CSIX-to-PI40 IP Core

User's Guide

October 2005
ipug17_02.0

Introduction

Lattice's CSIX-to-PI40 core provides a customizable solution allowing a CSIX interface to Agere Systems' PI40 switch fabric. This user's guide explains the functionality of the CSIX-to-PI40 core and how it can be implemented to provide that interface.

The CSIX-to-PI40 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-PI40 Bridge
- Supports standard 32-bit, 100MHz CSIX-L1 interfaces.
- Supports PI40 dual-SERDES interface links
- Interfaces on PI40 side to ORSO82G5 embedded core.
- Supports CSIX Idle, Unicast, Multicast ID and Flow Control frames.
- Interprets CSIX-L1 Cframe header, translates relevant information to PI40. Fabric Input Port Interface (FIPI) compatible header and encapsulates Cframes in PI40 payload for transport through fabric.
- Interprets PI40 Fabric Output Port Interface (FOPI) Format 1 header, translates appropriate cell type and flow control information and extracts Cframes from PI40 payload.
- Supports CSIX/PI40 flow control interworking; generates CSIX-L1 Flow Control Cframes in egress direction.
- Ingress data Cframe FIFO size of 1,024 bytes.
- Egress data Cframe FIFO size of 4,096 bytes.
- Egress control Cframe FIFO size of 1,024 bytes.
- Parameterizable PI40 user payload size (64, 72 or 80 octets) and corresponding Cframe MAX_FRAME_PAYLOAD_SIZE (56, 64 or 72 octets, respectively).
- Parameterizable number of CSIX/PI40 link instantiations (one or two).
- Parameterizable support for SERDES protection switching.
- Parameterizable type of I/O buffers for CSIX interface (LVCMOS or HSTL).
- Internal register set for control and status management.
- 8-bit register interfacing via built-in ORCA® System Bus.

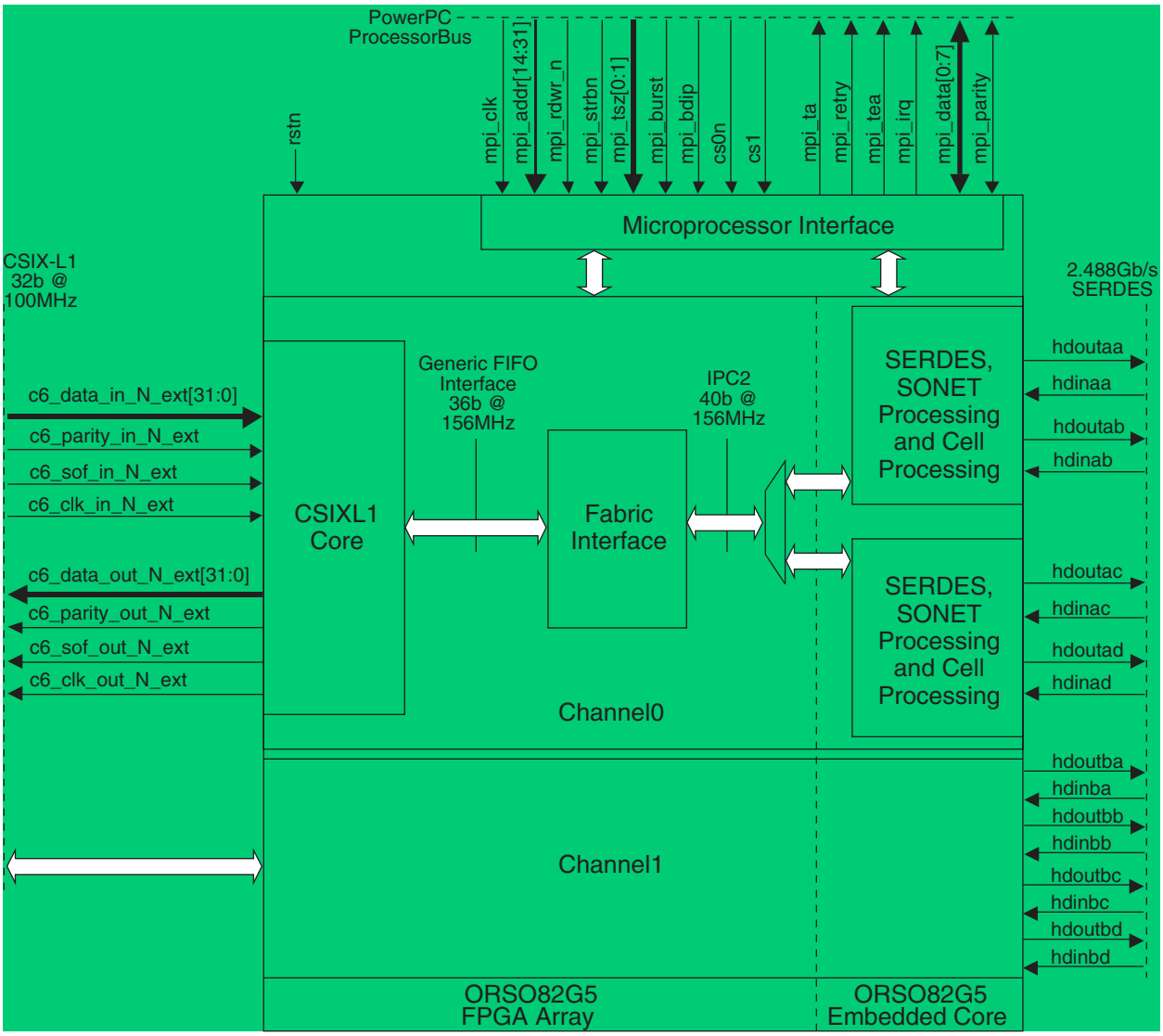
General Description

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; and provides 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 data and control queues 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 Semiconductor's CSIX-to-PI40 core links a compliant CSIX-L1 interface to Lattice's dual SERDES interface (compatible with PI40 interface). Inbound data frames from the CSIX port are deposited into the core's inbound FIFO. These CSIX frames are converted to PI40 cells and driven onto the dual SERDES interface. PI40 cells received on the dual SERDES interface are converted to CSIX frames and placed in the outbound FIFO. CSIX frames stored in the core's out-bound FIFOs are driven onto the outbound CSIX interface.

