

By default, the TRRE and CID bits are programmed using DSICR0[TRRE] and DSICR0[CID]. When TRIG[ENABLE] is set, the TRRE and CID bits are given by TRIG[TRRE] and TRIG[CID] respectively and the bits DSICR0[TRRE], DSICR0[CID] have no effect.

The following table lists the transfer initiation conditions.

**Table 40-8. DSI Data Transfer Initiation Control**

| TRRE | CID | ASDR_WT | Transfer Initiation Control |
|---|---|---|---|
| 0 | 0 | 0 | Continuous |
| 0 | 0 | 1 | ASDR Write |
| 0 | 1 | 0 | Change in Data |
| 0 | 1 | 1 | Change in Data or ASDR Write |
| 1 | 0 | 0 | Triggered |
| 1 | 0 | 1 | Triggered or ASDR Write |
| 1 | 1 | 0 | Triggered or Change in Data |
| 1 | 1 | 1 | Triggered or Change in Data or ASDR Write |

#### 40.5.3.5.1 Continuous Control

For Continuous Control a new DSI frame shifts out when the previous transfer cycle has completed and the Delay after Transfer ($t_{DT}$) has elapsed. This has no effect when DSPI is in Slave Mode.

#### 40.5.3.5.2 ASDR Write Control

For ASDR Write Control, a DSI frame shifts out whenever a write operation to ASDR0 is detected. For ASDR_WT to have any affect, TRIG[ASDR_WT] bit must be set. Note that ASDR0 write operations will not generate ASDR write triggers when the module is in STOPPED state, including when MCR[HALT] is asserted.

**NOTE**

If a DSI frame > 32 bits needs to be transmitted, the user must update ASDR1 before updating ASDR0.

#### 40.5.3.5.3 Change In Data Control

The $\overline{\text{MTRIG}}$ output signal is asserted every time a change in data is detected.

For change in data control a transfer is initiated when the data to be serialized has changed since the transfer of the last DSI frame. A copy of the previously transferred DSI data is stored in DSPI_COMPR0/1. When the data selected for the transfer from the DSPI_SDR0/1 and DSPI_ASDR0/1 registers is different from the data in DSPI_COMPR0/1 a new DSI frame is transmitted. The $\overline{\text{MTRIG}}$ output signal is asserted every time a change in data is detected.

**NOTE**

It is not advisable to configure a DSPI slave with the combination DSICR0[DCONT] = 1, DSICR0[CID] = 1 ; since the following statement then holds true.

Once slave select is asserted, there is a very small window (of up to t module clocks) after every SOF (Start of Frame) during which if a write on ASDR/SDR registers occur, the next frame to be transmitted would be loaded into the shift register. However, it is not feasible that such a write be initiated and completed within this short span of time; hence the data transmitted by DSPI slave would lag by one data frame.

#### 40.5.3.5.4 Change In Data or ASDR Write Control

For Change in Data or ASDR Write Control, initiation of a transfer is controlled by the detection of a change in data to be serialized or whenever a write operation to ASDR0 is detected.

**NOTE**

For any DSI transfer initiation combination where TRIG[ASDR_WT] and DSICR0[HT] are both enabled, it is possible that the ASDR0 write operation causes a Change of DSI Data. This special condition results in transmission of a single DSI frame even though 2 separate triggers were generated.

#### 40.5.3.5.5 Triggered Control

For Triggered Control initiation of a transfer is controlled by the Hardware Trigger signal (HT). The TPOL bit in the DSPI_DSICR0 selects the active edge of HT. For HT to have any affect, DSICR0[TRRE] bit must be set. Note that HT will not have any effect when the module is in STOPPED state, including when MCR[HALT] is asserted.

#### 40.5.3.5.6 Triggered or ASDR Write Control

For Triggered or ASDR Write Control, initiation of a transfer is controlled by the HT signal or whenever a write operation to ASDR0 is detected.

#### 40.5.3.5.7 Triggered or Change In Data Control

For Triggered or Change in Data Control, initiation of a transfer is controlled by the HT signal or by the detection of a change in data to be serialized.

#### 40.5.3.5.8 Triggered or Change In Data or ASDR Write Control

For Triggered or Change in Data or ASDR Write Control, initiation of a transfer is controlled by the HT signal or by the detection of a change in data to be serialized or whenever a write operation to ASDR0 is detected.

### 40.5.3.6 Multiple Transfer Operation (MTO)

In DSI Configuration the MTO feature allows for multiple DSPIs within a device to be chained together in a parallel or serial configuration. The parallel chaining allows multiple DSPIs internal to a device and multiple SPI devices external to the device to

share SCK and PCS signals thereby saving the device pins. The serial chaining allows bits from multiple DSPIs to be concatenated into a single DSI frame. MTO is enabled by setting the MTOE bit in the DSPI_DSICR0.

In parallel and serial chaining there is one bus master and multiple bus slaves. The bus master initiates and controls the transfers, but the DSPI slaves generate trigger signals for the bus DSPI master when an internal condition in the slave warrants a transfer. The DSPI slaves also propagate triggers from other slaves to the master. When a DSPI slave detects a trigger signal on its HT input, the slave generates a trigger signal on the $\overline{\text{MTRIG}}$ output.

Serial and parallel chaining require multiplexing of signals external to the DSPI.

**Note**

TSB operation is not available in MTO mode. DSICR0[TSBC] and DSICR0[MTOE] bits should not be set simultaneously.

**Note**

DSI64 mode is not available in MTO mode. DSICR1[DSI64] and DSICR0[MTOE] bits must not be set simultaneously.

#### 40.5.3.6.1 Parallel Chaining

Parallel chaining allows multiple DSPIs internal to a device and multiple SPI/DSI devices external to a device to share common SCK and PCS signals thereby saving pins. Two pins are saved per pair of DSPI/SPI. The following figure shows an example of how the blocks can be connected in a device.

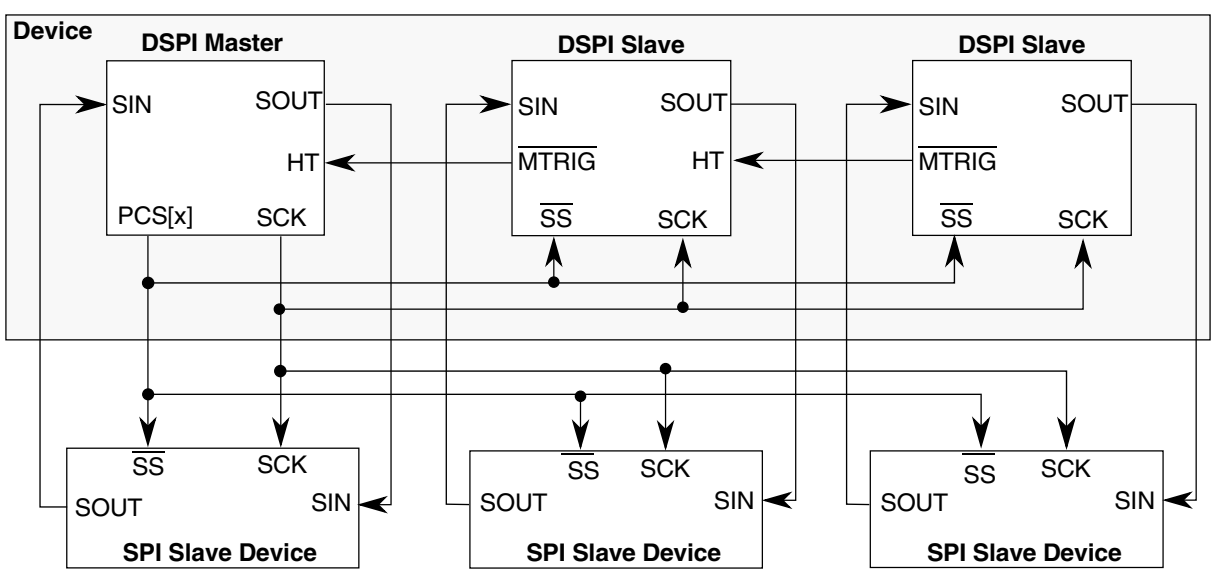



**Figure 40-6. Example of Parallel Chaining of DSPIs**

The DSPI master controls and initiates all transfer, but the DSPI slaves have a trigger output signal $\overline{\text{MTRIG}}$ that indicates to the master DSPI to start a transfer. When the DSPI slave has a change in its data to be serialized, it generates a pulse on the $\overline{\text{MTRIG}}$ signal to the master DSPI, which initiates the transfer. When a DSPI slave has its HT signal asserted it also generates a pulse on its $\overline{\text{MTRIG}}$ signal thereby propagating trigger signals from other DSPI slaves to the DSPI master.

To enable $\overline{\text{MTRIG}}$ signals generation the MTO has to be enabled for the DSPI slaves. The chained DSPI modules also should be in DSI configuration. The DSPI_DSICR0[MTOCNT] field must be written with the same number of bits to be transferred for each slave DSPI. The MTOCNT value must also match the FMSZ field of the DSPI_CTAR1 in the DSPI slaves and selected DSPI_CTAR register for the master DSPI module.

### 40.5.3.6.2 Serial Chaining

The serial chaining allows transfers of DSI frames of up to a total of 64 bits, using transfers of smaller DSI frames concatenated together by multiple DSPIs. The following figure shows an example of how the blocks can be connected in a device.

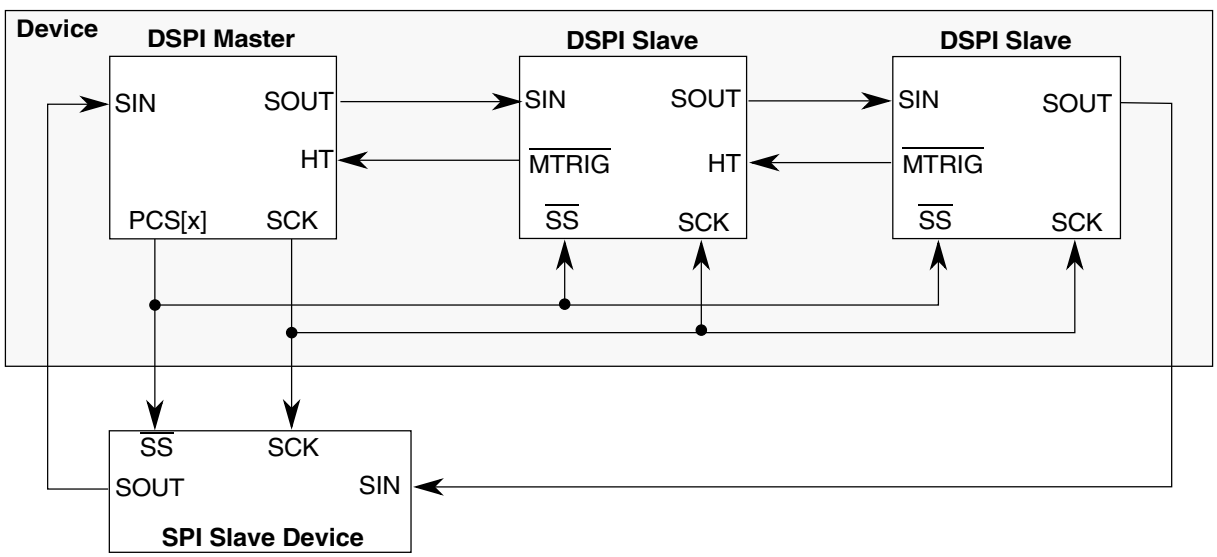



**Figure 40-7. Example of Serial Chaining of DSPIs**

The SOUT of the DSPI master is connected to the SIN of the first DSPI slave. The SOUT of the first DSPI slave is connected to the SIN input of the second slave and so on. The SOUT of the last DSPI slave is connected to the SIN of the external SPI slave. The SOUT of the external SPI slave is connected to the SIN of the DSPI master.

The DSPI master controls and initiates all transfers, but the slave DSPIs use the trigger output signal $\overline{\text{MTRIG}}$ to indicate to the DSPI master that a trigger condition has occurred. When a DSPI slave has a change in data to be serialized it asserts the $\overline{\text{MTRIG}}$ signal to

the DSPI master which initiates the transfer. When a DSPI slave has its HT signal asserted it will assert its $\overline{\text{MTRIG}}$ signal thereby propagating trigger signals from other DSPI slaves to the DSPI master.

The DSPI_DSICR0[MTOE] bit must be set in master and all slave devices. The DSPI_DSICR0[MTOCNT] field for the master and all slave devices must be written with the same value. The value must equal the sum of all FMSZ fields in the selected DSPI_CTAR registers for the DSPI master and all DSPI slaves. For example if one 16-bit DSI frame is created by concatenating eight bits from the DSPI master, and four bits from each of the DSPI slaves in the above figure, the DSPI master frame size must be set to eight in the FMSZ field, and the DSPI slaves frame size must be set to four. The largest DSI frame supported by the MTOCNT field is 64 bits. Any number of DSPIs can be connected together to concatenate DSI frames, as long as each DSPI transfers a minimum of 4 bits and a maximum of 32 bits and the total size of the concatenated frame is less than 65 bits long.

## 40.5.4 Combined Serial Interface (CSI) Configuration

The CSI Configuration of the DSPI is used to support SPI and DSI functions on a frame by frame basis. CSI Configuration allows interleaving of DSI data frames from the parallel input signals with SPI data frames from the TX FIFO. The data returned from the bus slave is either used to drive the Parallel Output signals or it is stored in the RX FIFO. The CSI Configuration allows serialized data and configuration or diagnostic data to be transferred to a slave device using only one serial link. The DSPI is in CSI Configuration when the DCONF field in the DSPI_MCR is 0b10. The following figure shows an example of how a DSPI can be used with a deserializing peripheral that supports SPI control for control and diagnostic frames.

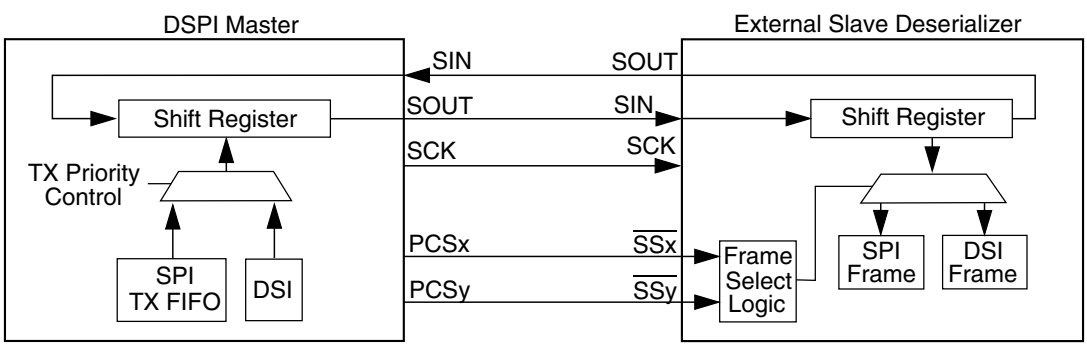



**Figure 40-8. Example of System using DSPI in CSI Configuration**

In CSI Configuration the DSPI transfers DSI data based on DSI Transfer Initiation Control .

When DSICR1[CSI_PRTY] is set to zero and there are SPI commands in the TX FIFO, the SPI frames have priority over the DSI frames. When the TX FIFO or CMD FIFO is empty, DSI transfer resumes.

When DSICR1[CSI_PRTY] bit is set, the DSI frames have priority over SPI frames. Thus, only when the DSI frame initiation requests have been catered to, will the SPI frames be transmitted, if available.

**NOTE**

The following combination is not a legal scenario and should be avoided - DSICR1[CSI_PRTY] = 1, DSICR1[TRRE] = 0 and DSICR1[CID] = 0, since such a configuration prevents SPI frames from ever getting transmitted.

