SERDES Eye/Backplane Demo for the LatticeECP3 Versa Evaluation Board

User's Guide

May 2011
UG44_01.1

Introduction

This document provides technical information and instructions on using the LatticeECP3™ SERDES Eye/Backplane Demo. The demo has been designed to demonstrate the performance of the LatticeECP3 SERDES I/O at 3.125 Gbps. The document provides a circuit description as well as instructions for running the demo on the LatticeECP3 Versa Evaluation Board (See EB62, LatticeECP3 Versa Evaluation Board User's Guide). The LatticeECP3 Versa Evaluation Board is part of the LatticeECP3 Versa Development Kit.

The demo walks you through the entire process in both Windows and Linux platforms. Please note that the operation in a Linux platform is limited to implementing the design process through bitstream generation only. In a Windows platform, further simulation and hardware evaluation of the demo can be accomplished.

The SERDES Eye/Backplane Demo installs into the default location of **C:\Lattice_DevKits\DK-ECP3-SERDES-010\.** However, you can install the demo files in a directory of your choice. The demo directory includes the following in Windows and root/DK-ECP3-SERDES-010 in Linux:

- Verilog source code for the FPGA design
- Lattice Diamond™ implementation Project files and Aldec Active-HDL simulation script file
- Bitstream (in format of *.bit)
- ORCAstra plug-in GUI files specifically for this SERDES Eye/Backplane Demo
  Note: The ORCAstra plug-in is not available on the Linux platform
- This SERDES Eye/Backplane Demo User's Guide

Hardware requirement for this loopback application test design:

- LatticeECP3 Versa Evaluation Board with LatticeECP3-35EA, 484-ball fpBGA device
- 12V AC/DC power adapter (with international plug adapters)
- PC with ORCAstra (PC not provided)
- 156.25 MHz on-board source for SERDES/PCS QUAD reference clock
- USB cable to download bitstream
- Backplane with SMAs (not provided)
- Pair of DC Blocks (not provided)
- Eye viewing instrument - DCA, DSO, etc. (not provided)
- SMA cables (not provided)

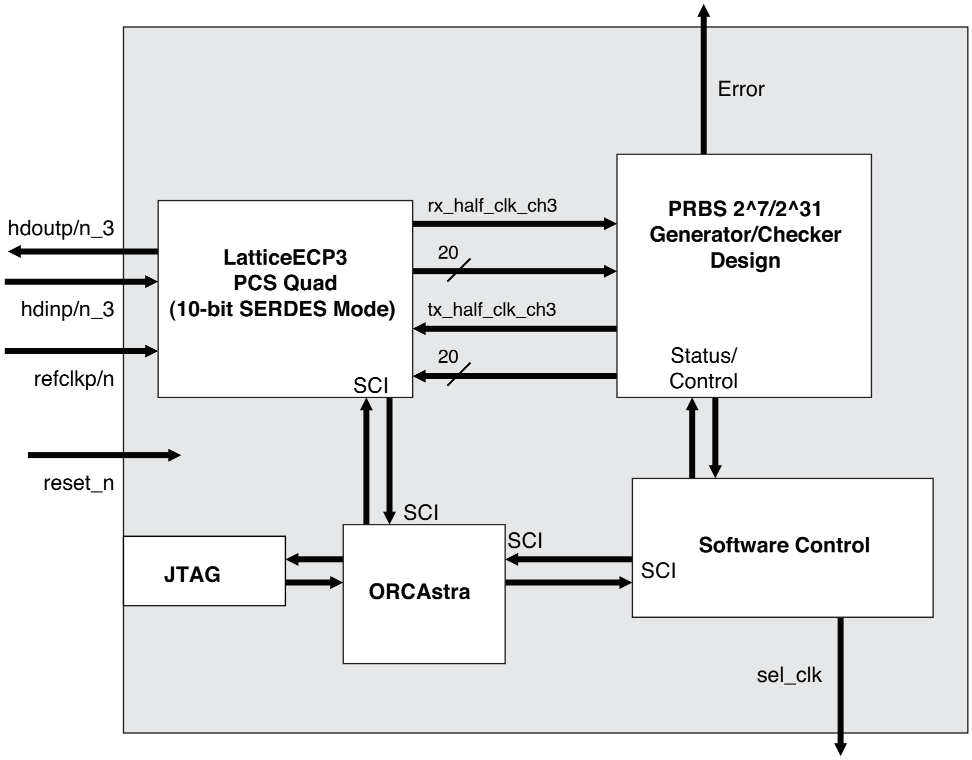
Software application and driver requirements include:

- ispVM™ System software (version 18 or later) for FPGA bitstream download to the LatticeECP3 Versa Evaluation Board
- Lattice Diamond design software version 1.2 (or later)
- ORCAstra software for user control interface - Included with Diamond 1.2 (or later on Windows only)

SERDES Eye/Backplane Demo Design Overview

A block diagram of the demo design is provided in Figure 1.

Figure 1. SERDES Eye/Backplane Demo Design



The basic concept of the design is a quad-based PRBS Generator/checker that transmits parallel data to a PCS quad. In turn, the PCS SERDES channels serialize the data in the transmit direction, and de-serialize it in the receive direction. The serial data stream can be:

- Looped back via a cable on the evaluation board, or
- Sent to a DCA or DSO for eye viewing.

In both cases, a backplane of variable length can be included in the serial path.

The PCS quad location PCSA is used for this SERDES Eye/Backplane demo. This is the only PCS Quad available with the ECP3-35EA device. The demo has been generated in 10-bit SERDES-only mode (20-bit data), at 3.2 Gbps per channel (actual demo rate is 3.125 Gbps or equivalent). On the LatticeECP3 Versa Evaluation Board, only PCSA Channel 3 is available to the user via SMA connectors. Therefore, all demo procedures use Channel 3 of PCSA. Note that the reference design implements the LatticeECP3 PCS TX and RX Reset State machines described under the SERDES/PCS RESET section of TN1176, LatticeECP3 SERDES/PCS Usage Guide.

Clock Sources

PCS quad A is clocked by its designated differential clock input, refclkp/n. The reference clock frequency is 156.25 MHz and has one source: the ispClock™5406A device (U13 on the LatticeECP3 Versa Evaluation Board). Internally to the PCS, the reference clock is multiplied by a factor of 20 to generate the 3.125 Gbps per channel data rate. At the PCS/FPGA interface, a 20-bit data interface is used. This requires 156.25 MHZ receive and transmit clocks. The PCS-generated rx_half_clk_ch3 and tx_half_clk_ch3 (see Figure 1) clock the PCS interface as well as the PRBS Generator/Checker RX and TX data.

PRBS Generator/Checker Quad

There is one PRBS Generator/Checker quad block in the demo design. It is associated with PCS Quad A Channel 3.

The PRBS generator/checker quad has the following characteristics:

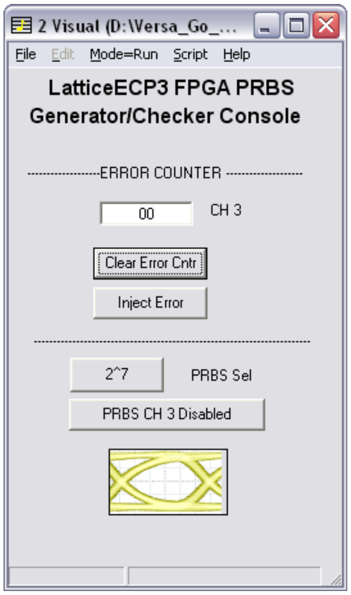
- 20-bit wide data
- One PRBS Generator (2^7 and 2^31)
- One PRBS Checker (2^7 and 2^31)
- Control/Status interface to user registers.
- One Error Counter connected to a user register for monitoring.
- One real time Error signal indicator connected to an on-board LED.

The ORCAstra block controls all user registers as well as the LatticeECP3 PCS QUAD via the SCI interface. ORCAstra is in turn controlled via the JTAG interface. See Figure 1.