Block Diagram

Figure 1. CSIX-to-PI40 Block Diagram



Parameter Descriptions

The list of parameters used for configuring the CSIX-to-PI40 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

No.	Parameter	Description	Choice	Default
1	NUM_OF_CHANNELS	Number of CSIX channels	1 or 2	1
2	PI40_CELL_SIZE	PI40 cell size and corresponding Cframe MAX_FRAME_PAYLOAD_SIZE (56,64,72).	76, 84 or 92 bytes	92 bytes
3	PROTECTION	Support ability to switch between two fabrics.	Yes or no	yes
4	BUFFER_TYPE	Buffer type used for external CSIX Pins.	LVCMOS or HSTL	LVCMOS

Signal Descriptions

Table 2. Signal Definitions of CSIX-to-PI40 Core

Signal Name[1]	Direction	Width (Bits)	Description
CSIX Interface (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)
SERDES Interface (FPGA Primary I/Os)			
hdinna[a:d]	Input	1	SERDES receive data input
hdinpa[a:d]	Input	1	SERDES receive data input
hdoutna[a:d]	Output	1	SERDES receive data output
hdoutpa[a:d]	Output	1	SERDES receive data output
hdinnb[a:d]	Input	1	SERDES receive data input
hdinpb[a:d]	Input	1	SERDES receive data input
hdoutnb[a:d]	Output	1	SERDES receive data output
hdoutpb[a:d]	Output	1	SERDES receive data output
Register Interface (FPGA Primary I/Os, Do Not Replicate)			
mpi_clk	Input	1	Clock (50 MHz)
mpi_addr[14:31]	Input	18	Address bits
mpi_rdwr_n	Input	1	Write High / Read Low
mpi_strbn	Input	1	Data transfer strobe
mpi_tsz[0:1]	Input	2	Data transfer size
mpi_burst	Input	1	Active low burst transfer indicator
mpi_bdip	Input	1	Active low for processor request of second beat
cs0n	Input	1	Active low chip select
cs1	Input	1	Active high chip select
mpi_ta	Output	1	Active low acknowledge to processor
mpi_retry	Output	1	Active low retry request to processor
mpi_tea	Output	1	Active low indicator to processor of internal bus error

Table 2. Signal Definitions of CSIX-to-PI40 Core (Continued)

Signal Name[1]	Direction	Width (Bits)	Description
mpi_irq	Output	1	Active low interrupt request to processor
mpi_data[0:7]	Input	8	Most significant byte of processor data bus
mpi_parity	Input	1	Parity bit for most significant byte of processor data bus
Global (FPSC Primary I/Os, Do Not Replicate)			
rstn	Input	1	Active low global reset
us_clk	Input	1	Clock (50 MHz)

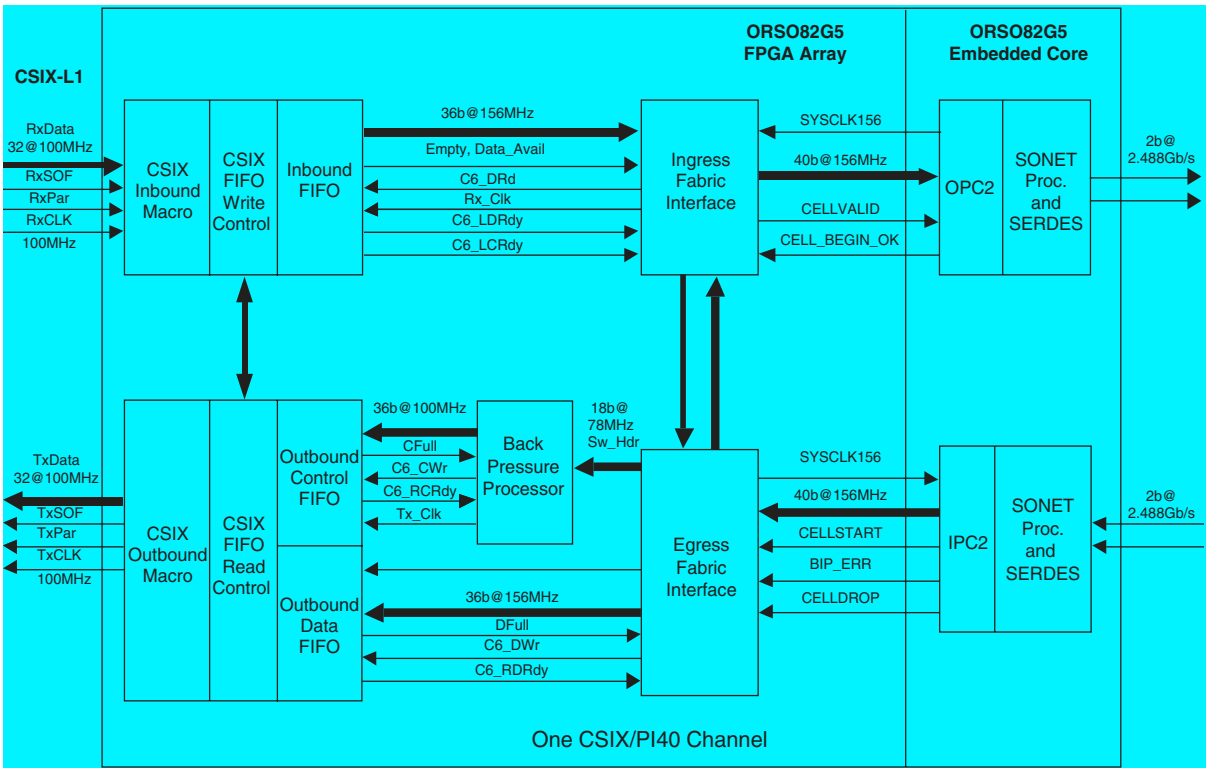
1. "N" denotes instantiated channel number (0, 1).