Two peripheral chip select signals indicate whether DSI data or SPI data is transmitted. The user must configure the DSPI so that the two CTAR registers associated with DSI data and SPI data assert different peripheral chip select signals denoted in the figure as PCSx and PCSy. The CSI Configuration is only supported in master mode.

Data returned from the external slave while a DSI frame is transferred is placed on the Parallel Output signals. Data returned from the external slave while a SPI frame is transferred is moved to the RX FIFO. The TX FIFO, CMD FIFO and RX FIFO are fully functional in CSI mode.

### 40.5.4.1 CSI Serialization

Serialization in the CSI configuration is similar to serialization in DSI Configuration. The transfer attributes for SPI frames are determined by the DSPI_CTAR register selected by the CTAS field in the SPI command halfword. The transfer attributes for the DSI frames are determined by the DSPI_CTAR register selected by the DSICTAS field in the DSPI_DSICR0.

The Parallel Inputs signal states are latched into the DSPI DSI Serialization Data Register (DSPI_SDR0/1) and serialized based on the transfer initiation control settings in the DSPI_DSICR0.When SPI frames are written to the TX FIFO (and when CMD FIFO is not empty) they have priority over DSI data from the DSPI_SDR and are transferred at the next frame boundary.A copy of the most recently transferred DSI frame is stored in DSPI_COMPR0/1. The Transfer Priority Logic selects the source of the serialized data and asserts the appropriate PCS signal.

Separate frame completion interrupts are available to indicate frame transmission completion from SPI and DSI frames.

### 40.5.4.2 CSI Deserialization

The deserialized frames in CSI Configuration go into the DSPI_DDR0/1 or the RX FIFO based on the transfer priority logic.When DSI frames are transferred the returned frames are deserialized and latched into the DSPI_DDR0/1. When SPI frames are transferred the returned frames are deserialized and written to the RX FIFO.

## 40.5.5 Module baud rate and clock delay generation

The SCK frequency and the delay values for serial transfer are generated by dividing the protocol clock frequency by a prescaler and a scaler with the option for doubling the baud rate. The following figure shows conceptually how the SCK signal is generated.

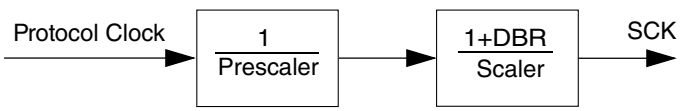



**Figure 40-9. Communications clock prescalers and scalers**

### 40.5.5.1 Baud rate generator

The baud rate is the frequency of the SCK. The protocol clock is divided by a prescaler (PBR) and scaler (BR) to produce SCK with the possibility of halving the scaler division. The DBR, PBR, and BR fields in the CTARs select the frequency of SCK by the formula in the BR field description. The following table shows an example of how to compute the baud rate.

**Table 40-9. Baud rate computation example**

| $f_P$ | PBR | Prescaler | BR | Scaler | DBR | Baud rate |
|---|---|---|---|---|---|---|
| 100 MHz | 0b00 | 2 | 0b0000 | 2 | 0 | 25 Mb/s |
| 20 MHz | 0b00 | 2 | 0b0000 | 2 | 1 | 10 Mb/s |

**NOTE**

The clock frequencies mentioned in the preceding table are given as an example. Refer to the clocking chapter for the frequency used to drive this module in the device.

### 40.5.5.2 PCS to SCK Delay ($t_{CSC}$)

The PCS to SCK delay is the length of time from assertion of the PCS signal to the first SCK edge. See Figure 40-11 for an illustration of the PCS to SCK delay. The PCSSCK and CSSCK fields in the CTAR*x* registers select the PCS to SCK delay by the formula in the CSSCK field description. The following table shows an example of how to compute the PCS to SCK delay.

**Table 40-10. PCS to SCK delay computation example**

| $f_{SYS}$ | PCSSCK | Prescaler | CSSCK | Scaler | PCS to SCK Delay |
|---|---|---|---|---|---|
| 100 MHz | 0b01 | 3 | 0b0100 | 32 | 0.96 µs |

PCSCSK and CSSCK fields have no effect in TSB configuration.

**NOTE**

The clock frequency mentioned in the preceding table is given as an example. Refer to the clocking chapter for the frequency used to drive this module in the device.

### 40.5.5.3 After SCK Delay ($t_{ASC}$)

The After SCK Delay is the length of time between the last edge of SCK and the negation of PCS. See Figure 40-11 and Figure 40-12 for illustrations of the After SCK delay. The PASC and ASC fields in the CTAR*x* registers select the After SCK Delay by the formula in the ASC field description. The following table shows an example of how to compute the After SCK delay.

**Table 40-11. After SCK Delay computation example**

| $f_P$ | PASC | Prescaler | ASC | Scaler | After SCK Delay |
|---|---|---|---|---|---|
| 100 MHz | 0b01 | 3 | 0b0100 | 32 | 0.96 µs |

PCASC and ASC fields have no effect in TSB configuration.

**NOTE**

The clock frequency mentioned in the preceding table is given as an example. Refer to the clocking chapter for the frequency used to drive this module in the device.

### 40.5.5.4 Delay after Transfer ($t_{DT}$)

The Delay after Transfer is the minimum time between negation of the PCS signal for a frame and the assertion of the PCS signal for the next frame. See Figure 40-11 for an illustration of the Delay after Transfer. The PDT and DT fields in the CTAR*x* registers select the Delay after Transfer by the formula in the DT field description. The following table shows an example of how to compute the Delay after Transfer.

**Table 40-12. Delay after Transfer computation example**

| $f_P$ | PDT | Prescaler | DT | Scaler | Delay after Transfer |
|---|---|---|---|---|---|
| 100 MHz | 0b01 | 3 | 0b1110 | 32768 | 0.98 ms |

**NOTE**

The clock frequency mentioned in the preceding table is given as an example. Refer to the clocking chapter for the frequency used to drive this module in the device.

When in Non-Continuous Clock mode the $t_{DT}$ delay is configured according to the equation specified in the CTAR[DT] field description. When in Continuous Clock mode and TSB is not enabled, the delay is fixed at 1 SCK period.

In TSB mode the Delay after Transfer is equal to a number formed by concatenation of PDT and DT fields plus 1 of the SCK clock periods. See detailed information on Timed Serial Bus (TSB)".

### 40.5.5.5 Peripheral Chip Select Strobe Enable ($\overline{\text{PCSS}}$)

The $\overline{\text{PCSS}}$ signal provides a delay to allow the PCS signals to settle after a transition occurs thereby avoiding glitches. When the Module is in Master mode and the PCSSE bit is set in the MCR, $\overline{\text{PCSS}}$ provides a signal for an external demultiplexer to decode peripheral chip selects other than PCS5 into glitch-free PCS signals. The following figure shows the timing of the $\overline{\text{PCSS}}$ signal relative to PCS signals.

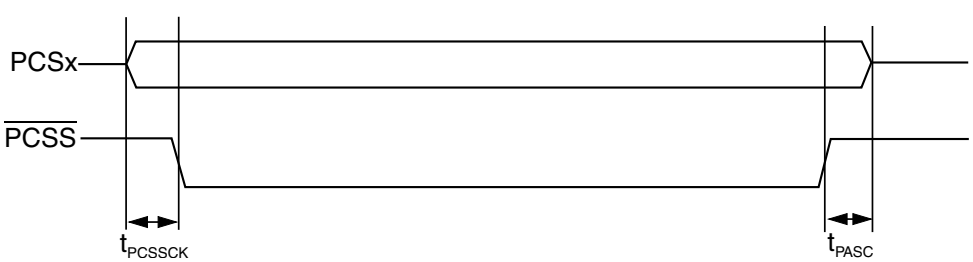



**Figure 40-10. Peripheral Chip Select Strobe timing**

The delay between the assertion of the PCS signals and the assertion of $\overline{\text{PCSS}}$ is selected by the PCSSCK field in the CTAR based on the following formula:

$$t_{\text{PCSSCK}} = \frac{1}{f_{\text{P}}} \times \text{PCSSCK}$$

At the end of the transfer, the delay between $\overline{\text{PCSS}}$ negation and PCS negation is selected by the PASC field in the CTAR based on the following formula:

$$t_{\text{PASC}} = \frac{1}{f_{\text{P}}} \times \text{PASC}$$

The following table shows an example of how to compute the $t_{pcssck}$ delay.

**Table 40-13. Peripheral Chip Select Strobe Assert computation example**

| $f_P$ | PCSSCK | Prescaler | Delay before Transfer |
|---|---|---|---|
| 100 MHz | 0b11 | 7 | 70.0 ns |

The following table shows an example of how to compute the $t_{pasc}$ delay.

**Table 40-14. Peripheral Chip Select Strobe Negate computation example**

| $f_P$ | PASC | Prescaler | Delay after Transfer |
|---|---|---|---|
| 100 MHz | 0b11 | 7 | 70.0 ns |

The $\overline{\text{PCSS}}$ signal is not supported when Continuous Serial Communication SCK or TSB mode is enabled.

**NOTE**

The clock frequency mentioned in the preceding tables is given as an example. Refer to the clocking chapter for the frequency used to drive this module in the device.

## 40.5.6 Transfer formats

The SPI serial communication is controlled by the Serial Communications Clock (SCK) signal and the PCS signals. The SCK signal provided by the master device synchronizes shifting and sampling of the data on the SIN and SOUT pins. The PCS signals serve as enable signals for the slave devices.

In Master mode, the CPOL and CPHA bits in the Clock and Transfer Attributes Registers (CTARn) select the polarity and phase of the serial clock, SCK.

- CPOL - Selects the idle state polarity of the SCK
- CPHA - Selects if the data on SOUT is valid before or on the first SCK edge

Even though the bus slave does not control the SCK signal, in Slave mode the values of CPOL and CPHA must be identical to the master device settings to ensure proper transmission. In SPI Slave mode, only CTAR0 is used. In DSI slave mode, only CTAR1 is used.

The module supports four different transfer formats:

- Classic SPI with CPHA=0
- Classic SPI with CPHA=1
- Modified Transfer Format with CPHA = 0
- Modified Transfer Format with CPHA = 1

A modified transfer format is supported to allow for high-speed communication with peripherals that require longer setup times. The module can sample the incoming data later than halfway through the cycle to give the peripheral more setup time. The MTFE bit in the MCR selects between Classic SPI Format and Modified Transfer Format.

In the interface configurations, the module provides the option of keeping the PCS signals asserted between frames. See Continuous Selection Format for details.

### 40.5.6.1 Classic SPI Transfer Format (CPHA = 0)

The transfer format shown in following figure is used to communicate with peripheral SPI slave devices where the first data bit is available on the first clock edge. In this format, the master and slave sample their SIN pins on the odd-numbered SCK edges and change the data on their SOUT pins on the even-numbered SCK edges.

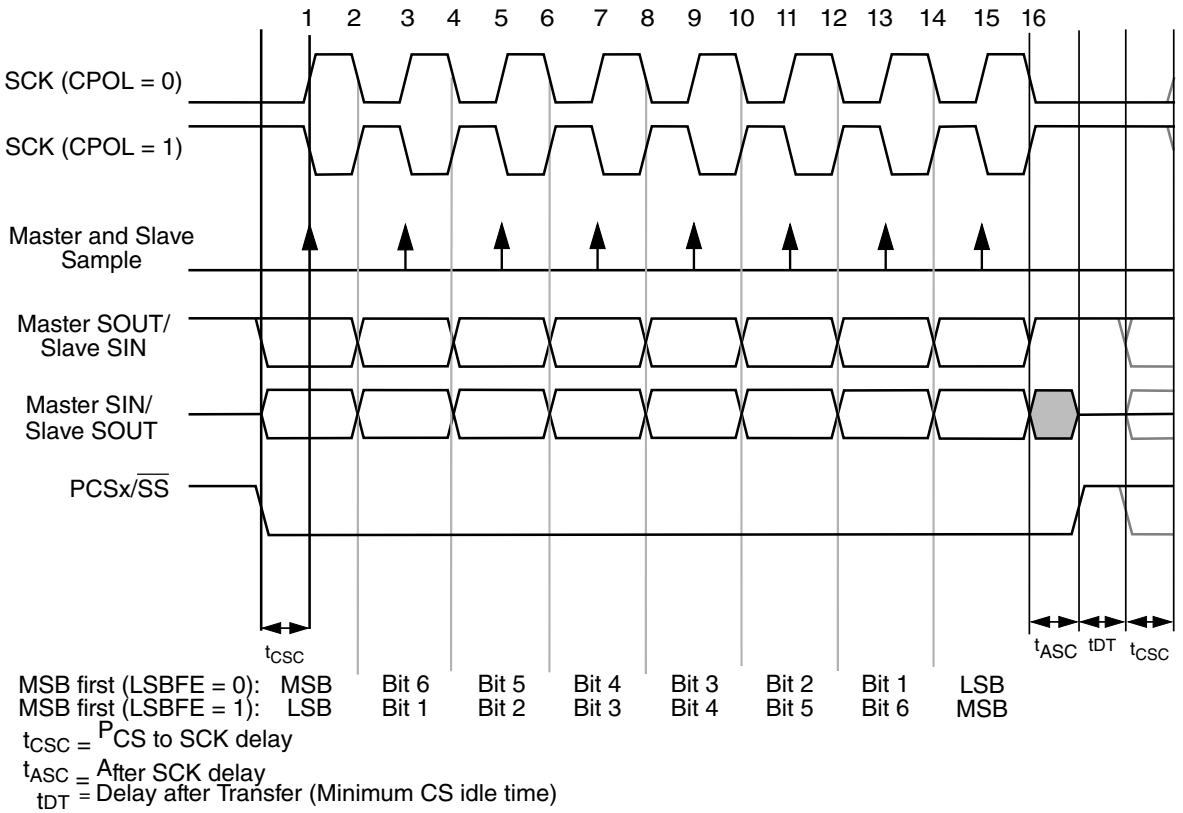



**Figure 40-11. Module transfer timing diagram (MTFE=0, CPHA=0, FMSZ=8)**

The master initiates the transfer by placing its first data bit on the SOUT pin and asserting the appropriate peripheral chip select signals to the slave device. The slave responds by placing its first data bit on its SOUT pin. After the $t_{CSC}$ delay elapses, the master outputs the first edge of SCK. The master and slave devices use this edge to sample the first input data bit on their serial data input signals. At the second edge of the SCK, the master and slave devices place their second data bit on their serial data output signals. For the rest of the frame the master and the slave sample their SIN pins on the odd-numbered clock edges and changes the data on their SOUT pins on the even-numbered clock edges. After the last clock edge occurs, a delay of $t_{ASC}$ is inserted before the master negates the PCS signals. A delay of $t_{DT}$ is inserted before a new frame transfer can be initiated by the master.

## 40.5.6.2 Classic SPI Transfer Format (CPHA = 1)

This transfer format shown in the following figure is used to communicate with peripheral SPI slave devices that require the first SCK edge before the first data bit becomes available on the slave SOUT pin. In this format, the master and slave devices change the data on their SOUT pins on the odd-numbered SCK edges and sample the data on their SIN pins on the even-numbered SCK edges.

