CSIX Level 1 IP Core

User's Guide

October 2005
ipug16_02.0

Introduction

Lattice's CSIX Level 1 core provides an ideal solution that meets the needs of today's CSIX applications. The CSIX Level 1 core provides a customizable solution allowing CSIX interfacing in many design applications. This core allows designers to focus on the application rather than the CSIX interface, resulting in a faster time to market.

This User's Guide explains the functionality of the CSIX Level 1 core and how it can be implemented to provide CSIX interfacing for any application. It also explains how to achieve the maximum level of performance.

The CSIX Level 1 core comes with the documents and files listed below:

- Data sheet
- Encrypted gate level netlist
- Secured RTL simulation model
- Core instantiation template

Features

- Implements a CSIX-L1 to Generic FIFO Bridge
- Supports 32-bit, 100MHz CSIX-L1 Interface
- Up to Four Parameterizable 32-bit channel instantiations
- Parameterizable channel aggregation (32-bit to 128-bit)
- Parameterizable FIFO size (up to 2,048 bytes)
- Programmable FIFO thresholds
- Supports MAX_FRAME_PAYLOAD_SIZE from 1 to 256 bytes
- Transports unicast, multicast, broadcast, and flow control frames
- Filters idle frames (not transported through core)
- Delineates Cframes at Generic FIFO Bridge interface with Start/End Flags (SOF and EOF)
- Passes entire CSIX frame structure across Generic FIFO Bridge interface
- Passes CSIX link level control directly to Generic FIFO Bridge (bypasses FIFOs)
- Supports CSIX-L1 link-layer flow control (XON/XOFF)
- Programmable horizontal and vertical parity check enables
- Internal register set for control and status management
- 8-bit register interface compatible with ORCA System Bus

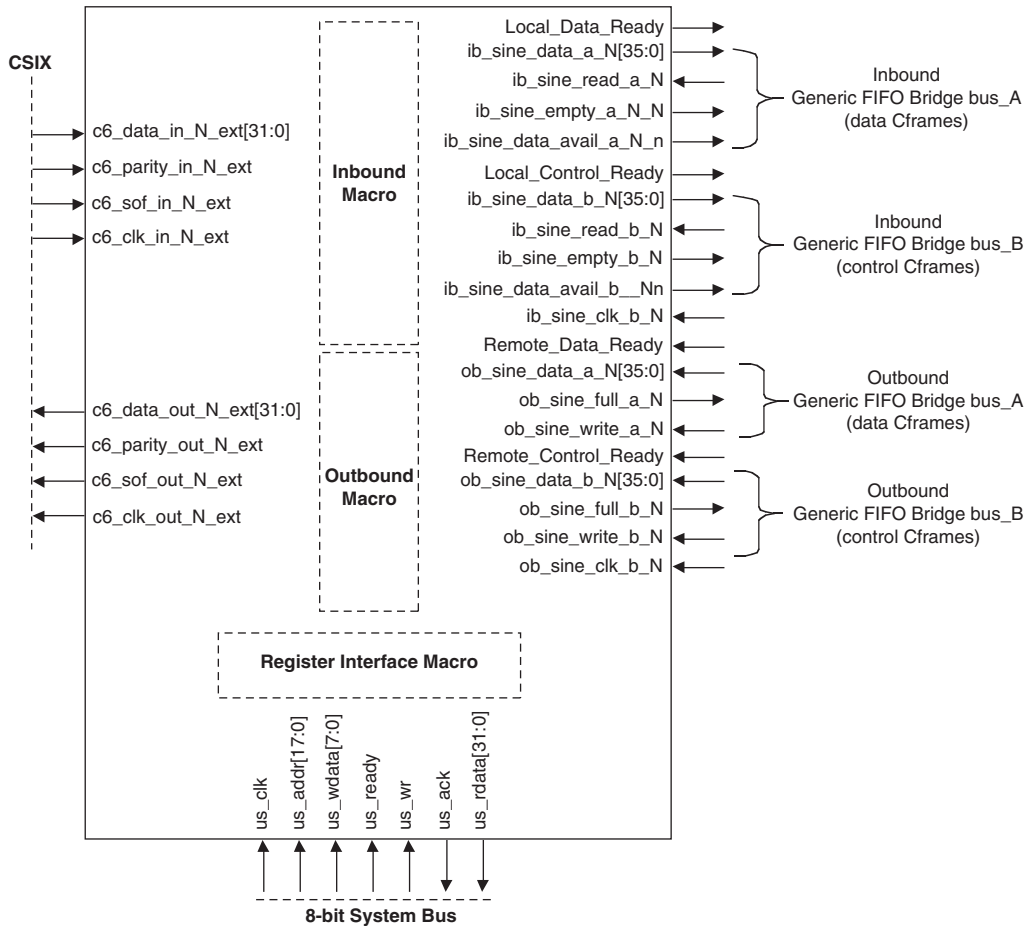
General Description

As stated by the CSIX Forum, the CSIX standard defines the physical and message layers of the interconnect between traffic managers (TM) and the switching fabric. The CSIX interface is designed to support a wide variety of system architectures and markets, providing a framework with a common set of mechanisms for enabling a fabric and a TM to communicate. This includes unicast addressing for up to 4,096 fabric ports, and multiple traffic classes that isolate data going to the same fabric port. Link level flow control is in-band and broken into a data and a control queue to isolate traffic based on this granular type. Flow control between the fabric and TM is defined and is relative to both fabric port and class. Three multicast approaches are defined. The interface assumes cell segmentation in the TM, but allows compression of the transfer.

Lattice's CSIX Level 1 core links a compliant CSIX-L1 interface to Lattice's Generic FIFO Bridge interface (a simple FIFO interface). Inbound control and data frames from the CSIX port are deposited into the core's inbound FIFOs. CSIX frames stored in the core's outbound FIFOs are driven onto the outbound CSIX interface. The Generic FIFO Bridge interface directly accesses the core's inbound and outbound FIFOs.

Block Diagram

Figure 1. Simplified Block Diagram - CSIX Level 1 Core



Parameter Descriptions

The list of parameters used for configuring the CSIX Level 1 core is listed below. The values of these parameters are to be set and must be done prior to synthesis or functional verification.

Table 1. User Configurable Parameters

Number	Parameter	Description	Choice	Default
1	NUM_OF_CHANNELS	Number of IP 32-bit channel instantiations.	1,2,3,4	1
2	FIFO_SIZE	FIFO storage capacity. Note that maximum FIFO size is dependent on the number of channel instantiations. For number of channels = 1, FIFO max. size = 2,048 For number of channels > 1, FIFO max. size = 1,024	1,024 or 2,048 bytes	1,024 bytes
3	AGGREGATION	Specifies whether aggregation is active.	yes, no	no
4	AGG_SPAN_0	Specifies number of channels combined to form the first aggregate group.	0,2,3,4	0
5	AGG_SPAN_1	Specifies number of channels combined to form the second aggregate group.	0,2	0
6	BUFFER_TYPE	Buffer type for CSIX interface primary I/O.	LVCMOS, HSTL	LVCMOS

Signal Descriptions

Table 2 defines all I/O interface ports available in this core.