Functional Description

A block diagram of one CSIX-to-PI40 Bridge channel is shown in Figure 2. Cframes received at the CSIX port are checked by the CSIX Inbound Macro and loaded into the Inbound CSIX FIFO. The CSIX information is then passed to the Ingress Fabric Interface (IFI) block that interprets the Cframe header, generates the appropriate PI40 header and maps the Cframe into PI40 cell payload. The PI40 cells are then written to the ORSO82G5 embedded core, which serializes and formats the SONET-based data frame structure and manages transmission on the SERDES links.

PI40 cells received in the egress direction on SERDES links are extracted and deserialized by the ORSO82G5 embedded core and passed to the Egress Fabric Interface (EFI) block. The EFI extracts the Unicast and Multicast ID Cframes from the PI40 cells and writes them into the Outbound CSIX Data FIFO. The EFI extracts backpressure information from the PI40 cell headers and passes it to the Backpressure Processor (BPP). The BPP generates the appropriate CSIX Flow Control Cframes that are written into the Outbound CSIX Control FIFO. Cframe transmission at the CSIX interface is managed by the CSIX Outbound Macro.

Figure 2. CSIX/PI40 Data Flow



The basic PI40 cell structure and CSIX Cframe structure are shown in Tables 2 and 3, respectively. PI40 cells have a fixed, parameterizable payload length that may be set to 64, 72 or 80 bytes in length. CSIX Cframes have a variable, parameterizable maximum length that may be set between 0 and 256 bytes. CSIX specifies that a Traffic Manager (TM) shall support a programmable MAX_FRAME_PAYLOAD_SIZE parameter between 1 and 256 bytes and that fabrics shall support Cframe sizes of any size equal to or less than the maximum value specified for MAX_FRAME_PAYLOAD_SIZE. CSIX also specifies that system integrators should use the smaller of the maximum values for MAX_FRAME_PAYLOAD_SIZE specified by the TM and fabric being integrated. The CSIX/PI40 Bridge is parameterizable to support PI40 cell size and CSIX MAX_FRAME_PAYLOAD_SIZE as specified in Table 5.

Table 3. PI40 Cell Structure

Cell Component	Size
Link Header	1 byte
Switch Header	11 bytes
PI40 Payload	64, 72 or 80 bytes
BIP	1 byte

Table 4. CSIX Cframe Structure

Cframe Component	Size
Base Header	2 bytes
Extension header	0-4 bytes
CSIX Payload	Variable
Vertical Parity	2 bytes

Table 5. CSIX/PI40 Bridge Payload Parameter

PI40 Payload Size	CSIX Maximum Payload Size (MAX_FRAME_PAYLOAD_SIZE)
64 bytes	56 bytes
72 bytes	64 bytes
80 bytes	72 bytes

Data Flow

Ingress (CSIX-to-PI40) Direction Data Flow

Cframes received at the CSIX interface are loaded into a FIFO in the CSIX-L1 core. The CSIX-L1 core supports two FIFOs, one for data and one for control Cframes. The only control Cframes presently defined are flow control Cframes. In the CSIX/PI40 Bridge application, no flow control Cframes are received at the CSIX interface, so only the data FIFO is implemented in the ingress direction.

Cframes are read from the ingress data FIFO by the IFI and converted to PI40 cells as shown in Figure 3. Each Cframe is mapped to a separate PI40 cell.

Figure 3. PI40 Ingress Cell Structure (FIPI)

PI40 Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	RES				EBP	TRCLASS[2:0]		
1	MC	PSW_ID	RQN[9:4]					
2	RQN[3:0]				MCTAG[19:16]/DESTP[9:6]			
3	MCTAG[15:10]/DESTP[5:0]						MCTAG[9:8]/RQN3ST[9:8]	
4	MCTAG[7:0]/ RQN3ST[7:0]							
5	RES							
6	RES							
7	RES							
8	RES							
9	RES							
10	RES							
11	CSIX Base Header							
12	CSIX Base Header							
13	CSIX Extension Header							
14	CSIX Extension Header							
15	CSIX Extension Header							
16	CSIX Extension Header							
17 to (N+16)	CSIX Payload (N bytes, where N ≤ MAX_FRAME_PAYLOAD_SIZE)							
N+17	CSIX Vertical Parity							
N+18	CSIX Vertical Parity							
(N+19) to 74, 82 or 90	Remaining PI40 Payload (pad if needed)							
75, 83 or 91	BIP8							

Cframe-to-PI40 Cell Conversion

Three types of Cframes are supported:

- Idle
- Unicast
- Multicast ID

These Cframe types are shown in Figures 4, 5 and 6. For Unicast and Multicast ID Cframes, the entire Cframe is mapped to the PI40 cell payload intact and Cframe base and extension header information is processed and mapped to corresponding PI40 header data. After mapping the Cframe to the fixed-length payload, pad is added to fill any unused locations.

