Hspice Differential IO Kit User's Manual

Simulation of Lattice SC Product LVDS and other differential Interfaces

OVERVIEW

The Lattice HSpice IO Kit contains a collection of HSpice model files that allow LVDS and other differential data link simulation across a PCB module or backplane hardware system. Examples of other differential buffers that can be simulated with this model are HyperTransport, RSDS, mini-LVDS inputs/outputs and LVPECL inputs. The differential buffer models are extracted from product final design files and are believed to closely reflect real device performance. The flexible, high-resolution capabilities of HSpice simulation allow signal integrity assessment of hardware device and interconnection designs. The HSpice IO Kit is available to Lattice customers and prospective customers under Non-Disclosure Agreement (NDA).

This document describes the HSpice simulation files provided in the IO Kit and how to use them to perform some basic PCB interconnection design signal analysis. The IO Kit's provides a single channel, top level HSpice simulation file. It allows selection and adjustment of hardware elements, operational parameters, and simulation conditions, extending from the Tx output buffer to the Rx input buffer. The signal path can include different device packages, backplane connectors, and PCB transmission line traces, which are included in the IO Kit model library. Signal eye-diagram waveform analysis is used to display data signals as they propagate across the signal path of the system under test (SUT).

Several simulation examples are presented in this document. Differential signal patterns and PCB interconnection design schemes as presented, to demonstrate a range of possible application simulations. The customer is encouraged to replace or modify the board level element HSpice models with one that appropriately represents his specific application.

SIMULATOR SOFTWARE AND FILE STRUCTURE

The Synopsys Hspice Circuit Simulator, Version 2002.2.2 or later, is recommended for use with this kit. It is assumed that the user has a working knowledge of the HSpice tool. Performing a simulation requires a top-level (.sp type) file that is loaded by the HSpice simulator, prior to running. It in turn selects all the appropriate element model files that are needed. "**.include**" statements in the top file identify actually link the element model files. These files reside in the LIBR directory of the IO Kit. The top file also contains simulation selection statements and output waveform statements that control the simulator operation and output file generation.