Table 2. Signal Definitions of CSIX Level 1 Core

Signal Name[1]	Direction	Width (Bits)	Description
CSIX Interface (These signals are FPGA primary I/Os)			
c6_data_in_N_ext[31:0]	Input	32	Inbound Data
c6_parity_in_N_ext	Input	1	Inbound Parity (Odd)
c6_sof_in_N_ext	Input	1	Inbound Start of Frame
c6_clk_in_N_ext	Input	1	Inbound Clock (100MHz)
c6_data_out_N_ext[31:0]	Output	32	Outbound Data
c6_parity_out_N_ext	Output	1	Outbound Parity (odd)
c6_sof_out_N_ext	Output	1	Outbound Start of Frame
c6_clk_out_N_ext	Output	1	Outbound Clock (100MHz)
Generic FIFO Bridge Interface (These signals are buried inside the FPGA)			
ib_sine_clk_a_N	Input	1	Inbound Clock, BUS A (100MHz)
ib_sine_data_a_N[35:0]	Output	36	Inbound Data, BUS A
ib_sine_read_a_N	Input	1	Inbound Active High Read Enable, Bus A
ib_sine_empty_a_N	Output	1	Inbound Active High Empty Flag, Bus A
ib_sine_data_avail_a_N_n	Output	1	Inbound Active Low Data Available, Bus A
ib_sine_clk_b_N	Input	1	Inbound Clock, Bus B (100MHz)
ib_sine_data_b_N[35:0]	Output	36	Inbound Data, Bus B
ib_sine_read_b_N	Input	1	Inbound Active High Read Enable, Bus B
ib_sine_empty_b_N	Output	1	Inbound Active High Empty Flag, Bus B
ib_sine_data_avail_b_N_n	Output	1	Inbound Active Low Data Available, Bus B
ob_sine_clk_a_N	Input	1	Outbound Clock, Bus A (100MHz)
ob_sine_data_a_N[35:0]	Input	36	Outbound Data, Bus A
ob_sine_write_a_N	Input	1	Outbound Active High Write Enable, Bus A
ob_sine_full_a_N	Output	1	Outbound Active High Full Flag, Bus A
ob_sine_clk_b_N	Input	1	Outbound Clock, Bus B (100MHz)
ob_sine_data_b_N[35:0]	Input	36	Outbound Data, Bus B
ob_sine_write_b_N	Input	1	Outbound Active High Write Enable, Bus B
ob_sine_full_b_N	Output	1	Outbound Active High Full Flag, Bus B

Table 2. Signal Definitions of CSIX Level 1 Core (Continued)

Signal Name[1]	Direction	Width (Bits)	Description
Register Interface (These signals are buried inside the FPGA – do not replicate)			
us_clk	Input	1	Clock (50MHz)
us_addr[17:0]	Input	18	Address bits
us_wdata[7:0]	Input	8	Input Data Bus
us_rdata[31:0]	Output	32	Output Data Bus, HiZ except during read cycle and us_addr matches one of the implemented register addresses. Note the core only implements 8-bit registers. For read accesses, byte wide register outputs are copied to all four bytes of the sysbus interface.
us_ready	Input	1	Bus Cycle Enable (Active Low)
us_wr	Input	1	Write High/ Read Low
us_ack	Output	1	Bus Cycle Acknowledge
Global (These signals are buried inside the FPGA – do not replicate)			
rst_n	input	1	Active Low Global Reset

1. N denotes instantiated channel number (0, 1, 2, 3).

Functional Description

Inbound Path

The inbound path monitors the inbound CSIX port and writes valid Cframes into control and data FIFOs. User applications may access the stored Cframes via the two inbound Generic FIFO Bridge interfaces. Major functions of inbound path include: distinguishing control and data Cframes, directing Cframe writes to the appropriate FIFO, filtering idle Cframes from the FIFOs, delineating stored Cframes with end-of-frame flags, checking horizontal and vertical parity, and facilitating link-level flow control and backpressure.

The inbound core utilizes two FIFOs; one for data Cframes, the other for control Cframes. Each FIFO has programmable high and low watermarks. When the low watermark is asserted, the Generic FIFO Bridge bus is signaled that a frame is available for reading. When the high watermark is asserted, a "nearly-full" signal is sent to the IP core's outbound path for flow control. In cases where a FIFO either underflows or overflows, a corresponding error signal is sent to the register interface for storage.

An inbound write controller monitors the external CSIX port and writes CSIX data frames into one FIFO, and CSIX control frames into another FIFO.

Horizontal and vertical parity are checked within the inbound data path. The checkers can be enabled via control signals from the register interface. Any detected errors cause a corresponding error signal to be sent to the register interface for storage.

There are three flow control mechanisms associated with the inbound path. The first mechanism passes ready bits from the CSIX interface to the Local_Data_Ready or Local_Control_Ready signals alongside the Generic FIFO Bridge interface. These signals are directly connected to the CSIX ready bits at the second stage of the inbound data pipeline, and therefore lag the external CSIX interface by two clock cycles. The local ready signals give user applications direct access to the CSIX ready status without having to wait for the ready status to percolate through the FIFOs.

The second mechanism is associated with the inbound FIFOs. If either FIFO passes its high watermark, a "nearly-full" signal is passed to the outbound datapath. In turn, the outbound datapath will deassert the ready bits on the next outbound frame, signaling a pause to the transmitting inbound network element. This mechanism prevents the inbound FIFOs from overflowing.

The third mechanism is associated with the outbound FIFOs. If either outbound FIFO exceeds its high watermark, a "nearly-full" signal is sent to the inbound datapath. In turn, the inbound datapath deasserts the associated local

ready signal. The deassertion of the local ready signal should signal the user application to pause outbound dataflows, thereby preventing the outbound FIFOs from overflowing.

The inbound Generic FIFO Bridge interface employs two Generic FIFO Bridge buses so that user applications can process the control and data streams independently. The inbound Generic FIFO Bridge bus implements the following signals: 36-bit databus, data available, read enable, empty flag, and clock. The data available signal asserts when the associated FIFO's low watermark is exceeded. When the read enable is asserted, dataflow begins one clock later. When the read enable is deasserted, dataflow halts one clock later. The empty flag asserts when the FIFO occupancy is zero. The clock signal can be any frequency up to 100MHz. However, keep in mind that the CSIX interface operates at 100MHz. Therefore, if the Generic FIFO Bridge interface operates more slowly than proper system FIFO depth, watermark settings, and flow control mechanisms must all function appropriately to prevent the core's FIFOs from overflowing.

Outbound Path

The outbound path monitors the internal FIFOs and passes all available Cframes to the external CSIX interface. User applications must load valid Cframes into the outbound FIFOs via the two outbound Generic FIFO Bridge interfaces. Major functions of the outbound path include: obeying the CSIX link startup process, continuously checking frame availability from control and data FIFOs in round robin fashion, unloading internal FIFOs and driving the external CSIX interface, and facilitating link-level flow control and backpressure.

As stated in the Network Processor Forum's CSIX-L1 Specification, version 1.0, dated 8/5/2000, the outbound CSIX interface must follow a prescribed startup process. Three states are defined, "reset", "transmit_idle", and "normal". After exiting a reset state, the outbound CSIX interface must enter the "transmit_idle" state and begin transmitting idle frames with ready bits deasserted. If the inbound interface detects idle frames (ready bit states irrelevant) then the outbound path must send idle frames with the ready bits asserted. The outbound interface assumes a "normal" state only after the inbound path receives valid idle frames with the ready bits asserted high.

The core's startup FSM follows a modified form of this process. While in the "transmit_idle" state, the core will only send idle frames with ready bits asserted after both receiving idle frames on the inbound CSIX interface and receiving asserted remote control/data signals from the outbound Generic FIFO Bridge interface.

The outbound Generic FIFO Bridge interface is similar to the inbound path except that the direction of dataflow is reversed. The following signals are implemented: 36-bit databus, write enable, full flag, and clock. FIFO data writes occur coincident with write enable assertion. The EOF flag should be asserted during the penultimate word as shown in Figure 8. The full flag is asserted when the FIFO occupancy is almost full (as defined by the high watermark for the FIFO). The clock can be any frequency up to 100MHz. However, at lower frequencies, the average CSIX throughput rate is limited by the Generic FIFO Bridge clock rate.

The outbound FIFOs are similar to those used in the inbound path. An outbound transmit controller reads outbound FIFOs and determines which type of frame appears on the external CSIX interface.

There are four flow control mechanisms associated with the outbound path. The first mechanism responds to control or data ready-bit deassertions from the inbound CSIX interface. If either of these bits is deasserted, the corresponding outbound FIFO is prohibited from further reads after the current Cframe ends. This effects a CSIX "pause" and remains active as long as the inbound ready bit is deasserted.