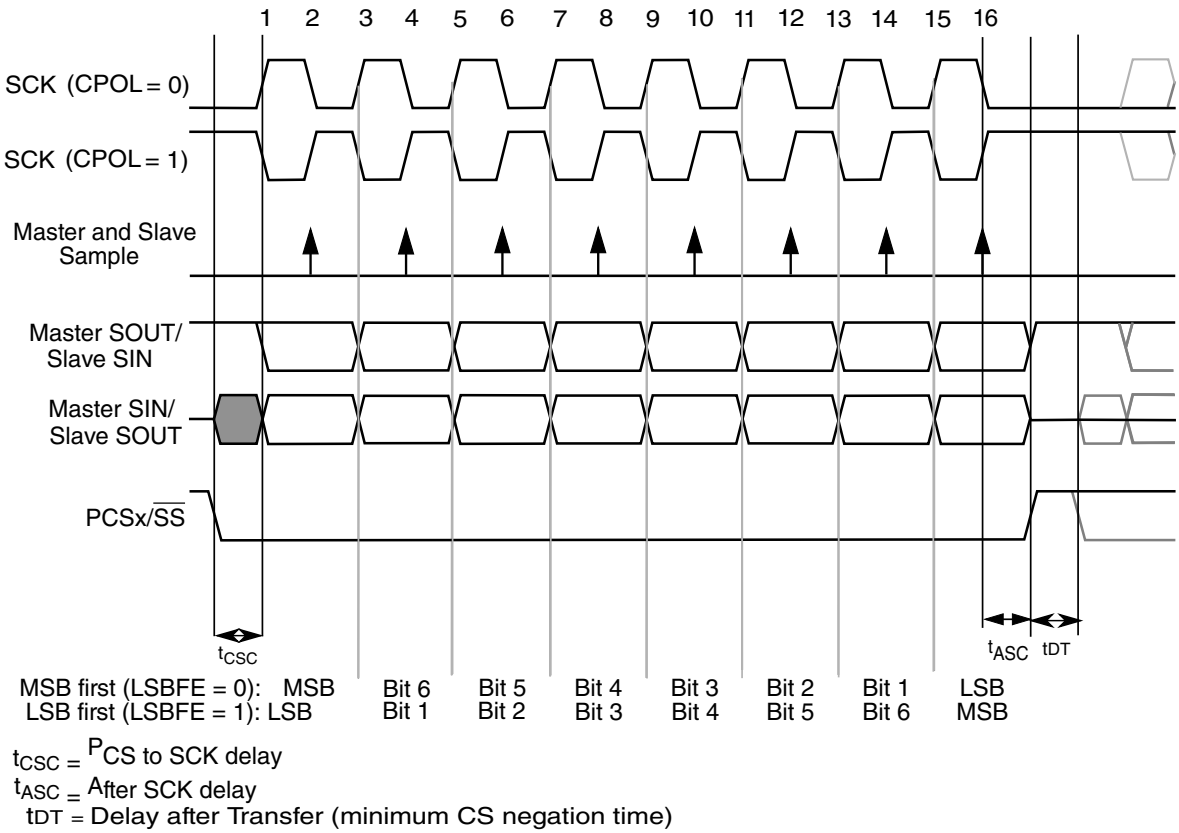



**Figure 40-12. Module transfer timing diagram (MTFE=0, CPHA=1, FMSZ=8)**

The master initiates the transfer by asserting the PCS signal to the slave. After the $t_{CSC}$ delay has elapsed, the master generates the first SCK edge and at the same time places valid data on the master SOUT pin. The slave responds to the first SCK edge by placing its first data bit on its slave SOUT pin.

At the second edge of the SCK the master and slave sample their SIN pins. For the rest of the frame the master and the slave change the data on their SOUT pins on the odd-numbered clock edges and sample their SIN pins on the even-numbered clock edges. After the last clock edge occurs, a delay of $t_{ASC}$ is inserted before the master negates the PCS signal. A delay of $t_{DT}$ is inserted before a new frame transfer can be initiated by the master.

### 40.5.6.3 Modified SPI/DSI Transfer Format (MTFE = 1, CPHA = 0)

In this Modified Transfer Format both the master and the slave sample later in the SCK period than in Classic SPI mode to allow the logic to tolerate more delays in device pads and board traces. These delays become a more significant fraction of the SCK period as the SCK period decreases with increasing baud rates.

The master and the slave place data on the SOUT pins at the assertion of the PCS signal. After the PCS to SCK delay has elapsed the first SCK edge is generated. The slave samples the master SOUT signal on every odd numbered SCK edge. The DSPI in the slave mode when the MTFE bit is set also places new data on the slave SOUT on every odd numbered clock edge. Regular external slave, configured with CPHA=0 format drives its SOUT output at every even numbered SCK clock edge.

The DSPI master places its second data bit on the SOUT line one protocol clock after odd numbered SCK edge if the protocol clock frequency to SCK frequency ratio is higher than three. If this ratio is below four the master changes SOUT at odd numbered SCK edge. The point where the master samples the SIN is selected by the DSPI_MCR[SMPL_PT] field. The master sample point can be delayed by one or two protocol clock cycles. The SMPL_PT field should be set to 0 if the protocol to SCK frequency ratio is less than 4. However if this ratio is less than 4, the actual sample point is delayed by one protocol clock cycle automatically by the design.

The following timing diagrams illustrate the DSPI operation with MTFE=1. Timing delays shown are:

- $T_{csc}$ - PCS to SCK assertion delay
- $T_{acs}$ - After SCK PCS negation delay
- $T_{su\_ms}$ - master SIN setup time
- $T_{hd\_ms}$ - master SIN hold time
- $T_{vd\_sl}$ - slave data output valid time, time between slave data output SCK driving edge and data becomes valid.
- $T_{su\_sl}$ - data setup time on slave data input
- $T_{hd\_sl}$ - data hold time on slave data input
- $T_{sys}$ - protocol clock period.

The following figure shows the modified transfer format for CPHA = 0 and Fsys/Fsck = 4. Only the condition where CPOL = 0 is illustrated. Solid triangles show the data sampling clock edges. The two possible slave behavior are shown.

- Signal, marked "SOUT of Ext Slave", presents regular SPI slave serial output.
- Signal, marked "SOUT of DSPI Slave", presents DSPI in the slave mode with MTFE bit set.

Other MTFE = 1 diagrams show DSPI SIN input as being driven by a regular external SPI slave, configured according DSPI master CPHA programming.

## Note

In the following diagrams, $f_{sys}$ represents the protocol clock frequency from which the Baud frequency $f_{sck}$ is derived.

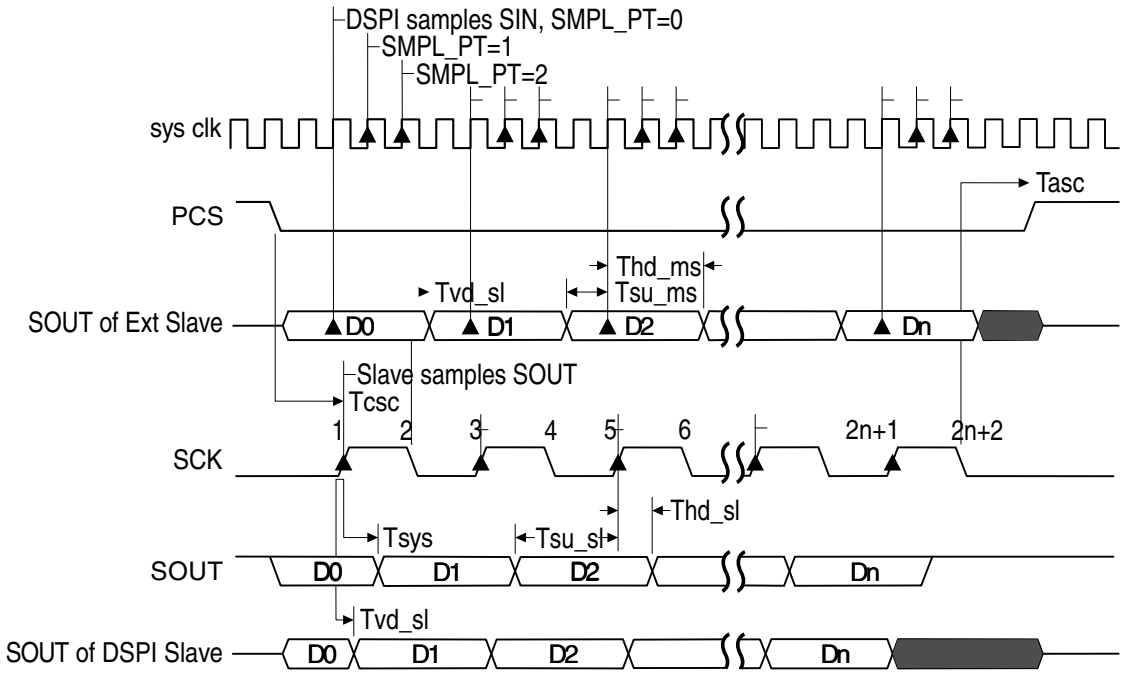



**Figure 40-13. DSPI Modified Transfer Format (MTFE=1, CPHA=0, $f_{sck} = f_{sys}/4$)**

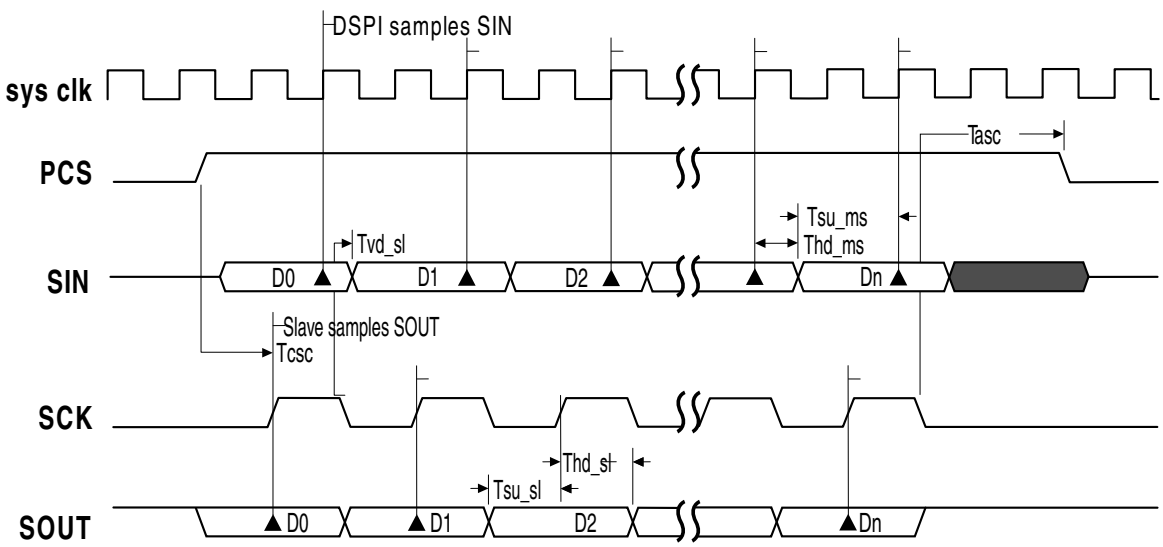



**Figure 40-14. DSPI Modified Transfer Format (MTFE=1, CPHA=0, $f_{sck} = f_{sys}/2$)**

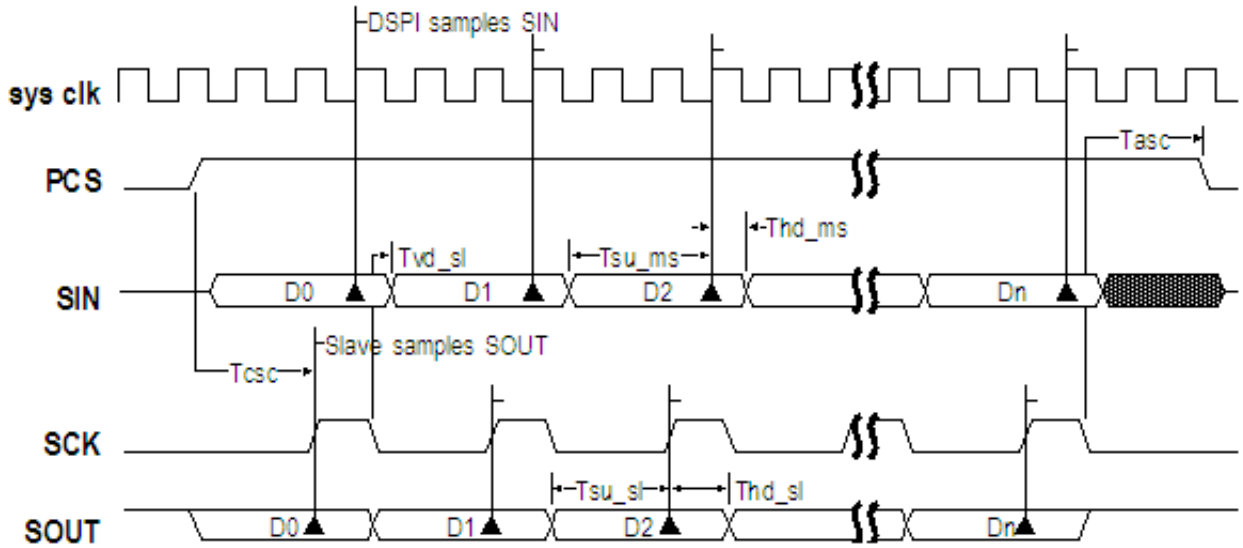



**Figure 40-15. DSPI Modified Transfer Format (MTFE=1, CPHA=0, $f_{sck} = f_{sys}/3$)**

### 40.5.6.4 Modified SPI/DSI Transfer Format (MTFE = 1, CPHA = 1)

The following figures show the Modified Transfer Format for CPHA = 1. Only the condition, where CPOL = 0 is shown. At the start of a transfer the DSPI asserts the PCS signal to the slave device. After the PCS to SCK delay has elapsed the master and the slave put data on their SOUT pins at the first edge of SCK . The slave samples the master SOUT signal on the even numbered edges of SCK. The master samples the slave SOUT signal on the odd numbered SCK edges starting with the third SCK edge. The slave samples the last bit on the last edge of the SCK. The master samples the last slave SOUT bit one half SCK cycle after the last edge of SCK. No clock edge will be visible on the master SCK pin during the sampling of the last bit. **The SCK to PCS delay and the After SCK delay must be greater or equal to half of the SCK period.**

**NOTE**

When MTFE=1 with continuous SCK enabled (MCR [CONT_SCKE] =1) in master mode, configure CTAR[LSBFE]=0 for correct operations while receiving unequal length frames. If PUSHR[CONT] is also set for back to back frame transfer, also configure the frame size of the first frame as less than or equal to the frame size of the next frame. In this scenario, make sure that for all received frames, the bits are read equal to their respective frame sizes and any extra bits during POP operation are masked.

