Quad SPI-3 to SPI-4 PHY Layer Bridge

User's Guide

October 2005
ipug24_02.0

Introduction

Lattice's Quad SPI-3 (System Packet Interface Level 3) to SPI-4 (System Packet Interface Level 4) Bridge is an IP core which serves as a bridge between one SPI-4 and one to four SPI-3 links.

Lattice's Quad SPI-3 to SPI-4 Bridge core is a core developed in conjunction with the Lattice ORCA® ORSPI4 FPSC to provide a full solution. For more information on these and other Lattice products refer to the Lattice web site at www.latticesemi.com.

This user's guide explains the functionality of the Quad SPI-3 to SPI-4 Bridge core and how it can be implemented to provide a full SPI-3 to SPI-4 bridging solution. It also explains how to achieve the maximum level of performance.

The Quad SPI-3 to SPI-4 Bridge core comes with the documentation and the files listed below:

- Data sheet
- Lattice gate level netlist
- ModelSim simulation models and test benches available for free evaluation
- Core instantiation template

Features

- Quad full-featured SPI-3 PHY Interfaces as defined by the OIF specifications
- Supports full clock rates for SPI-3 core: 104MHz
- Each SPI-3 PHY can support up to eight ports
- Seamless integration with the SPI-4.2 embedded section in the ORSPI4 FPSC
- 10 Gbps aggregate throughput
- Parameterizable number of SPI-3 PHY interfaces (1 to 4)
- Parameterizable FIFO size selection
- Parameterizable SPI-3 BYTE_MODE or PKT_MODE selection
- Configurable through MicroProcessor Interface (MPI) ORCA 4 System Bus
- Programmable parity type on SPI-3 bus. Default is ODD

General Description

The Quad SPI-3 to SPI-4 Bridge Intellectual Property (IP) Core targets the programmable array section of the ORCA ORSPI4 FPSC and provides a bridging function between one to four SPI-3 links and a SPI-4 link.

The ORSPI4 is an FPSC built on the Series 4 re-configurable embedded System-on-a-Chip (SoC) architecture and intended for high-speed data transmission. The SPI-4.2 interface block provides a 10 Gbps physical to link layer interfaces in conformance to the OIF-SPI4-02.0 specification and bi-directional interfaces with an aggregate bandwidth of 13.6 Gbps.

SPI-4 is an interface for packet and cell transfer between a physical layer (PHY) device and a link layer device, for applications such as OC-192 ATM and Packet over SONET/SDH (POS), as well as 10 Gbps Ethernet applications.

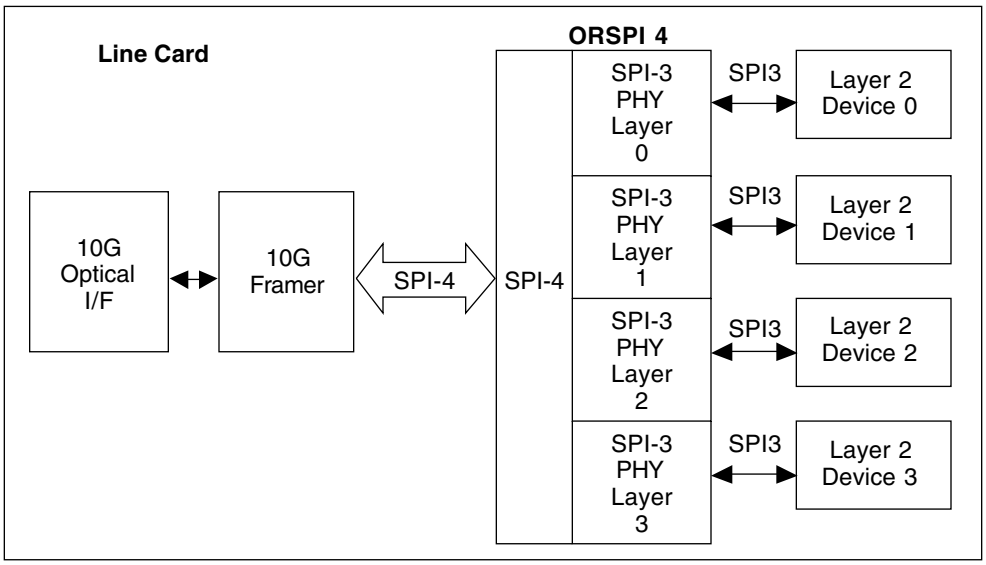
The SPI-3 interface defines the interface between Physical layer and Link layer devices, and can be used to implement several packet-based protocols. The SPI-3 interface supports clock transfer rates of 104MHz and an aggregate bit rate of 2.4 Gbps with a 32-bit wide bus.

The Quad SPI-3 to SPI-4 Bridge IP core is provided with implementation scripts, test benches, and documentation to allow customers to bridge multiple 2.5 Gbps ports (SPI-3) to a 10 Gbps (SPI-4) pipe.

Quad SPI-3 to SPI-4 Bridge Application Overview

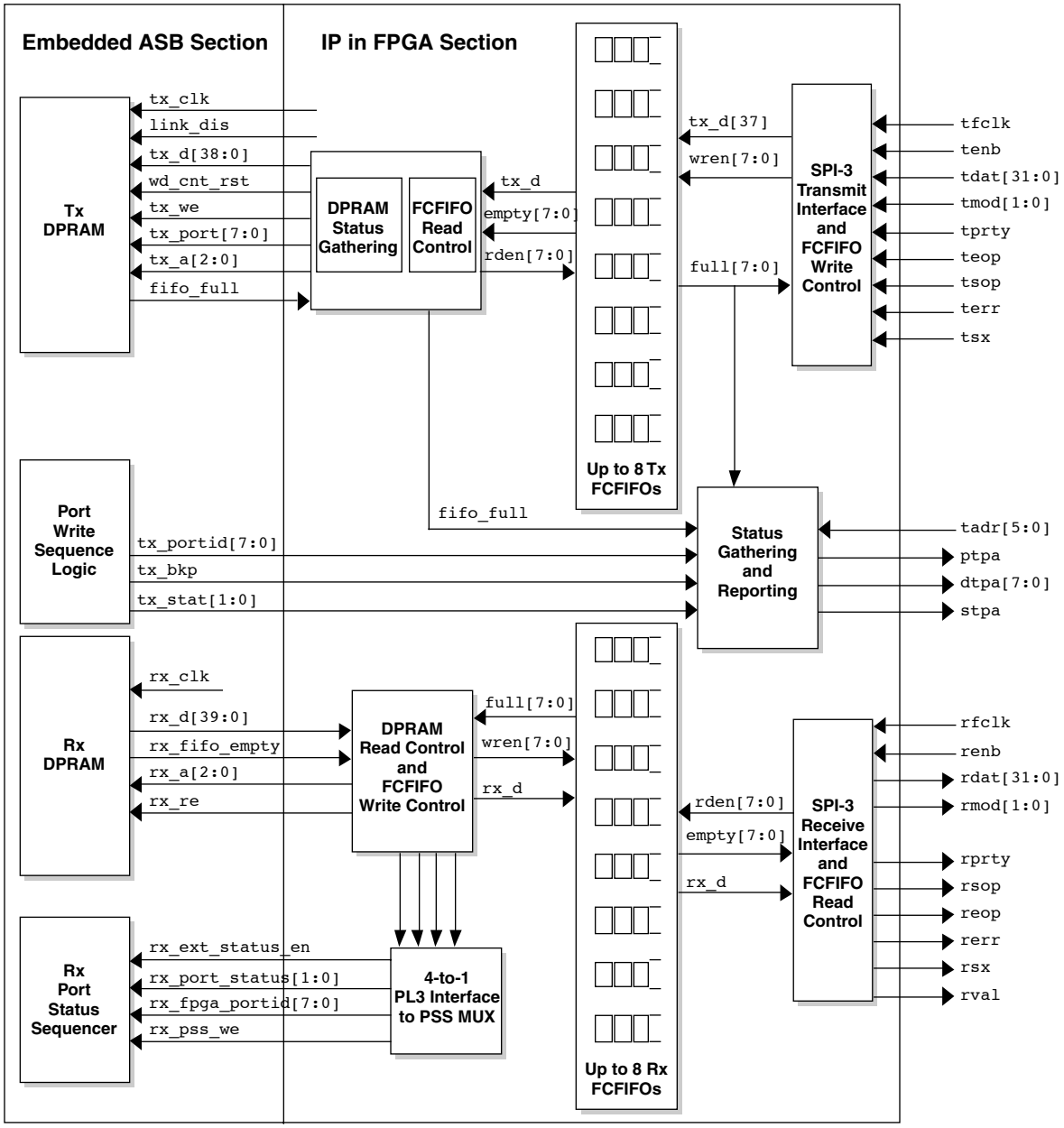
The SPI-3 interface (referred to interchangeably as PL-3 or POS-PHY Level 3) will comply with the OIF implementation agreement (OIF-SPI3-01.0). Version 1 of the bridge design will implement an SPI-3 PHY layer. The primary application for such a bridge design is to allow customers to bridge multiple 2.5 Gbps ports (SPI-3) to a 10 Gbps pipe (SPI4) as shown in Figure 1.

Figure 1. Application of Quad SPI-3 (PHY) to SPI-4 Bridge



Block Diagram

Figure 2. Quad SPI-3 to SPI-4 Bridge Solution Simplified Block Diagram



Note: SPI-3 interface logic can be replicated up to 4x (quad) for 10G applications.

Functional Description

The major blocks in the Quad SPI-3 to SPI-4 Bridge core are shown in Figure 2. Detailed descriptions of these blocks follow.

Transmit Section

The transmit section bridges up to four SPI-3 links to a single SPI-4. As shown in Figure 1, the transmit section consists of the following functional blocks:

- Tx SPI-3 Interface
- Tx Flow Control FIFO
- FCFIFO Read Control
- Status Gathering and Reporting

Tx SPI-3 Interface

This block interfaces one SPI-3 link to as many as eight Tx direction FCFIFOs. There may be from one to four Tx SPI-3 Interface blocks provisioned, depending on the desired number of SPI-3 links. This block receives transmit data (32-bit bus) from the link layer device along with `tsx`, `teop`, `tsop`, `terr`, and `tmod` signals and writes them all into the FCFIFO. Even or odd parity checking (user-programmable) over the Tx data is supported. The port address on `tdat[7:0]` is extracted when the `tsx` input is active. Optionally, it supports `tadr[2:0]` to poll for status of up to eight ports, and optionally supports up to eight DTPA direct polled status outputs (depending on pin availability). This block supports either byte-level or packet-level transfers from the link layer device (DTPA/PTPA). Note: Though the TADR Bus is six bits wide (in `tadr[5:0]`), only bits [2:0] are potentially used.

The quad-SPI-3/SPI-4 Bridge IP Core is designed at the RTL level to support up to eight SPI-3 ports per interface. The interface conforms to the OIF-SPI3-01.0 specification.