The second flow control mechanism is associated with the remote data/control ready signals that enter the outbound path along with the Generic FIFO Bridge interfaces. These signals indicate the CSIX link ready statuses from remotely connected CSIX entities. For example, a traffic manager and a fabric device interconnected by two back-to-back CSIX IP cores. The remote ready signals connect directly to the outgoing ready bits of the CSIX interface. A deasserted remote ready signal causes the external CSIX network element to pause transmission on the inbound data path, thereby easing the congestion being experienced by the remote device.

The third flow control mechanism is associated with the inbound FIFOs. If either of the inbound "nearly-full" signals are asserted, the outgoing ready bits of the CSIX interface are deasserted. This in-turn causes the external CSIX

network element to pause transmission on the inbound data path, thereby preventing the inbound FIFO from overflowing.

The fourth flow control mechanism is associated with the outbound FIFOs. If either of the associated "near-full" signals is asserted, the inbound path deasserts the local control/data ready signal. This should signal the user application or remote CSIX element to pause transmission on the outbound path, thereby preventing the outbound FIFOs from overflowing.

Port Aggregation

The core supports port aggregation. If an application instantiates more than one 32-bit CSIX core, two or more cores can be aggregated into a larger CSIX interface. For example, two cores can be aggregated to operate as a 64-bit CSIX interface, three cores can operate as a 96-bit interface, and four cores can operate as a 128-bit interface. Proper operation of the port aggregation mode requires that the external CSIX ports conform to the clock domain skew limits listed in the CSIX-L1 Specification. Also, the internal user application for the CSIX core must use a single timing domain to clock the aggregated cores.

Three user parameters control the configuration of port aggregation:

1. Port_Aggregation (yes, no)
2. Aggregation_Span_0 (2,3,4)
3. Aggregation_Span_1 (0,2)

If port aggregation is to be instantiated, then the first parameter must be yes. If aggregation is instantiated, then the last two parameters must be specified. Typically, only one aggregation function will be specified (Span_0). However, if four cores are instantiated for an application, it is possible to create two 64-bit CSIX interfaces. In this case Span_0 = 2, and Span_1 = 2. The highest numbered core is always the most significant 32-bit group of the aggregate. The other cores decrease in precedence, down to core 0 as the least significant group.

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 as shown in Figure 1. When a user instantiates an ORCA® SYSBUS slave interface to control the core registers, the external FPGA control interface is compatible with a Motorola MPC860 Power PC interface. The description below is specific to this IP core. Therefore, the bus widths may not match the generic bus widths shown in Figure 1.

The core maintains an 8-bit implementation of the system bus. A 9-bit address bus (us_addr[8:0]) specifies the locations of the registers (0x000 - 0x1FF). An active low `us_ready` signal enables a register access cycle. An `us_wr` signal enables writing when high, reading when low. Data to be written to registers appears on the 8-bit bus "us_wdata[7:0]". Data read from registers appears on the 8-bit bus "us_rdata[7:0]", and is driven only during access to one of the implemented registers. The `us_clk` signal synchronizes register accesses and can be any frequency up to 50MHz. The active high `us_ack` signal asserts when the core is ready to end the current bus cycle.

Register Descriptions

Table 3. Register Map

Register Name	Register Address	Description
Global Control Register	0x8000	Holds global control functions
Reset Control Reg	0x8004	Resets logic for IP Channels [3:0]
FIFO Flush Control Reg	0x8008	Flushes all FIFOs for IP Channels [3:0]
Force H Parity Error Reg	0x800C	Force Horizontal Par Errs for Channels[3:0]
Force V Parity Error Reg	0x8010	Force Vertical Par Errs for Channels[3:0]
Reserved	0x8014 - 0x801c	Reserved
In Data Lo Watermark 0	0x8040	Low watermark for the inbound data FIFO, Channel 0
In Cntrl Lo Watermark 0	0x8044	Low watermark for the inbound cntrl FIFO, Channel 0
In Data Hi Watermark 0	0x8048	High watermark for the inbound data FIFO, Channel 0
In Cntrl Hi Watermark 0	0x804C	High watermark for the inbound cntrl FIFO, Channel 0
Out Data Lo Watermark 0	0x8050	Low watermark for the outbound data FIFO, Channel 0
Out Cntrl Lo Watermark 0	0x8054	Low watermark for the outbound cntrl FIFO, Channel 0
Out Data Hi Watermark 0	0x8058	High watermark for the outbound data FIFO, Channel 0
Out Cntrl Hi Watermark 0	0x805C	High watermark for the outbound cntrl FIFO, Channel 0
In Data Lo Watermark 1	0x8080	Low watermark for the inbound data FIFO, Channel 1
In Cntrl Lo Watermark 1	0x8084	Low watermark for the inbound cntrl FIFO, Channel 1
In Data Hi Watermark 1	0x8088	High watermark for the inbound data FIFO, Channel 1
In Cntrl Hi Watermark 1	0x808C	High watermark for the inbound cntrl FIFO, Channel 1
Out Data Lo Watermark 1	0x8090	Low watermark for the outbound data FIFO, Channel 1
Out Cntrl Lo Watermark 1	0x8094	Low watermark for the outbound cntrl FIFO, Channel 1
Out Data Hi Watermark 1	0x8098	High watermark for the outbound data FIFO, Channel 1
Out Cntrl Hi Watermark 1	0x809C	High watermark for the outbound cntrl FIFO, Channel 1
In Data Lo Watermark 2	0x80C0	Low watermark for the inbound data FIFO, Channel 2
In Cntrl Lo Watermark 2	0x80C4	Low watermark for the inbound cntrl FIFO, Channel 2
In Data Hi Watermark 2	0x80C8	High watermark for the inbound data FIFO, Channel 2
In Cntrl Hi Watermark 2	0x80CC	High watermark for the inbound cntrl FIFO, Channel 2
Out Data Lo Watermark 2	0x80D0	Low watermark for the outbound data FIFO, Channel 2
Out Cntrl Lo Watermark 2	0x80D4	Low watermark for the outbound cntrl FIFO, Channel 2
Out Data Hi Watermark 2	0x80D8	High watermark for the outbound data FIFO, Channel 2
Out Cntrl Hi Watermark 2	0x80DC	High watermark for the outbound cntrl FIFO, Channel 2
In Data Lo Watermark 3	0x8100	Low watermark for the inbound data FIFO, Channel 3
In Cntrl Lo Watermark 3	0x8104	Low watermark for the inbound cntrl FIFO, Channel 3
In Data Hi Watermark 3	0x8108	High watermark for the inbound data FIFO, Channel 3
In Cntrl Hi Watermark 3	0x810C	High watermark for the inbound cntrl FIFO, Channel 3
Out Data Lo Watermark 3	0x8110	Low watermark for the outbound data FIFO, Channel 3
Out Cntrl Lo Watermark 3	0x8114	Low watermark for the outbound cntrl FIFO, Channel 3
Out Data Hi Watermark 3	0x8118	High watermark for the outbound data FIFO, Channel 3
Out Cntrl Hi Watermark 3	0x811C	High watermark for the outbound cntrl FIFO, Channel 3
FIFO Underflow Errors	0x8140	FIFO underflow error status for Channels[3:0]
FIFO Overflow Errors	0x8144	FIFO overflow error status for Channels[3:0]
Parity Errors	0x8148	Horizontal and vertical parity errors for Channels[3:0]
Miscellaneous Errors	0x814C	Receiver synchronization error

Detailed Register Descriptions

Address: **0x8000**		Name: **Global Control Register**					
D7	D6	D5	D4	D3	D2	D1	D0
—	—	—	—	Reset Enable	H_Parity Enable	V_Parity Enable	Transmission Enable
Default value: 0x01				Mode: Read/Write			
Description:							
Transmission_Enable	When high, CSIX and SINE transmission logic is enabled to operate for all instantiated channels.						
V_Parity_Enable	When high, vertical parity error detectors on inbound CSIX ports are enabled to operate for all instantiated channels.						
H_Parity _Enable	When high, horizontal parity error detectors on inbound CSIX ports are enabled to operate for all instantiated channels.						
Reset_Enable	When high, this bit initializes all registers and resets all transmission logic. Note that this reset function is self clearing.						