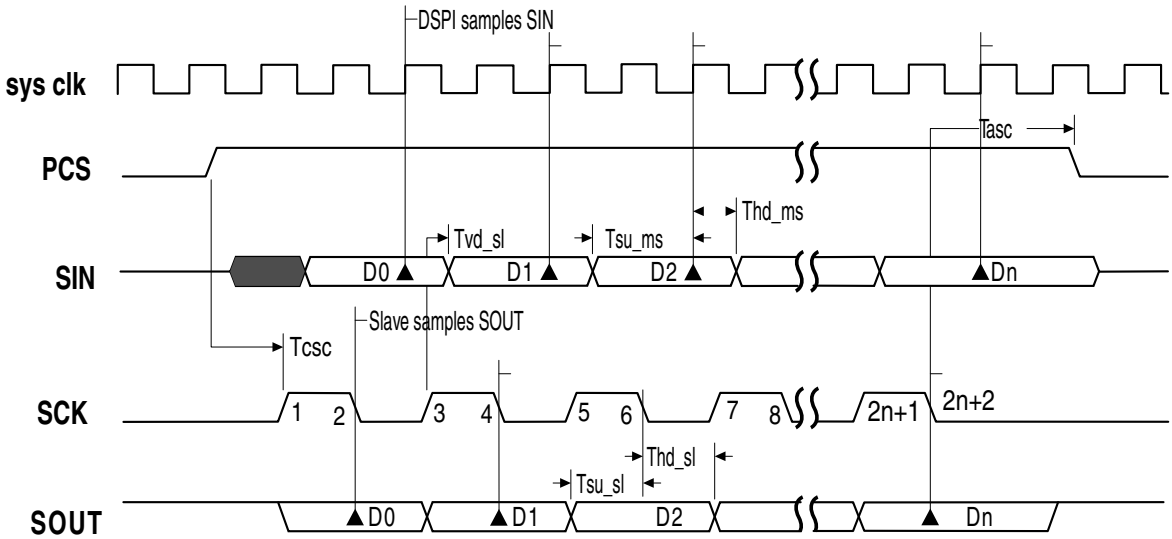



**Figure 40-16. DSPI Modified Transfer Format (MTFE=1, CPHA=1, $f_{sck} = f_{sys}/2$)**

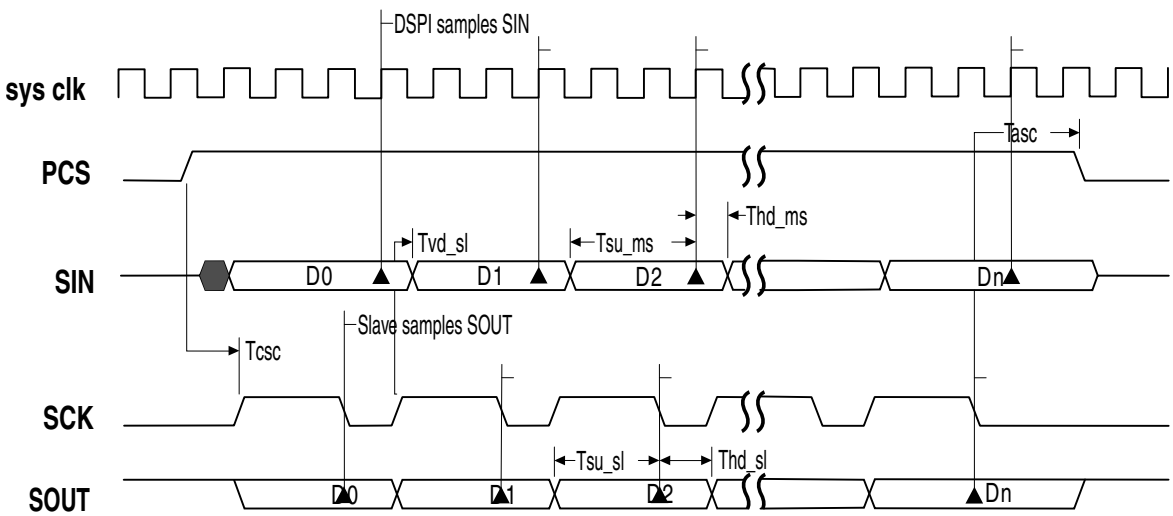



**Figure 40-17. DSPI Modified Transfer Format (MTFE=1, CPHA=1, $f_{sck} = f_{sys}/3$)**

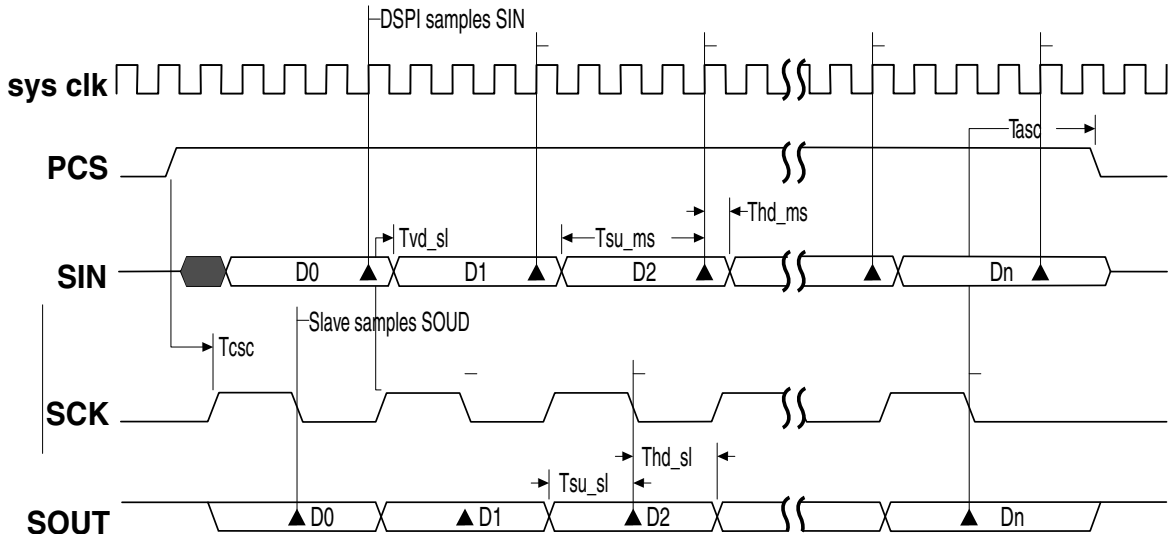



**Figure 40-18. DSPI Modified Transfer Format (MTFE=1, CPHA=1, $f_{sck} = f_{sys}/4$)**

## 40.5.6.5 Continuous Selection Format

Some peripherals must be deselected between every transfer. Other peripherals must remain selected between several sequential serial transfers. The Continuous Selection Format provides the flexibility to handle both the following cases. The Continuous Selection Format is enabled for the SPI configuration by setting the CONT bit in the SPI command. Continuous Selection is enabled for the DSI Configuration by setting the DCONT bit in the DSICR0.The behavior of the PCS signals in the two configurations is identical so only SPI configuration will be described.

When the CONT bit = 0, the module drives the asserted Chip Select signals to their idle states in between frames. The idle states of the Chip Select signals are selected by the PCSISn bits in the MCR. The following timing diagram is for two four-bit transfers with CPHA = 1 and CONT = 0.

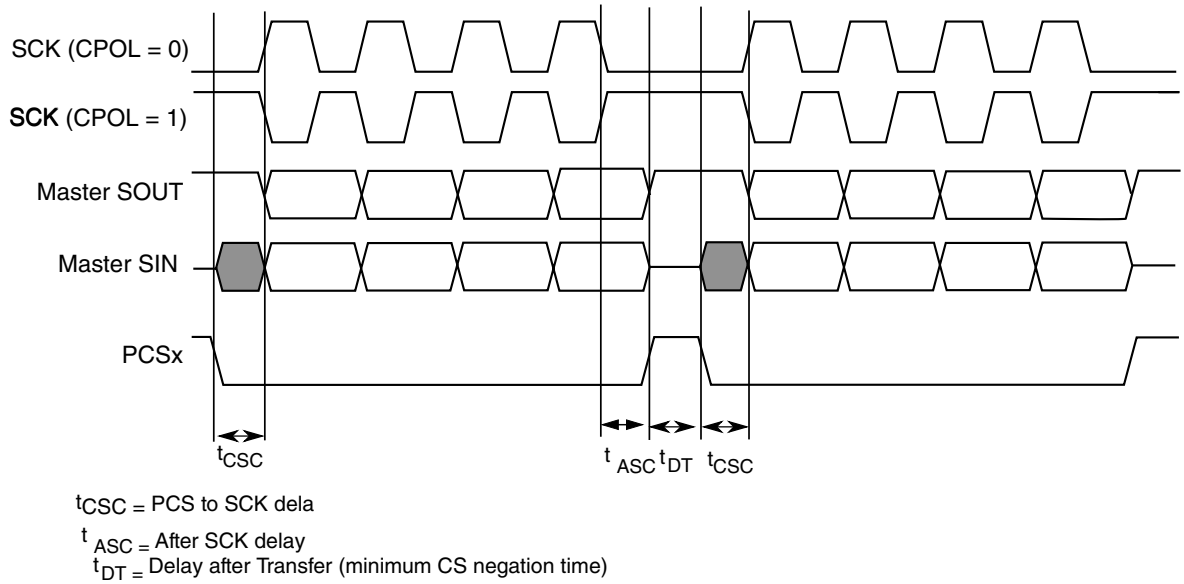



**Figure 40-19. Example of non-continuous format (CPHA=1, CONT=0)**

When the CONT bit = 1, the PCS signal remains asserted for the duration of the two transfers. The Delay between Transfers ($t_{DT}$) is not inserted between the transfers. The following figure shows the timing diagram for two four-bit transfers with CPHA = 1 and CONT = 1.

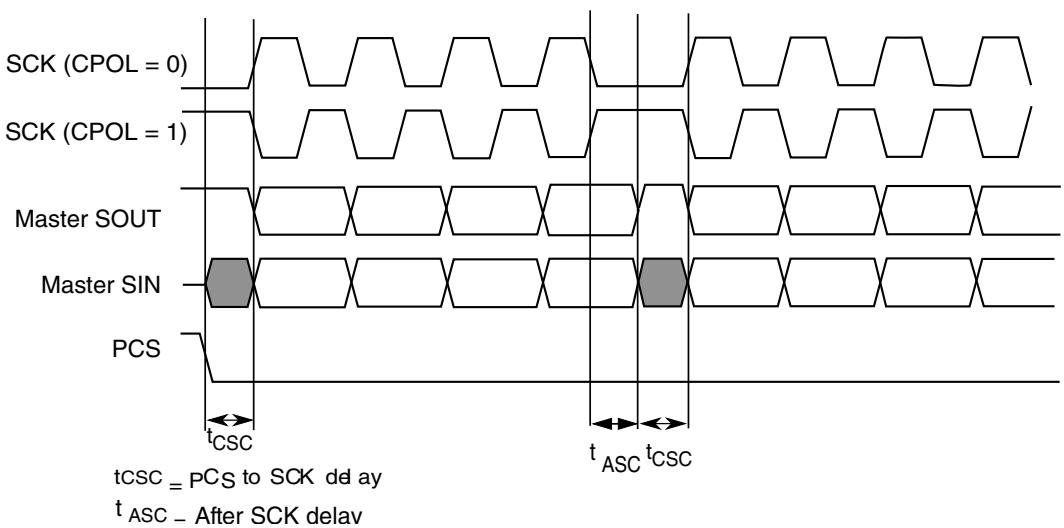



**Figure 40-20. Example of continuous transfer (CPHA=1, CONT=1)**

When using the module with continuous selection follow these rules:

- All transmit commands must have the same PCSn bits programming.
- The CTARs, selected by transmit commands, must be programmed with the same transfer attributes. Only FMSZ field can be programmed differently in these CTARs.

- When transmitting multiple frames in this mode, the user software must ensure that the last frame has the PUSHR[CONT] bit deasserted in Master mode and the user software must provide sufficient frames in the TX_FIFO to be sent out in Slave mode and the master deasserts the PCSn at end of transmission of the last frame.
- PUSHR[CONT] and DSICR0[DCONT] must be deasserted before asserting MCR[HALT] in master mode. This will make sure that the PCSn signals are deasserted. Asserting MCR[HALT] during continuous transfer will cause the PCSn signals to remain asserted and hence Slave Device cannot transition from Running to Stopped state.

**NOTE**

User must fill the TX FIFO with the number of entries that will be concatenated together under one PCS assertion for both master and slave before the TX FIFO becomes empty.

When operating in Slave mode, ensure that when the last entry in the TX FIFO is completely transmitted, that is, the corresponding TCF flag is asserted and TXFIFO is empty, the slave is deselected for any further serial communication; otherwise, an underflow error occurs.

### 40.5.6.6 Fast Continuous Selection Format

The Fast Continuous Selection Format functions similar to Continuous Selection Format except that the inter command delays, $t_{ASC}$ and $t_{CSC}$, can be masked out and are not inserted by the hardware.

**NOTE**

The Fast Continuous Selection Format is available in the SPI configuration only and when Continuous Serial Communication Clock mode is disabled. Masking of delays is not allowed in DSI and CSI configurations or if the transfer is non-continuous.

The Fast Continuous Selection Format is enabled by writing ‘1’ into FCPCS bit of the MCR register. When this bit is asserted, MASC and MCSC bits of the PUSHR register perform the function of mask bits for the transmit frame. These bits individually mask the $t_{ASC}$ and $t_{CSC}$ delays as programmed by the user software. A normal Continuous Selection Format has these two delays for each frame that is transmitted with the CONT

bit asserted. In order to avoid these delays and to speed up the transfer process, the software can simply mask these delays while programming the command in the PUSHR register.

While masking the delays, the software must follow the following masking rules, else correct operation is not guaranteed.

- MASC bit masks the “After SCK” delay for the current frame.
- MCSC bit masks the “PCS to SCK” delay for the next frame.
- “After SCK” ($t_{ASC}$) delay must not be masked when the current frame is the last frame in the continuous selection format.
- The “PCS to SCK” delay for the first frame in the continuous selection format cannot be masked.
- Masking of only $t_{ASC}$ is not allowed. If $t_{ASC}$ is masked then $t_{CSC}$ must be masked too.
- Masking of both $t_{ASC}$ and $t_{CSC}$ delays is allowed. In this case, the delay between two frames is equal to half the baud rate set by the user software.
- Masking of only $t_{CSC}$ is allowed. In this case, the delay between two frames is equal to the $t_{ASC}$ time and thus the user software must ensure that the $t_{ASC}$ time is greater than the baud rate.
- The user software must not mask these delays if the continuous selection format is not used and MCR[FCPCS] is asserted.
- Rules applicable to the Continuous Selection Format are applicable here too.

The following figure shows the timing for a Fast Continuous Selection Format transfer. Here seven frames are transferred with both $t_{ASC}$ and $t_{CSC}$ delays masked except for the last frame that terminated the transfer. The last frame has $t_{ASC}$ delay at its end.

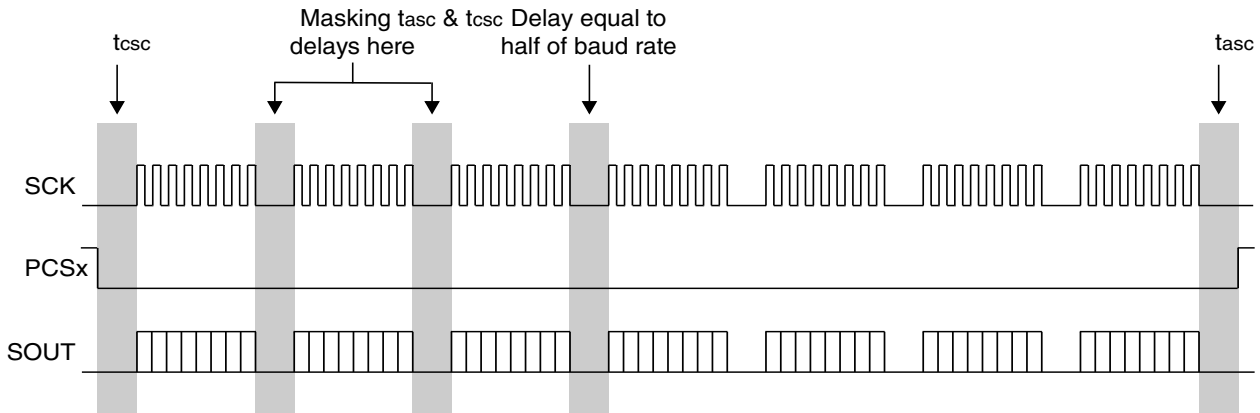



**Figure 40-21. Example of Fast Continuous Selection Format**

In case any chip select is to be changed, then the fast continuous selection format should be terminated and then the chips selects should change and appropriate delays must be introduced.

## 40.5.7 Continuous Serial Communications Clock

The module provides the option of generating a Continuous SCK signal for slave peripherals that require a continuous clock.

Continuous SCK is enabled by setting the CONT_SCKE bit in the MCR. Enabling this bit generates the Continuous SCK only if MCR[HALT] bit is low. Continuous SCK is valid in all configurations.

Continuous SCK is only supported for CPHA=1. Clearing CPHA is ignored if the CONT_SCKE bit is set. Continuous SCK is supported for Modified Transfer Format.