Tx Flow Control FIFO (FCFIFO)

Each SPI-3 interface has up to eight Tx Flow Control FIFOs (FCFIFO). The almost-full thresholds are programmable and are used to feed back status information to the Status Gathering and Reporting Block. The FCFIFO is written with the following information: `tx_d[31:0]`, `tsop`, `teop`, `terr`, `tmod[1:0]`, and `tsx`.

When the Transmit Start of Transfer (`tsx`) signal is high and `tenb` is low on the incoming SPI-3 interface, bits `tdat[7:0]` contain the in-band port address. This value is used to select the FCFIFO into which the data will be written.

The FCFIFOs selection is configurable. From one to eight FCFIFOs may be configured in each SPI-3 interface in each direction. Due to the finite amount of storage space in the FPGA, the selection of the number of FCFIFOs determines the available storage space per FCFIFO. Table 1 shows the available choices.

Table 1. FCFIFO Sizes

Number of FCFIFOs per SPI-3 Interface	FIFO Size Max. (Bytes)
1	2,048
2	1,024
4	512
8	256

FCFIFO Read Control

The FCFIFO read control circuit is constantly checking the FCFIFOs for available data. The FCFIFO are accessed in a sequence controlled by the provisioning registers. Whenever a FCFIFO has sufficient data it is read and the data is written into the corresponding DPRAM FIFO. All `tsop`, `teop`, and `terr` signals are also passed to the DPRAM. If a `tsop` is detected without being preceded by a `teop`, the `wd_cnt_rst` signal (word count reset signal

from the FPGA to the ASB) is set active to force the WCL (write control logic block inside the ASB) to write the virtual FIFO and clear the internal word counter.

If the Read Control logic detects that a DPRAM FIFO is partially full, it will cease all writes to that DPRAM FIFO until the `fifo_full` indicator for that DPRAM FIFO clears. The `fifo_full` threshold for the DPRAM FIFOs should be configured to the 3/4 +1 level in the ASB section of the ORSPI4.

When the FCFIFO Read Control block is sending data to the DPRAM it also provides an 8-bit port address in parallel. Each SPI-3 interface has an allowed range of 0-7 for the incoming port numbers. Since only 1 to 3 bits of the address are needed to encode the 0-7 port numbers, the next two bits of the 8-bit port address are translated by the circuit to identify the SPI-3 interface on which the packet arrived. In the 8 ports per interface case, bits 0-2 are the incoming port number and bits 3-4 are used to designate the interface number. In the 4 port per interface case, bits 0-1 are the incoming port number and bits 2-3 are used to designate the interface number. This is a hard-coded translation and is not programmable by the user.

Status Gathering and Reporting
The quad SPI-3/SPI-4 bridge stores status information for all active ports. This information is reported as a 1-bit full or not full condition for each port using STPA/DTPA/PTPA on the SPI-3 interface. Each per-port bit is a summation of all of the following:

- Translation of 2 bits per port SPI-4 status (from Port Write Sequence Logic of the ASB) to 1 bit per port
- `fifo_full` bits from the configured DPRAM virtual FIFOs
- `fifo_full` status from the Tx FCFIFO

Receive Section

The receive section can be divided into the following functional blocks:

- DPRAM Read Control and FCFIFO Write Control
- Rx Flow Control FIFO (FCFIFO)
- Rx SPI-3 Interface
- 4-to-1 SPI-3 to PSS MUX

DPRAM Read Control and FCFIFO Write Control
This block uses a round-robin method to check the DPRAM FIFOs for data. The DPRAM FIFOs are accessed in a sequence controlled by the provisioning registers. Whenever a DPRAM FIFO contains data, it is read until 16 bytes of data are read, an `eop` is read, or until the FIFO is empty. The data read from the DPRAM FIFO is written into the corresponding Rx FCFIFO. The status of the DPRAM FIFOs (empty or not empty) is latched and passed to the 4-to-1 MUX block. Whenever an `eop` is read from a DPRAM FIFO and stored in a FCFIFO, this block passes an `eop` signal to the RX SPI-3 Interface circuit so that it knows how many EOP bits are currently stored in each FCFIFO. The DPRAM Read Control circuit will stop reading a DPRAM before 16 EOPs are stored in the corresponding FCFIFO.

The DPRAM Read Control Block overwrites, with zeros, the two bits encoded with the interface number.

Rx Flow Control FIFO
Each SPI-3 interface has up to eight Rx Flow Control FIFOs (FCFIFO). The almost-full and almost-empty thresholds are configurable by the user. The almost-full threshold determines how close to full the FCFIFO Write Control Block will try to fill each FCFIFO. The almost-empty threshold determines how large the SPI-3 burst size will be, since once the almost-empty threshold is exceeded the SPI-3 interface will be ready to send a burst of data from that FCFIFO.

Rx PL3 Interface
The PL3 Interface circuit monitors the Rx FCFIFOs to determine when a data burst may be sent on the SPI-3 link. Each FCFIFO is examined in a round-robin fashion to determine if it has either a full burst of data (almost-empty

threshold exceeded) or it contains at least one EOP. When either condition is met, data is read from that FIFO and sent on the SPI-3 link until an entire burst has been sent or the EOP is reached. Then the next FIFO is examined in turn.

Parity is calculated and set on the RPRTY output during each word of the burst. Parity type may be set to either even or odd and is configurable.

4-to-1 PL3 to PSS MUX
This circuit muxes the status information from each of the four possible SPI-3 Bridges to the single Rx Port Status Sequencer. The PL3 to PSS MUX Block uses a round-robin method to send status information from the individual DPRAM FIFOs to the SPI-4 interface.

Register Interface

A bank of registers is implemented to manage various programmable control functions and store various error and status signals. These registers are controlled by a register interface that is compatible with the ORCA system bus interface. The ORCA SYSBUS slave interface is instantiated to control the core registers. The external FPGA control interface is compatible with a Motorola MPC860 Power PC interface.

The core maintains an 8-bit implementation of the system bus. Details of the operation of the system bus are available in Lattice technical note TN1017, *ORCA Series 4 MPI System Bus,* available from the Lattice web site at www.latticesemi.com.

DPRAM R/W Provisioning Registers
The register interface contains a set of sixteen provisioning registers which control which DPRAM virtual FIFOs are accessed, and in what order. These registers control both the reading of the DPRAMs on the receive side, and the writing of the transmit side DPRAMs. The receive and transmit sides are provisioned together, and may not be set independently.

The DPRAM virtual FIFOs are accessed in the order specified by the sixteen provisioning registers. The registers are read sequentially by the IP core, one at a time, and the value contained in each register determines which DPRAM virtual FIFO is accessed. After each of the sixteen registers have been read once in order, the IP core starts over at register 0 and continues reading all sixteen in a repeating fashion.

For example, for a 4 FIFO configuration, the user may write the provisioning registers with a pattern of 0,1,2,3,0,1,2,3,0,1,2,3,0,1,2,3 (DPRAM provisioning reg. 0 = 0x0, DPRAM provisioning reg. 1 = 0x1, …. DPRAM provisioning reg. 15 = 0x3). This causes the IP core to access all DPRAM virtual FIFOs in a simple round robin fashion with each FIFO getting _ of the total 2.5 Gbps bandwidth for the interface, and with no back-to-back accesses occurring on any virtual FIFO since the pattern specifies a sequentially increasing order of FIFO accesses.

As a second example, assuming the same 4 FIFO configuration as the first example, the user may write the 16 provisioning registers with a pattern of 0,1,0,1,0,1…0,1 (DPRAM provisioning reg. 0 = 0x0, DPRAM provisioning reg. 1 = 0x1, DPRAM provisioning reg. 2 = 0x0, …etc.). This causes the IP core to read only from virtual FIFOs 0 and 1, even though 4 FIFOs have been configured. FIFOs 2 and 3 are effectively "shut off," and the 2.5 Gbps bandwidth for the interface is evenly divided across FIFOs 1 and 2.

As a third example, if the user has configured the IP core for 8 FIFOs per interface, the user could write the 16 provisioning registers with a pattern of 0,1,2,3,0,1,2,3,0,1,2,3,4,5,6,7. This turns on all 8 of the FIFOs, but allocates only 1/16 of the total 2.5 Gbps bandwidth to each of the FIFOs 4,5,6, and 7. Whereas FIFOs 0,1,2, and 3 each get 3/16 of the total bandwidth since they are each assigned to be read 3 times during the repeating sixteen element pattern.

The user may write the sixteen provisioning registers using any pattern desired with the following restrictions:

1. The user should not assign a FIFO numbers to any provisioning register which exceeds the number of FIFOs configured.

2. The user should avoid back to back assignments to the same FIFO as this results in less than optimal data transfers for small packets.

3. All 16 registers must be provisioned. If the user chooses not to write to the registers, they will automatically reset to the values which provide for a round-robin scheme which includes all of the configured FIFOs. The user does not need to write the provisioning registers at all if the simple round-robin scheme is all that is desired.

4. The provisioning registers may be modified at any time, however any data in the FIFOs may be lost if they are modified while the circuit is in operation.

The IP Core provides a total of 2.5 Gbps bandwidth for each of the interfaces (1 to 4) configured. As described above, this bandwidth is divided across all of the FIFOs configured for a particular interfaces depending on the DPRAM R/W provisioning registers. Normally the full bandwidth will be achieved except for certain circumstances. First, if the user sends packets smaller than 64 bytes then there will be some degradation in bandwidth due to inefficiencies in handling small packets. Second, if the user configures multiple FIFOs per interface, but provisions only 1 FIFO (effectively turning off the other FIFOs using the provisioning registers) then there may be some degradation in bandwidth for packets smaller than 128 bytes. This situation can be avoided by configuring the IP Core for only 1 FIFO per interface (as opposed to "provisioning" for only 1 FIFO per interface) since the 1 FIFO interface design allows for full bandwidth for packets larger than 64 bytes.

Design Parameters

Table 2. Parameters

No.	Parameter	Dependency	Choice	Configuration 1 Default
1	Number of SPI3 Interfaces	Note 1	1, 2, 3, 4	4
2	Number of FIFOs per Interface	See Table 1	1, 2, 4, 8	4
3	Burst Size	Note 2	32, 64, 128, 256	256
4	Minimum Interburst Pause	—	0, 2	2
5	DTPA Selected	—	Yes/No	Yes
6	PTPA Selected	—	Yes/No	No
7	STPA Selected	—	Yes/No	No

1. Number of interfaces can only be 1 or 2 if the number of ports per interface is 8.
2. Burst size cannot be 256 if number of FIFOs is 8.