Address: **0x8004**		Name: **Reset Control Register**					
D7	D6	D5	D4	D3	D2	D1	D0
—	—	—	—	Reset_3	Reset_2	Reset_1	Reset_0
Default value: 0x00				Mode: Read/Write			
Description:							
Reset_N	When high, transmission logic for channel N is reset. Note that this bit is not self clearing. The bit must be written to 0 to deassert the associated reset.						

Address: **0x8008**		Name: **FIFO Flush Register**					
D7	D6	D5	D4	D3	D2	D1	D0
—	—	—	—	Flush_3	Flush_2	Flush_1	Flush_0
Default value: 0x00				Mode: Read/Write			
Description:							
Flush_N	When high, all FIFOs for channel N are flushed. Note that this bit is not self clearing. The bit must be written to zero to deassert the associated flush.						

Address: **0x800C**		Name: **Force H Parity Error Register**					
D7	D6	D5	D4	D3	D2	D1	D0
—	—	—	—	Force HPERR_3	Force HPERR_2	Force HPERR_1	Force HPERR_0
Default value: 0x00				Mode: Read/Write			
Description:							
Force_HPERR_N	When high, the horizontal parity checker on channel N is forced to detect a parity error. Note this bit is not self clearing. The bit must be written to zero to deassert the associated error condition.						

Detailed Register Descriptions (Continued)

Address: **0x8010**		Name: **Force V Parity Error Register**					
D7	D6	D5	D4	D3	D2	D1	D0
—	—	—	—	Force VPERR_3	Force VPERR_2	Force VPERR_1	Force VPERR_0
Default value: 0x00				Mode: Read/Write			
Description:							
Force_VPERR_N		When high, the vertical parity checker on channel N is forced to detect a parity error. Note this bit is not self clearing. The bit must be written to zero to deassert the associated error condition.					

Address: **0x8040**		Name: **In_Data_Lo_Watermark_0**					
D7	D6	D5	D4	D3	D2	D1	D0
IBD_LOW_WMARK_0							
Default value: 0x08				Mode: Read/Write			
Description:							
IBD_LOW_WMARK_0		Channel 0, low watermark for inbound data FIFO. This value represents the number of FIFO words (4 bytes) that assert the FIFO low watermark. When the number of words in the FIFO exceeds this value, the FIFO's frame available signal asserts.					

Address: **0x8044**		Name: **In_Control_Lo_Watermark_0**					
D7	D6	D5	D4	D3	D2	D1	D0
IBC_LOW_WMARK_0							
Default value: 0x08				Mode: Read/Write			
Description:							
IBC_LOW_WMARK_0		Channel 0, low watermark for inbound control FIFO. This value represents the number of FIFO words (4 bytes) that assert the FIFO low watermark. When the number of words in the FIFO exceeds this value, the FIFO's frame available signal asserts.					

Address: **0x8048**		Name: **In_Data_Hi_Watermark_0**					
D7	D6	D5	D4	D3	D2	D1	D0
IBD_HI_WMARK_0							
Default value: 0x3f				Mode: Read/Write			
Description:							
IBD_HI_WMARK_0		Channel 0, high watermark for inbound data FIFO. This value represents the number of FIFO words (4 bytes) that assert the FIFO high watermark. When the number of words in the FIFO exceeds this value, the FIFO's PARTIAL_FULL signal asserts.					

Detailed Register Descriptions (Continued)

Address: **0x804C**		Name: **In_Cntrl_Hi_Watermark_0**					
D7	D6	D5	D4	D3	D2	D1	D0
IBC_HI_WMARK_0							
Default value: 0x3f				Mode: Read/Write			
Description:							
IBC_HI_WMARK_0	Channel 0, high watermark for inbound control FIFO. This value represents the number of FIFO words (4 bytes) that assert the FIFO high watermark. When the number of words in the FIFO exceeds this value, the FIFO's PARTIAL_FULL signal asserts.						

Address: **0x8050**		Name: **Out_Data_Lo_Watermark_0**					
D7	D6	D5	D4	D3	D2	D1	D0
OBD_LOW_WMARK_0							
Default value: 0x08				Mode: Read/Write			
Description:							
OBD_LOW_WMARK_0	Channel 0, low watermark for outbound data FIFO. This value represents the number of FIFO words (4 bytes) that assert the FIFO low watermark. When the number of words in the FIFO exceeds this value, the FIFO's frame available signal asserts.						

Address: **0x8054**		Name: **Out_Cntrl_Lo_Watermark_0**					
D7	D6	D5	D4	D3	D2	D1	D0
OBC_LOW_WMARK_0							
Default value: 0x08				Mode: Read/Write			
Description:							
OBC_LOW_WMARK_0	Channel 0, low watermarked for outbound control FIFO. This value represents the number of FIFO words (4 bytes) that assert the FIFO low watermark. When the number of words in the FIFO exceeds this value, the FIFO's frame available signal asserts.						

Address: **0x8058**		Name: **Out_Data_Hi_Watermark_0**					
D7	D6	D5	D4	D3	D2	D1	D0
OBD_HI_WMARK_0							
Default value: 0x3f				Mode: Read/Write			
Description:							
OBD_HI_WMARK_0	Channel 0, high watermark for outbound data FIFO. This value represents the number of FIFO words (4 bytes) that assert the FIFO high watermark. When the number of words in the FIFO exceeds this value, the FIFO's PARTIAL_FULL signal asserts.						

Detailed Register Descriptions (Continued)

Address: **0x805C**		Name: **Out_Cntrl_Hi_Watermark_0**					
D7	D6	D5	D4	D3	D2	D1	D0
OBC_HI_WMARK_0							
Default value: 0x3f				Mode: Read/Write			
Description:							
OBC_HI_WMARK_0	Channel 0, high watermark for outbound control FIFO. This value represents the number of FIFO words (4 bytes) that assert the FIFO high watermark. When the number of words in the FIFO exceeds this value, the FIFO's PARTIAL_FULL signal asserts.						

Watermark registers for channels 1 through 3 reside at locations 0x8080 - 0x811C as shown in the register map. The detailed descriptions for these register are the same as shown for channel 0 except that the registers control their associated channel instantiation. For brevity, these register descriptions are not shown.

Note that when channel aggregation is used, the watermark settings of the channel with highest precedence control the behavior of all the aggregated FIFOs. The watermark settings of the lower order channels have no affect on FIFO operation.

Address: **0x8140**		Name: **FIFO Underflow Error Register**					
D7	D6	D5	D4	D3	D2	D1	D0
—	—	—	—	UF_ERR_3	UF_ERR_2	UF_ERR_1	UF_ERR_0
Default value: n/a				Mode: Clear on Read			
Description:							
UF_ERR_N	When high, indicates that one of the FIFOs for channel N experienced an underflow error. The associated channel's FIFOs should be flushed to guarantee proper operation after the underflow. This bit clears upon reading. If the stimulus that caused the error is removed, then the error bit in this register will clear after reading.						

Address: **0x8144**		Name: **FIFO Overflow Error Register**					
D7	D6	D5	D4	D3	D2	D1	D0
—	—	—	—	OF_ERR_3	OF_ERR_2	OOF_ERR_1	OF_ERR_0
Default value: n/a				Mode: Clear on Read			
Description:							
OF_ERR_N	When high, indicates that one of the FIFOs for channel N experienced an overflow error. The associated channel's FIFOs should be flushed to guarantee proper operation after the underflow. This bit clears upon reading. If the stimulus that caused the error is removed, then the error bit in this register will clear after reading.						