Clock and transfer attributes for the Continuous SCK mode are set according to the following rules:

- When the module is in SPI configuration, CTAR0 is used initially. At the start of each SPI frame transfer, the CTAR specified by the CTAS for the frame is used.
- When the module is in DSI configuration, the CTAR specified by the DSICTAS field is used at all times.
- When the module is in CSI configuration, the CTAR selected by the DSICTAS field is used initially.

- At the start of a SPI frame transfer, the CTAR specified by the CTAS value for the frame is used.
- At the start of a DSI frame transfer, the CTAR specified by the DSICTAS field is used.

- In all configurations, the currently selected CTAR remains in use until the start of a frame with a different CTAR specified, or the Continuous SCK mode is terminated.

It is recommended to keep the baud rate the same while using the Continuous SCK. Switching clock polarity between frames while using Continuous SCK can cause errors in the transfer. Continuous SCK operation is not guaranteed if the module is put into the External Stop mode or Module Disable mode.

Enabling Continuous SCK disables the PCS to SCK delay and the Delay after Transfer ($t_{DT}$) is fixed to one SCK cycle. When TSB configuration is enabled the $t_{DT}$ is programmable from 1 to 64 SCK cycles. The following figure is the timing diagram for Continuous SCK format with Continuous Selection disabled.

**NOTE**

In Continuous SCK mode, for the SPI transfer CTAR0 should always be used, and the TX FIFO must be cleared using the MCR[CLR_TXF] field before initiating transfer.

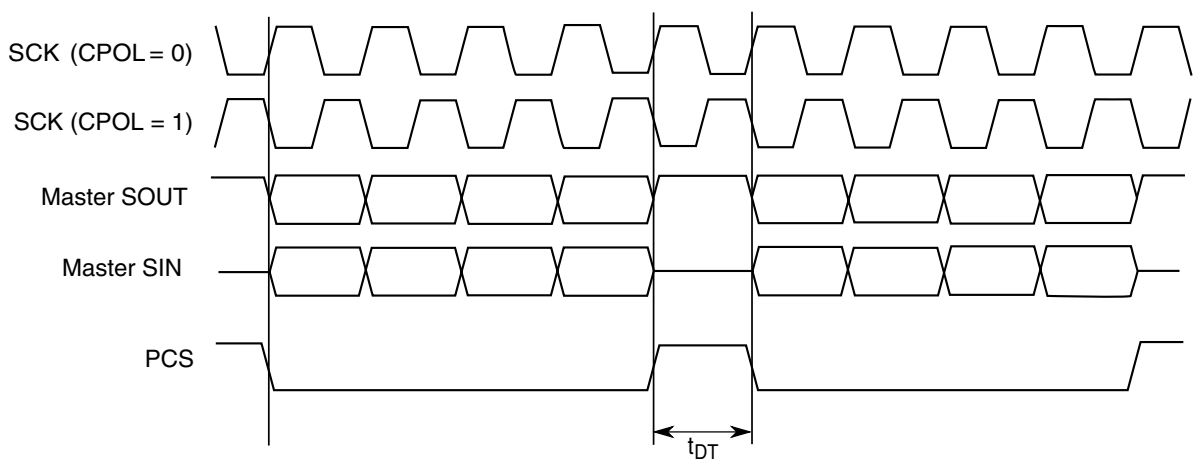



**Figure 40-22. Continuous SCK Timing Diagram (CONT=0)**

If the CONT bit in the TX FIFO entry is set or the DCONT in DSICR0 is set, PCS remains asserted between the transfers. Under certain conditions, SCK can continue with PCS asserted, but with no data being shifted out of SOUT, that is, SOUT pulled high. This can cause the slave to receive incorrect data. Those conditions include:

- Continuous SCK with CONT bit set, but no data in the TX FIFO.

- Continuous SCK with CONT bit set and entering Stopped state (refer to Start and Stop of module transfers).
- Continuous SCK with CONT bit set and entering Stop mode or Module Disable mode.

The following figure shows timing diagram for Continuous SCK format with Continuous Selection enabled.

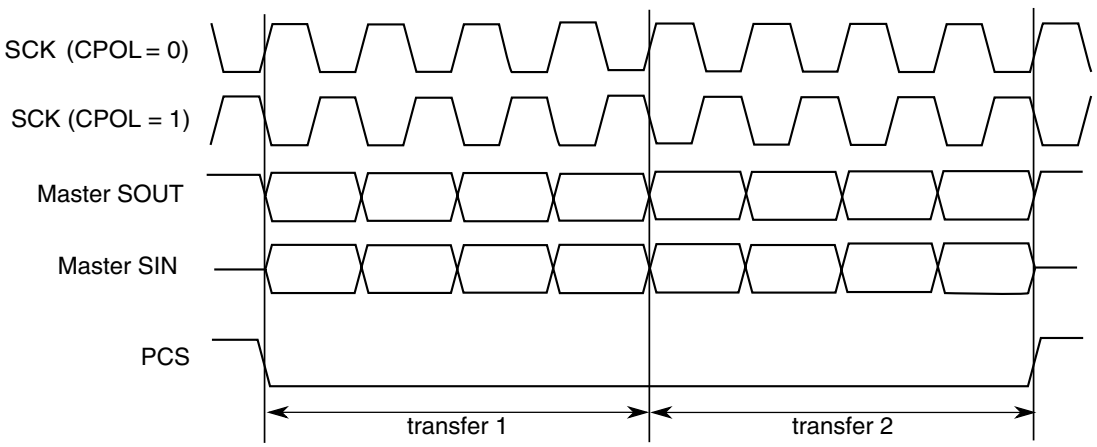



**Figure 40-23. Continuous SCK timing diagram (CONT=1)**

## 40.5.8 Slave Mode Operation Constraints

Slave mode logic shift register is buffered. This allows data streaming operation, when the module is permanently selected and data is shifted in with a constant rate.

The transmit data is transferred at second SCK clock edge of the each frame to the shift register if the $\overline{SS}$ signal is asserted and any time when transmit data is ready and $\overline{SS}$ signal is negated.

Received data is transferred to the receive buffer at last SCK edge of each frame, defined by frame size programmed to the CTAR0/1 register. Then the data from the buffer is transferred to the RXFIFO or DDR register.

If the $\overline{SS}$ negates before that last SCK edge, the data from shift register is lost.

## 40.5.9 Timed Serial Bus (TSB)

The DSPI can be programmed in Timed Serial Bus configuration by setting the TSBC bit in the DSPI_DSICR0 register. See DSPI DSI Configuration Register 0 (DSPI_DSICR0) for details.

TSB configuration provides the Micro Second Channel (MSC) downstream channel support.

The MSC upstream channel is not supported by the DSPI, but can be supported by any available Serial Communication Controller (SCI or UART) in the SoC.

The DSI Continuous Frame Selection Format is not supported by the TSB Mode (DSICR0[DCONT] bit cannot not be asserted).

To work in TSB mode the DSPI must be in master mode and in DSI (DCONF = 0b01) or CSI (DCONF = 0b10) configuration. Both Continuous and Non Continuous Serial Communication Clock (controlled by the DSPI_MCR[CONT_SCKE] bit) are supported in the TSB mode.

The following figure shows the signals used in the TSB interface.

In the TSB configuration the DSPI is able to send from 4 to 66 bits MSC data frames (4 to 64 serialized data bits and up to 2 Data Selection zero bits). The serialized data bits source can be either:

- the DSPI DSI Alternate Serialization Data Register (DSPI_ASDR0/1), written by the host software,
- Parallel Input pin states latched into the DSPI DSI Serialization Data Register (DSPI_SDR0/1).

DSPI_DSICR0 TXSS bit or DSPI_SSR0–1 register bits define the source of the data.

The Least Significant Bits of the DSPI_ASDR0 or DSPI_SDR0 registers are selected to be serialized if the data frame is set to less than 32 bits.

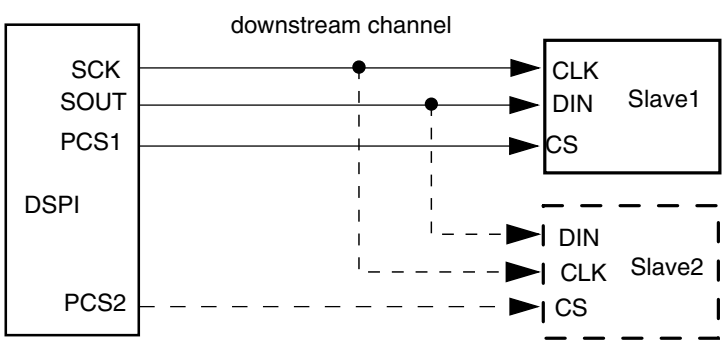



**Figure 40-24. DSPI usage in the TSB Configuration**

The PCS signals are driven together with SOUT. The $t_{CSC}$ and $t_{ASC}$ delays are not available. Delay after Transfer (DT) is set in SCK clock periods as a binary number formed by concatenation of the DSPI_CTARn PDT and DT fields plus one, allowing to set DT from 1 to 64 serial clock periods. DT field provides least significant bits and PDT field provides most significant bits of the Delay after Transfer.

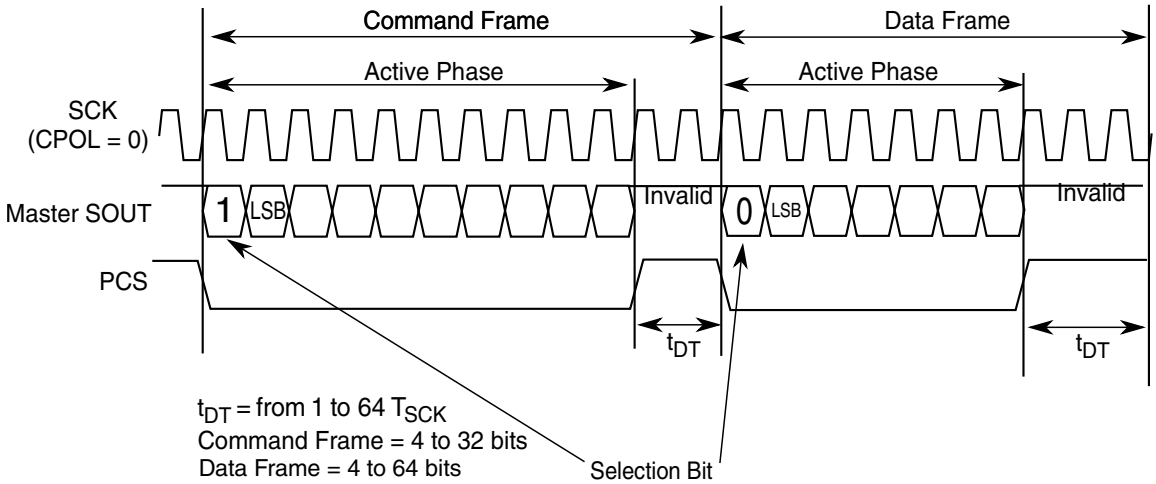



**Figure 40-25. TSB Downstream Frames**

Above figure shows the two types of MSC downstream frames - command frame, and data frame.

The first transmitted bit, called the selection bit, determines the frame type:

- The selection bit "0" indicates a data frame
- The selection bit "1" indicates a command frame

Data frame may contain up to 2 selection bits to support two external slave devices, (so called dual receiver configuration) or no selection bits at all.

The command frame can be written by software, through SPI TX FIFO, using one or two FIFO entries with help of the CONT bit or MCR[XSPI]. The data frame consists of up to 64 bits from the SDR0–1 or ASDR0–1 registers and up to two selection bits (0). Number of data bits in the data frame is defined by the DSCICR1[TSBCNT] field.

The selection bit of the MSC command frames (1) can be implemented by software.

The selection bits in the data frames are enabled by DSPI_DSICR1 DSE0 and DSE1 bits. Each DSEn bit set increases the data frame size by one bit.

To comply with MSC specification, set DSPI_CTARn[LSBFE] to transmit least significant bit first.

Regardless the LSBFE bit setting, the Data Frame Selection Bits, if enabled, are always transmitted first, before corresponded data sub frames.

### 40.5.9.1 MSC Dual Receiver Support with PCS Switch Over

When in TSB mode it is possible to switch the set of PCS signals that are driven during the first part of the frame to a different set of PCS signals during the second part of the frame. The bit, at which this switch over occurs, is defined by FMSZ field of the DSPI_CTARn register, selected by DSICTAS field of the DSICR0 register. If DSICR1[DSI64E] is enabled, then the bit at which the PCS switch occurs, is defined by the concatenation of {DSICR0[FMSZ4], CTAR[FMSZ]} fields.

Number of the bits, not including the Data Selection Bit, in the first part of the frame is equal to value of the FMSZ field plus one. (This value changes to {FMSZ4,FMSZ} plus one, when DSICR1[DSI64E] is enabled). During this part of the frame the PCS signal levels are controlled by DSPI_DSICR0 DPCSn bits, after that by DSPI_DSICR1 DPCS1_n bits.

The PCS switch over occurs at driving edge of the SCK clock output.

The second Data Selection Bit is inserted after the PCS switch over if enabled.

Data Frame with PCS switch over is shown in the following figure.

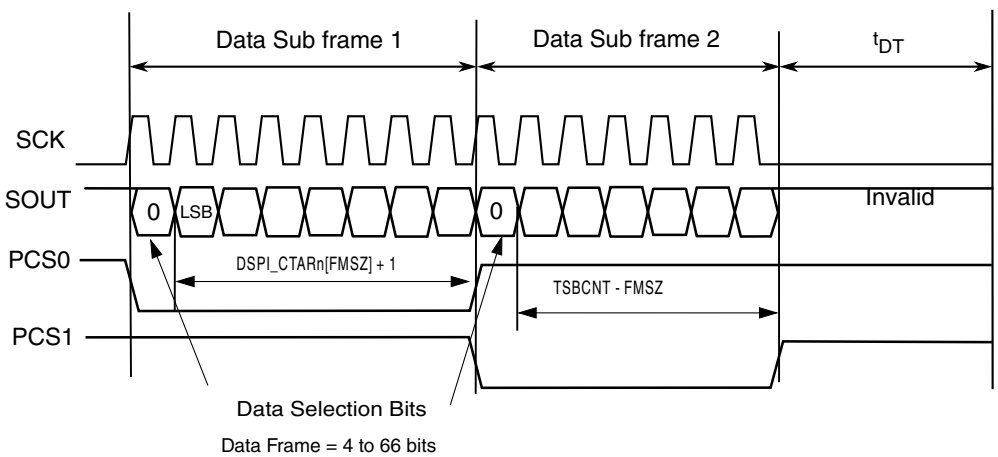



**Figure 40-26. TSB Data Frame Format for MSC Dual Receiver Operation**

## 40.5.10 Interleaved TSB (ITSB) Mode

This mode is similar to the TSB mode in some aspects except for the differences mentioned in the table below. The major difference is that the ITSB mode is highly generic in its approach to be compliant with the Downstream operation of the

Microsecond Channel Protocol. It includes a triggered time-slot based system with user configurable options for the frames to be transmitted in each time slot. (The time period between any two triggers is known as a time-slot).

The DSPI can be programmed to the ITSB Mode by asserting DSICR0[TSBC], DSICR0[ITSB]. To work in ITSB Mode, the DSPI must be in Master Mode and MCR[DCONF] must be set to CSI mode. (DCONF = 0b10). Both continuous and non-continuous clock serial communication clock (MCR[CONT_SCKE] are supported in this mode.