PRBS Generator/Checker Quad ORCAstra Console Window

This demo utilizes a visual window plug-in to the base ORCAstra installation as shown in Figure 2. The LatticeECP3 FPGA PRBS Generator/Checker Console window is associated with logical LatticeECP3 PCS Quad 1 in the main ORCAstra software application as described later in this document.

Figure 2. PRBS Generator/Checker Quad Console Window



2 Visual (D:\Versa_Go_...

LatticeECP3 FPGA PRBS Generator/Checker Console

ERROR COUNTER

00 CH 3

2^7 PRBS Sel

PRBS CH 3 Disabled

ERROR COUNTER: There is one ERROR COUNTER display corresponding to channel 3 in the PCSA Quad. When PRBS Channel 3 is Enabled, the error counter increments every time errors are detected at the PRBS checker. The counter is only eight bits wide so the maximum count reached is hFF. When the counter reaches hFF, it does not roll back to zero unless the Clear Error Counter button is selected. The counter will not increment unless the corresponding PRBS channel is enabled.

Clear Error Counters: When the Clear Error Counter button is selected, it asynchronously clears the content of the PRBS checker error counter.

Inject Error button: The PRBS channel needs to be enabled for this error injection feature to work. The demo design injects a single incorrect parallel data word in the transmitted PRBS data every time a positive edge occurs on the register bit associated with the Inject Error button. So, an incorrect data word is inserted every time the Inject Error button is selected. When a channel is in a SERDES near-end (HDOUT->HDIN) loopback, a single incorrect data word injected by a channel generator does not always correspond to a single count increment in the checker error counter. The injected error can cause the error counter to increment by as much as three counts. This is attributed to the nature of the PRBS checker design.

PRBS Selection button: This button allows the selection of either 2^7 or 2^31 PRBS generation and checking.

PRBS CH DISABLED/ENABLED: This enables/disables both the transmission of PRBS data from the generator, and the detection of errors by the checkers of a quad. When this button is disabled, the PRBS ERROR counters in the visual window stop incrementing, and the real time output PRBS Error signal indicator remains low (see Figure 1 and Table 2).

PRBS Generator/Checker User Registers Map

The user-accessible registers for the PRBS generator/checker quad are defined in Table 1. All register addresses are in hexadecimal. Also note that register address h00800 (not shown in Table 1) is a read-only register that contains the version number of the design.

Table 1. User Registers Map

GUI Option	PCSA FPGA Register	Description
	Ch3	
PRBS SEL	08000, bit 0	0=2^7-1 1=2^31-1
PRBS CH 3 Enabled/Disabled	08000, bit 7	0=disable 1=enable
PRBS ERR CNT	08004, bits [0:7]	Count up to flip-flop. Clear on read.
Inject Error	08000, bit 2	Write 0 then 1 to inject error.
Clear Error Cntr	08000, bit 3	Write 0 to clear.

LatticeECP3 Versa Evaluation Board Setup

There are two evaluations that can be done using the SERDES Eye/Backplane Demo design. The first demo is to loop the PRBS data back to the LatticeECP3 and check the data. The second demo evaluates the CML eye diagram of the high-speed data signal to a DSO. These setups assume the following.

1. ispVM System software is installed on a PC. The ispVM installation executable is included with the Diamond design software.

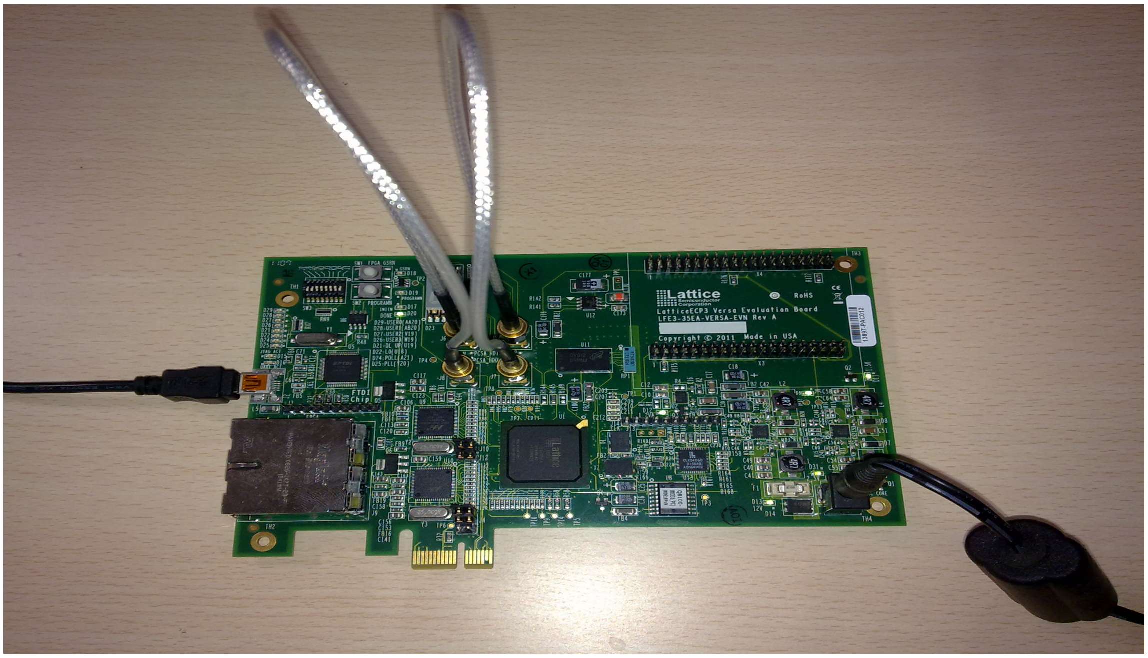
2. ORCAstra is installed on a PC. The ORCAstra installation executable is included with the Diamond design software.

3. The board is connected to the PC using the USB cable supplied. The USB cable is connected to J2 on the board.

4. Power is applied to the board via the provided power supply.

Figure 3 shows the board setup for Loopback test.

Figure 3. LatticeECP3 Versa Evaluation Board Setup for Loopback Test



SERDES Eye/Backplane Demo Design Signal Descriptions

Table 2 lists all the SERDES Eye/Backplane Demo Design signals that are connected on the LatticeECP3 Versa Evaluation Board.

Table 2. Signal Descriptions

Signal Name	Type	Board Connection	Description
reset_n	I	SW1 Push Button	FPGA global active low reset
tck	I	To on-board JTAG logic	JTAG pins
tdi	I		
tdo	O		
tms	I		
Error_3	O	Red LED D27 lights when errors occur	PRBS error indicator for channel 3. Will not light if PRBS channel is disabled in the ORCAstra PRBS Console window.
hdinp_3/ hdinn_3	I	SMA J5 and J6	Channel 3 differential high-speed SERDES inputs
hdoutp_3/ hdoutn_3	O	SMA J7 and J8	Channel 3 differential high-speed SERDES outputs

Loading the LatticeECP3 SERDES Eye Demo Bitstream with ispVM

Follow the instructions below to load the SERDES Eye demo bitstream.

1. Start ispVM System software by selecting **Start >Programs >Lattice Diamond 1.2 >Accessories >ispVM System**.

   For Linux , type **ispvm** in the command line. Please refer to the Lattice Diamond Installation Notice for proper setting of the Environment Variables.