Detailed Register Descriptions (Continued)

Address: **0x8148**		Name: **Parity Error Register**					
D7	D6	D5	D4	D3	D2	D1	D0
VPERR_3	VPERR_2	VPERR_1	VPERR_0	HPERR_3	HPERR_2	HPERR_1	HPERR_0
Default value: n/a				Mode: Clear on Read			
Description:							
HPERR_N		When high, indicates that a horizontal parity error occurred on the inbound CSIX port for channel N. This bit clears upon reading. If the stimulus that caused the error is removed, then the error bit in this register will clear after reading.					
VPERR_N		When high, indicates that a vertical parity error occurred on the inbound CSIX port for channel N. This bit clears upon reading. If the stimulus that caused the error is removed, then the error bit in this register will clear after reading.					

Address: **0x814C**		Name: **Miscellaneous Error Register**					
D7	D6	D5	D4	D3	D2	D1	D0
—	—	—	—	—	—	SYNC_ERR_1	SYNC_ERR_0
Default value: n/a				Mode: Clear on Read			
Description:							
SYNC_ERR_0		This bit operates only when port aggregation is instantiated. When high, indicates that the channels associated with aggregation span 0 experienced a sync error on the inbound CSIX ports. This bit clears upon reading. If the stimulus that caused the error is removed, then the error bit in this register will clear after reading.					
SYNC_ERR_1		This bit operates only when aggregation span 1 is instantiated. When high, indicates that the channels associated with aggregation span 1 experienced a sync error on the inbound CSIX ports. This bit clears upon reading. If the stimulus that caused the error is removed, then the error bit in this register will clear after reading.					

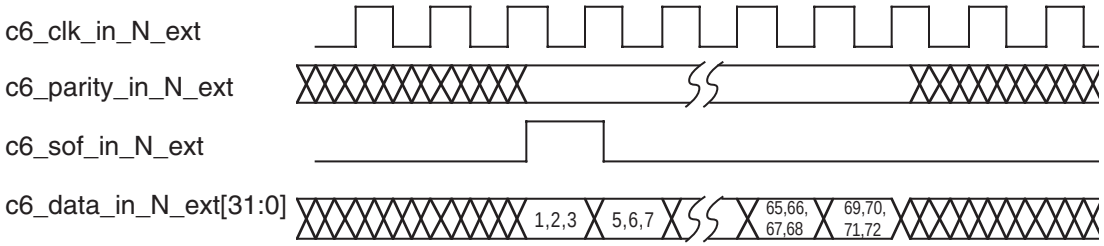
Interface Timing and Electrical Specifications

CSIX Ports

The core's CSIX ports are characterized to operate at 100MHz. Incoming signals are sampled on the rising edge of the CSIX input clock. Outgoing signals are clocked on the rising edge of the outgoing CSIX clock. The instantiated I/O buffers for the CSIX ports are compatible with LVCMOS and HSTL levels. Detailed timing and electrical specifications are shown in the paragraphs below.

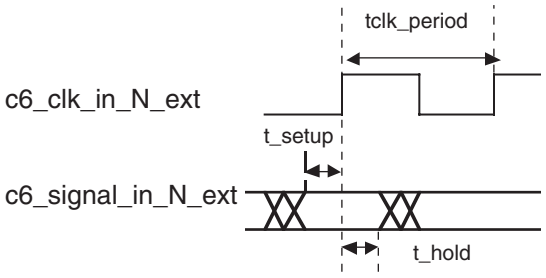
The following figure shows a 64-byte payload arriving at the inbound CSIX interface. Note that the frame is actually 72 bytes long with header and vertical parity added.

Figure 2. CSIX Inbound Frame Transfer



The AC timing specifications for the inbound CSIX port are as follows:

Figure 3. CSIX Inbound AC Timing Specifications



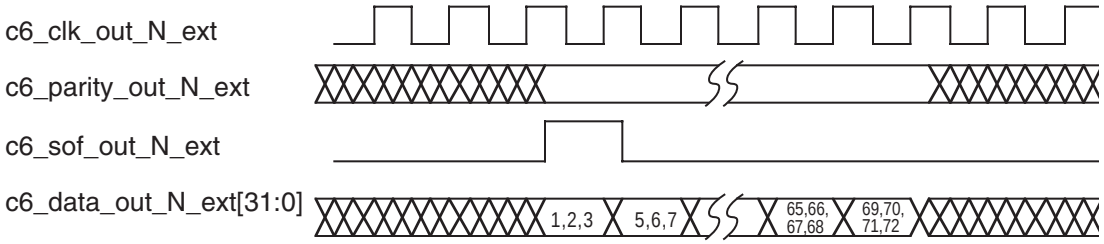
Name	Description	Min.	Max.
tclk_freq	Clock frequency	—	100MHz
tclk_period	Clock period	10ns	—
t_setup	Setup time to rising edge of clock	1.5ns	—
t_hold	Hold time to rising edge of clock	0ns	—

The DC electrical specifications for the inbound CSIX port are as follows:

Name	Description	Min.	Typ.	Max.
V_{DDIO}	I/O supply voltage	2.3V	2.5V	2.7V
V_{IH}	V_{IN} high threshold	2.0V	—	V_{DDIO} + 0.3V
V_{IL}	V_{IN} low threshold	-0.5V	—	0.8V

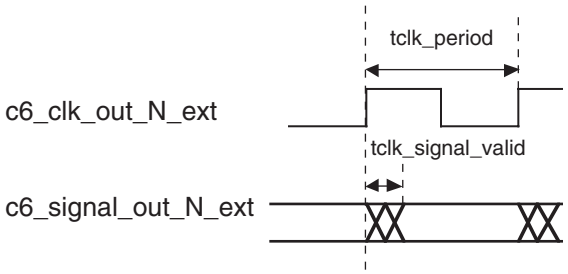
The following figure shows a 64-byte payload leaving the outbound CSIX interface. Note that the frame is actually 72 bytes long with header and vertical parity added.

Figure 4. CSIX Outbound Frame Transfer



The AC timing specifications for the outbound CSIX port are as follows:

Figure 5. CSIX Outbound AC Timing Specifications



Name	Description	Min.	Max.
tclk_freq	Clock frequency	—	100MHz
tclk_period	Clock period	10ns	—
tclk_signal_valid	Signal valid from rising edge of clock	0.5ns	2.4ns

The DC electrical specifications for the inbound CSIX port are as follows:

Name	Description	Min.	Typ.	Max.
V_{DDIO}	I/O supply voltage	2.3V	2.5V	2.7V
V_{OH}	V_{OUT} high voltage	V_{DDIO} - 0.2V	—	—
V_{OL}	V_{OUT} low voltage	—	—	0.2V
I_{OH}	I_{OUT} at V_{OH}	12mA	—	—
I_{OL}	I_{OUT} at V_{OL}	6mA	—	—

Generic FIFO Bridge Ports

The Generic FIFO Bridge interface is buried inside the FPGA. No I/O buffers are allocated to any signals of the Generic FIFO Bridge interface. Timing specifications are tightly coupled to FPGA placement and routing and therefore cannot be documented here. However, this design has been verified to route in ORCA FPGAs at a Generic FIFO Bridge clock maximum frequency of 100MHz. Consider the following characteristics about the interface in designing circuits to read and write frames from/to the Generic FIFO Bridge interface: 1) all outputs are clocked by the rising edge of the clock, 2) all inputs are sampled by the rising edge of the clock, 3) all outputs originate at Q outputs (sequential), 4) all inputs should be driven by sources that originate at Q outputs.

The table below lists various hardware aspects of the IP core's Generic FIFO Bridge interface signals.

The following figures show timing diagrams for a 64-byte payload read from the Generic FIFO Bridge interface. Note that due to a peculiar characteristic of the Block RAM based FIFO of the ORCA Series 4, read transfer timing differs between the first read after the FIFO is empty and a read anytime thereafter. The two cases are shown below.