Register Descriptions

Table 3. Register Descriptions

Register Name	Register Address	Description
ID/Version Register	0x8000	ID/Version Number of Core
Parity Control Register	0x8001	Selects Odd or Even parity on SPI-3 interface
Transmit Parity Error Register	0x8002	Indicates parity error on a SPI-3 interface
Miscellaneous Control Register	0x8003	Controls side selection and flow control information
Receive Pause Control Register	0x8004	Sets minimum between transfers on receive side
Rx FIFO Partial Full Threshold Register	0x8005	Sets RX Flow control FIFO partial full threshold
Tx FIFO Partial Full Threshold Register	0x8006	Sets TX Flow control FIFO partial full threshold
Rx FIFO Partial Empty Threshold Register	0x8007	Sets RX Flow control FIFO partial empty threshold
TX FIFO Overflow Error Register 0	0x8008	Indicates FCFIFO overflow condition in the TX Interface 0
TX FIFO Overflow Error Register 1	0x8009	Indicates FCFIFO overflow condition in the TX Interface 1
TX FIFO Overflow Error Register 2	0x800A	Indicates FCFIFO overflow condition in the TX Interface 2
TX FIFO Overflow Error Register 3	0x800B	Indicates FCFIFO overflow condition in the TX Interface 3
DPRAM R/W Provisioning Register 0	0x800C	Selects Virtual FIFO to be accessed during cycle 1 of 16
DPRAM R/W Provisioning Register 1	0x800D	Selects Virtual FIFO to be accessed during cycle 2 of 16
DPRAM R/W Provisioning Register 2	0x800E	Selects Virtual FIFO to be accessed during cycle 3 of 16
DPRAM R/W Provisioning Register 3	0x800F	Selects Virtual FIFO to be accessed during cycle 4 of 16
DPRAM R/W Provisioning Register 4	0x8010	Selects Virtual FIFO to be accessed during cycle 5 of 16
DPRAM R/W Provisioning Register 5	0x8011	Selects Virtual FIFO to be accessed during cycle 6 of 16
DPRAM R/W Provisioning Register 6	0x8012	Selects Virtual FIFO to be accessed during cycle 7 of 16
DPRAM R/W Provisioning Register 7	0x8013	Selects Virtual FIFO to be accessed during cycle 8 of 16
DPRAM R/W Provisioning Register 8	0x8014	Selects Virtual FIFO to be accessed during cycle 9 of 16
DPRAM R/W Provisioning Register 9	0x8015	Selects Virtual FIFO to be accessed during cycle 10 of 16
DPRAM R/W Provisioning Register 10	0x8016	Selects Virtual FIFO to be accessed during cycle 11 of 16
DPRAM R/W Provisioning Register 11	0x8017	Selects Virtual FIFO to be accessed during cycle 12 of 16
DPRAM R/W Provisioning Register 12	0x8018	Selects Virtual FIFO to be accessed during cycle 13 of 16
DPRAM R/W Provisioning Register 13	0x8019	Selects Virtual FIFO to be accessed during cycle 14 of 16
DPRAM R/W Provisioning Register 14	0x801A	Selects Virtual FIFO to be accessed during cycle 15 of 16
DPRAM R/W Provisioning Register 15	0x801B	Selects Virtual FIFO to be accessed during cycle 16 of 16

Table 4. Register Map for Quad SPI-3 to SPI-4 PHY Layer Bridge Core

Address: **0x8000**		Name: **ID/Version Register**					
D7	D6	D5	D4	D3	D2	D1	D0
ID/VER							
Default value: 0x01				Mode: Read Only			
Description:							
ID/VER	The ID and version number of the core.						

Address: **0x8001**		Name: **Parity Control Register**					
D7	D6	D5	D4	D3	D2	D1	D0
—		—			—	RX_PAR	TX_PAR
Default value: 0x03				Mode: Read Only			
Description:							
TX_PAR	When high, odd parity is supported on the SPI-3 transmit interfaces.						
RX_PAR	When high, odd parity is supported on the SPI-3 receive interfaces.						

Address: **0x8002**		Name: **Transmit Parity Error Register**					
D7	D6	D5	D4	D3	D2	D1	D0
—		—		PERR_3	PERR_2	PERR_1	PERR_0
Default value: n/a				Mode: Clear on Read			
Description:							
PERR_N	Detection of a parity error on any port on interface N sets this bit. This bit clears upon reading.						

Address: **0x8003**		Name: **Miscellaneous Control Register**					
D7	D6	D5	D4	D3	D2	D1	D0
—		—			—	SEL_AB	PSS_EN
Default value: 0x00				Mode: Read/Write			
Description:							
PSS_EN	When high, the flow control information from the core overrides that generated by the ASB.						
SEL_AB	When high, the core connects side B or the ASB (default is side A).						

Address: **0x8004**		Name: **Receive Pause Control Register**					
D7	D6	D5	D4	D3	D2	D1	D0
—		—			—		RX_PAUSE
Default value: 0x00				Mode: Read/Write			
Description:							
RX_PAUSE	Selects the minimum pause (0 or 2 clock cycles) between receive side transfers. When high, the minimum pause is 2 clock cycles.						

Table 4. Register Map for Quad SPI-3 to SPI-4 PHY Layer Bridge Core (Cont.)

Address: **0x8005**		Name: **Receive FIFO Partial Full Threshold**					
D7	D6	D5	D4	D3	D2	D1	D0
RX_FIFO_PFT							
Default value: 3/4 (FIFO Size)				Mode: Read/Write			
Description:							
RX_FIFO_PFT		When the number of words (8 bytes) in the FIFO exceeds this value, the FIFO's frame available signal asserts.					

Address: **0x8006**		Name: **Transmit FIFO Partial Full Threshold**					
D7	D6	D5	D4	D3	D2	D1	D0
TX_FIFO_PFT							
Default value: (FIFO Size - Burst Size)				Mode: Read/Write			
Description:							
TX_FIFO_PETW		When the number of words (8 bytes) in the FIFO exceeds this value, the FIFO's PARTIAL_FULL signal asserts.					

Address: **0x8007**		Name: **Receive FIFO Partial Empty Threshold**					
D7	D6	D5	D4	D3	D2	D1	D0
RX_FIFO_PET							
Default value: Burst Size				Mode: Read/Write			
Description:							
TX_FIFO_PET		When the number of words in the FIFO is less than or equal to this value, the FIFO's almost empty signal asserts.					

Address: **0x8008**		Name: **Transmit FIFO Overflow Error Register Interface 0**					
D7	D6	D5	D4	D3	D2	D1	D0
OF_ERR7	OF_ERR6	OF_ERR5	OF_ERR4	OF_ERR3	OF_ERR2	OF_ERR1	OF_ERR0
Default value: n/a				Mode: Clear on Read			
Description:							
OF_ERR_N		When high, indicates that the FIFO for port N on interface 0 experienced an overflow error. This bit clears upon reading. For configurations with less than 8 FIFOs, all unused bits are set to zero.					

Address: **0x8009**		Name: **Transmit FIFO Overflow Error Register Interface 1**					
D7	D6	D5	D4	D3	D2	D1	D0
OF_ERR7	OF_ERR6	OF_ERR5	OF_ERR4	OF_ERR3	OF_ERR2	OF_ERR1	OF_ERR0
Default value: n/a				Mode: Clear on Read			
Description:							
OF_ERR_N		When high, indicates that the FIFO for port N on interface 1 experienced an overflow error. This bit clears upon reading. For configurations with less than 8 FIFOs, all unused bits are set to zero.					

Table 4. Register Map for Quad SPI-3 to SPI-4 PHY Layer Bridge Core (Cont.)

Address: **0x800A**		Name: **Transmit FIFO Overflow Error Register Interface 2**					
D7	D6	D5	D4	D3	D2	D1	D0
OF_ERR7	OF_ERR6	OF_ERR5	OF_ERR4	OF_ERR3	OF_ERR2	OF_ERR1	OF_ERR0
Default value: n/a				Mode: Clear on Read			
Description:							
OF_ERR_N		When high, indicates that the FIFO for port N on interface 2 experienced an overflow error. This bit clears upon reading. For configurations with less than 8 FIFOs, all unused bits are set to zero.					

Address: **0x800B**		Name: **Transmit FIFO Overflow Error Register Interface 3**					
D7	D6	D5	D4	D3	D2	D1	D0
OF_ERR7	OF_ERR6	OF_ERR5	OF_ERR4	OF_ERR3	OF_ERR2	OF_ERR1	OF_ERR0
Default value: n/a				Mode: Clear on Read			
Description:							
OF_ERR_N		When high, indicates that the FIFO for port N on interface 3 experienced an overflow error. This bit clears upon reading. For configurations with less than 8 FIFOs, all unused bits are set to zero.					

Address: **0x800C**		Name: **DPRAM R/W Provisioning Register 0**					
D7	D6	D5	D4	D3	D2	D1	D0
—		—			RSEl_0		
Default value: Depends on configuration. See Functional Description section.				Mode: Read/Write			
Description:							
RSEL_0		Selects Virtual FIFO to be accessed during cycle 1 of 16.					

Address: **0x800D**		Name: **DPRAM R/W Provisioning Register 1**					
D7	D6	D5	D4	D3	D2	D1	D0
—		—			RSEl_1		
Default value: Depends on configuration. See Functional Description section.				Mode: Read/Write			
Description:							
RSEL_1		Selects Virtual FIFO to be accessed during cycle 2 of 16.					

Address: **0x800E**		Name: **DPRAM R/W Provisioning Register 2**					
D7	D6	D5	D4	D3	D2	D1	D0
—		—			RSEl_2		
Default value: Depends on configuration. See Functional Description section.				Mode: Read/Write			
Description:							
RSEL_2		Selects Virtual FIFO to be accessed during cycle 3 of 16.					

Table 4. Register Map for Quad SPI-3 to SPI-4 PHY Layer Bridge Core (Cont.)

Address: **0x800F**		Name: **DPRAM R/W Provisioning Register 3**					
D7	D6	D5	D4	D3	D2	D1	D0
—		—			RSEl_3		
Default value: Depends on configuration. See Functional Description section.				Mode: Read/Write			
Description:							
RSEL_3	Selects Virtual FIFO to be accessed during cycle 4 of 16.						

Address: **0x8010**		Name: **DPRAM R/W Provisioning Register 4**					
D7	D6	D5	D4	D3	D2	D1	D0
—		—			RSEl_4		
Default value: Depends on configuration. See Functional Description section.				Mode: Read/Write			
Description:							
RSEL_4	Selects Virtual FIFO to be accessed during cycle 5 of 16.						

Address: **0x8011**		Name: **DPRAM R/W Provisioning Register 5**					
D7	D6	D5	D4	D3	D2	D1	D0
—		—			RSEl_5		
Default value: Depends on configuration. See Functional Description section.				Mode: Read/Write			
Description:							
RSEL_5	Selects Virtual FIFO to be accessed during cycle 6 of 16.						

Address: **0x8012**		Name: **DPRAM R/W Provisioning Register 6**					
D7	D6	D5	D4	D3	D2	D1	D0
—		—			RSEl_6		
Default value: Depends on configuration. See Functional Description section.				Mode: Read/Write			
Description:							
RSEL_6	Selects Virtual FIFO to be accessed during cycle 7 of 16.						