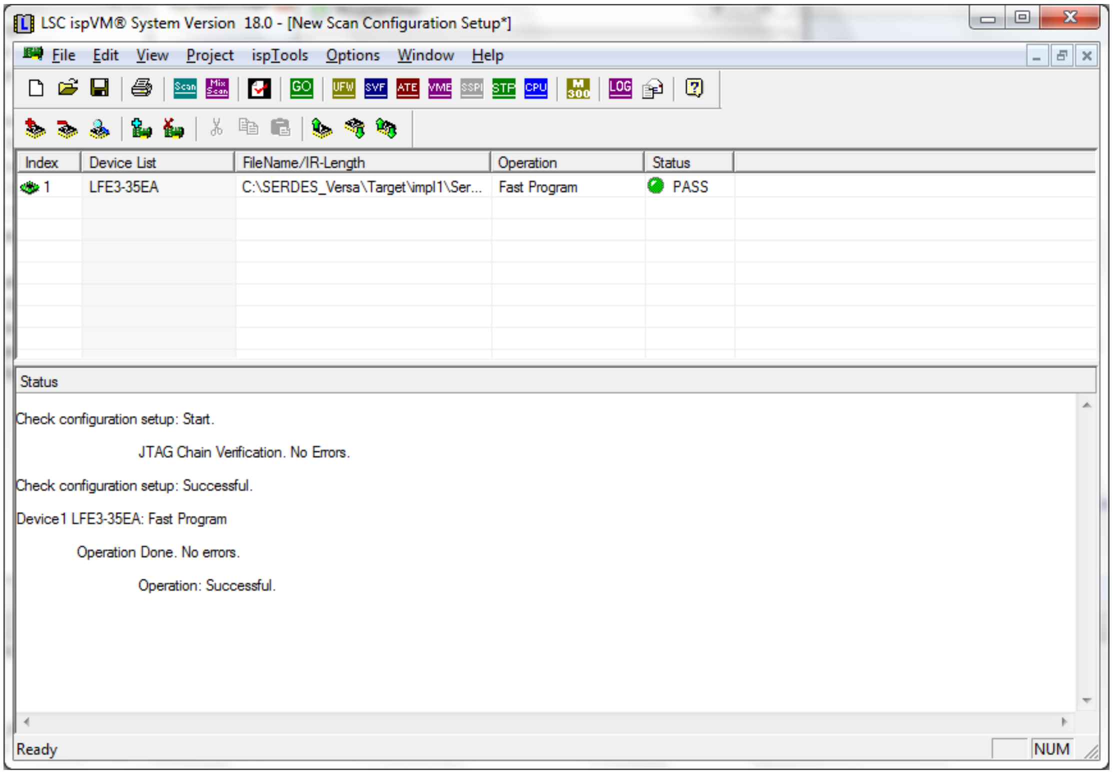
2. In the ispVM window, select the green **Scan** button (see Figure 4). LFE3-35EA will appear in the device list.

3. If ispVM does not detect the board or the device, select **Options >Cable and I/O Port Setup**. The Cable and I/O Port Setup window opens. Select **Auto Detect** and select **USB2**. Select **OK**. Select the green **Scan** button again and ensure LFE3-35EA is in the device list.

   If the board is not detected in the Linux platform, refer to the ispVM System Linux Installation and Setup Manual (ispVMLinuxInstallation.pdf) in the ispVM System installation directory.

4. Double-click on **LFE3-35EA** in the device list. The Device Information window will appear.
5. Set Device Access Options to **JTAG1532 Mode** and Operation to **Fast Program** (default setting).
6. Set the Data File to <*demo_directory*>**\Bitstreams\Serdes_Eye_Demo_impl1.bit** and select **OK**.
7. Select **Go** in the ispVM window. ispVM will download the bitstream to the LatticeECP3 device. A successful download of the bitstream is indicated by a green icon and “PASS” in the Status column as shown in Figure 4.

Figure 4. ispVM Setup



Index	Device List	FileName/IR-Length	Operation	Status
1	LFE3-35EA	C:\SERDES_Versa\Target\impl1\Ser...	Fast Program	● PASS

Status

```
Check configuration setup: Start.
                    JTAG Chain Verification. No Errors.
Check configuration setup: Successful.
Device1 LFE3-35EA: Fast Program
          Operation Done. No errors.
                    Operation: Successful.
```

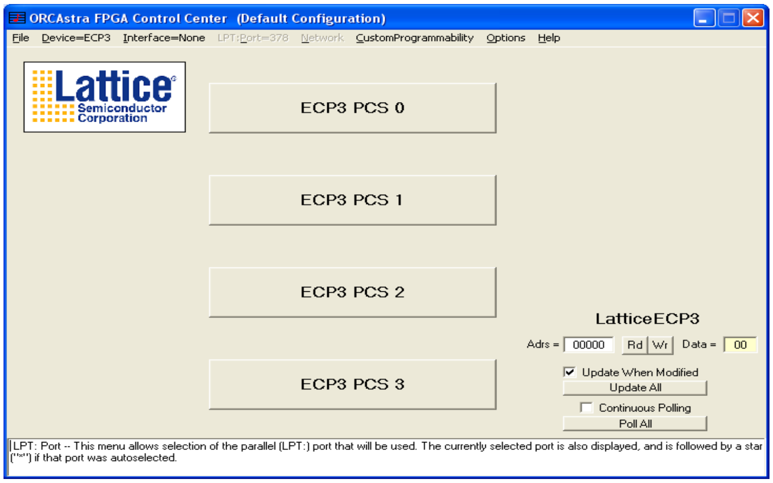
Running a Demo with the ORCAstra PCS View

This section describes the use of ORCAstra to interactively change and monitor the LatticeECP3 PCS and user-accessible registers. This section is for the Windows platform only.

Starting ORCAstra

Ensure the LatticeECP3 Demo bitstream has been loaded, then:

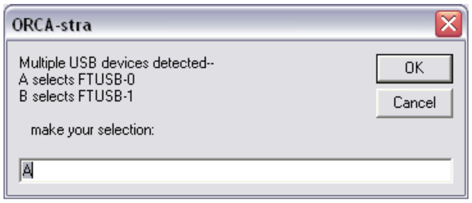
1. Start the ORCAstra software by invoking **ORCA-stra.exe** from the ORCAstra directory (ex: **C:\lscc\diamond\1.2\orcastra\**. The following ORCAstra FPGA Control Center window appears:



LPT: Port -- This menu allows selection of the parallel (LPT:) port that will be used. The currently selected port is also displayed, and is followed by a star ("*") if that port was autoselected.

2. Select **Interface > 1 ispVM JTAG Hub USB Interface**. The system detects two Devices. Select **A** and select **OK**, as shown in Figure 5.

Figure 5. Multiple Devices Window

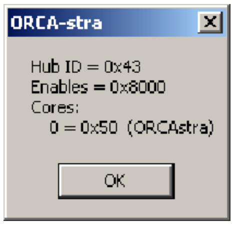


Multiple USB devices detected--
A selects FTUSB-0
B selects FTUSB-1

make your selection:

A

3. You will see the window shown below. Select **OK**.

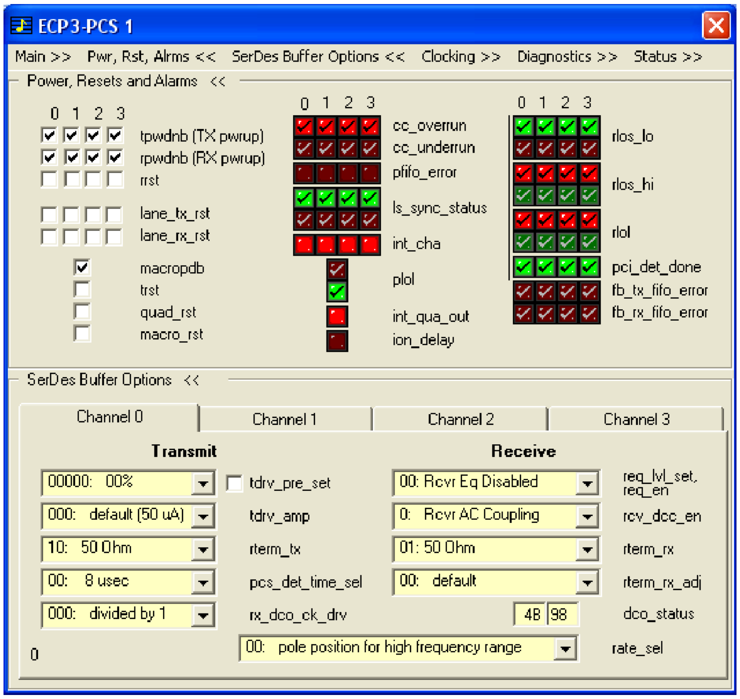


Hub ID = 0x43
Enables = 0x8000
Cores:
0 = 0x50 (ORCAstra)

4. Select **Device >4 Lattice ECP3**. Also select **Options** and un-check **Display Data in [7:0] Order in Data Box**.

5. Select the **ECP3 PCS1** button. The ECP3-PCS 1 window will appear. Select the **Main** tab, and select the **Pwr, Rst, Alrms**, and **SerDes Buffer Options** tabs. The resulting ECP3-PCS 1 window is shown in Figure 6.