Ingress Idle Cframes are dropped by the CSIX-L1 core. The IFI generates PI40 idle cells when no valid data cells are available. These cells contain no user data, but do contain port egress direction backpressure information.

Figure 4. CSIX Unicast Cframe

CSIX Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	DReady	CReady	Type				CR	P
1	Payload Length							
2	Vertical Parity							
3	Vertical Parity							

Figure 5. CSIX Idle Cframe

CSIX Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	DReady	CReady	Type				CR	P
1	Payload Length							
2	Class							
3	P		CSIX Reserved					
4	CSIX Reserved				Destination Address[11:8][1]			
5	Destination Address[7:0][1]							
6	Payload Byte 1							
—	—							
N+5	Payload Byte N							
	Vertical Parity							
	Vertical Parity							

1. Undefined in egress direction.

Figure 6. CSIX Multicast ID Cframe

CSIX Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	DReady	CReady	Type				CR	P
1	Payload Length							
2	Class							
3	P	CR	Multicast ID[21:16][1]					
4	Multicast ID[15:8][1]							
5	Multicast ID[7:0][1]							
6	Payload Byte 1							
—	—							
N+5	Payload Byte N							
	Vertical Parity							
	Vertical Parity							

1. Undefined in egress direction.

CSIX-L1 Multicast Mask, Multicast Binary Copy and Broadcast frames are not supported by this bridge since there are no equivalent features in the PI40 protocol. CSIX Flow Control Cframes are never received in the CSIX inbound direction in this application.

Cframe data is read from the ingress FIFO and inspected for SOF and EOF indications as long as the FIFO empty flag is not asserted (i.e. as long as there is data in the FIFO). Detection of an SOF indicates the beginning of a new Cframe/cell.

The Cframe header information is inspected and mapped to PI40 cell header as follows:

- Unicast Cframe mapping:
  - RES = 0
  - EBP = 1 if CSIX Ready = 0 or outbound FIFOs exceed high watermark, else EBP = 0
  - TRCLASS = top 3 bits of CSIX Class
  - MC = 0
  - PSW_ID = provisionable
  - RQN = 0
  - DESTP = bottom ten bits of CSIX Destination

- Multicast ID Cframe mapping:
  - RES = 0
  - EBP = 1 if CSIX Ready = 0 or outbound FIFOs exceed high watermark, else EBP = 0
  - TRCLASS = top 3 bits of CSIX Class
  - MC = 1
  - PSW_ID = provisionable
  - RQN = 0
  - MCTAG = bottom 20 bits of CSIX Multicast ID

If the CSIX Ready bits (CRdy or DRdy) go low (0) or the outbound FIFOs exceed their high watermark, Egress Backpressure (EBP) is asserted in the ingress cells.

If there are no Unicast or Multicast cells, Idle cells are generated as follows:

- Idle cell format with egress back pressure:
  - RES = 0
  - EBP = 1 if CSIX Ready = 0 or outbound FIFOs exceed high watermark, else EBP = 0
  - TRCLASS = 111
  - MC = 0
  - PSW_ID = provisionable
  - RQN = 0
  - DEST/MCTAG = 0

PI40 Cells to ORSO Embedded Core

PI40 Unicast or Multicast cells are passed to the 2-link ORSO Output Port Controller (OPC2) whenever available. PI40 Idle cells are sent when no valid data cells are available. The OPC2 automatically prepends PI40 Link Header Byte and appends PI40 BIP8 to each PI40 cell. The OPC2 manages cell transmission on the corresponding SERDES links. Complete details on the OPC2 interface and functionality are given in the ORCA ORSO82G5 data sheet.

Egress (PI40-to-CSIX) Direction Data Flow

In the egress direction, Unicast and Multicast Cframes are extracted from PI40 cells received on the SERDES links and written to the CSIX-L1 Outbound Data FIFO and then transmitted on the CSIX interface. Flow control information extracted from the egress PI40 cells is used to generate CSIX Flow Control Cframes that are written to the CSIX Outbound Control FIFO and then transmitted on the CSIX interface. The CSIX/PI40 Bridge IP supports Cframes mapped to PI40 FOPI Format 1 cells as shown in Figure 7.

Figure 7. PI40 Egress Cell Structure (FOPI Format 1)

PI40 Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	RES	PBKP[3:0]				CELLTYPE[2:0]		
1	BKPQ1[11:4]							
2	BKPQ1[3:0]				BKPQ2[11:8]			
3	BKPQ2[7:0]							
4	BKPQ3[11:4]							
5	BKPQ3[3:0]				BKPQ4[11:8]			
6	BKPQ4[7:0]							
7	BKPQ5[11:4]							
8	BKPQ5[3:0]				BKP1SC	BKP1ON	BKP2SC	BKP2ON
9	BKP3SC	BKP3ON	BKP4SC	BKP4ON	BKP5SC	BKP5ON	MCBKP[1:0]	
10	DIAG[7:0]							
11	CSIX Base Header							
12	CSIX Base Header							
13	CSIX Extension Header							
14	CSIX Extension Header							
15	CSIX Extension Header							
16	CSIX Extension Header							
17 to (N+16)	CSIX Payload (N bytes, where N ≤ MAX_FRAME_PAYLOAD_SIZE)							
N+17	CSIX Vertical Parity							
N+18	CSIX Vertical Parity							
(N+19) to 74, 82 or 90	Remaining PI40 Payload (pad if needed)							
75, 83 or 91	BIP8							

PI40 Cells from ORSO Embedded Core

Egress direction PI40 cell reception is managed by the ORSO 2-link Input Port Controller (IPC2). The IPC2 manages cell reception on the corresponding SERDES links, automatically extracting the cells and checking BIP8. Valid cells are passed to the IP core EFI block. The IPC2 indicates when a cell is available by asserting the CELLSTART signal. It also provides BIP_ERR and CELLDROP indications. Complete details on the IPC2 interface and functionality are given in the ORCA ORSO82G5 data sheet.