Signal Name	In/Out	f_{MAX}	Active Clock Edge	Max LUT Depth to Din	Output Type
any sine input clock	In	100MHz	—	—	—
any sine input signal	In	—	↑	4	—
ib_sine_empty	Out	—	↑	—	EBR Qout
ib_sine_data_avail_n	Out	—	↑	—	EBR Qout
ib_sine_data	Out	—	↑	—	EBR Qout
ob_sine_full	Out	—	↑	—	PFU Qout

Figure 6. Inbound Generic FIFO Bridge Read Transfer (First Read After Empty)

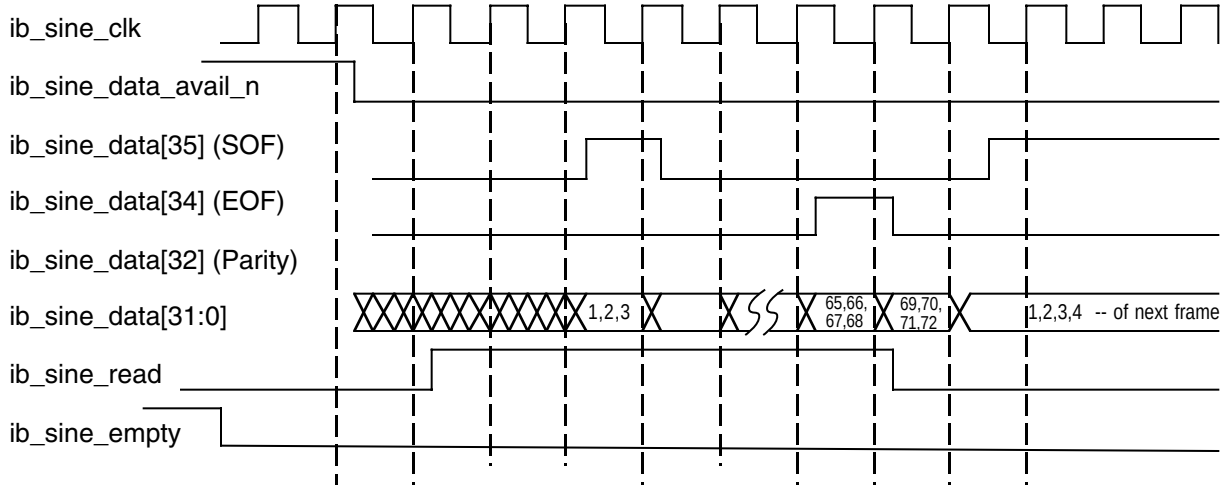


Figure 7. Inbound Generic FIFO Bridge Read Transfer (Ordinary)

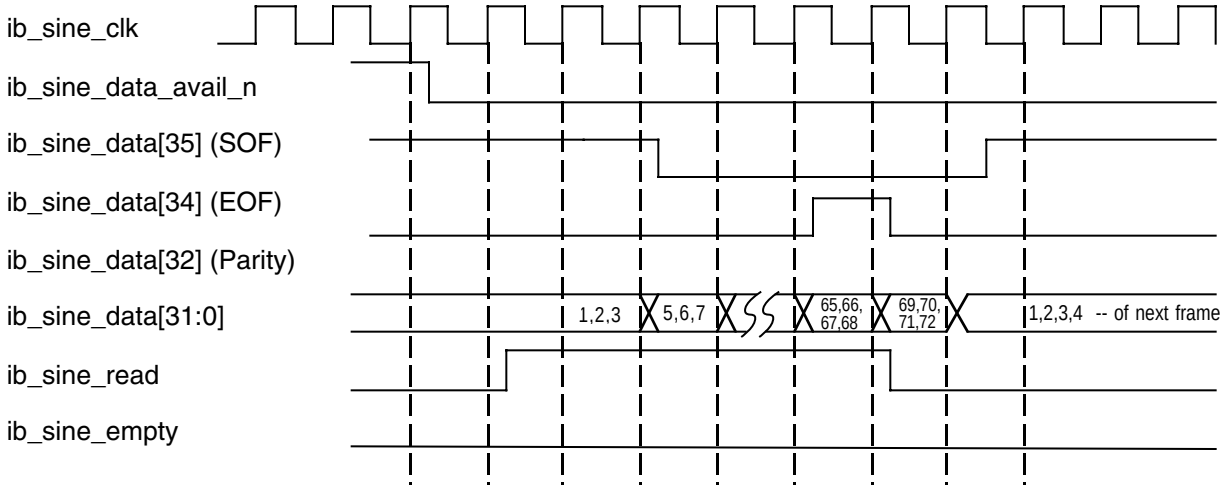
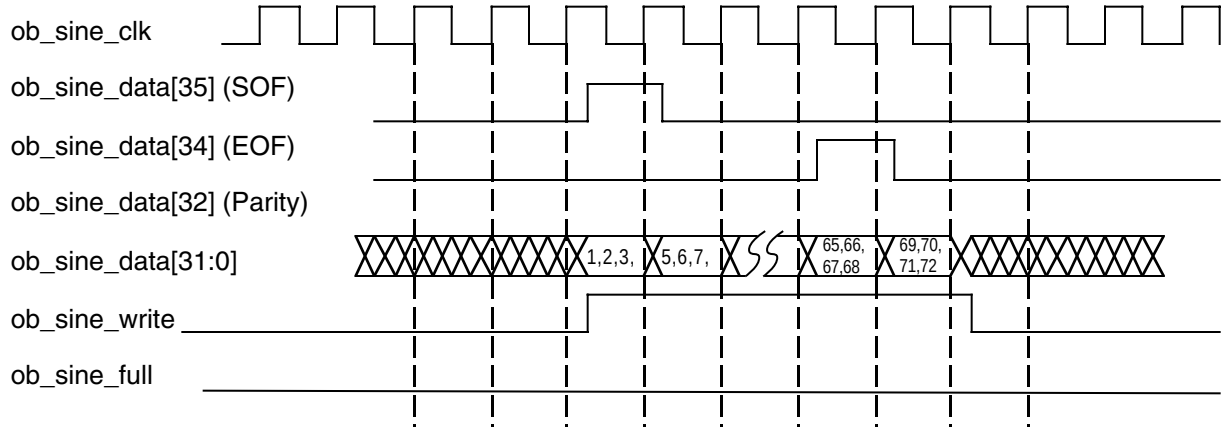


Figure 8. Outbound Generic FIFO Bridge Write Transfer



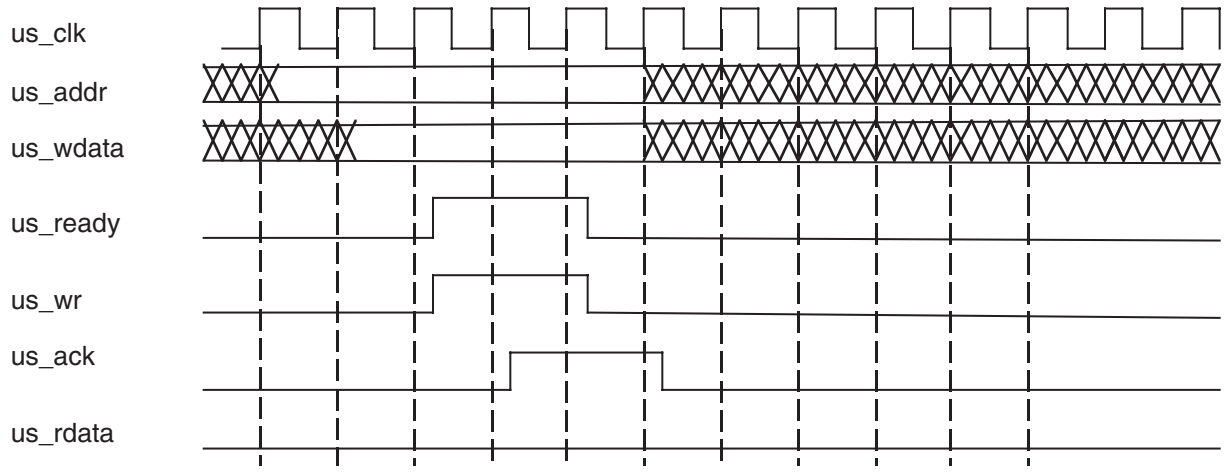
Register Access Port

The register interface is buried inside the FPGA. No I/O buffers are allocated to any signals of the register interface. Timing specifications are tightly coupled to FPGA placement and routing and therefore cannot be documented here. However, this design has been verified to route in ORCA FPGAs at a us_clk maximum frequency of 50MHz. Consider the following characteristics about the interface in designing circuits to read and write data from/to the register interface: 1) all outputs are clocked by the rising edge of the clock, 2) all inputs are sampled by rising edge of clock, 3) some outputs originate at Q-outputs (sequential) and other outputs originate at F-outputs (combinational).