There are three types of frames which can be transmitted in a time slot i.e. Data Frame, Command Frame and Passive (Idle) frame. Similar to the TSB Mode, SPI frames are used for MSC Command frames, while DSI frames are used for MSC Data frames. A Passive Frame is an Idle frame in which no chip selects are asserted and no data is transmitted. The frame transmission is broadly governed by the following rules:

- Transmission of frames is done on trigger pulses that can be generated internal to the DSPI. The internal trigger is periodic with a period of 4. Thus there are 4 time-slots which appear periodically in the ITSB mode. The frame rules to be followed during each of the 4 time-slots can be programmed via TS_CONF[TSn] fields. The length of each time slot can be programmed via TSL[TSx_LEN] fields. The minimum length of a time slot is governed by the following formulae :
  - Minimum Length for Time slot X = Frame size of the frame to be transmitted in Time slot X + no. of selection bits, if any + 2 (Inherent time(Delay) changed by baud rate(Fp/n)).
  - The above constraint is only true for Data and Command frames. Passive frames do not have a minimum trigger length constraint. The user software shall configure the trigger lengths and trigger source.
- If the previously transmitted frame was a Command (SPI) frame or the TX_FIFO (or CMD FIFO) is empty, the non-passive frame transmitted on a subsequent trigger will not be a command frame.
- If the previously transmitted frame was not a Command (SPI) frame and the TX_FIFO (and CMD FIFO) have sufficient data to allow a SPI frame transmission, the frame transmitted on subsequent trigger can be a Command (SPI) frame.
- A Dual Receiver Frame (DRF) is a single Data (DSI) Frame which is split and transmitted to two different receivers (via occurrence of PCS switchover at the split) - the former part being called DRF1 and the latter part as DRF2 (Dual Receiver Frames 1 & 2). Using TS_CONF[TSn] fields, a time-slot can be programmed to transmit a sole DRF1 or a DRF2 frame. A sole DRF1 frame followed by any other frame is a valid scenario. A DRF2 frame programmed for a time slot is only valid if

the immediate previous time-slot was DRF1 - an invalid DRF2 frame, if programmed results in the transmission of a Passive frame (PF).

This mode does not change the way DSI and SPI frame transmissions are done except that ITSB mode interleaves the source (SPI or DSI) with a configurable set of rules based on triggered time-slot system. All features of SPI and DSI mode can be used except for the Continuous Frame Selection format (PUSHR[CONT] and DSICR0[DCONT] bits should not be asserted).

MCR[MTFE] should be set to 0 during ITSB Mode of operation, since MicroSecond Channel upstream operation is not supported.

The following table lists the differences between the TSB and ITSB modes:

**Table 40-15. Differences between TSB and ITSB modes**

| TSB Mode | ITSB Mode |
|---|---|
| Priority available to frames from SPI. Back to back SPI frames are sent out. | No priority. Messages are interleaved such that there are no back-to-back SPI frames |
| Selection Bit inserted for DSI frames only. Selection bit for SPI frames is inserted by software. | Selection bit for both DSI and SPI frames inserted automatically. Encoding of selection bit same as TSB mode (0 = DSI; 1 = SPI). |
| Occasionally triggered transmission (that is trigger is received via hardware trigger input). | Triggered time-slot based transmission (that is frames are transmitted continuously on every trigger). |
| Gap between frames is configurable. Time between frames is $t_{DT}$ which is configurable from 1 to 64 baud rate clock cycles. | Gap between frames is configurable and can be introduced in any of the two ways - By programming a Passive Frame or by increasing the length of a time-slot. Time $t_{DT}$ is not used in this mode as it can be made part of the trigger period (as passive time). |

In both TSB & ITSB mode, separate frame completion interrupts will be available to indicate frame transmission completion from SPI & DSI frames. MSC dual receiver support with PCS switchover is also supported in ITSB mode. See MSC Dual Receiver Support with PCS Switch Over for more details.

The ITSB mode is suitable to implement the Downstream Channel for the MSC Bus because of the above features.

### 40.5.10.1 Configuring DSPI for ITSB mode

The following steps give the recommended way to initialize the DSPI for the ITSB mode of operation:

- The DSPI should be put in HALT mode (by asserting the MCR[HALT] bit) before programming the registers.

- Set the DSICR0[TSBC] and DSICR0[ITSB] bits. The DSICR0[TRRE] and DSICR0[CID] bits should be cleared. DSICR0[DCONT] and PUSHR[CONT] should not be used in ITSB mode. The bit MCR[MTFE] should be cleared as Upstream operation is not supported in ITSB Mode. If DSICR0[ITSB] bit is set without setting DSICR0[TSBC], then the DSPI operates in the Normal Mode depending on MCR[DCONF] bit.
- Configure the Time Slot Length's via TSL register and the frame rules for each time slot via TS_CONF register.
- Based on trigger source requirements, set the values in the DSICR0[TRG] bit.
- Enable frame completion interrupts for either any frame completion (TCF) or separate frame complete (DSITCF and SPITCF) as required by application.
- When remainder configuration of DSPI is complete, clear the MCR[HALT] bit to allow DSPI to start operations.

Once configured and DSPI is brought out of HALT, the ITSB mode shall start sequencing the frames from either DSI or SPI as per the rules mentioned in Interleaved TSB (ITSB) Mode.

### 40.5.10.2 Variable Length Time Slots

In the Interleaved TSB Mode, the TSL[TSx_LEN] registers are used to configure the time slot length's as required by the application. There are up to 4 time slots named TS0, TS1, TS2 and TS3 respectively, which repeat periodically. To achieve a periodicity less than 4, any time slot length can be programmed as 0 (which eliminates this time slot). The following figures illustrate this concept.

In the figure below, the fields TSL[TSx_LEN] are programmed to be non-zero for x = 0,1,2,3. Hence, the time slots repeat after every 4 trigger pulses as shown.

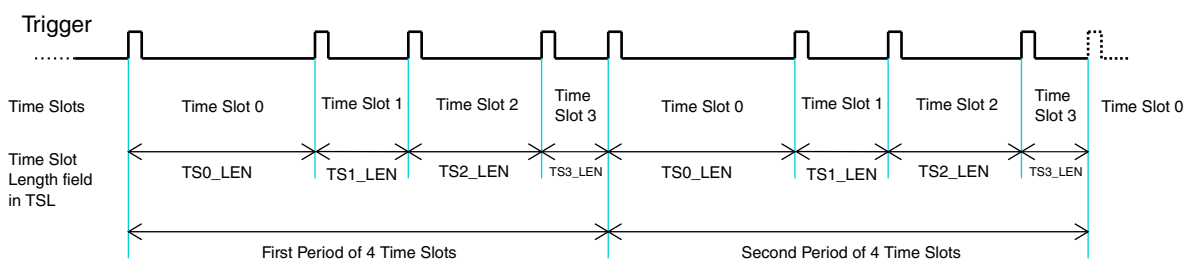



**Figure 40-27. Periodic Time Slots with period = 4**

In the figure below, the fields TSL[TSx_LEN] are programmed to be non-zero for x = 0,1,2, while TSL[TS3_LEN] = 0. Hence, the time slots repeat after every 3 trigger pulses as shown.

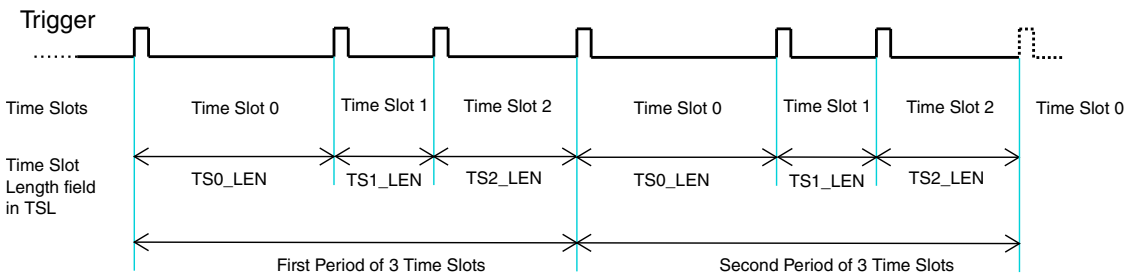



**Figure 40-28. Periodic Time Slots with period = 3**

The following figure shows time slots with periodicity of 1. Here the fields TSL[TSx_LEN] are programmed to zero for x = 1,2 and 3 while TSL[TS0_LEN] is non-zero. Hence, the time slots repeat after every trigger as shown.

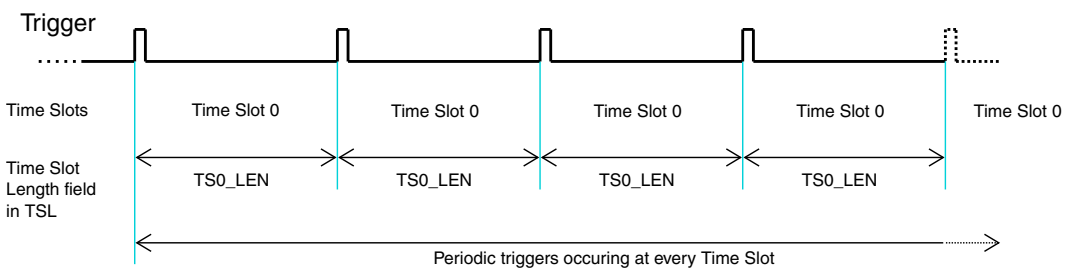



**Figure 40-29. Periodic Time Slots with period = 1**

Similarly, any TSL[TSx_LEN] can be made 0 to vary periodicity of time slots, but it is an illegal scenario to have all TSL[TSx_LEN] fields as 0.

## NOTE

If external triggering mechanism is enabled (DSICR[TRG] is set), then it must be ensured that the minimum distance between two trigger pulses is at least 5P, where P is the protocol clock period.

When internal triggering mechanism is enabled (DSICR[TRG] is zero), the minimum value which can be programmed in the field TSL[TSx_LEN] is given by L = F + 1 + m; where F is the frame size of the frame programmed to be transmitted for the time slot TSx. The variable 'm' equals 1 when maximum baud rate is programmed (i.e. SCK = protocol clock frequency/ 2) and 0 otherwise.

### 40.5.10.3 Using ITSB mode for MSC Downstream Channel

MSC Downstream Channel uses three types of frames: command frames, data frames and passive frames. The Downstream operation of these frames require:

- Data frames can be sent back to back.
- Back to back command frames are not a legal scenario.
- Passive frames are used when no activity is desired for any time slot. In a passive frame, chip selects are de-asserted, and no SCK or data is transmitted. For this reason Command Frames cannot be separated by Passive frames as it would still be an illegal scenario.
- In dual MSC ASIC, some bits of the data frame goes to first ASIC and remainder bits to second ASIC. The first part is referred to as DRF1 (Dual Receiver Frame 1) and the second part as DRF2 (Dual Receiver Frame 2).
- A DRF2 programmed for a time slot is only valid if the immediately preceding time slot was DRF1. If DRF2 is valid, it would be transmitted seamlessly after DRF1 without waiting for the trigger pulse for the time slot of DRF2. For that reason, the TSL[TSx_LEN] for DRF1 must be programmed to the minimum possible value. If DRF2 is not valid, a Passive Frame is transmitted instead.

The ITSB mode can help implement the MSC Downstream Channel by allowing the command frames to be sent via the SPI interface and data frames to be sent via the DSI interface. Once the DSPI has been configured to work in ITSB mode (See Configuring DSPI for ITSB mode), the user software can program the DSI and SPI sides to send data and command frames respectively. The programming of frames for transmission via DSI or SPI is same as explained in Combined Serial Interface (CSI) Configuration.

The following figure shows a sample MSC Downstream transmission for the ITSB mode with the following configuration:

- All TSL[LEN] are programmed to be non-zero. Note that the length between triggers are not drawn to scale in the figure. The lengths of all 4 time slots shown are unequal.
- Data and Command selection bits are enabled, i.e. DSICR1[DSE0] and DSICR1[CSE] are set.
- TS_CONF[TS0] = CMD_SRDF
- TS_CONF[TS1] = CMD_SRDF
- TS_CONF[TS2] = CMD_PF

- TS_CONF[TS3] = CMD_SRDF
- PCS0 is used for Data Frames while PCS1 is used for Command Frames. Both chip select's are active low.

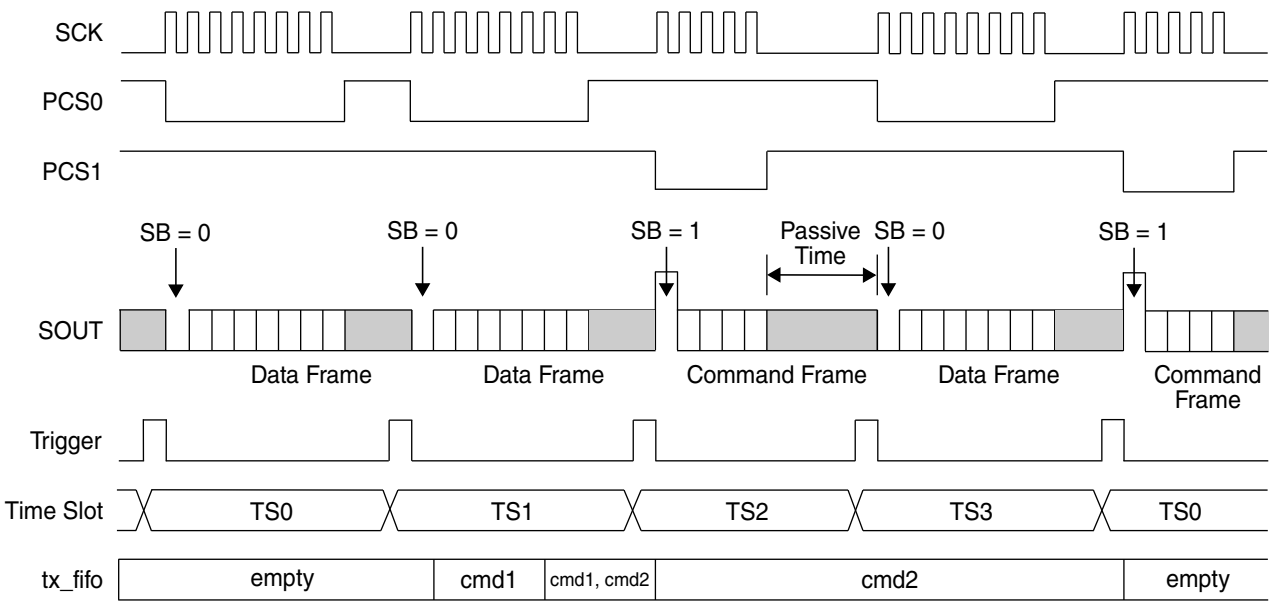



**Figure 40-30. Sample MSC Downstream transmission using ITSB mode**

Brief description of operation includes:

- At the first trigger, TS0 is initiated. The frame rule for TS0 is configured as CMD_SRDF (Command if valid, else Single Receiver Data Frame). Since there is no data present in the TXFIFO, a valid command frame is not available to be transmitted. Hence a single receiver data frame is transmitted instead.
- The second trigger initiates TS1. Again a single receiver data frame is transmitted since the frame rule for TS1 is programmed to CMD_SRDF and a command frame is not valid as there is no data present in the TXFIFO when TS1 is initiated.
- During TS1, two frames are pushed into the TX_FIFO.
- At third trigger, TS2 is initiated , the frame rule for which is programmed to CMD_PF (Command if valid. else Passive Frame). A Command frame is transmitted as the TX_FIFO is not empty.
    - Since the frame size of the command frame is less than the trigger length for TS2, no transmission happens till next trigger. This is marked as the “Passive Time”. Such passive time exists in all frames after frame transmission completes and until next trigger pulse is detected.

- For TS3, the frame rule programmed is CMD_SRDF. Since the previous transmission was a command frame, a single receiver data frame is transmitted during TS3.
- The time slot TS0 repeats, and this time a command frame is transmitted during TS0 since the Command Frame is eligible to be transmitted.
- At sixth trigger (not shown), TS1 occurs and the process continues.