Address: **0x8013**		Name: **DPRAM R/W Provisioning Register 7**					
D7	D6	D5	D4	D3	D2	D1	D0
—		—			RSEl_7		
Default value: Depends on configuration. See Functional Description section.				Mode: Read/Write			
Description:							
RSEL_7	Selects Virtual FIFO to be accessed during cycle 8 of 16.						

Table 4. Register Map for Quad SPI-3 to SPI-4 PHY Layer Bridge Core (Cont.)

Address: **0x8014**		Name: **DPRAM R/W Provisioning Register 8**					
D7	D6	D5	D4	D3	D2	D1	D0
—		—			RSEl_8		
Default value: Depends on configuration. See Functional Description section.				Mode: Read/Write			
Description:							
RSEL_8	Selects Virtual FIFO to be accessed during cycle 9 of 16.						

Address: **0x8015**		Name: **DPRAM R/W Provisioning Register 9**					
D7	D6	D5	D4	D3	D2	D1	D0
—		—			RSEl_9		
Default value: Depends on configuration. See Functional Description section.				Mode: Read/Write			
Description:							
RSEL_9	Selects Virtual FIFO to be accessed during cycle 10 of 16.						

Address: **0x8016**		Name: **DPRAM R/W Provisioning Register 10**					
D7	D6	D5	D4	D3	D2	D1	D0
—		—			RSEl_10		
Default value: Depends on configuration. See Functional Description section.				Mode: Read/Write			
Description:							
RSEL_10	Selects Virtual FIFO to be accessed during cycle 11 of 16.						

Address: **0x8017**		Name: **DPRAM R/W Provisioning Register 11**					
D7	D6	D5	D4	D3	D2	D1	D0
—		—			RSEl_11		
Default value: Depends on configuration. See Functional Description section.				Mode: Read/Write			
Description:							
RSEL_11	Selects Virtual FIFO to be accessed during cycle 12 of 16.						

Address: **0x8017**		Name: **DPRAM R/W Provisioning Register 11**					
D7	D6	D5	D4	D3	D2	D1	D0
—		—			RSEl_11		
Default value: Depends on configuration. See Functional Description section.				Mode: Read/Write			
Description:							
RSEL_11	Selects Virtual FIFO to be accessed during cycle 12 of 16.						

Table 4. Register Map for Quad SPI-3 to SPI-4 PHY Layer Bridge Core (Cont.)

Address: **0x8018**		Name: **DPRAM R/W Provisioning Register 12**					
D7	D6	D5	D4	D3	D2	D1	D0
—		—			RSEl_12		
Default value: Depends on configuration. See Functional Description section.				Mode: Read/Write			
Description:							
RSEL_12	Selects Virtual FIFO to be accessed during cycle 13 of 16.						

Address: **0x8019**		Name: **DPRAM R/W Provisioning Register 13**					
D7	D6	D5	D4	D3	D2	D1	D0
—		—			RSEl_13		
Default value: Depends on configuration. See Functional Description section.				Mode: Read/Write			
Description:							
RSEL_13	Selects Virtual FIFO to be accessed during cycle 14 of 16.						

Address: **0x801A**		Name: **DPRAM R/W Provisioning Register 14**					
D7	D6	D5	D4	D3	D2	D1	D0
—		—			RSEl_14		
Default value: Depends on configuration. See Functional Description section.				Mode: Read/Write			
Description:							
RSEL_14	Selects Virtual FIFO to be accessed during cycle 15 of 16.						

Address: **0x801B**		Name: **DPRAM R/W Provisioning Register 15**					
D7	D6	D5	D4	D3	D2	D1	D0
—		—			RSEl_15		
Default value: Depends on configuration. See Functional Description section.				Mode: Read/Write			
Description:							
RSEL_15	Selects Virtual FIFO to be accessed during cycle 16 of 16.						

Signal Descriptions

Table 5. Signal Definitions for Quad SPI-3 to SPI-4 Bridge Solution I/O

Signal Name	Direction	Description
SPI-3 Signals[1]		
rfclk	Input	SPI-3 Interface Clock
rerr	Output	Receive Error Indicator
renb	Input	Receive Read Enable
renb_a	Input	Internal RENB (connect to renb externally)
renb_b	Input	Internal RENB (connect to renb externally)
rval	Output	Receive Data Valid
rdat [31:0]	Output	Receive Packet Data Bus
rmod [1:0]	Output	Receive Word Modulo
rprty	Output	Receive Bus Parity
rsx	Output	Receive Start of Transfer
rsop	Output	Receive Start of Packet
reop	Output	Receive End of Packet
tsx	Input	Transmit Start of Transfer
tenb	Input	Transmit Write Enable
terr	Input	Transmit Error Indicator
tsop	Input	Transmit Start of Packet
teop	Input	Transmit End of Packet
tdat [31:0]	Input	Transmit Packet Data Bus
tmod [1:0]	Input	Transmit Word Modulo
tprty	Input	Transmit Bus Parity
tadr [5:0]	Input	Transmit PHY Address Bus (only bits [2:0] used)
dtpa [7:0]	Output	Direct Transmit Packet Available
stpa	Output	Selected PHY Transmit Packet Available
ptpa	Output	Polled PHY Transmit Packet Available
SPI-4 Signals[2]		
ardat[15:0]p	Input	LVDS signals for SPI-4.2 RXA
ardat[15:0]n		
arctlp	Input	LVDS signal for RXA Control
arctln		
ardclkp	Input	Differential RXA Data clock signal
ardclkn		
arstat[1:0]p	Output	LVDS Status Output
arstat[1:0]n		
arsclkp	Output	LVDS Status Output clock
arsclkn		
rstat[1:0]a	Output	LVTTL Status Output
rsclka	Output	LVTTL Status Clock Output
ardat[15:0]p	Output	LVDS signals for SPI-4.2 TXA
ardat[15:0]n		
atctlp	Output	Transmit Control for SPI-4.2 TXA
atctln		

Table 5. Signal Definitions for Quad SPI-3 to SPI-4 Bridge Solution I/O (Continued)

Signal Name	Direction	Description
atdclkp	Output	Differential TXA clock signal
atdclkn		
atstat[1:0]p	Input	LVDS Status Input
atstat[1:0]n		
atsclkp	Input	LVDS Status Input clock
atsclkn		
tstat[1:0]a	Input	LVTTL Status Input
tsclka	Input	LVTTL Status Clock Input
ORSPI4 Embedded Core Control, Global I/O and FPGA Configuration I/O		
Refer to the ORCA Series 4 FPGA Data Sheet and the ORSPI4 Data Sheet for information on the various configuration options.		

1. The signals listed here are required for a single SPI-3 interface. The signals should be replicated for each additional SPI-3 Interface instantiation.
2. The signals listed here are required for the SPI-4 interface A. Please refer to the ORSPI4 Data Sheet for additional information on configuring the SPI-4 interface for specific applications.

Table 6. Signal Definitions for Quad SPI-3 to SPI-4 Bridge Solution - FPGA/Embedded ASB Interface (Internal to ORSPI4 Device)

Signal Name	FPGA Direction	Description
Receive Interface Signals		
rx_clk	Output	Read Clock to DPRAM Bank 0
rx_d[31:0]	Input	Read Data from DPRAM Bank 0
rx_d[32]	Input	SOP Indicator from DPRAM Bank 0
rx_d[33]	Input	EOP Indicator from DPRAM Bank 0
rx_d[37:34]	Input	Byte Valid Indicator from DPRAM Bank 0
rx_d[38]	Input	Error Indication for Read Data
rx_d[39]	Input	Port ID Indicator (RSX in SPI-3)
rx_fifo_empty	Input	FIFO Empty Flag from DPRAM Bank 0
rx_a[2:0]	Output	Read Address to DRPAM Bank 0
rx_re	Output	Read Enable to DPRAM Bank 0
Receive Status Signals		
rx_pss_clk	Output	Write Clock to PSS Memory
rx_ext_status_en	Output	Indicates Valid Status
rx_port_status[1:0]	Output	2-Bit Status for Port Specified by Port Id
rx_fpga_portid[7:0]	Output	Address of Port for which Status is provided
rx_pss_we	Output	Write Enable to PSS Memory
Transmit Status Signals		
tref_clk	Input	SPI-4 Transmit Reference Clock
tx_spi_clk	Output	SPI-3 Transmit Clock
tx_port_id[7:0]	Input	Address of Port for which Status is provided
tx_bkp	Input	SPI-4 Backpressure to FPGA
tx_stat[1:0]	Input	Status of Port specified by TX_PORT_ID
Transmit Interface Signals		
tx_clk	Output	Write Clock to DPRAM Bank 0

Table 6. Signal Definitions for Quad SPI-3 to SPI-4 Bridge Solution - FPGA/Embedded ASB Interface (Internal to ORSPI4 Device) (Continued)

Signal Name	FPGA Direction	Description
tx_d[31:0]	Input	Write Data to DPRAM Bank 0
tx_d[32]	Input	SOP Indicator to DPRAM Bank 0
tx_d[33]	Input	EOP Indicator to DPRAM Bank 0
tx_d[37:34]	Input	Byte Valid Indicator to DPRAM Bank 0
tx_d[38]	Input	Error Indication for Write Data
wd_cnt_rst	Output	WCL Word Count Reset 0
tx_we	Output	Write Enable for DPRAM Bank 0
tx_a[2:0]	Output	Write Address for DPRAM Bank 0
tx_port[7:0]	Output	Port ID for DPRAM Bank 0
fifo_full	Output	FIFO Full Flag from DPRAM Bank 0

1. The signals listed here are required for a single SPI-3 interface. The signals should be replicated for each additional SPI-3 Interface instantiation. The status signals in both receive and transmit directions between the FPGA and the embedded Application Specific Block (ASB) are not replicated when the core has multiple SPI-3 interfaces.

I/O Timing and Electrical Specifications

SPI-3 Specifications

The examples in the Transmit Logical Timing and Receive Logical Timing sections are provided only to aid in the visualization of the interface operation.

Transmit Logical Timing

Figure 3 shows transactions on a SPI-3 Transmit interface with two ports. The SPI-3 transmit interface is controlled by the Link Layer device using the tenb signal. All signals must be updated and sampled using the rising edge of the transmit FIFO clock, tfclk. The PHY layer device indicates that a FIFO is not full by asserting the appropriate transmit packet available signal dtpa. dtpa remains asserted until the transmit FIFO is almost full. "Almost full" implies that the PHY layer device can accept, at most, a predefined number of writes after the current write.

If dtpa is asserted and the Link Layer device is ready to write a word, tsx will be asserted, tenb deasserted and the port address presented on the tdat bus if required. Subsequent data transfers with tenb low are treated as packet data which is written to the selected FIFO. At any time, if the Link Layer device does not have data to write, tenb may be deasserted. The tsop and teop signals must be appropriately marked at the start and end of packets on the tdat bus.