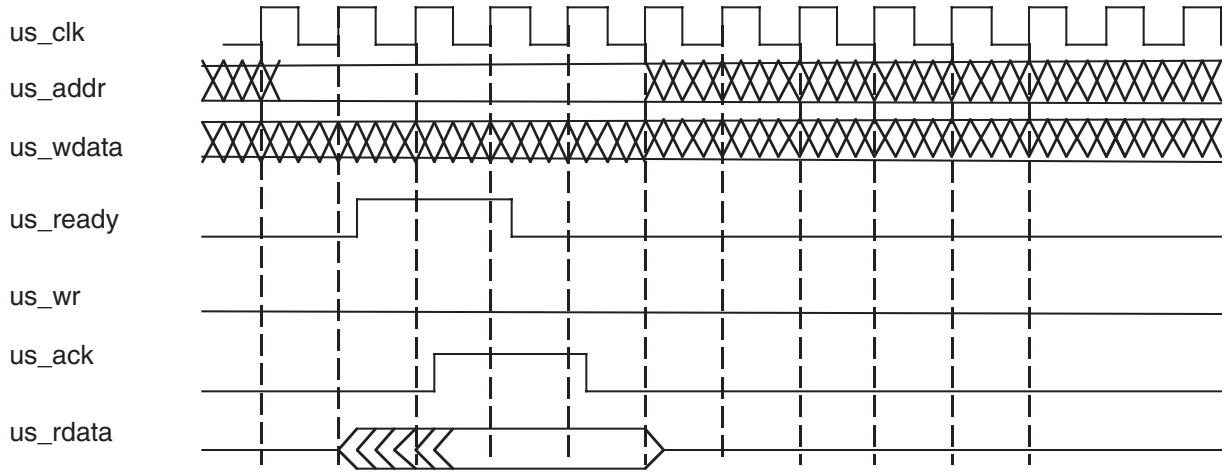
The table below lists various hardware aspects of the IP core's register interface signals.

Signal Name	In/Out	f_{MAX}	Active Clock Edge	Max LUT Depth to Din	Output Type
us_clk	In	50MHz	—	—	—
Any register input signal	In	—	↑	4	—
us_ack	Out	—	↑	—	PFU Qout
us_rdata	Out	—	↑	—	Tbuff Out

The following figure shows a timing diagram for a register write cycle.



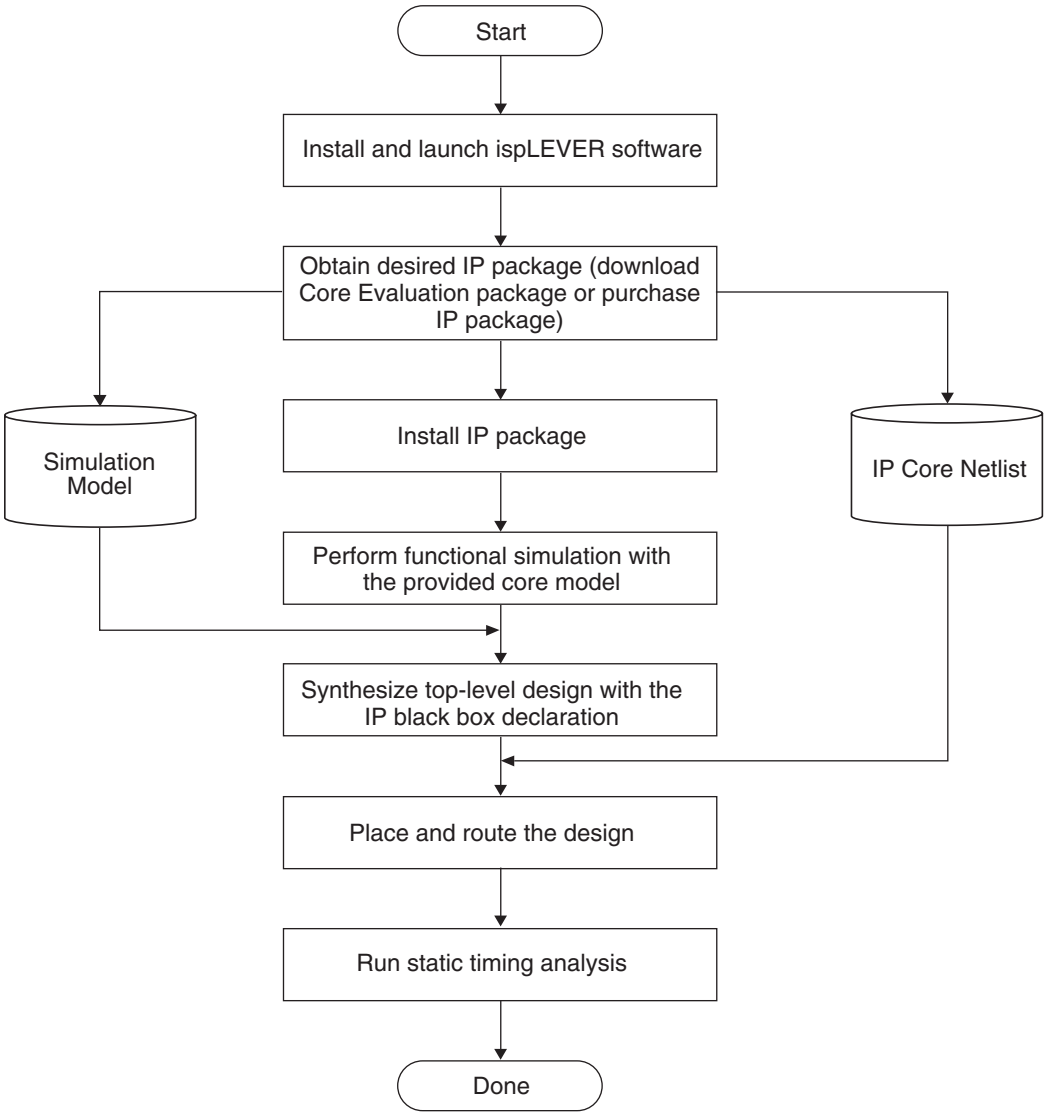
The following figure shows a timing diagram for a register read cycle.



CSIX Level 1 Core Design Flow

The CSIX Level 1 IP Core can be implemented using various methods. The scope of this document covers only the push-button Graphical User Interface (GUI) flow. Figure 9 illustrates the software flow model used when evaluating with the CSIX Level 1 core.

Figure 9. Lattice Core Implementation Flow



IPexpress™

The Lattice IP configuration tool, IPexpress, is incorporated in the ispLEVER® software. IPexpress includes a GUI for entering the required parameters to configure the core. For more information on using IPexpress and the ispLEVER design software, refer to the software help and tutorials included with ispLEVER. For more information on ispLEVER, see the Lattice web site at: www.latticesemi.com/software.

Functional Simulation under ModelSim (PC Platform)

Note: The following procedures are shown using the ORCA® Series 4 version of the CSIX-to-PI40 core. For other device versions, refer to the Readme release notes included in that evaluation package.

The RTL simulation environment contains a testbench and a simple application that uses the CSIX Level 1 IP core. The application consists of the Generic FIFO Bridge loopback function. The application instantiates the IP core, a Generic FIFO Bridge loopback module, and an ORCA SYSBUS module. The instantiated name of the application is called "top". The testbench includes a CSIX driver, a CSIX monitor, a Motorola Power PC driver, and an instantiation of the "top" application. The CSIX driver sends 28 CSIX frames to the user application, based on the file: ~eval\testbench\vectors\vectors_32.v. The CSIX monitor inspects CSIX frames transmitted by the application and dumps the results to a file called "cmon_out.dat" in the local simulation directory. The PowerPC driver gets its instructions from the vectors_32.v file and dumps its results to a file called "mpu_out" in the local simulation directory. The following procedure describes the method for running a simulation of the user application.