The figure below shows another sample MSC Downstream transmission for the ITSB mode with the following configuration:

- All TSL[LEN] are programmed to be non-zero. Note that the length between triggers are not drawn to scale in the figure. The lengths of all 4 time slots shown are unequal.
- Data and Command selection bits are enabled, i.e. DSICR1[DSE0] and DSICR1[CSE] are set.
- TS_CONF[TS0] = CMD_DRF1
- TS_CONF[TS1] = DRF2
- TS_CONF[TS2] = CMD_DRF1
- TS_CONF[TS3] = CMD_DRF2
- PCS0 is used for DRF1 while PCS1 is used for DRF2 or Command Frames. Both chip select's are active low.

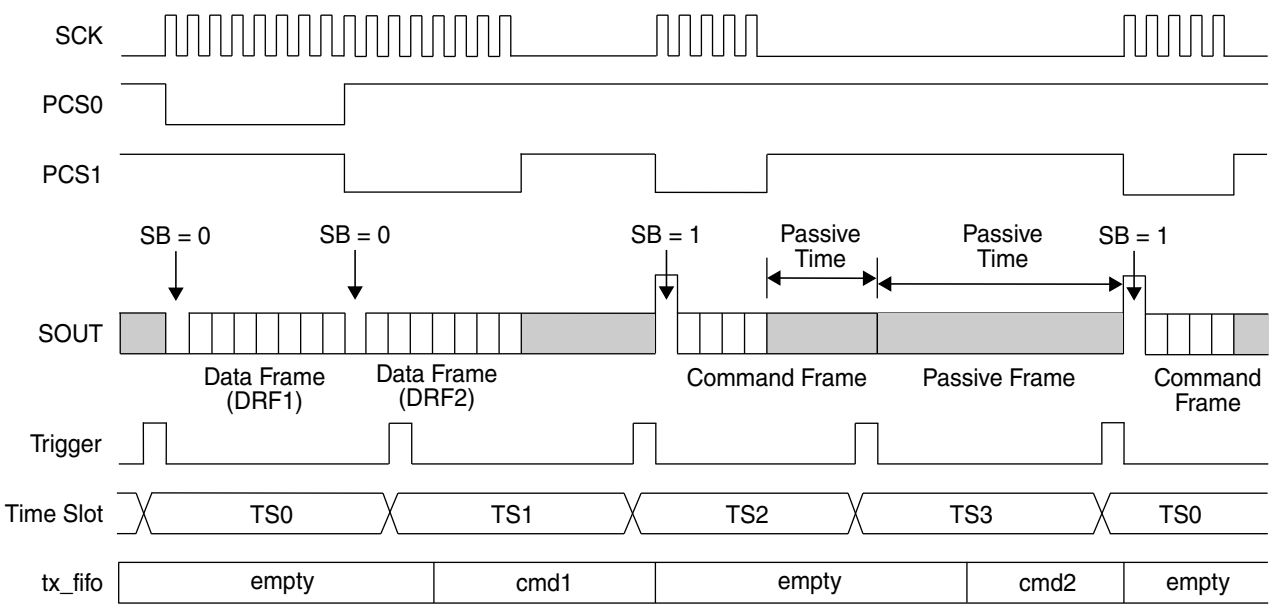



**Figure 40-31. Sample MSC Downstream transmission using ITSB mode**

Brief description of operation includes:

- At the first trigger, TS0 is initiated. The frame rule for TS0 is configured as CMD_DRF1 (Command if valid, else Dual Receiver Frame 1). Since there is no data present in the TXFIFO, a valid command frame is not available to be transmitted. Hence DRF1 is selected for transmission. Now it is checked if the next trigger is DRF2 which is true and hence DRF1 and DRF2 are transmitted back-to-back seamlessly with the PCS switch-over occurring in between. Note that this happens independent of the next trigger. If the next trigger were configured to CMD_DRF2 instead, then at the current trigger, the validity of CMD is checked to confirm the presence of DRF2 for next trigger. If presence of DRF2 is confirmed for next trigger, it is transmitted along with DRF1 seamlessly as shown here, otherwise only DRF1 would be transmitted in the current trigger and the CMD frame would be transmitted in the next trigger.
- During TS1, one frame is pushed into the TX_FIFO.
- Since DRF2 is transmitted immediately after DRF1, the occurrence second trigger does not cause any operation. Hence it must be ensured by the user that Trigger Length of TS0 must be programmed to minimum possible value - during which DRF1 can be successfully transmitted. This ensures that trigger for TS1 occurs sometime after the switch-over and not after DRF2 has been transmitted.
- The third trigger initiates TS2, the frame rule for which is configured as CMD_DRF1. A Command frame is transmitted as the data required to transmit a command frame is available (TX_FIFO and CMD_FIFO is not empty) and the previous transmitted frame was not a command frame.
- The fourth trigger initiates TS3, the frame rule for which is configured as CMD_DRF2. Since a CMD frame isn't available and DRF2 is not valid (since DRF1 wasn't transmitted in the previous trigger), a passive frame is transmitted instead.
- The time slot TS0 repeats, and this time a command frame is transmitted during TS0 since the Command Frame is eligible to be transmitted.
- At sixth trigger (not shown), TS1 occurs and the process continues.

**NOTE**

For a command/data frame to be transmitted on a trigger, they should be available some time before the occurrence of the trigger. A DSI frame must be updated/present "X" time units before the occurrence of the trigger for the time slot in which this data is to be transmitted. This is true for all DSI frames and also implies that when DRF1 and DRF2 are positioned to be

transmitted seamlessly, both DRF1 and DRF2 frames must be available 'X' time units before occurrence of trigger pulse for the time slot of DRF1 itself.

Similarly, a command frame must be available "Y" time units before the occurrence of the trigger for the time slot in which this data is to be transmitted.

Here X = 2R + 1P and Y = 1R + 3P; where R = register clock period, P = protocol clock period.

## 40.5.11 Parity Generation and Check

The module can generate and check parity in the serial frame. The parity bit replaces the last transmitted bit in the frame. The parity is calculated for all transmitted data bits in frame, not including the last data bit that would be transmitted. The parity generation/control is done on frame basis. The registers field setting frame size defines the total number of bits in the frame, including the parity bit. Thus, to transmit/receive the same number of data bits with parity check, increase the frame size by one versus the same data size frame without the parity check.

Parity can be selected as odd or even. Parity Errors in the received frame set Parity Error flags in the Status register. The Parity Error Interrupt Requests are generated if enabled. The module can be programmed to stop frame transmission in case of a frame reception with parity error.

### 40.5.11.1 Parity for SPI Frames

When the module is in the master mode the parity generation is controlled by PE and PP bits of the CMD FIFO entries (PUSHR). Setting the PE bit enables parity generation for transmitted SPI frames and parity check for received frames. PP bit defines polarity of the parity bit.

When continuous PCS selection is used to transmit SPI data, two parity generation scenarios are available:

- Generate/check parity for the whole frame
- Generate/check parity for each sub-frame separately.

To generate/check parity for the whole frame set PE bit only in the last command/TX FIFO entry, forming this frame (with the PUSHR register).

To generate/check parity for each sub-frame set PE bit in each command/TX FIFO entry, forming this frame.

If the parity error occurs for received SPI frame, the SR[SPEF] bit is set. If MCR[PES] bit is set, the module stops SPI frames transmission. To resume SPI operation clear the SR[SPEF] or the MCR[PES] bits.

In slave mode the parity is controlled by the PE and PP bits of the CTAR0 register similar to the master mode parity generation without continuous PCS selection.

### 40.5.11.2 Parity for DSI Frames

When DSPI is in Master Mode, parity generation is controlled by PE and PP bits of the DSPI_DSICR0 register similar to the SPI frames. The parity is calculated and checked for each DSI frame. DSPI_DSICR0[DCONT] bit has no effect on parity generation. In slave mode the parity is controlled by the PE and PP bits of the CTAR1 register.

If the parity error occurs for received DSI frame, the DSPI_SR[DPEF] bit is set. If DSPI_DSICR0[PES] bit is set, the DSPI stops DSI frames transmission. To resume DSI operation clear the DSPI_SR[DPEF] or the DSPI_DSICR0[PES] bits.

## 40.5.12 Interrupts/DMA requests

The module has several conditions that can generate only interrupt requests and two conditions that can generate interrupt or DMA requests. The following table lists these conditions.

**Table 40-16. Interrupt and DMA request conditions**

| Condition | Flag | Interrupt | DMA |
|---|---|---|---|
| End of Queue (EOQ) | EOQF | Yes | - |
| TX FIFO Fill | TFFF | Yes | Yes |
| CMD FIFO Fill | CMDFFF | Yes | Yes |
| TX FIFO Invalid Write | TFIWF | Yes | - |
| Transfer Complete | TCF | Yes | - |
| CMD Transfer Complete | CMDTCF | Yes | - |
| SPI Transfer Complete | SPITCF | Yes | - |
| DSI Transfer Complete | DSITCF | Yes | - |
| TX FIFO Underflow | TFUF | Yes | - |
| RX FIFO Drain | RFDF | Yes | Yes |
| RX FIFO Overflow | RFOF | Yes | - |
| SPI Parity Error | SPEF | Yes | - |

*Table continues on the next page...*

**Table 40-16. Interrupt and DMA request conditions (continued)**

| Condition | Flag | Interrupt | DMA |
|---|---|---|---|
| DSI Parity Error | DPEF | Yes | - |
| DSI Deserialized Data Match | DDIF | Yes | Yes |

Each condition has a flag bit in the module Status Register (SR) and a Request Enable bit in the DMA/Interrupt Request Select and Enable Register (RSER). Certain flags (as shown in above table) generate interrupt requests or DMA requests depending on configuration of RSER register.

The module also provides a global interrupt request line, which is asserted when any of individual interrupt requests lines is asserted.

### 40.5.12.1 End Of Queue interrupt request

The End Of Queue (EOQ) interrupt request indicates that the end of a transmit queue is reached. The module generates the interrupt request when EOQ interrupt requests are enabled (RSER[EOQF_RE]) and the EOQ bit in the executing SPI command is 1.

When Extended SPI mode is enabled (MCR[XSPI]) and the EOQ bit in the executing SPI command is 1, the module generates the EOQ interrupt request when the last bit of the last data frame in the command cycle has been transmitted.

When Extended SPI mode is disabled (MCR[XSPI]), the module generates the interrupt request when the last bit of the SPI frame with EOQ bit set is transmitted.

### 40.5.12.2 Transmit FIFO Fill Interrupt or DMA Request

The Transmit FIFO Fill Request indicates that the TX FIFO is not full. The Transmit FIFO Fill Request is generated when the number of entries in the TX FIFO is less than the maximum number of possible entries, and the TFFF_RE bit in the RSER is set. The TFFF_DIRS bit in the RSER selects whether a DMA request or an interrupt request is generated.

**NOTE**

TFFF flag clears automatically when DMA is used to fill TX FIFO. Configure the DMA to fill only one FIFO location per transfer.

To clear TFFF when not using DMA, follow these steps for every PUSH performed using CPU to fill TX FIFO:

1. Wait until TFFF = 1.
2. Write data to PUSHR using CPU.
3. Clear TFFF by writing a 1 to its location. If TX FIFO is not full, this flag will not clear.

### 40.5.12.3 Command FIFO Fill Interrupt or DMA Request

The Command FIFO Fill Request indicates that the CMD FIFO is not full. The Command FIFO Fill Request is generated when the number of entries in the CMD FIFO is less than the maximum number of possible entries, and the CMDFFF_RE bit in the RSER is set. The CMDFFF_DIRS bit in the RSER selects whether a DMA request or an interrupt request is generated.

This Request is useful when MCR[XSPI] is enabled, since TX FIFO and CMD FIFO can be filled independantly. If MCR[XSPI] is disabled, then 'TX FIFO Fill Interrupt or DMA Request' will suffice to fill both FIFO's since both FIFO's must be filled simultaneously.

**NOTE**

CMDFFF flag clears automatically when DMA is used to fill CMD FIFO. Configure the DMA to fill only one FIFO location per transfer.

To clear CMDFFF when not using DMA, follow these steps for every PUSH performed using CPU to fill CMDFIFO:

1. Wait until CMDFFF = 1
2. Write data to PUSHR using CPU.
3. Clear CMDFFF by writing a 1 to its location. If CMD FIFO is not full, this flag will not clear.

### 40.5.12.4 Transmit FIFO Invalid Write Interrupt Request

The Transmit FIFO Invalid Write Request is valid only when MCR[XSPI] is enabled. This Request indicates that Data exists in the TX FIFO while the CMD FIFO is empty. Since no Command Fields are associated with the Data present in TX FIFO, this data is considered invalid until a Command Entry becomes available. The Transmit FIFO Invalid Write Request is generated for the above condition when TFIWF_RE bit is set in the RSER.

### 40.5.12.5 Transfer Complete Interrupt Request

The Transfer Complete Request indicates the end of the transfer of a serial frame. The Transfer Complete Request is generated at the end of each frame transfer when the TCF_RE bit is set in the RSER.

### 40.5.12.6 Command Transfer Complete Interrupt Request

The Command Transfer Complete Request indicates the end of transfer of the last SPI frame in a Command Cycle. The Transfer Complete Request is generated for the above condition when the CMDTCF_RE bit is set in the RSER.

### 40.5.12.7 SPI Transfer Complete Interrupt Request

The SPI Transfer Complete Request indicates the end of transfer of an SPI frame in any Transmission Mode which uses DSPI in CSI Mode. The SPI Transfer Complete Request is generated when SR[SPITCF] flag asserts and SPITCF_RE bit is set in the RSER.

### 40.5.12.8 DSI Transfer Complete Interrupt Request

The DSI Transfer Complete Request indicates the end of transfer of a DSI frame in any Transmission Mode which uses DSPI in CSI Mode. The DSI Transfer Complete Request is generated when SR[DSITCF] flag asserts and DSITCF_RE bit is set in the RSER.

### 40.5.12.9 Transmit FIFO Underflow Interrupt Request

The Transmit FIFO Underflow Request indicates that an underflow condition in the TX FIFO has occurred. The transmit underflow condition is detected only for the module operating in Slave mode and SPI configuration. The TFUF bit is set when the TX FIFO of the module is empty, and a transfer is initiated from an external SPI master. If the TFUF bit is set while the TFUF_RE bit in the RSER is set, an interrupt request is generated.

### 40.5.12.10 Receive FIFO Drain Interrupt or DMA Request

The Receive FIFO Drain Request indicates that the RX FIFO is not empty. The Receive FIFO Drain Request is generated when the number of entries in the RX FIFO is not zero, and the RFDF_RE bit in the RSER is set. The RFDF_DIRS bit in the RSER selects whether a DMA request or an interrupt request is generated. Configure the DMA to drain only one FIFO location per transfer.

### 40.5.12.11 Receive FIFO Overflow Interrupt Request

The Receive FIFO Overflow Request indicates that an overflow condition in the RX FIFO has occurred. A Receive FIFO Overflow request is generated when RX FIFO and shift register are full and a transfer is initiated. The RFOF_RE bit in the RSER must be set for the interrupt request to be generated.

Depending on the state of the ROOE bit in the MCR, the data from the transfer that generated the overflow is either ignored or shifted in to the shift register. If the ROOE bit is set, the incoming data is shifted in to the shift register. If the ROOE bit is cleared, the incoming data is ignored.

### 40.5.12.12 SPI Frame Parity Error Interrupt Request

The SPI Frame Parity Error Flag indicates that a SPI frame with parity error had been received. The SPEF_RE bit in the RSER must be set for the interrupt request to be generated.

### 40.5.12.13 DSI Frame Parity Error Interrupt Request

The DSI Frame Parity Error Flag indicates that a DSI frame with parity error has been received. The DPEF_RE bit in the DSPI_RSER must be set for the interrupt request to be generated.