Egress PI40 Cell Processing

The EFI extracts switch header information from all valid egress cells and passes it to the Back Pressure Processor (BPP). The switch header CELLTYPE field is examined and all cells other than Unicast or Multicast ID are discarded. Unicast and Multicast ID Cframes are extracted from the cell payload, using the Cframe payload length information in the CSIX Base Header to determine the valid Cframe data in the cell payload. Any remaining PI40 payload pad is discarded. The Unicast and Multicast ID Cframes are then written to the CSIX Outbound Data FIFO.

Backpressure Processing

Every PI40 FOPI Format 1 cell header contains Port Backpressure (PBKP) and Multicast Backpressure (MCBKP) information and Queue Backpressure information for five different queues as shown in Figure 7. The information is passed to the BPP block, where it is processed and used to generate CSIX Flow Control Cframes that are written to the CSIX Outbound Control FIFO. The CSIX Flow Control Cframe format is shown in Figure 8. Entries 1 through 5 contain flow control information about the 5 queues detailed in the PI40 FOPI Format 1 cell header. Entry 6 contains flow control information for low priority Multicast and entry 7 contains flow control information for high priority Multicast.

Figure 8. CSIX Flow Control Cframe

	CSIX Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Base Header	0	DReady	CReady	Type				CR	P
	1	Payload Length							
Flow Control Entry 1	2	Class							
	3	FC Entry Type		CWC	PWC	Speed			
	4	P		CSIX		Destination Address[11:8]			
	5	Destination Address[7:0]							
Flow Control Entry 2	(4*2)-2	Class							
	(4*2)-1	FC Entry Type		CWC	PWC	Speed			
	(4*2)	P		CSIX		Destination Address[11:8]			
	(4*2)+1	Destination Address[7:0]							
Flow Control Entries 3-6	:								
Flow Control Entry 7	(4*7)-2	Class							
	(4*7)-1	FC Entry Type		CWC	PWC	Speed			
	(4*7)	P		CSIX		Destination Address[11:8]			
	(4*7)+1	Destination Address[7:0]							
		Vertical Parity							
		Vertical Parity							

Register Descriptions

Table 6. Register Map

Register Name	Register Address	Description
Global Control Register	0x8000	Holds global control functions
Reset Control Reg	0x8004	Resets logic for IP Channels [1:0]
FIFO Flush Control Reg	0x8008	Flushes all FIFOs for IP Channels [1:0]
Force H Parity Error Reg	0x800C	Force Horizontal Par Errs for Channels[1:0]
Force V Parity Error Reg	0x8010	Force Vertical Par Errs for Channels[1:0]
Protection Control Ch0	0x8014	Protection fabric switch control for Channel 0
Protection Control Ch1	0x8018	Protection fabric switch control for Channel 1
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
FIFO Underflow Errors	0x8140	FIFO underflow error status for Channels[1:0]
FIFO Overflow Errors	0x8144	FIFO overflow error status for Channels[1:0]
Parity Errors	0x8148	Horizontal and vertical parity errors for Channels[1:0]
Miscellaneous Errors	0x814C	Receiver synchronization error

Detailed Register Descriptions

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description
Global Control Register (Read/Write)				
8000	[0:3]	Not Used	01	Not Used
	[4]	Reset_Enable		When high, this bit initializes all registers and resets all transmission logic. Note that this reset function is self clearing.
	[5]	H_Parity_Enable		When high, horizontal parity error detectors on inbound CSIX ports are enabled to operate for all instantiated channels.
	[6]	V_Parity_Enable		When high, vertical parity error detectors on inbound CSIX ports are enabled to operate for all instantiated channels.
	[7]	Transmission_Enable		When high, CSIX transmission logic is enabled to operate for all instantiated channels.
Reset Control Register (Read/Write)				
8004	[0:5]	Not Used	00	Not Used
	[6]	Reset_1		When high, transmission logic for channel 1 is reset. Note this bit is not self clearing. The bit must be written to "0" to deassert the associated reset.
	[7]	Reset_0		When high, transmission logic for channel 0 is reset. Note this bit is not self clearing. The bit must be written to "0" to deassert the associated reset.
FIFO Flush Register (Read/Write)				
8008	[0:5]	Not Used	00	Not Used
	[6]	Flush_1		When high, all FIFOs for channel 1 are flushed. Note this bit is not self clearing. The bit must be written to "0" to deassert the associated flush.
	[7]	Flush_0		When high, all FIFOs for channel 0 are flushed. Note this bit is not self clearing. The bit must be written to "0" to deassert the associated flush.
Force H Parity Error Registration (Read/Write)				
800C	[0:5]	Not Used	00	Not Used
	[6]	Force HPERR_1		When high, the horizontal parity checker on channel 1 is forced to detect a parity error. Note this bit is not self clearing. The bit must be written to "0" to deassert the associated error condition.
	[7]	Force HPERR_0		When high, the horizontal parity checker on channel 0 is forced to detect a parity error. Note this bit is not self clearing. The bit must be written to "0" to deassert the associated error condition.
Force V Parity Error Registration (Read/Write)				
8010	[0:5]	Not Used	00	Not Used
	[6]	Force VPERR_1		When high, the vertical parity checker on channel 0 is forced to detect a parity error. Note this bit is not self clearing. The bit must be written to "0" to deassert the associated error condition.
	[7]	Force VPERR_0		When high, the vertical parity checker on channel 1 is forced to detect a parity error. Note this bit is not self clearing. The bit must be written to "0" to deassert the associated error condition.