A simulation script file is provided in the "eval" directory for RTL simulation. The script file eval_sim_csix2sine.do uses precompiled models provided with this package. The pre-compiled library of models is located in the directory `eval\lib\modelsim\work`.

Simulation Procedures

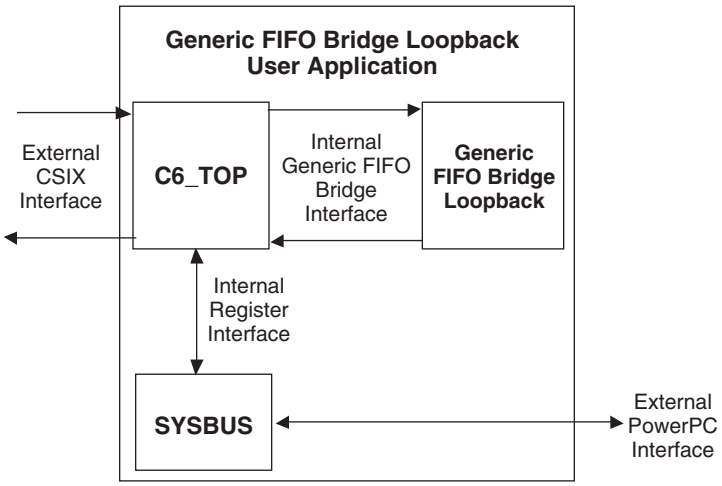
1. Launch ModelSim.
2. Using the main GUI, change the directory location.
   Select: File_Change Directory_eval\simulation
3. Execute Simulation Macro
   Select: Macro _ Execute Macro _ `scripts\eval_sim_csix2sine.do`

The pre-compiled model provided in this IP evaluation package does not work with the OEM version of ModelSim embedded in the ispLEVER 3.0 software. For more information on how to use ModelSim, please refer to the *ModelSim User's Manual.*

Core Implementation

Lattice's CSIX Level 1 evaluation package includes a simple CSIX user application to demonstrate the process of synthesizing, mapping, and routing a design using the CSIX Level 1 IP core. The application consists of the basic CSIX Level 1 IP core, a verilog module that loops the Generic FIFO Bridge output interface to the Generic FIFO Bridge input interface, and a verilog module that instantiates the ORCA4 SYSBUS component, thereby providing a Motorola Power PC interface to the core's register interface. This example application is illustrated in Figure 10. Once familiar with the core implementation process, the "real" application can replace the example application.

Figure 10. Example Application



The following Verilog files for CSIX Level 1 core are provided:

- csix_lev1_o4_01_001.v: top level for the CSIX Level 1 IP core
- sine_loopback.v: example user application for CSIX Level 1 IP core
- mpi_synth.v: top level for ORCA PowerPC-to-Register Interface module
- lb_top_1.v: top-level module that uses ties all the application components together

Users can use the CSIX top level as a black box in system designs. Users may also use lb_top_1.v as a template for an application. However, all inappropriate modules (e.g. the Generic FIFO Bridge loopback module) must be replaced by the real application modules. Also, any invalid internal signal names (e.g. Generic FIFO Bridge interface, register interface) must be replaced with the actual names from the real application.

Black Box Considerations

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.*

The core implementation consists of synthesis and place and route sections. Each section is described below. Two synthesis tools, Synplicity® Synplify® and LeonardoSpectrum™, are included in Lattice's ispLEVER software for seamless processing of designs. The current IP cores are being tested with EDIF flow. The following are the step-by-step procedures for each synthesis tool to generate the EDIF netlist containing the IP core as a black box.

Synthesis using Synplicity's Synplify

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

1. Launch the Synplify synthesis tool.
2. Select -> **Open Project -> Existing Project**
   navigate to select the following file: eval\synthesis\synplicity\user_application\top_001.prj
3. Click on the **RUN** button. This starts the synthesis process. When complete, the resulting synthesized design resides in the file: TOP.edn.

Synthesis using LeonardoSpectrum

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

1. Launch the Leonardo Spectrum synthesis tool.

2. Select -> **File -> Run Script**
   navigate to select the following file: eval\synthesis\exemplar\user_application\fpga_syn_001.tcl

   This automatically starts the synthesis process. When complete, the resulting synthesized design resides in the file: TOP.edf.

Place and Route for ORCA Series 4 Devices

Once the EDIF netlist is generated, the next step is to 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. Copy the following files to the Place and Route working directory: eval\par
   a) eval\ngo\csix_lev1_o4_01_001.ngo
   b) eval\prf\csix_lev1_o4_01_001.prf
   c) The top-level EDIF netlist generated from running synthesis
   Rename the copied file: csix_lev1_o4_01_001.prf to TOP.prf.

2. Launch the ispLEVER software.

3. Select -> **New Project**
   navigate to: eval\par
   type in the project name: TOP
   select -> **Project type -> EDIF**
   click on the SAVE button.

4. In the project window, right click on the listed Lattice device
   Select -> **Select New Device**
   Choose -> ORCA or4e404, -2 speed, BM680 package.

5. In the project window, right click on the listed or4e04 device
   Select -> **Import**
   Choose -> **TOP.edf** (or TOP.edn if you used synplicity)

6. In the ispLEVER Project Navigator, select **Tools**->Timing **Checkpoint Options**. The Timing Checkpoint Options window will pop-up. In both Checkpoint Options, select **Continue**.

7. In the ispLEVER Project Navigator, highlight **Place & 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: 1
   • Routing Resource Optimization: 1
   • Routing Delay Reduction Passes: 5
   • Routing Passes: 30
   • Placement Effort Level: 5

All other options remain at their default values. The properties shown above are the settings for single channel 32-bit mode. Each core configuration has its own properties settings. For the appropriate settings for specific configuration, please refer to the Readme.htm included in the downloaded package.

To start the place and route, choose -> **Start** while the highlighted "Place and Route" item is right clicked.

8. Select the **Place & Route Trace Report** in the project navigator to execute Place and Route and generate a timing report for ORCA.
9. If the f_{MAX} for the core meets the required static timing then the process is complete. Otherwise proceed to step 11.
10. Select the **Cycle Stealing** process in the Project Navigator.
11. Highlight **Place and Route TRACE Report**, with a right mouse click and select **Force One Level**. A new timing report is generated.

References and Related Information

- CSIX-L1 Common Switch Interface Specification-L1, August 5, 2000, Network Processing Forum
- ORCA Series 4 FPGAs Data Sheet, January 2002, Lattice Semiconductor
- ORCA Series 4 MPI/System Bus Technical Note TN1017, March 2002, Lattice Semiconductor
- Standard Interface to External FIFO (SINE) Interface Specification, Version 1.1, August 2002, Lattice Semiconductor

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 FPGAs and FPSCs

Table 4. Performance and Utilization[1]

Configuration Number	Core Description	FIFO Size	PFUs	LUTs	Regs	EBR	PIO	Buried Generic FIFO Bridge I/O	Buried Reg I/O	f_{MAX} (MHz)
csix_lev1_o4_1_001.lpc	one 32-bit csix	1024	222	818	1198	4	112	156	26	100

1. Performance and utilization characteristics are generated using an OR4E04-2BM680C in Lattice's ispLEVER 3.0 software. When using this IP core in a different density, package, speed, or grade within the ORCA family, results may vary.

Note: See parameter table for default values.

Supplied Netlist Configurations

The Ordering Part Number (OPN) for all configurations of the CSIX Level 1 IP Core core ORCA Series 4 devices is CSIX-LEV1-O4-N1. Table 4 lists the netlists available as Evaluation Packages for the ORCA Series 4 devices, which can be downloaded from the Lattice web site at www.latticesemi.com.