Figure 6. Lattice PCS ORCAstra View



0 1 2 3
☑ ☑ ☑ ☑ tpwdnb (TX pwrup)
☑ ☑ ☑ ☑ rpwdnb (RX pwrup)
☐ ☐ ☐ ☐ rrst
☐ ☐ ☐ ☐ lane_tx_rst
☐ ☐ ☐ ☐ lane_rx_rst
☑ macropdb
☐ trst
☐ quad_rst
☐ macro_rst

Transmit		Receive	
00000: 00%	☐ tdrv_pre_set	00: Rcvr Eq Disabled	req_lvl_set, req_en
000: default (50 uA)	tdrv_amp	0: Rcvr AC Coupling	rcv_dcc_en
10: 50 Ohm	rterm_tx	01: 50 Ohm	rterm_rx
00: 8 usec	pcs_det_time_sel	00: default	rterm_rx_adj
000: divided by 1	rx_dco_ck_drv	48 98	dco_status
0		00: pole position for high frequency range	rate_sel

6. From the main ORCAstra Window, select **CustomProgrammability-> Visual Window**.
7. In the new window, select **File->Open** and navigate to select <*demo_directory*>**\ORCAstra Plug-ins\EyeDemo.vis**. You will see the PRBS Gen./Check Visual Window shown in Figure 2.
8. Make sure **Continuous Polling** is checked in the main ORCAstra window. This ensures that changes in the GUI are instantly reflected in the hardware registers.

Configuring PCS 1 Options in ORCAstraECP3-PCS 1 Window

Power, Reset and Alarms

The default Pwr, Resets and Alarms section contains the following important information (See Figure 6):

- A single red or green LED indicates the status of the entire quad or all of the four channels of the quad.
- There are two LEDs, red and green for plol. If the green LED glows, it means a successful PLL lock. If the red LED glows, it means the PLL failed to lock. Be sure PLL is locked before continuing with the demo.
- The green and red LEDs (one per channel) indicate Receive CDR lock (rlol). Green indicates a successful lock.

This view also allows the user to identify which channels (or the entire QUAD), are powered down or reset. This view also allows users to reset PCS digital logic (lane_tx_rst and lane_rx_rst), as well as SERDES logic (macro_rst) and the whole QUAD (quad_rst). For more details, refer to TN1176, LatticeECP3 SERDES/PCS Usage Guide.

SERDES Buffer Options View

This view allows controlling the characteristics of output, input, and reference clock buffers: TX pre-emphasis, TX amplitude, RX equalization, TX and RX buffer termination and coupling. These options can be changed by clicking the Channel 3 tab in the ECP3-PCS1 window. Refer to SERDES Buffer Options Section shown in Figure 6.

Typical Backplane Demo Applications

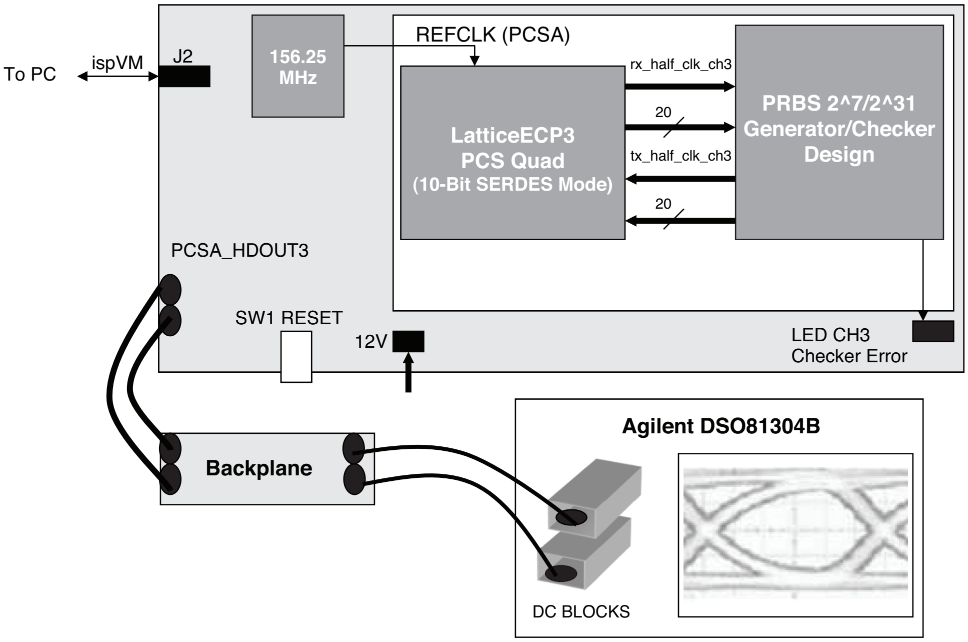
As mentioned earlier, the serial data stream can be:

- Looped back via a cable on the evaluation board, or
- Sent to a DSO or DCA for eye viewing. This document describes using a DSO. The DCA requires an external trigger input.

SERDES Eye/Backplane Demo on PCS Quad PCSA using DSO

A typical DSO SERDES Eye/Backplane Demo application with PCS PCSB is illustrated in Figure 7.

Figure 7. Typical DSO SERDES Backplane Eye Demo on PCS Quad PCSA



The hardware setup is as follows:

- The LatticeECP3 Versa Evaluation Board has an on-board reference clock (ispClock5406A device) that applies differential clock inputs to the PCSA reference clock input.
- Connect the SMA cables from the channel 3 high speed outputs, J7 and J, to SMA terminals on the backplane. If you do not want to use a backplane, connect the SMA cables from J7 and J8 directly to the DC blocks on the DSO.
- Setup the Agilent DSO81304B instrument using the setup file: <*demo_directory*>**\Misc\ DSO81394B_setup.set**

In previous sections of this user's guide, you have already connected the power supply to the LatticeECP3 Versa Evaluation Board, loaded the LatticeECP3 SERDES Eye/Backplane demo bitstream to the board, invoked ORCAstra, and customized the ORCAstra interface with the PRBS Generator/Checker Console window and ECP3-PCS 1 windows on your computer.

Perform the following steps to complete the demo:

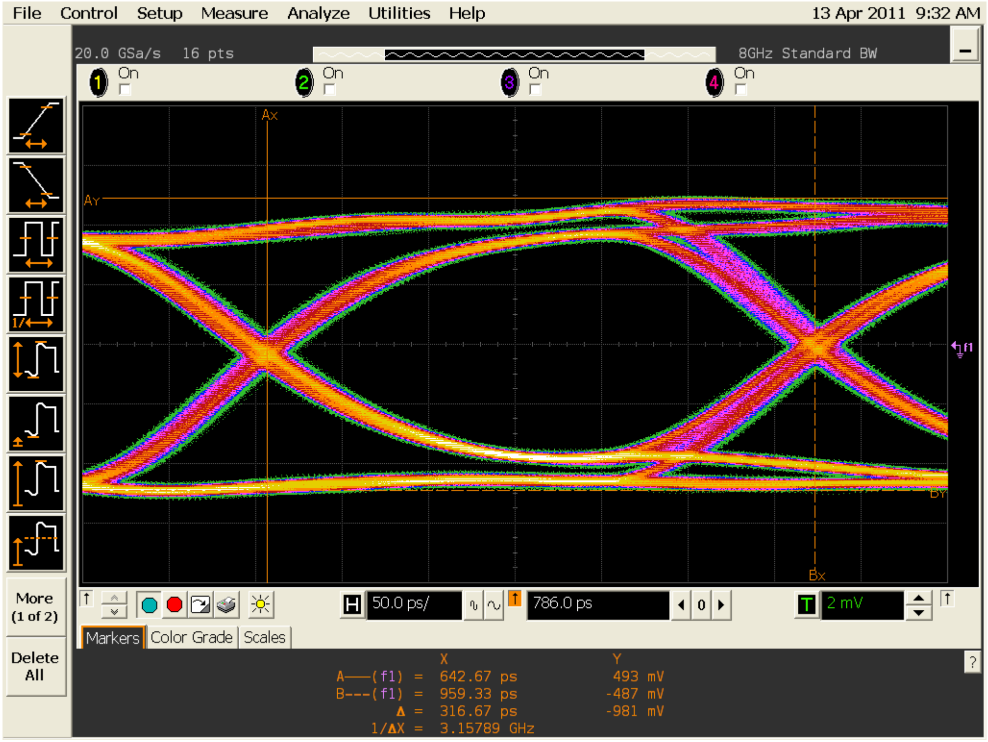
1. In the ORCAstra FPGA Control Center window, be sure to check the **Continuously Polling** check box.
2. Press **SW1 FPGA GSRN** on the LatticeECP3 Versa Evaluation Board to re-acquire the ref clock.