Detailed Register Descriptions (Continued)

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description
Protection Control Channel xx (Read/Write), xx = [0, 1]				
8014 - 0 8018 - 1	[0:4]	Not Used	00	Not Used
	[5]	PSW_ SELECT_xx		This bit does the selection of which fabric to receive data from for channel xx. (0 use fabric 0, 1 use fabric 1)
	[6]	PSW_ ID_1_xx		This bit is the source for bit PSW_ID in the PI40 cell overhead (byte 1 bit 6 of FIPI cell format) sent to fabric 1 for channel xx.
	[7]	PSW_ ID_0_xx		This bit is the source for bit PSW_ID in the PI40 cell overhead (byte 1 bit 6 of FIPI cell format) sent to fabric 0 for channel xx.
In_Data_Lo_Watermark_xx (Read/Write), xx = [0, 1]				
8040 - 1 8080 - 1	[0:7]	IBD_LOW_WMARK_xx	08	Channel xx, 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.
In_Cntrl_Lo_Watermark_xx Read/Write), xx = [0, 1]				
8044 - 0 8084 - 1	[0:7]	IBC_LOW_WMARK_xx	08	Channel xx, 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.
In_Data_Hi_Watermark_xx (Read/Write), xx = [0, 1]				
8048 - 0 8088 - 1	[0:7]	IBD_HI_WMARK_xx	3f	Channel xx, 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.
In_Cntrl_Hi_Watermark_xx (Read/Write), xx = [0, 1]				
804C - 0 808C - 1	[0:7]	IBC_HI_WMARK_xx	3f	Channel xx, 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.
Out_Data_Lo_Watermark_xx (Read/Write), xx = [0, 1]				
8050 - 1 8090 - 1	[0:7]	OBD_LOW_WMARK_xx	08	Channel xx, 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.
Out_Cntrl_Lo_Watermark_xx (Read/Write), xx = [0, 1]				
8054 - 0 8094 - 1	[0:7]	OBC_LOW_WMARK_xx	08	Channel xx, low watermark 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.
Out_Data_Hi_Watermark_xx (Read/Write), xx = [0, 1]				
8058 - 0 8098 - 1	[0:7]	OBD_HI_WMARK_xx	3f	Channel xx, 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)

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description
Out_Cntrl_Hi_Watermark_xx (Read/Write), xx = [0, 1]				
805C - 0 809C - 1	[0:7]	OBC_HI_WMARK_xx	3f	Channel xx, 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.
FIFO Underflow Error Register (clear on read)				
8140	[0:5]	Not Used	n/a	Not Used
	[6]	UF_ERR_1		When high, indicates that one of the FIFOs for channel 1 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.
	[7]	UF_ERR_0		When high, indicates that one of the FIFOs for channel 0 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.
FIFO Overflow Error Register (clear on read)				
8144	[0:5]	Not Used	n/a	Not Used
	[6]	OF_ERR_1		When high, indicates that one of the FIFOs for channel 1 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
	[7]	OF_ERR_0		When high, indicates that one of the FIFOs for channel 0 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
Parity Error Register (clear on read)				
8148	[0:1]	Not Used	n/a	Not Used
	[2]	VPERR_1		When high, indicates that a vertical parity error occurred on the inbound CSIX port for channel 1. 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
	[3]	VPERR_0		When high, indicates that a vertical parity error occurred on the inbound CSIX port for channel 0. 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
	[4:5]	Not Used		Not Used
	[6]	HPERR_1		When high, indicates that a horizontal parity error occurred on the inbound CSIX port for channel 1. 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.
	[7]	HPERR_0		When high, indicates that a horizontal parity error occurred on the inbound CSIX port for channel 0. 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.

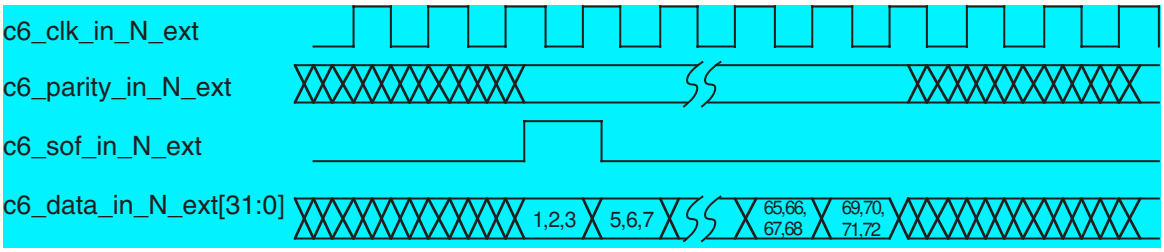
Interface Timing and Electrical Specifications

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.