When dtpa transitions low and it has been sampled, the Link Layer device can write no more than a predefined number of bytes to the selected FIFO. In this particular example, the predefined value is two double-words or eight bytes. In the IP Core implementation if dtpa is high the PHY layer can accept a complete burst. When the dtpa is deasserted the PHY layer can accept the burst being transferred, but no more. If the Link Layer writes more than that predefined number of words and dtpa remains deasserted throughout the PHY layer device should indicate an error condition and ignore additional writes until dtpa is asserted again.

Figure 3. Transactions on the SPI-3 Transmit Interface

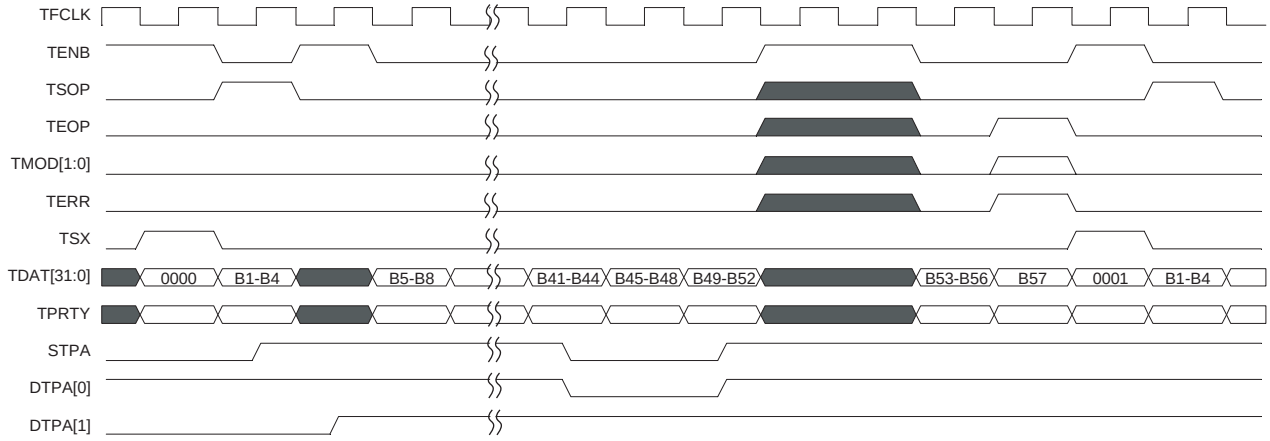
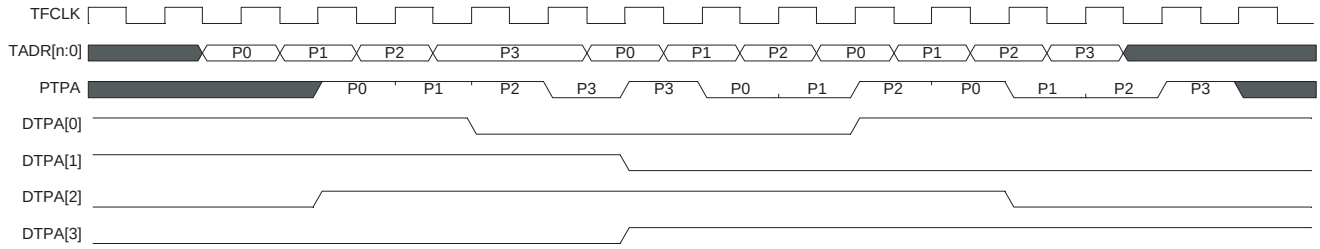


Figure 4 shows the use of the polling feature of the transmit interface. For comparison purposes, the "direct transmit packet available" signals for the example ports are provided in the diagram. The status of a given PHY port may be determined by setting the polling address `tadr` bus to the port address. The "polled transmit packet available" signal `ptpa` is updated with the transmit FIFO status in a pipelined manner. The Link Layer device is not restricted in its polling order. The "selected transmit packet available" `stpa` signal allows monitoring of the selected PHY status and halting of the data transfer once the FIFO is full. The `ptpa` signal allows polling other PHY's at any time, including while a data transfer is in progress.

Figure 4. Packet Level Transmit Polling

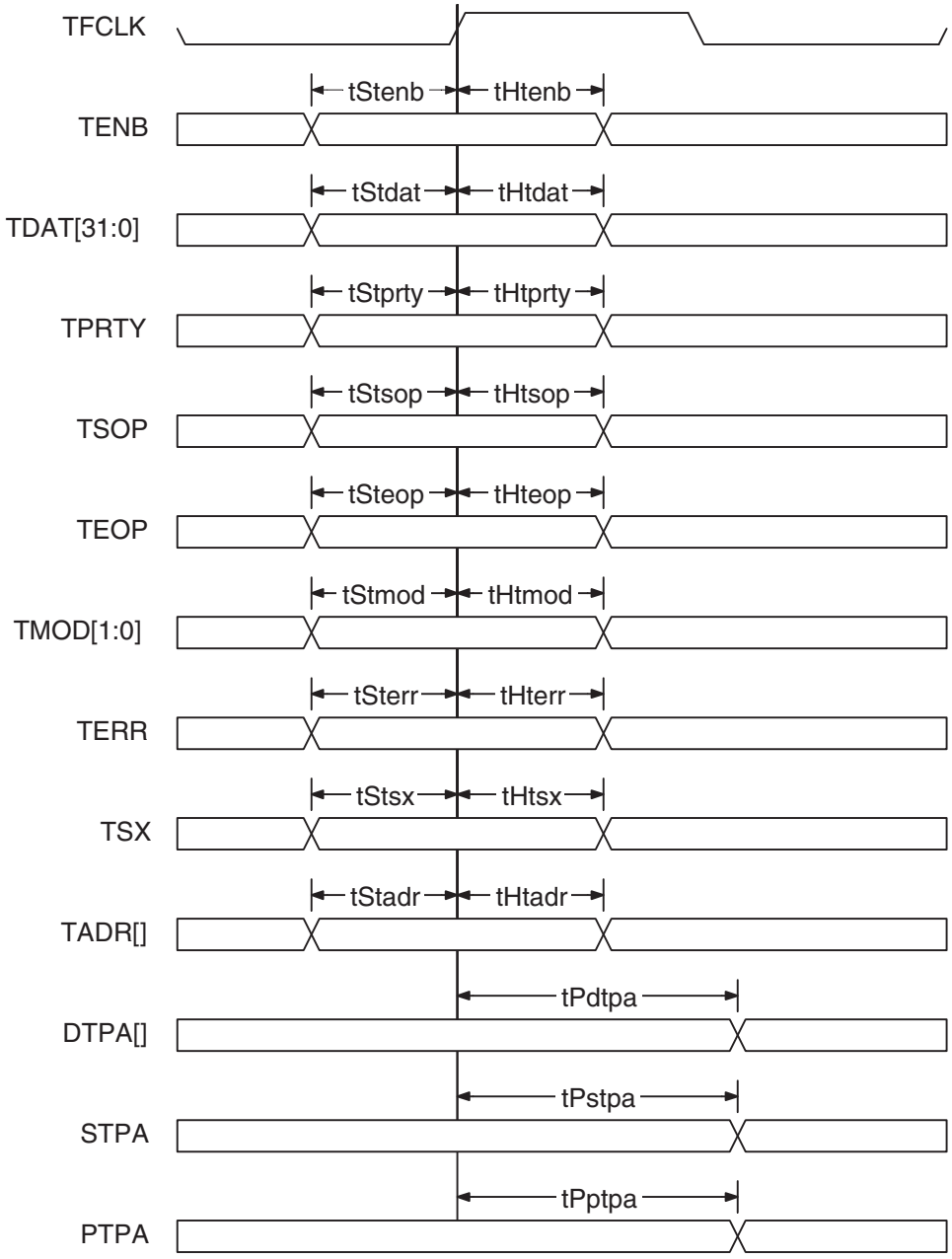


Transmit Interface AC Timing

Table 7. Transmit Interface Timing

Symbol	Description	Min.	Max.	Units
	tfclk frequency	—	104	MHz
	tfclk duty cycle	40	60	%
t_{STENB}	tenb set-up time to tfclk	2	—	ns
t_{HTENB}	tenb hold time to tfclk	0.5	—	ns
t_{STDAT}	tdat[31:0] set-uo time to tfclk	2	—	ns
t_{HTDAT}	tdat[31:0] hold time to tfclk	0.5	—	ns
t_{STPRTY}	tprty set-up time to tfclk	2	—	ns
t_{HTPRTY}	tprty hold time to tfclk	0.5	—	ns
t_{STSOP}	tsop set-up time to tfclk	2	—	ns
t_{HTSOP}	tsop hold time to tfclk	0.5	—	ns
t_{STEOP}	teop set-up time to tfclk	2	—	ns
t_{HTEOP}	teop hold time to tfclk	0.5	—	ns
t_{STMOD}	tmod set-up time to tfclk	2	—	ns
t_{HTMOD}	tmod hold time to tfclk	0.5	—	ns
t_{STERR}	terr set-up time to tfclk	2	—	ns
t_{HTERR}	terr hold time to tfclk	0.5	—	ns
t_{STSX}	tsx set-up time to tfclk	2	—	ns
t_{HTSX}	tsx hold time to tfclk	0.5	—	ns
t_{STADR}	tadr[5:0] set-up time to tfclk (only signals [2:0] are used)	2	—	ns
t_{HTADR}	tadr[5:0] hold time to tfclk (only signals [2:0] are used)	0.5	—	ns
t_{PDTPA}	tfclk high to dtpa valid	1.5	6	ns
t_{PSTPA}	tfclk high to stpa valid	1.5	6	ns
t_{PPTPA}	tfclk high to ptpa valid	1.5	6	ns

Figure 5. Transmit Physical Timing



Notes on Transmit Interface I/O Timing

1. When a set-up time is specified between an input and a clock, the set-up time is the time in nanoseconds from the 1.4V point of the input to the 1.4V point of the clock.
2. When a hold time is specified between an input and a clock, the hold time is the time in nanoseconds from the 1.4V point of the clock to the 1.4V point of the input.
3. Output propagation delay time is the time in nanoseconds from the 1.4V point of the reference signal to the 1.4V point of the output.
4. Maximum output propagation delays are measured with a 30 pF load on the outputs.

Receive Logical Timing

Figure 6 shows transactions on a SPI-3 Receive interface with two ports. The SPI-3 Receive Interface is controlled by the Link Layer device using the `renb` signal. All signals must be updated and sampled using the rising edge of the receive FIFO clock. The `rdat` bus, `rptry`, `rmod`, `rsop`, `reop` and `rerr` signals are valid in cycles for which `rval` is high and `renb` was low in the previous cycle. When transferring data, `rval` is asserted and remains high until the internal FIFO of the PHY layer device is empty or an end of packet is transferred. The `rsx` signal is valid in the cycle for which `rval` is low and `renb` was low in the previous cycle.