3. In the Power, Resets, and Alarms tab of the ECP3-PCS 1 window, power down channels 0, 1 and 2 (uncheck tpwdnb and rpwdnb for channels 0, 1, and 2; Channel 3 must remain checked).
4. In the PRBS Generator/checker Console window, select **2^7** or **2^31** for the PRBS pattern. If you select 2^31, then also set Receive equalization to **11: Long-Reach Eq** in the SERDES Buffer Option tab and Channel 3 tab, of the ECP3-PCS 1 window.
5. In the PRBS Generator/checker Console window, make sure PRBS channel 3 is enabled.

As a result of running the demo steps, a PRBS eye should appear on the DSO screen. Figure 8 illustrates a 3.125 Gbps differential eye based on 2^7 PRBS mode, the SERDES buffer settings, and the SERDES output SMAs J7 and J8 connected directly to the DC blocks (no backplane).

In a typical backplane application, the SERDES Buffer Options of PCS1 in ORCAstra (Figure 6) allows real-time tweaking of output buffer characteristics, as different backplane lengths are utilized to maintain a clean SERDES eye diagram at the DSO input. The eye in Figure 8 can change if any of the TX buffer options, such as amplitude, pre-emphasis, and coupling, are changed.

Figure 8. DSO Differential Eye Diagram with PCS Quad PCSA and No Backplane



```
                 X               Y
A——(f1) =  642.67 ps         493 mV
B---(f1) =  959.33 ps        -487 mV
       Δ =  316.67 ps        -981 mV
    1/ΔX =  3.15789 GHz
```

SERDES Eye/Backplane PRBS Loopback Demo

In a typical PRBS backplane loopback application:

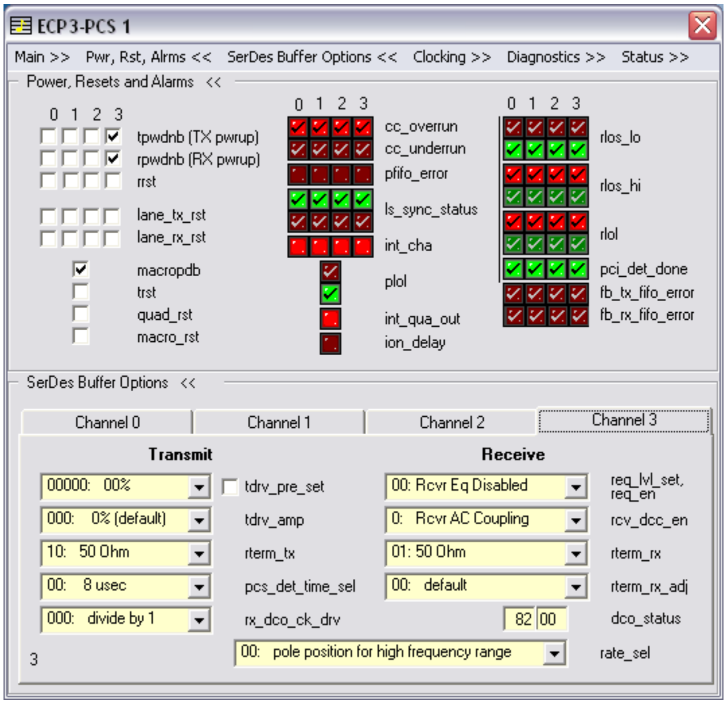
- The FPGA PRBS Generator is used to transmit data from the FPGA to the PCS TXD ports on a given channel.
- The PCS SERDES HDOUT pins are connected to a backplane through SMA cables. The other backplane terminals connect to the PCS HDIN channel inputs through SMA cables. *Note: To do the loopback demo without a backplane, use the SMA cables to directly connect HDOUT to HDIN on the LatticeECP3 Versa Evaluation Board.*
- The PCS RXD ports then feed the recovered data to the FPGA PRBS Quad Checker.

This type of application is illustrated in Figure 3. This setup applies to PCSA Quad, Channel 3 which has been brought out on the LatticeECP3 Versa Evaluation Board.

This application can make use of the ORCAstra PRBS Generator/Checker Quad Console window from Figure 2 and the SERDES Buffer Options in the ORCAstra ECP3-PCS 1 window (see Figure 6). While looping PRBS data on channel 3 the PRBS Generator/checker Console window is used to verify error-free PRBS data is received, while the ECP3-PCS 1 window is used to tweak output and input buffer options (as different backplane lengths are used).

The following steps describe how to run through a PRBS loopback demo on channel 3. The HDIN and HDOUT SMA (P, N) pairs are (J5 and J6) and (J7 and J8) respectively.

1. Make sure power is supplied to the LatticeECP3 Versa Evaluation Board.
2. Make sure HDOUT3 is looped back to HDIN3. *Note: To do the loopback demo without a backplane, use the SMA cables to directly connect HDOUT to HDIN on the LatticeECP3 Versa Evaluation Board.*
3. Press **SW1 FPGA GSRN** on the LatticeECP3 Versa Evaluation Board to re-acquire the reference clock.
4. Load the LatticeECP3 SERDES Demo bitstream as previously described.
5. Start an ORCAstra session and load the ECP3-PCS 1 window and the PRBS Generator/Checker Console window as previously described.
6. Make sure the **Continuous Polling** box is checked in the ORCAstra FPGA Control Center window.
7. In the Power, Resets, and Alarms tab of the ECP3-PCS 1 window, power down channels 0, 1 and 2 (uncheck tpwdnb and rpwdnb for channels 0, 1 and 2).



0	1	2	3	
□	□	□	☑	tpwdnb (TX pwrup)
□	□	□	☑	rpwdnb (RX pwrup)
□	□	□	□	rrst
□	□	□	□	lane_tx_rst
□	□	□	□	lane_rx_rst

☑ macropdb
□ trst
□ quad_rst
□ macro_rst

Transmit

00000: 00% — □ tdrv_pre_set
000: 0% (default) — tdrv_amp
10: 50 Ohm — rterm_tx
00: 8 usec — pcs_det_time_sel
000: divide by 1 — rx_dco_ck_drv

Receive

00: Rcvr Eq Disabled — req_lvl_set, req_en
0: Rcvr AC Coupling — rcv_dcc_en
01: 50 Ohm — rterm_rx
00: default — rterm_rx_adj
82 00 — dco_status
00: pole position for high frequency range — rate_sel

3

8. In the PRBS Generator/Checker Quad Visual Window:
   A. Select 2^7 or 2^31 for PATTERN. If you select 2^31, then also set Receive equalization to **11: Long-Reach Eq** in the SERDES Buffer Option tab, channel 3, of the ECP3-PCS 1 window.

B. ENABLE channel 3. The generator and checker for channel 3 are now transmitting and checking PRBS packets. After this step, rlol should be solid green. Also verify that plol is solid green. If any of these indicators are red, then proceed to debugging these indicators as indicated in step 7D.