### 40.5.12.14 Deserialized Data Match Interrupt or DMA Request

The Deserialized Data Match Flag (DDIF) indicates that a DSI frame matches DSPI_DPIR0–1 data, masked with DSPI_DIMR0–1, had been received. The DDIF_RE bit in the DSPI_RSER must be set for the request to be generated. The DDIF_DIRS bit in the RSER selects whether a DMA request or an interrupt request is generated.

### 40.5.13 Power saving features

The module supports following power-saving strategies:

- External Stop mode
- Module Disable mode – Clock gating of non-memory mapped logic

#### 40.5.13.1 Stop mode (External Stop mode)

This module supports the Stop mode protocol. When a request is made to enter External Stop mode, the module acknowledges the request . If a serial transfer is in progress, then this module waits until it reaches the frame boundary before it is ready to have its clocks shut off . While the clocks are shut off, this module's memory-mapped logic is not accessible. This also puts the module in STOPPED state. The SR[TXRXS] bit is cleared to indicate STOPPED state. The states of the interrupt and DMA request signals cannot be changed while in External Stop mode.

#### 40.5.13.2 Module Disable mode

Module Disable mode is a block-specific mode that the module can enter to save power. Host CPU can initiate the Module Disable mode by setting the MDIS bit in the MCR. The Module Disable mode can also be initiated by hardware.

When the MDIS bit is set, the module negates the Clock Enable signal at the next frame boundary. Once the Clock Enable signal is negated, it is said to have entered Module Disable Mode. This also puts the module in STOPPED state. The SR[TXRXS] bit is cleared to indicate STOPPED state. If implemented, the Clock Enable signal can stop the clock to the non-memory mapped logic. When Clock Enable is negated, the module is in a dormant state, but the memory mapped registers are still accessible. Certain read or write operations have a different effect when the module is in the Module Disable mode. Reading the RX FIFO Pop Register does not change the state of the RX FIFO. Similarly, writing to the PUSHR Register does not change the state of the TX FIFO or CMD FIFO. Clearing either of the FIFOs has no effect in the Module Disable mode. Changes to the DIS_TXF and DIS_RXF fields of the MCR have no effect in the Module Disable mode. In the Module Disable mode, all status bits and register flags in the module return the correct values when read, but writing to them has no effect. Writing to the TCR during Module Disable mode has no effect. Interrupt and DMA request signals cannot be cleared while in the Module Disable mode.

## 40.6 Initialization/application information

This section describes how to initialize the module.

### 40.6.1 How to manage queues

The queues are not part of the module, but it includes features in support of queue management. Queues are primarily supported in SPI configuration.

1. When module executes last command word from a queue, the EOQ bit in the command word is set to indicate it that this is the last entry in the queue.
2. At the end of the transfer, corresponding to the command word with EOQ set is sampled, the EOQ flag (EOQF) in the SR is set.
3. The setting of the EOQF flag disables serial transmission and reception of data, putting the module in the Stopped state. The TXRXS bit is cleared to indicate the Stopped state.
4. The DMA can continue to fill TX FIFO until it is full or step 5 occurs.
5. Disable DMA transfers by disabling the DMA enable request for the DMA channel assigned to TX FIFO (and CMD FIFO) and RX FIFO. This is done by clearing the corresponding DMA enable request bits in the DMA Controller.
6. Ensure all received data in RX FIFO has been transferred to memory receive queue by reading the RXCNT in SR or by checking RFDF in the SR after each read operation of the POPR.
7. Modify DMA descriptor of TX and RX channels for new queues
8. Flush TX FIFO (and CMD FIFO) by writing a 1 to the CLR_TXF bit in the MCR. Flush RX FIFO by writing a '1' to the CLR_RXF bit in the MCR.
9. Clear transfer count either by setting CTCNT bit in the command word of the first entry in the new queue or via CPU writing directly to SPI_TCNT field in the TCR.
10. Enable DMA channel by enabling the DMA enable request for the DMA channel assigned to the module TX FIFO, (and CMD FIFO) and RX FIFO by setting the corresponding DMA set enable request bit.
11. Enable serial transmission and serial reception of data by clearing the EOQF bit.

## 40.6.2 Switching Master and Slave mode

When changing modes in the module, follow the steps below to guarantee proper operation.

1. Halt it by setting MCR[HALT].
2. Clear the transmit and receive FIFOs by writing a 1 to the CLR_TXF and CLR_RXF bits in MCR.
3. Set the appropriate mode in MCR[MSTR] and enable it by clearing MCR[HALT].

## 40.6.3 Initializing Module in Master/Slave Modes

Once the appropriate mode in MCR[MSTR] is configured, the module is enabled by clearing MCR[HALT]. It should be ensured that module Slave is enabled before enabling it's Master. This ensures the Slave is ready to be communicated with, before Master initializes communication.

## 40.6.4 Baud rate settings

The following table shows the baud rate that is generated based on the combination of the baud rate prescaler PBR and the baud rate scaler BR in the CTARs. The values calculated assume a 100 MHz protocol frequency and the double baud rate DBR bit is cleared.

**Table 40-17. Baud rate values (bps)**

| | | Baud rate divider prescaler values | | | |
|---|---|---|---|---|---|
| | | 2 | 3 | 5 | 7 |
| Baud Rate Scaler Values | 2 | 25.0M | 16.7M | 10.0M | 7.14M |
| | 4 | 12.5M | 8.33M | 5.00M | 3.57M |
| | 6 | 8.33M | 5.56M | 3.33M | 2.38M |
| | 8 | 6.25M | 4.17M | 2.50M | 1.79M |
| | 16 | 3.12M | 2.08M | 1.25M | 893k |
| | 32 | 1.56M | 1.04M | 625k | 446k |
| | 64 | 781k | 521k | 312k | 223k |
| | 128 | 391k | 260k | 156k | 112k |
| | 256 | 195k | 130k | 78.1k | 55.8k |
| | 512 | 97.7k | 65.1k | 39.1k | 27.9k |

*Table continues on the next page...*

**Table 40-17. Baud rate values (bps) (continued)**

| | | Baud rate divider prescaler values | | | |
|---|---|---|---|---|---|
| | | 2 | 3 | 5 | 7 |
| | 1024 | 48.8k | 32.6k | 19.5k | 14.0k |
| | 2048 | 24.4k | 16.3k | 9.77k | 6.98k |
| | 4096 | 12.2k | 8.14k | 4.88k | 3.49k |
| | 8192 | 6.10k | 4.07k | 2.44k | 1.74k |
| | 16384 | 3.05k | 2.04k | 1.22k | 872 |
| | 32768 | 1.53k | 1.02k | 610 | 436 |

## 40.6.5 Delay settings

The following table shows the values for the Delay after Transfer ($t_{DT}$) and CS to SCK Delay ($T_{CSC}$) that can be generated based on the prescaler values and the scaler values set in the CTARs. The values calculated assume a 100 MHz protocol frequency.

**NOTE**

The clock frequency mentioned above is given as an example in this chapter. See the clocking chapter for the frequency used to drive this module in the device.

This table does not apply for TSB Continuous mode.

**Table 40-18. Delay values**

| | | Delay prescaler values | | | |
|---|---|---|---|---|---|
| | | 1 | 3 | 5 | 7 |
| Delay scaler values | 2 | 20.0 ns | 60.0 ns | 100.0 ns | 140.0 ns |
| | 4 | 40.0 ns | 120.0 ns | 200.0 ns | 280.0 ns |
| | 8 | 80.0 ns | 240.0 ns | 400.0 ns | 560.0 ns |
| | 16 | 160.0 ns | 480.0 ns | 800.0 ns | 1.1 µs |
| | 32 | 320.0 ns | 960.0 ns | 1.6 µs | 2.2 µs |
| | 64 | 640.0 ns | 1.9 µs | 3.2 µs | 4.5 µs |
| | 128 | 1.3 µs | 3.8 µs | 6.4 µs | 9.0 µs |
| | 256 | 2.6 µs | 7.7 µs | 12.8 µs | 17.9 µs |
| | 512 | 5.1 µs | 15.4 µs | 25.6 µs | 35.8 µs |
| | 1024 | 10.2 µs | 30.7 µs | 51.2 µs | 71.7 µs |
| | 2048 | 20.5 µs | 61.4 µs | 102.4 µs | 143.4 µs |
| | 4096 | 41.0 µs | 122.9 µs | 204.8 µs | 286.7 µs |
| | 8192 | 81.9 µs | 245.8 µs | 409.6 µs | 573.4 µs |
| | 16384 | 163.8 µs | 491.5 µs | 819.2 µs | 1.1 ms |

*Table continues on the next page...*

**Table 40-18. Delay values (continued)**

| | | Delay prescaler values | | | |
|---|---|---|---|---|---|
| | | 1 | 3 | 5 | 7 |
| | 32768 | 327.7 µs | 983.0 µs | 1.6 ms | 2.3 ms |
| | 65536 | 655.4 µs | 2.0 ms | 3.3 ms | 4.6 ms |

## 40.6.6 Calculation of FIFO pointer addresses

Complete visibility of the FIFO contents is available through the FIFO registers, and valid entries can be identified through a memory-mapped pointer and counter for each FIFO. The pointer to the first-in entry in each FIFO is memory mapped. For the TX FIFO the first-in pointer is the Transmit Next Pointer (TXNXTPTR). For the CMD FIFO the first-in pointer is the Command Next Pointer (CMDNXTPTR). For the RX FIFO the first-in pointer is the Pop Next Pointer (POPNXTPTR). The following figure illustrates the concept of first-in and last-in FIFO entries along with the FIFO Counter. The TX FIFO is chosen for the illustration, but the concepts carry over. See Transmit First In First Out (TX FIFO) buffering mechanism, Command First In First Out (CMD FIFO) Buffering Mechanism and Receive First In First Out (RX FIFO) buffering mechanism for details on the FIFO operation.

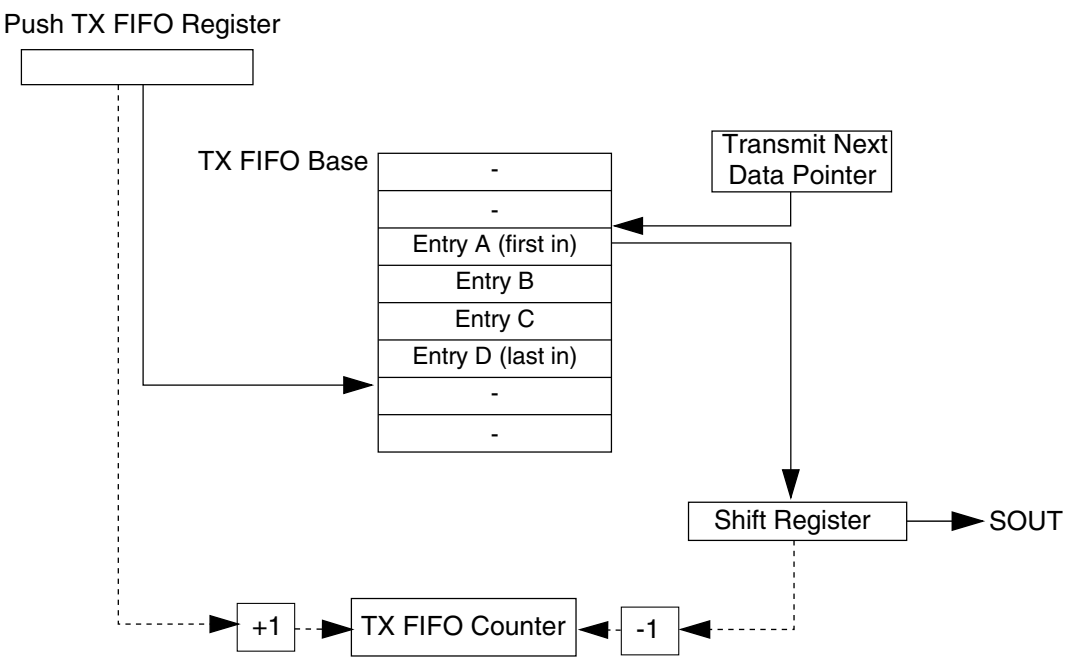



**Figure 40-32. TX FIFO pointers and counter**

#### 40.6.6.1 Address Calculation for the First-in Entry and Last-in Entry in the TX FIFO

The memory address of the first-in entry in the TX FIFO is computed by the following equation:

$$\text{First-in EntryAddress} = \text{TXFIFOBase} + (4 \times \text{TXNXTPTR})$$

The memory address of the last-in entry in the TX FIFO is computed by the following equation:

$$\text{Last-inEntryaddress} = \text{TXFIFOBase} + 4 \times (\text{TXCTR} + \text{TXNXTPTR} - 1) \bmod (\text{TXFIFOdepth})$$

TX FIFO Base - Base address of TX FIFO
TXCTR - TX FIFO Counter
TXNXTPTR - Transmit Next Pointer
TX FIFO Depth - Transmit FIFO depth, implementation specific

#### 40.6.6.2 Address Calculation for the First-in Entry and Last-in Entry in the CMD FIFO

The memory address of the first-in entry in the CMD FIFO is computed by the following equation:

$$\text{First-in EntryAddress} = \text{TXFIFOBase} + (4 \times \text{TXNXTPTR})$$

The memory address of the last-in entry in the CMD FIFO is computed by the following equation:

$$\text{Last-inEntryaddress} = \text{TXFIFOBase} + 4 \times (\text{TXCTR} + \text{TXNXTPTR} - 1) \bmod (\text{TXFIFOdepth})$$

CMD FIFO Base - Base address of CMD FIFO
CMDCTR - CMD FIFO Counter
CMDNXTPTR - Command Next Pointer
CMD FIFO Depth - Command FIFO depth, implementation specific

#### 40.6.6.3 Address Calculation for the First-in Entry and Last-in Entry in the RX FIFO

The memory address of the first-in entry in the RX FIFO is computed by the following equation:

$$\text{First-in EntryAddress} = \text{RX FIFOBase} + \left(4 \times \text{POPNXTPTR}\right)$$

The memory address of the last-in entry in the RX FIFO is computed by the following equation:

$$\text{Last-inEntryaddress} = \text{RX FIFO Base} + 4 \times \left(\text{RXCTR} + \text{POPNXTPTR} - 1\right) \text{mod} \left(\text{RXFIFOdepth}\right)$$

RX FIFO Base - Base address of RX FIFO
RXCTR - RX FIFO counter
POPNXTPTR - Pop Next Pointer
RX FIFO Depth - Receive FIFO depth, implementation specific

# Chapter 41
# Ethernet Switch (ESW)

## 41.1 ESW Configuration

### 41.1.1 ESW Interrupt mapping

The following tables lists the ESW interrupt mapping implemented in this device.

**Table 41-1. ESW Interrupt mapping**

| | |
|---|---|
| ESW Interrupt Group0 | Receive Buffer Done for Queue/Ring 0 |
| | Receive Frame Done for Queue/Ring 0 |
| | Transmit Buffer Done for Queue/Ring 0 |
| | Transmit Frame Done for Queue/Ring 0 |
| | Receive Flush Done for Queue/Ring 0 |
| ESW Interrupt Group1 | Receive Buffer Done for Queue/Ring 1 |
| | Receive Frame Done for Queue/Ring 1 |
| | Transmit Buffer Done for Queue/Ring 1 |
| | Transmit Frame Done for Queue/Ring 1 |
| | Receive Flush Done for Queue/Ring 1 |
| ESW Interrupt Group2 | Receive Buffer Done for Queue/Ring 2 |
| | Receive Frame Done for Queue/Ring 2 |
| | Transmit Buffer Done for Queue/Ring 2 |
| | Transmit Frame Done for Queue/Ring 2 |
| | Receive Flush Done for Queue/Ring 2 |
| ESW Interrupt Group3 | AHB Bus Error detected |
| | Low amount of free memory |
| | Output discarded for Port 0 |
| | Output discarded for Port 1 |
| | Output discarded for Port 2 |
| | Learning record available |