The PHY informs the Link Layer device of the port address of the selected FIFO by asserting `rsx` with the port address on the `rdat` bus. The Link Layer may pause the Receive Interface at any time by deasserting the `renb` signal. When the selected FIFO is empty, `rval` is deasserted. In this example, the `rval` is re-asserted, without changing the selected FIFO, transferring the last section of the packet. The end of the packet is indicated with the REOP signal. Thus, the next subsequent FIFO transfer for this port would be the start of the next packet. If an error occurred during the reception of the packet, the `rerr` would be asserted with `reop`. Since another port's FIFO has sufficient data to initiate a bus transfer, `rsx` is again asserted with the port address. In this case, an intermediate section of the packet is being transferred.

Figure 6. Transactions on the SPI-3 Receive Interface

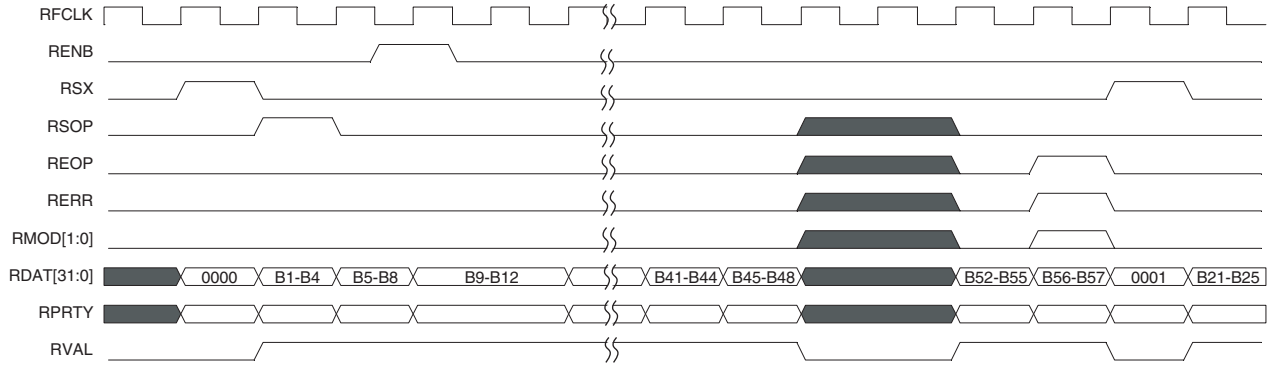


Figure 7 shows the use of the pause feature of the receive interface. The first transfer is a complete 3-byte packet and the second transfer is the end of a 36-byte packet. The pause allows the Link Layer device to halt data between transfers. In order to handle an end of packet, the Link Layer device may deassert the `renb` signal when it samples `reop` active. As shown in the diagram, the Link Layer device pauses the PHY device on the in-band address for two clock cycles.

Figure 7. Receive Interface Transactions with Pausing

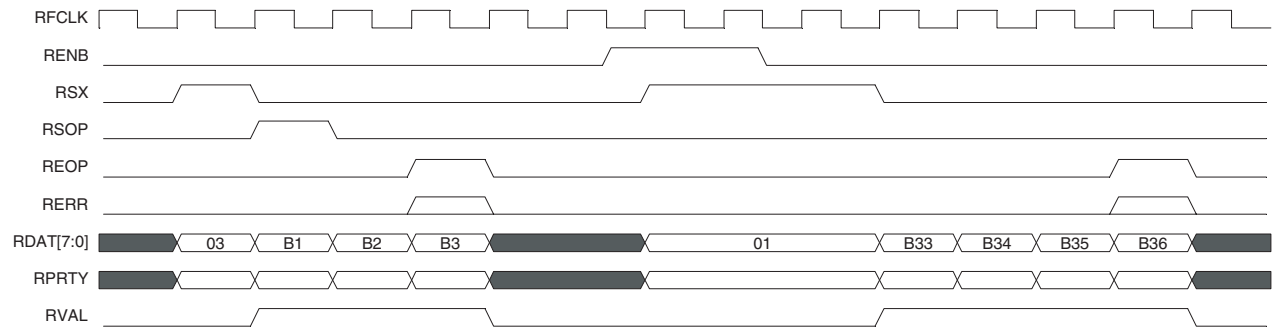
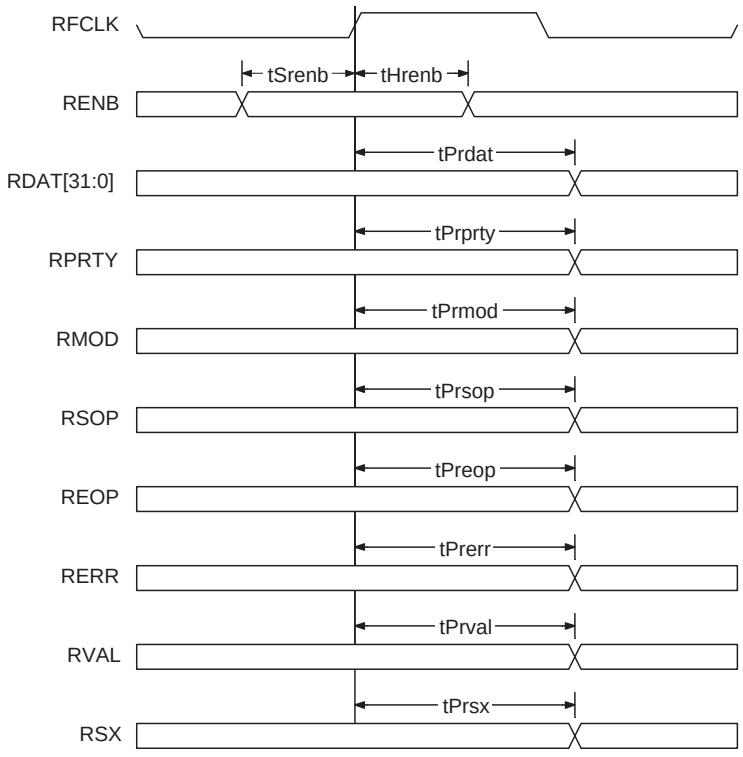


Table 8. Receive Interface Timing

Symbol	Description	Min.	Max.	Units
	`rfclk` frequency	—	104	MHz
	`rfclk` duty cycle	40	60	%
t_{SRENB}	`renb` set-up time to `rfclk`	2	—	ns
t_{HRENB}	`renb` hold time to `rfclk`	0.5	—	ns
t_{PRDAT}	`rfclk` high to `rdat` valid	1.5	6	ns
t_{PRPRTY}	`rfclk` high to `rprty` valid	1.5	6	ns
t_{PRSOP}	`rfclk` high to `rsop` valid	1.5	6	ns
t_{PREOP}	`rfclk` high to `reop` valid	1.5	6	ns
t_{PRMOD}	`rfclk` high to `rmod` valid	1.5	6	ns
t_{PRERR}	`rfclk` high to `rerr` valid	1.5	6	ns
t_{PRVAL}	`rfclk` high to `rval` valid	1.5	6	ns
t_{PRSX}	`rfclk` high to `rsx` valid	1.5	6	ns

Figure 8. Receive Physical Timing



Notes on Receive I/O Timing:

1. When a set-up time is specified between an input and a clock, the set-up time is the time from the 1.4V point of the input to the 1.4V point of the clock.
2. When a hold time is specified between an input and a clock, the hold time is the time from the 1.4V point of the clock to the 1.4V point of the input.
3. Output propagation delay time is the time from the 1.4V point of the reference signal to the 1.4V point of the output.
4. Maximum output propagation delays are measured with a 30 pF load on the outputs.

SPI-4 Specifications

System-level reference points for specified parameters in this section are shown in Figure 9. Corresponding reference points with respect to the clock edge are shown in Figures. 10 and 11.

Figure 9. System Level Reference Points

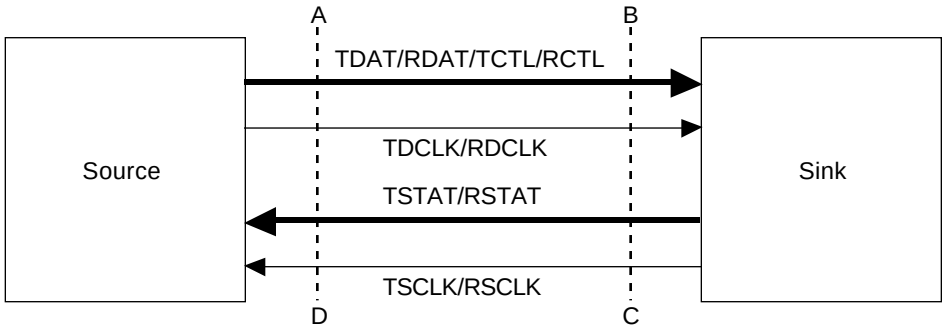


Figure 10. Reference Points for Data Path Timing Parameters

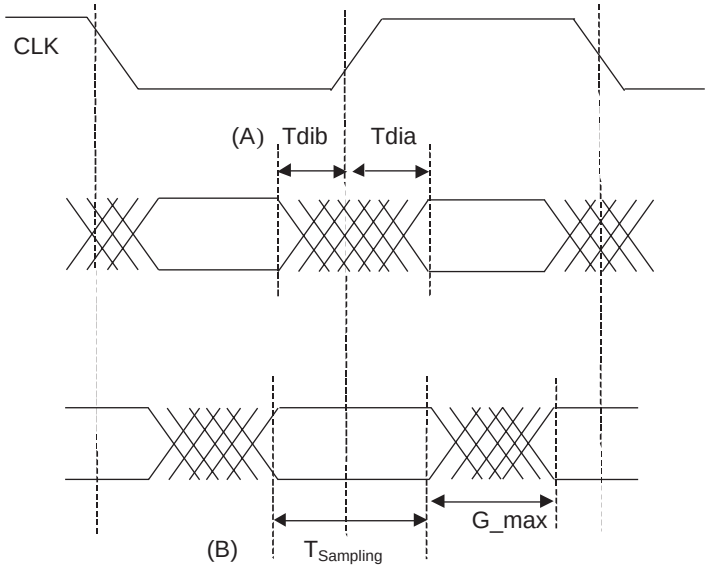
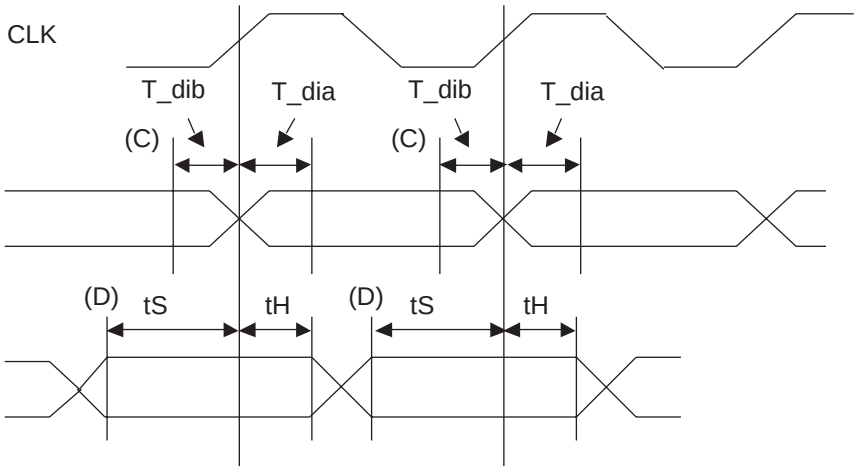


Figure 11. Reference Points for FIFO Status Channel Timing Parameters



Data Path

Two sets of data path timing parameters are specified to support different bit alignment schemes at the receiver. Table 9 gives the corresponding parameters for the case of "static alignment", in which the receiver latches data at a fixed point in time relative to clock (requiring a more precisely specified sampling window). Table 10 gives the corresponding parameters for the case of "dynamic alignment", in which the receiver has the capability of centering the data and control bits relative to clock. From an AC timing perspective, a compliant interface only needs to meet the parameters at the data path for either static or dynamic alignment, but may also comply to both sets of parameters. A compliant driver must meet both timing specifications to be interoperable with both types of receivers.