C. Clear all counters by pressing the **Clear Error Cntr** button.

D. If the ERR COUNTER for channel 3 increments and/or the D27 error LED lights up, then the PRBS checker is receiving patterns with errors. Note that the error counters will not rollover after reaching hFF. It is necessary to reset the counter to ensure no new errors are received.

E. Verify that the Power, Reset and Alarm section in the ECP3-PCS 1 window does not show any PLL loss of lock (plol) or CDR loss of lock (rlol).

   a. A red plol indicates that the reference clock source to the TX PLL is not stable or may have the incorrect frequency.

   b. A red rlol indicates incorrect activity on the HDIN* inputs. The input signals may be too attenuated by the medium.

F. Modifying some of the input and output buffer settings in the SERDES Buffer Options section of ECP3-PCS 1 window. (ex: TX pre-emphasis, TX amplitude, RX equalization) to address the rlol issue.

Implementing and Simulating the Reference Design

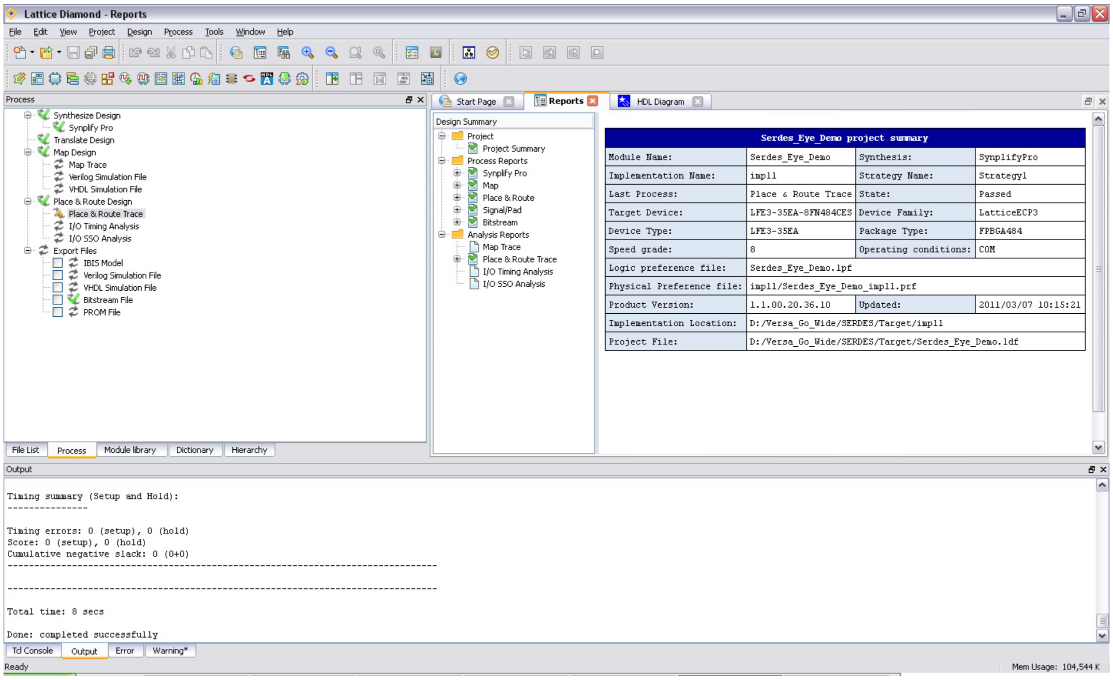
The steps below explain how to run the SERDES Eye/Backplane Demo reference design source code through Diamond map, place and route, bitstream generation, and simulation. Simulation is supported in Windows only.

Both implementation and simulation start with the same steps:

1. Open Lattice Diamond design software.

2. Open <*demo_directory*>**\Target\Serdes_Eye_Demo.ldf**. This will load the Verilog-based project, as shown in Figure 9.

Figure 9. Diamond Verilog HDL Design



Serdes_Eye_Demo project summary			
Module Name:	Serdes_Eye_Demo	Synthesis:	SynplifyPro
Implementation Name:	impl1	Strategy Name:	Strategy1
Last Process:	Place & Route Trace	State:	Passed
Target Device:	LFE3-35EA-8FN484CES	Device Family:	LatticeECP3
Device Type:	LFE3-35EA	Package Type:	FPBGA484
Speed grade:	8	Operating conditions:	COM
Logic preference file:	Serdes_Eye_Demo.lpf		
Physical Preference file:	impl1/Serdes_Eye_Demo_impl1.prf		
Product Version:	1.1.00.20.36.10	Updated:	2011/03/07 10:15:21
Implementation Location:	D:/Versa_Go_Wide/SERDES/Target/impl1		
Project File:	D:/Versa_Go_Wide/SERDES/Target/Serdes_Eye_Demo.ldf		

```
Timing summary (Setup and Hold):
---------------

Timing errors: 0 (setup), 0 (hold)
Score: 0 (setup), 0 (hold)
Cumulative negative slack: 0 (0+0)
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------

Total time: 8 secs

Done: completed successfully
```

Implementation

To implement the design and generate a bitstream, double-click **Bitstream File** in the **Process** window. This will run through the full synthesis, place and route flow and generate a new serdes_eye_demo_impl1.bit file in the impl1 folder.

Simulation

The simulation process uses the **custom.do** Aldec Active-HDL simulation script located under the **Target** directory. To ensure this script is used, follow the steps below, as illustrated in Figures 10 to 14:

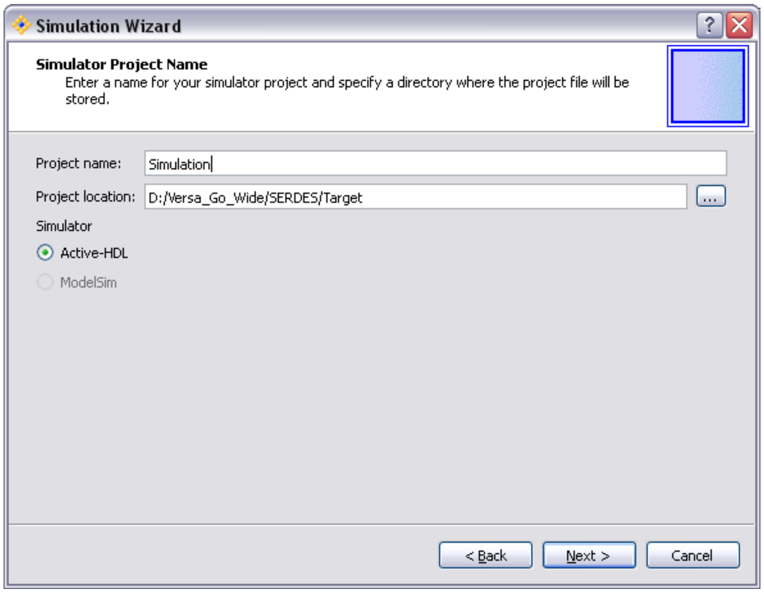
1. Open the Lattice Diamond design tool. Select **Tools->Simulation Wizard**. Select **Next** and provide the name of your choice as the project name. Select **Next**. **RTL** should be selected. Select **Next**.

2. In the **Add Source** window click the '**+**' button and add the test bench file **..SERDES\Sim\Src\aaa_SERDES_tb.v**. Select **Next** and **Finish**.

3. The Active-HDL environment will open. Select **Tool- >Execute Macro** and select the **custom.do** file.

4. The script compiles all necessary design and test bench files and runs the simulation into the Aldec Waveform window shown in Figure 15.

Figure 10. Configuring Aldec Simulation



Simulation Wizard

Simulator Project Name

Enter a name for your simulator project and specify a directory where the project file will be stored.