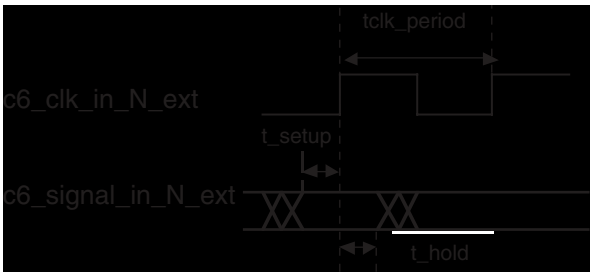
The following figure shows a 64-byte payload arriving at the inbound CSIX interface. Note that the frame is actually 72 bytes long with header and vertical parity added.

Figure 9. CSIX Inbound Frame Transfer



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

Figure 10. CSIX Inbound AC Timing Specifications



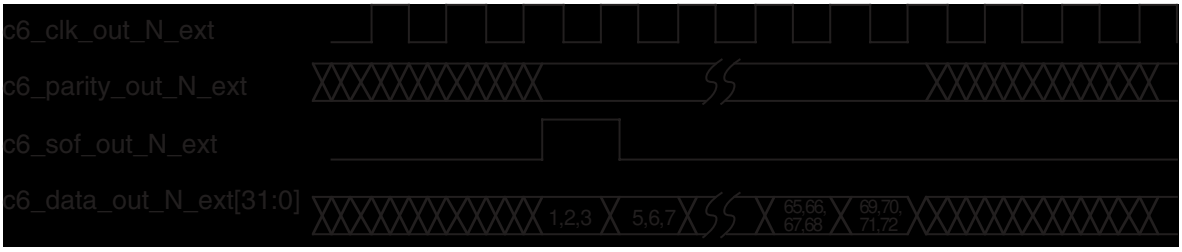
Name	Description	Min.	Max.
tclk_freq	Clock frequency	—	100MHz
tclk_period	Clock period	10ns	—
t_setup	Setup time to rising edge of clock	1.5ns	—
t_hold	Hold time to rising edge of clock	0ns	—

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

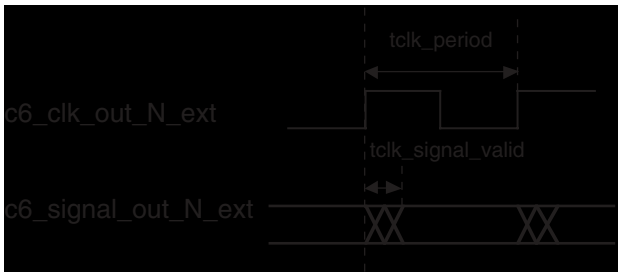
The following figure shows a 64-byte payload leaving the outbound CSIX interface. Note that the frame is actually 72 bytes long with header and vertical parity added.

Figure 11. CSIX Outbound Frame Transfer



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

Figure 12. 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	—	—

SERDES Ports

The timing and electrical specifications for this interface are covered in the ORCA ORSO82G5 data sheet.