Table 9. Data Path Interface Timing (Static Alignment)

Symbol	Description		Min.	Max.	Units
fD	TDCLK / RDCLK Frequency		311		MHz
	TDCLK / RDCLK Duty Cycle		45	55	%
T_dia, T_dib	Data invalid window with respect to clock edge (Ref. point A)			280	ps
Gmax	Worst-case skew and jitter contribution (Ref. Point B)			790	ps
Tsampling	Data valid window with respect to clock edge (Ref. point B)			1/2fD	ps
	20-80% rise and fall times UI = 1/2fD	(Reference point A)	100 ps	0.30 UI	
		(Reference point B)	100 ps	0.36 UI	

Table 10. Data Path Interface Timing (Dynamic Alignment)

Symbol	Description		Min.	Max.	Units
fD	TDCLK / RDCLK Frequency		311		MHz
	TDCLK / RDCLK Jitter (at Reference Point A)			0.1	UI
	TDAT / RDAT / TCTL / RCTL Jitter (at Reference Point A)			0.24	UI
	20-80% rise and fall times UI = 1/2fD	(Reference point A)	100 ps	0.30 UI	
		(Reference point B)	100 ps	0.36 UI	

Notes:

1. Rise and fall times assume nominal 100Ω termination and exclude reflections.
2. All timing parameters are measured relative to the differential crossing point of the corresponding clock signal.
3. Jitter parameters are peak-to-peak, measured above fD / 1000 and below fD.
4. Receiver sensitivity is assumed to be less than or equal to 100 mV.
5. Assumes a 5 pF output load at reference point A, a 10 pF load at reference point B, and a 50Ω transmission line in between.
6. Assumes up to 20 ps skew between traces of a differential pair.

FIFO Status Channel

The following section describes AC timing parameters for a FIFO status channel implemented using LVTTL I/O. For optional LVDS FIFO status channel implementations, the reader is referred to the LVDS data path parameters in the previous section. As noted in Table 11, the maximum clock frequency of the LVTTL FIFO Status Channel shall not exceed one quarter of the selected data path clock rate.

Table 11. FIFO Status Channels Interface Timing (LVTTL I/O)

Symbol	Description		Min.	Max.	Units
f_S	TSCLK Frequency			fD/4	
	TSCLK Duty Cycle		45	60	%
T_dia	Data invalid window with respect to clock edge	(Reference point C)	2.5		ns
T_dib		(Reference point D)	1		
t_{SSCLK}	TSTAT setup time to TSCLK RSTAT setup time to RSCLK	(Reference point D)	2		ns
t_{HSCLK}	TSTAT hold time to TSCLK RSTAT hold time to RSCLK	(Reference point D)	0.5		ns

Notes on LVTTL I/O Timing:

1. When a set-up time is specified between an input and a clock, the set-up time is the time from the 1.4V point of the input to the 1.4V point of the clock.
2. When a hold time is specified between an input and a clock, the hold time is the time from the 1.4V point of the clock to the 1.4V point of the input.
3. Assumes a 25 pF, 500Ω load.

DC Parameters

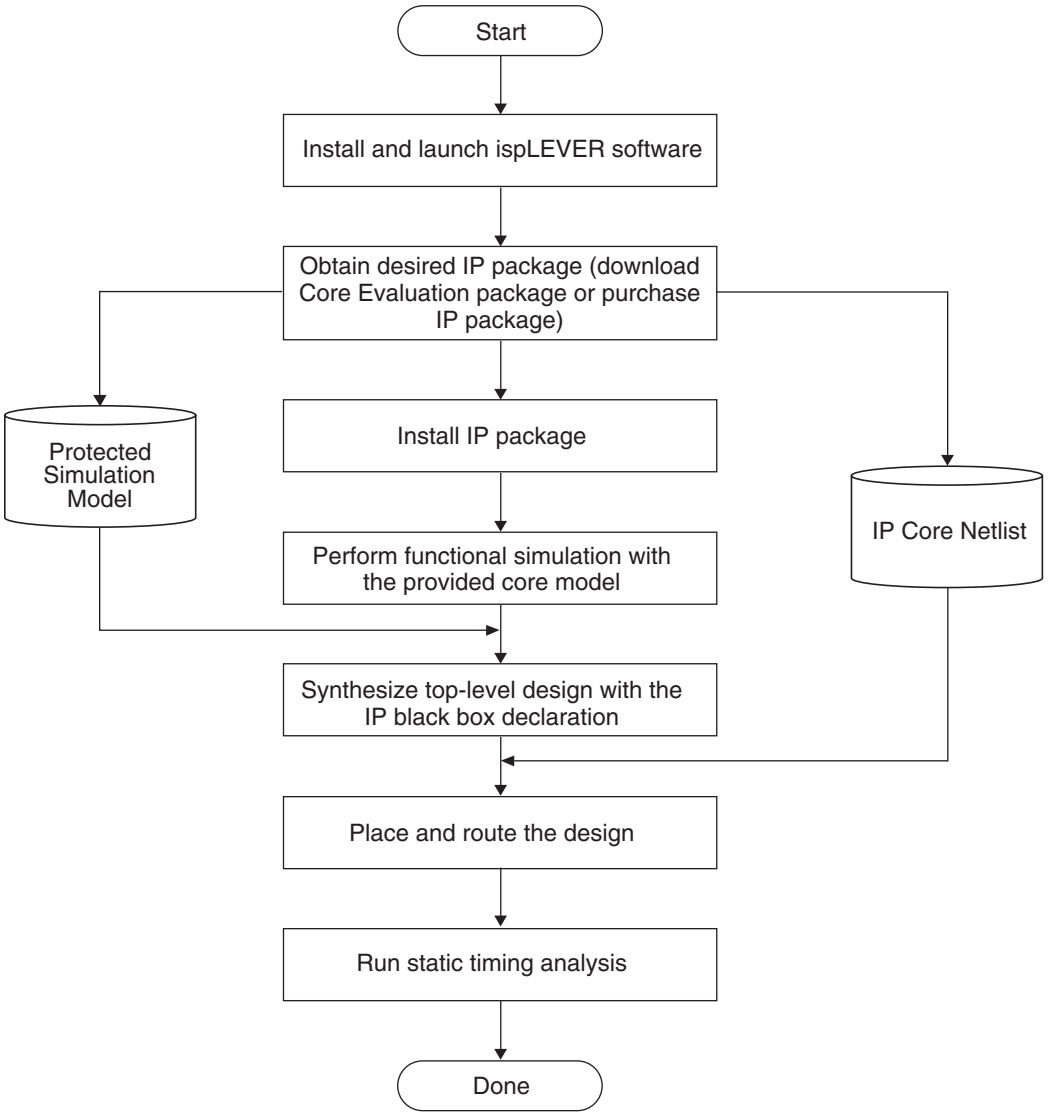
Table 12. DC Threshold for the LVTTL FIFO Status Channels

Parameter	Symbol	Min. (V)	Max. (V)
Output High Voltage	V_OH	2.4	3.6
Input High Voltage	V_IH	2	3.6
Output Low Voltage	V_OL		0.4
Input Low Voltage	V_IL		0.8

Quad SPI-3 to SPI-4 PHY Layer Bridge Core Design Flow

The Quad SPI-3 to SPI-4 Bridge IP Core can be implemented using various methods. The scope of this document covers only the push-button Graphical User Interface (GUI) flow. Figure 12 illustrates the software flow model used when evaluating with the Quad SPI-3 to SPI-4 Bridge core.

Figure 12. Lattice IP Core Evaluation Flow



Functional Simulation under ModelSim (PC Platform)

Once the Quad SPI-3 to SPI-4 Bridge core has been downloaded and unzipped to the designated directory, the core is ready for evaluation. The RTL simulation environment contains a testbench and a simple application that uses the Quad SPI-3 to SPI-4 Bridge design. The application instantiates the Quad SPI-3 to SPI-4 Bridge core, an ORCA ORSPI4 module and an ORCA SYSBUS module. The module name of the application is called "orspi_qspi3". The testbench includes a SPI-4 driver, a SPI-4 monitor, an instantiation of the user application and SPI-3 loopback models. In the simulation many packets of varying length generated by the SPI-4 driver are applied to the receive side of the SPI-4 bridge. A loopback is implemented at the SPI-3 interface. The packets received at the SPI-4 transmit interface are checked against the packets sent by the SPI-4 driver.

The Quad SPI-3 to SPI-4 PHY top level source code and testbench models have been compiled into the work directories "source\top", source\sim", and "eval\test*". The Quad SPI-3 to SPI-4 PHY IP CORE, ORCA4, ORSPI4 and SYSBUS MTI models have been provided in the directory "lib\modelsim" as zipped archives (spi_324p_04_1_001.zip, orca4_work.zip, orspi4_work.zip and sysbus_work.zip). These files should be unarchived into the directory "lib\modelsim". All these work directories will be refreshed by the simulation compilation script before simulation can be run. The simulation script file is provided in the "eval\ simulation\rtl\ scripts" directory for RTL simulation. The script file run_vsim.bat (PC) uses precompiled models provided with this package. The parameter "MODELTECH" in the script file should be set to the appropriate MTI Modelsim version value for the local system. The simulation can be run by executing "scripts\run_vsim.bat" from the "eval\modelsim" directory. A successful simulation is achieved when the "test passed" message is displayed at the end of the simulation.

For more information on how to use ModelSim, please refer to the *ModelSim User's Manual.*

Note that the pre-compiled ORSPI4 simulation models provided in this IP evaluation package do not work with the OEM version of ModelSim embedded in the ispLEVER® 3.0 software. The full, licensed version of ModelSim is required to run this simulation.