Project name: Simulation

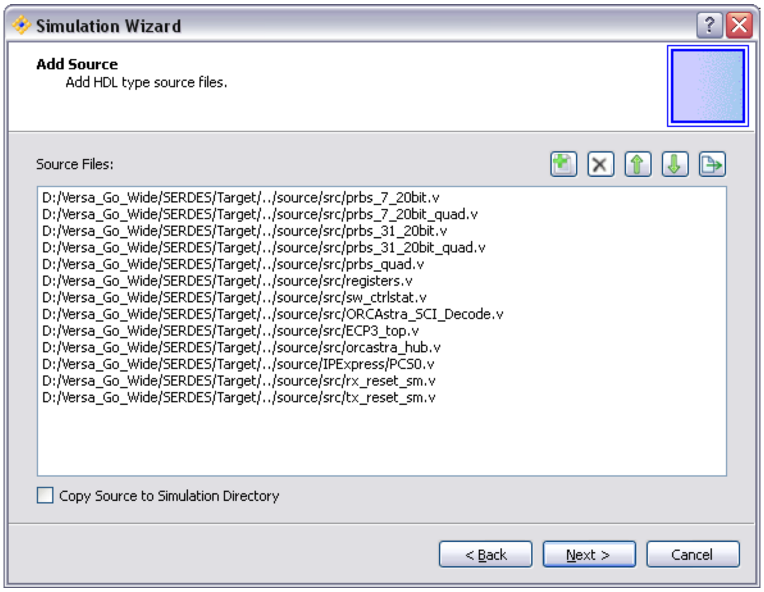
Project location: D:/Versa_Go_Wide/SERDES/Target

Simulator

● Active-HDL

○ ModelSim

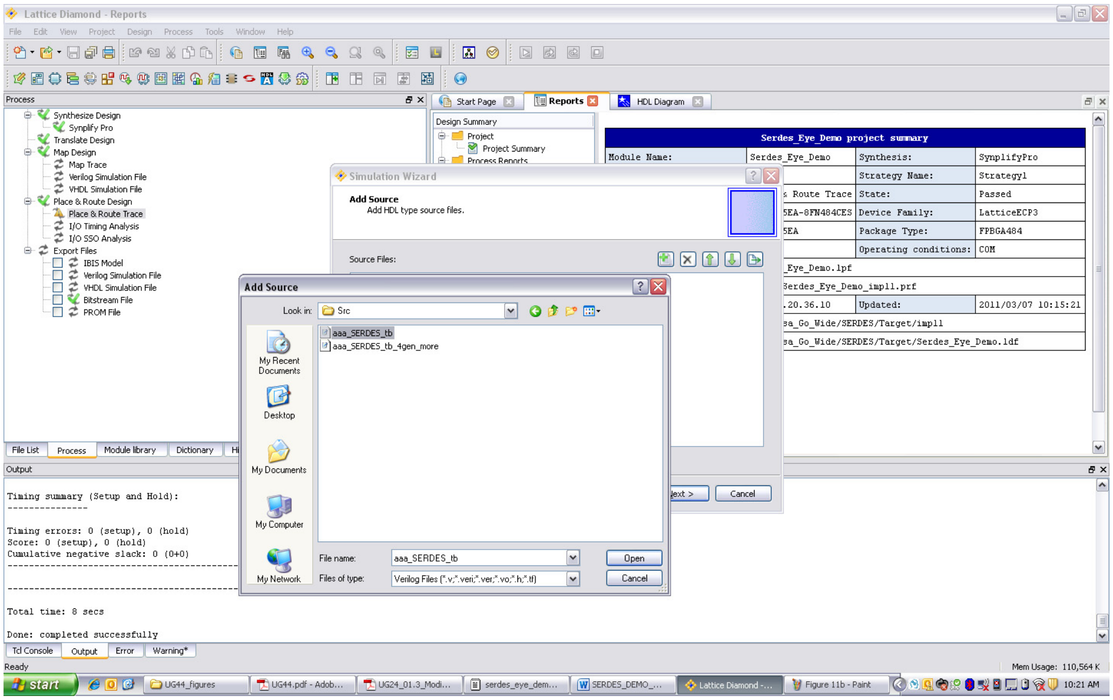
Figure 11. Add Source Window



```
D:/Versa_Go_Wide/SERDES/Target/../source/src/prbs_7_20bit.v
D:/Versa_Go_Wide/SERDES/Target/../source/src/prbs_7_20bit_quad.v
D:/Versa_Go_Wide/SERDES/Target/../source/src/prbs_31_20bit.v
D:/Versa_Go_Wide/SERDES/Target/../source/src/prbs_31_20bit_quad.v
D:/Versa_Go_Wide/SERDES/Target/../source/src/prbs_quad.v
D:/Versa_Go_Wide/SERDES/Target/../source/src/registers.v
D:/Versa_Go_Wide/SERDES/Target/../source/src/sw_ctrlstat.v
D:/Versa_Go_Wide/SERDES/Target/../source/src/ORCAstra_SCI_Decode.v
D:/Versa_Go_Wide/SERDES/Target/../source/src/ECP3_top.v
D:/Versa_Go_Wide/SERDES/Target/../source/src/orcastra_hub.v
D:/Versa_Go_Wide/SERDES/Target/../source/IPExpress/PCS0.v
D:/Versa_Go_Wide/SERDES/Target/../source/src/rx_reset_sm.v
D:/Versa_Go_Wide/SERDES/Target/../source/src/tx_reset_sm.v
```

□ Copy Source to Simulation Directory

Figure 12. Adding Test Bench



Serdes_Eye_Demo project summary			
Module Name:	Serdes_Eye_Demo	Synthesis:	SynplifyPro
		Strategy Name:	Strategy1
	& Route Trace	State:	Passed
	5EA-8FN484CES	Device Family:	LatticeECP3
	5EA	Package Type:	FPBGA484
		Operating conditions:	COM
	_Eye_Demo.lpf		
	Serdes_Eye_Demo_impl1.prf		
	.20.36.10	Updated:	2011/03/07 10:15:21
	sa_Go_Wide/SERDES/Target/impl1		
	sa_Go_Wide/SERDES/Target/Serdes_Eye_Demo.ldf		

Look in: Src

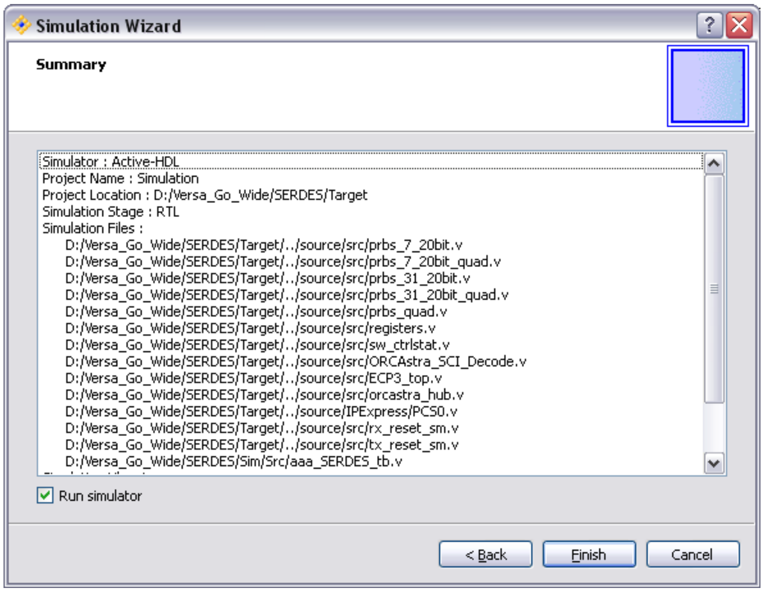
aaa_SERDES_tb
aaa_SERDES_tb_4gen_more

File name: aaa_SERDES_tb

Files of type: Verilog Files (*.v;*.veri;*.ver;*.vo;*.h;*.tf)

```
Timing summary (Setup and Hold):
---------------

Timing errors: 0 (setup), 0 (hold)
Score: 0 (setup), 0 (hold)
Cumulative negative slack: 0 (0+0)
-----------------------------------------


-----------------------------------------

Total time: 8 secs

Done: completed successfully
```