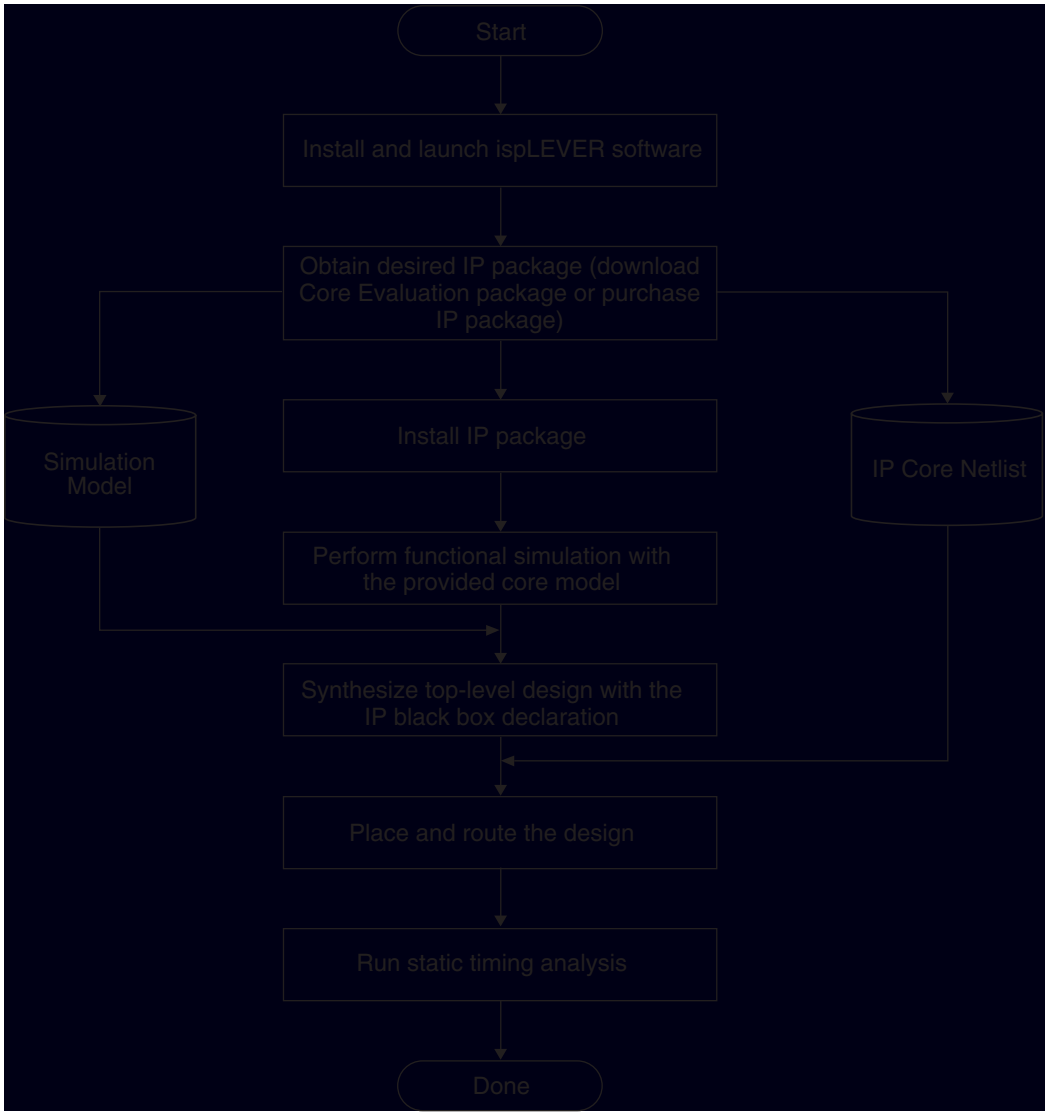
Microprocessor Interface

The timing and electrical specifications for this interface are covered in the ORCA Series 4 FPGA data sheet.

CSIX-to-PI40 Core Design Flow

The CSIX-to-PI40 IP Core can be implemented using various methods. The scope of this document covers only the push-button Graphical User Interface (GUI) flow. Figure 13 illustrates the software flow model used when evaluating with the CSIX-to-PI40 core.

Figure 13. Lattice IP Core Evaluation 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-to-PI40 IP core. The application consists of SERDES loopback function. The application instantiates the IP core, an ORCA ORSO82G5 module and an ORCA SYSBUS module. The instantiated name of the application is called "orso82g5_chip". The testbench includes a CSIX driver, a CSIX monitor, a Motorola Power PC driver, and an instantiation of the "orso82g5_chip" application. The CSIX driver sends 82 CSIX frames to the user application. 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 sends register initialization information 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\config1\simulation\modelsim\scripts" directory for RTL simulation. The script file vsim_rtl.bat uses precompiled models provided with this package. The CSIX-to-PI40 and testbench models have been compiled into the work directory in directory "eval\config1\simulation\modelsim\rtl". The ORCA gate models, ORSO82G5 models, and ORCA SYSBUS models are delivered with the ispLEVER software and the ORSO82G5 design kit.

Simulation Procedures

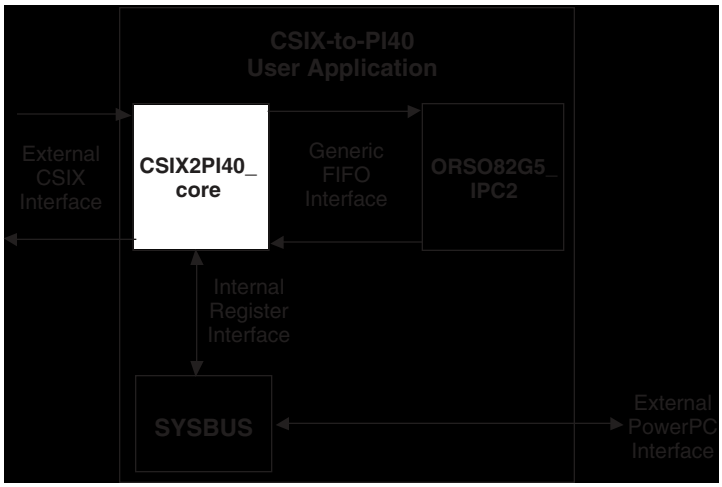
1. Launch ModelSim.
2. Using the main GUI, change the directory location.
   Select: File_Change Directory_eval\simulation
3. Execute Simulation Macro
   Select: Macro _ Execute Macro _ `scripts\eval_sim_csix2pi40.do`

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

Core Implementation

Lattice's CSIX-to-PI40 evaluation package includes a simple CSIX user application to demonstrate the process of synthesizing, mapping, and routing a design using the CSIX-to-PI40 IP core. The application consists of the basic CSIX-to-PI40 IP core, a verilog module that instantiates the ORSO82G5_IPC2 component, 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 user application is illustrated in the figure below. Once the user is familiar with the core implementation process, the "real" user application can replace the example application.

Figure 14. Example Application



The following Verilog files for CSIX-to-PI40 core are provided:

- csix2pi40_core.v for the CSIX-to-PI40 IP core
- orso82g5_chip for top-level module that ties all the application components together

Users can use the csix2pi40_core module as a black box in their system designs. Users may also use orso82g5_chip.v as a template for their own 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\csix2pi40_o4_01_001.ngo
   b) eval\prf\csix2pi40_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 or4e04, -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

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 ORSO82G5 FPSC

Table 7. Performance and Utilization

Configuration Number	Core Configuration	Number of Channels	PI40 Cell Size	Protection Switching	Buffer Type	PFUs	LUTs	EBR Blocks	PIO	f_{MAX} (MHz)
csix_pi40_o4_1_001.lpc	Default 1 channel	1	92	Yes	LVCMOS	626	2953	6	111	100

1. Results are generated using Lattice ispLEVER 3.0 software targeting an ORSO82G5-2BM680.

Supplied Netlist Configurations

The Ordering Part Number (OPN) for all configurations of the CSIX-to-PI40 IP Core core on ORCA Series 4 devices is CSIX-PI40-O4-N1. Table 7 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.