Core Implementation

Lattice's Quad SPI-3 to SPI-4 PHY Layer Bridge evaluation package includes a Quad SPI-3 to SPI-4 PHY Layer Bridge user application and scripts for synthesizing, mapping and routing the Quad SPI-3 to SPI-4 PHY Layer Bridge IP solution.

The Quad SPI-3 to SPI-4 PHY Layer Bridge evaluation package includes the following components:

- Basic Quad SPI-3 to SPI-4 PHY Layer Bridge IP core;
- Verilog module that instantiates the ORSPI4 component and the ORCA4 SYSBUS with User Slave component, providing a Motorola Power PC interface to the IP core's register interface, as well as registers in the ORSPI4 embedded core;

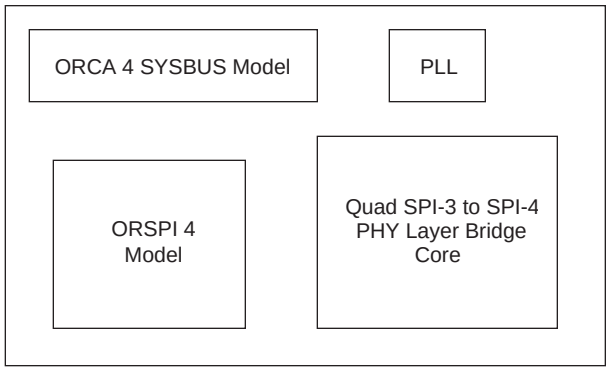
This evaluation package is illustrated in Figure 13. The following Verilog files are provided:

- qspi3_define.v for the Quad SPI-3 to SPI-4 PHY Layer Bridge core and top level parameters (Note: this file and all IP parameter files must not be modified in any way. If this file is modified, this IP core may not run at specification);
- spi_324p_o4_1_001.v for the Quad SPI-3 to SPI-4 PHY Layer Bridge core;
- mycore.v for the ORSPI4 module;
- sysbus_fpsc.v for the SYSBUS module;
- rfclk_hpll_ph_bo.v for PLL;
- rfclk_pll_ph_bo.v for PLL;
- orspi.v for "orspi_qspi3" top-level module that ties all the application components together.

Note that the Quad SPI-3 to SPI-4 PHY Layer Bridge Core is delivered as a gate-level netlist (spi_324p_o4_1_001.ngo). Users can compile the entire design shown in Figure 13 to realize a turnkey solution, or instantiate the Quad SPI-3 to SPI-4 PHY Layer Bridge Core as a block box together with any of the other blocks shown and/or their own designs, to realize a unique system-level project solution. Users may use orspi.v as a template for their own application.

Implementing a design in an ORSPI4 device requires the ispLEVER software and an ORSPI4 FPSC Design Kit. For more information, please contact your local Lattice sales representative or visit the Lattice website at www.latticesemi.com.

Figure 13. Quad SPI-3 to SPI-4 PHY Layer Bridge Top-Level Application



Black Box Consideration

Since the core is delivered as a gate-level netlist, the synthesis software will not re-synthesize the internal nets of the core. For more information regarding Synplify's black box declaration, please refer to the Instantiating Black Boxes in Verilog section of the Synplify reference manual.

Synthesis

The following sections provide procedures for synthesizing the Quad SPI-3 to SPI-4 PHY Layer Bridge Core IP solution with the Synplicity Synplify and LeonardoSpectrum synthesis tools, which are included in the ispLEVER software. These procedures generate an EDIF netlist containing the Quad SPI-3 to SPI-4 PHY Layer Bridge core as a black box.

Synthesis using Synplicity's Synplify

To synthesize the Quad SPI-3 to SPI-4 PHY Layer Bridge solution Synplicity's Synplify in one step, go to the directory "eval\synthesis\synplicity\" and enter "run_syn.bat". A top-level EDIF for the application will be produced in the rev_1 directory. Users may use run_syn.bat as a guide and template if they are creating their own unique system-level project solution.

The following step-by-step procedure may also be executed. Note that the step-by step flow results vary from those obtained with the scripted flow due to possible small differences in options between both flows.

1. Create a new working directory for synthesis.
2. Launch the Synplify synthesis tool.
3. Start a new project and add the specified files in the following order:
   eval\source\top\qspi3_define.v
   eval\source\synplicity\orca4_synplify.v
   eval\source\synplicity\rfclk_hpll_ph_bo.v
   eval\source\synplicity\rfclk_pll_ph_bo.v
   eval\source\synplicity\sysbus_fpsc.v
   eval\source\synplicity\mycore.v
   eval\source\top\spi_324p_o4_1_001.v
   eval\source\top\orspi.v
4. In the Implementation Options select the ORCA series 4 technology, the O4E06 part, speed grade -3 and package BA352. Note that these options are acceptable since synthesis target the series 4 based FPGA array of the device.
5. Specify an EDIF netlist filename and EDIF netlist output location in the Implementation Options. This top-level EDIF netlist will be used during Place and Route.

6. In the Implementation Options, set the following:
   • Fanout guide: 1000
   • Enable FSM Compiler
   • Enable Resource Sharing
   • Set the global frequency constraint to 104MHz.

7. Select run.

Synthesis using LeonardoSpectrum
To synthesize the Quad SPI-3 to SPI-4 PHY Layer Bridge solution using LeonardoSpectrum in one step, go to the directory "eval\synthesis\exemplar\" and enter "run_syn.bat". A top-level EDIF for the application will be produced. Users may use run_syn.bat as a guide and template if they are creating their own unique system-level project solution.

The following step-by-step procedure may also be executed. Note that the step-by step flow results vary from those obtained with the scripted flow due to possible small differences in options between both flows.

The step-by-step procedure provided below describes how to run synthesis using LeonardoSpectrum.

1. Create a new working directory for synthesis.

2. Launch the LeonardoSpectrum synthesis tool.

3. Start a new project and select Lattice device technology ORCA-4E.

4. Select Input tab, set the Working Directory path pointed to the source directory.

5. Open the specified files in the following order:
   eval\source\top\qspi3_define.v
   eval\source\exemplar\orca4_leonardo.v
   eval\source\exemplar\rfclk_hpll_ph_bo.v
   eval\source\exemplar\rfclk_pll_ph_bo.v
   eval\source\exemplar\sysbus_fpsc.v
   eval\source\exemplar\mycore.v
   eval\source\top\spi_324p_o4_1_001.v
   eval\source\top\orspi.v

6. Select orspi,v, use the right click button on your mouse, and choose from the list "Make orspi.v Top of the Design"

7. In the Constraints tab, set Clock Frequency as 104MHz.

8. Set the synthesis directory, created in step 1, as the path where you would like to save the output netlist.

9. Specify an EDIF netlist filename for the output file. This top-level EDIF netlist will be used during Place and Route.

10. Select Run Flow

Place and Route

Once the EDIF netlist is generated, the next step is to map, place and route the design.

The step-by-step procedure provided below explains how to run an EDIF based flow through place and route using ispLEVER Project Navigator

Once the EDIF netlist is generated, import the EDIF into the Project Navigator. The ispLEVER software automatically detects the provided EDIF netlist of the instantiated IP core in the design. The step-by-step procedure provided below describes how to perform Place and Route in ispLEVER for an ORCA device:

1. Create a new working directory for Place and Route.
2. Start a new project, assign a project name and select the project type as EDIF.
3. Select the ORSPI4 (or ORSPII if that is the option available) target device, with -3 speed grade and BS1036 package.
4. Copy the following files to the Place and Route working directory:
   - eval\ngo\spi_324p_o4_1_001.ngo
   - eval\prf\exemplar\orspi_qspi3.prf (with the LeonardoSpectrum EDIF)
   - eval\prf \synplicity\ orspi_qspi3.prf (with the Synplify EDIF)
   - The top-level EDIF netlist generated from running synthesis
5. Rename the orspi_qspi3.prf file (in step 4) to match the project name. For example, if the project name is "demo", then the .prf file must be renamed to demo.prf. The preference file name must match that of the project name.
6. Import the EDIF netlist into the project.
7. In the ispLEVER Project Navigator, select Tools->Timing Checkpoint Options. The Timing Checkpoint Options window will pop-up. In both Checkpoint Options, select Continue.
8. In the ispLEVER Project Navigator, highlight Place and Route Design, with a right mouse click select Properties. Set the following Properties:
   - Placement Iterations: 1
   - Placement Save Best Run: 1
   - Placement Iteration Start Point: 4
   - Routing Resource Optimization: 1
   - Routing Delay Reduction Passes: 1 for Synplify EDIF, 6 for Leonardo EDIF
   - Routing Passes: 10
   - Placement Effort Level: 5

All other options remain at their default values.

9. Select the Place and Route Trace Report in the project navigator to execute Place and Route and generate a timing report for ORCA.
10. Highlight Place and Route TRACE Report, with a right mouse click and select Force One Level. A new timing report is generated.

References

The SPI-3 PHY interfaces in the Quad SPI-3 to SPI-4 Bridge IP core solution are compliant with the standard OIF-SPI3-01.0. A complete description of this standard is given in the specification document.

- Optical Internetworking Forum (OIF). System Packet Interface Level 3 (SPI-3): OC-48 System Interface for Physical and Link Layer Devices. OIF-SPI3-01.0.

A complete description of the SPI-4 standard is given in the specification document

- Optical Internetworking Forum (OIF). System Packet Interface Level 4 (SPI-4) Phase 2: OC-192 System Interface for Physical and Link Layer Devices. OIF-SPI4-02.0

Additional information on implementing this solution is contained in the following documents:

- ORCA ORSPI4 FPSC Data Sheet
- ORCA Series 4 FPGAs Data Sheet
- Lattice Technical Note TN1017 *ORCA Series 4 MPI/System Bus*

These documents are available on the Lattice web site at www.latticesemi.com.

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Appendix for ORCA Series 4 ORSPI4 FPSC

Table 13. Performance and Utilization[1]

Configuration	PFUs	Block RAM	PLL	LUTs	f_{MAX}	Parameters			
						Interfaces	Ports	Burst Size	Polling
001	1706	16	4	7631	104MHz SPI-3 ref clk	4	4	256	DTPA

1. Performance and utilization characteristics are generated using an ORSPI4-2FE1036C in Lattice's ispLEVER 4.0 software. When using this IP core in a different density, package, speed, or grade within the ORCA 4 family, performance may vary.

Supplied Netlist Configurations

The Ordering Part Number (OPN) for all configurations of this core is SPI-324P-O4-N1.

You can use the IPexpress™ software tool to help generate new configurations of this IP core. IPexpress is the Lattice IP configuration utility, and is included as a standard feature of the ispLEVER design tools. Details regarding the usage of IPexpress can be found in the IPexpress and ispLEVER help system. For more information on the ispLEVER design tools, visit the Lattice web site at: www.latticesemi.com/software.