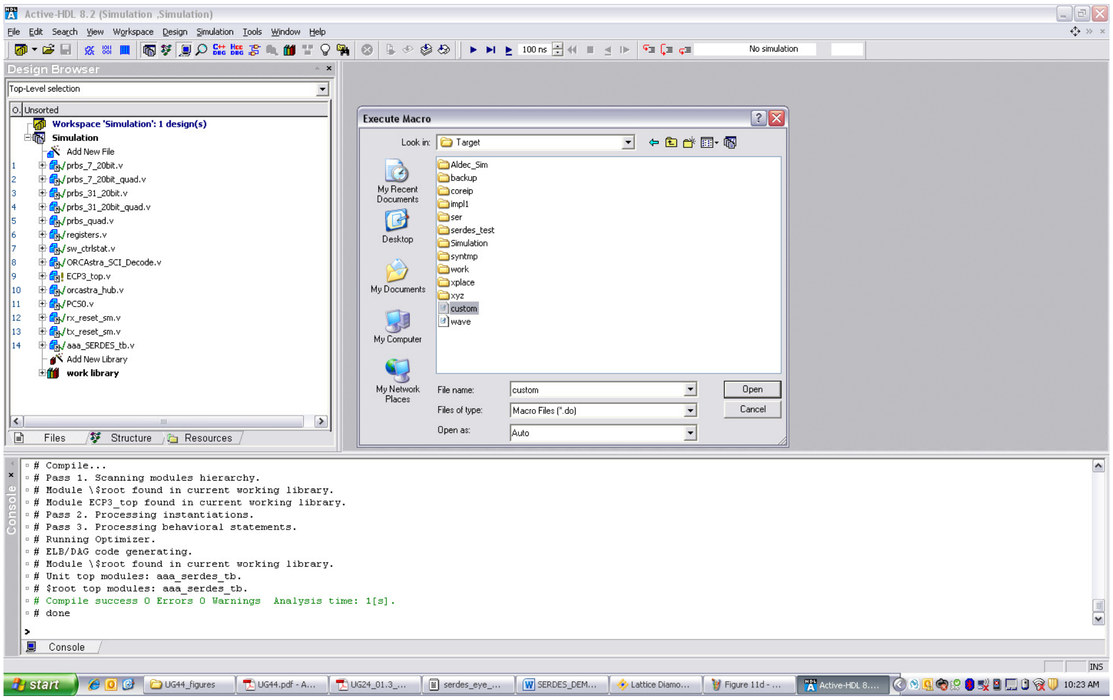
Figure 13. Summary of Sources



```
Simulator : Active-HDL
Project Name : Simulation
Project Location : D:/Versa_Go_Wide/SERDES/Target
Simulation Stage : RTL
Simulation Files :
    D:/Versa_Go_Wide/SERDES/Target/../source/src/prbs_7_20bit.v
    D:/Versa_Go_Wide/SERDES/Target/../source/src/prbs_7_20bit_quad.v
    D:/Versa_Go_Wide/SERDES/Target/../source/src/prbs_31_20bit.v
    D:/Versa_Go_Wide/SERDES/Target/../source/src/prbs_31_20bit_quad.v
    D:/Versa_Go_Wide/SERDES/Target/../source/src/prbs_quad.v
    D:/Versa_Go_Wide/SERDES/Target/../source/src/registers.v
    D:/Versa_Go_Wide/SERDES/Target/../source/src/sw_ctrlstat.v
    D:/Versa_Go_Wide/SERDES/Target/../source/src/ORCAstra_SCI_Decode.v
    D:/Versa_Go_Wide/SERDES/Target/../source/src/ECP3_top.v
    D:/Versa_Go_Wide/SERDES/Target/../source/src/orcastra_hub.v
    D:/Versa_Go_Wide/SERDES/Target/../source/IPExpress/PCS0.v
    D:/Versa_Go_Wide/SERDES/Target/../source/src/rx_reset_sm.v
    D:/Versa_Go_Wide/SERDES/Target/../source/src/tx_reset_sm.v
    D:/Versa_Go_Wide/SERDES/Sim/Src/aaa_SERDES_tb.v
```

☑ Run simulator

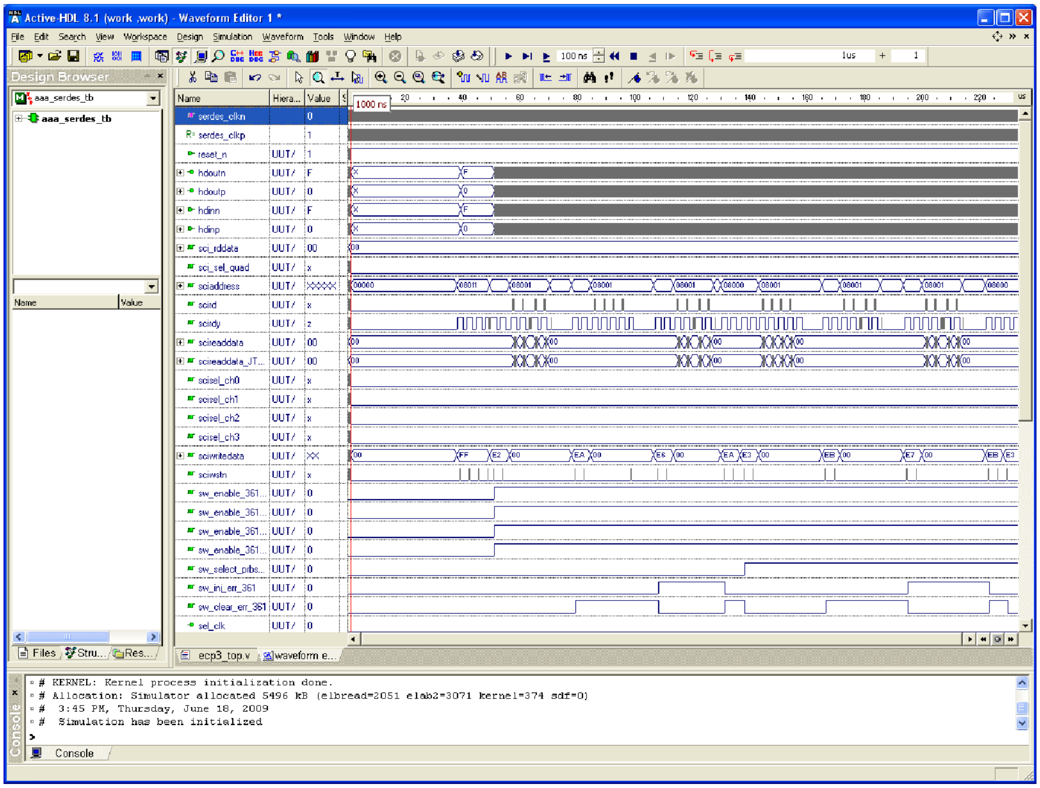
Figure 14. Executing Custom.do File



File name: custom
Files of type: Macro Files (*.do)
Open as: Auto

```
# Compile...
# Pass 1. Scanning modules hierarchy.
# Module \$root found in current working library.
# Module ECP3_top found in current working library.
# Pass 2. Processing instantiations.
# Pass 3. Processing behavioral statements.
# Running Optimizer.
# ELB/DAG code generating.
# Module \$root found in current working library.
# Unit top modules: aaa_serdes_tb.
# $root top modules: aaa_serdes_tb.
# Compile success 0 Errors 0 Warnings  Analysis time: 1[s].
# done
>
```

Figure 15. Aldec Active-HDL Simulation



```
# KERNEL: Kernel process initialization done.
# Allocation: Simulator allocated 5496 kB (elbread=2051 elab2=3071 kernel=374 sdf=0)
#  3:45 PM, Thursday, June 18, 2009
#  Simulation has been initialized
>
```

References

The following documents provide more information:

- TN1176, LatticeECP3 SERDES/PCS Usage Guide
- DS1021, LatticeECP3 Family Data Sheet
- EB62, LatticeECP3 Versa Evaluation Board User's Guide

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

Revision History

Date	Version	Change Summary
April 2011	01.0	Initial release.
May 2011	01.1	Updated steps for demos to include pressing of GSRN.