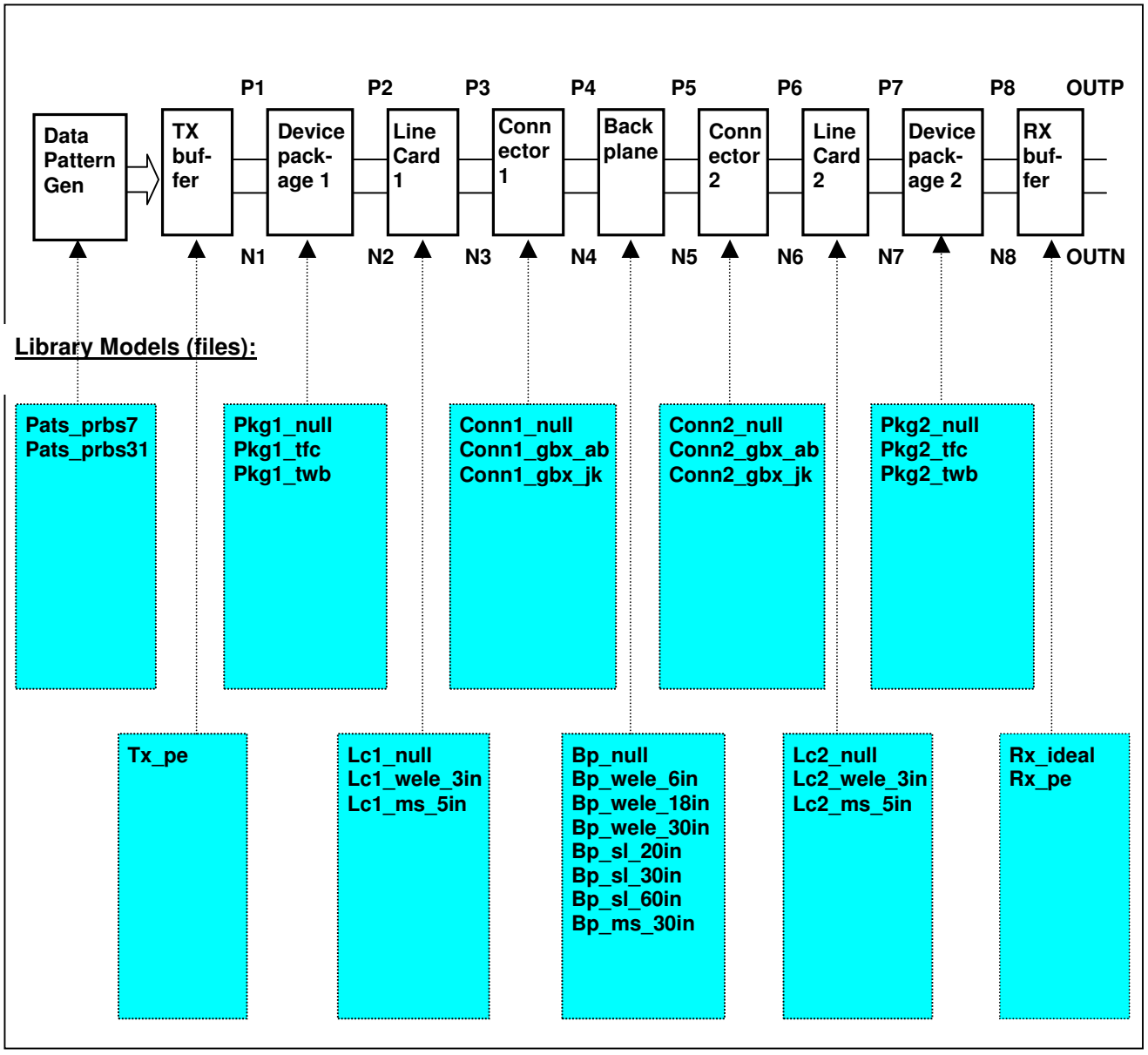
Experienced HSpice simulation users may generate or may already have their own top-level simulations files that include other portions of their application system. They may choose to just use the buffer and package model files that are provided in the IO Kit, calling them into their top-level simulation file. The user will simply need to identify the IO Kit files (models) of interest and provide appropriate instantiation, connection and stimulation. The information needed to do this can be found in the details of the IO Kit top file and the comments contained in the specific model files of interest. Note that some of the model files have been

encrypted to protect Lattice proprietary information. The encryption is transparent to the HSpice simulation tools.

TESTBENCH TOPOLOGY AND MODEL LIBRARY

A generic testbench.sp file is provided that configures a signal path topology across a backplane, as is shown at the top of the Figure 1.

FIGURE 1



This top-level file has provisions to select from most of the element models included in the IO Kit. One model from each of the library model boxes in Figure 1 is selected to configure a complete system for simulation. The left-most box of library files, in Figure 1, lists the available data test patterns to provide stimulus for the data link. Note that in each library file block a _null suffix file is included. Selection of this file eliminates the model

for that block element and simply connects the input terminals to the output terminals. This configuration option allows the user to remove a specific path from the simulation. Comparing this result to that with the element present can often provide useful information about the incremental effect of an element insertion into a system.

The testbench.sp file specifies a transient type simulation analysis, with voltage waveform outputs saved for each nodes indicated. Additional output parameters are defined to permit observation of Different signals and eye-diagram signals. The simulation results may be displayed with the HSpice viewer Awaves, or other compatible viewers.

In the simulation analysis & control section, the transient simulation time step size and overall time interval are specified in the .tran statement. The user as appropriate in each application should set these values. The other control parameters of interest to the user are in various model files, as listed in the SIMULATION PARAMETER CONTROL section of this file.

When the extracted buffer models are used, the device operating temperature may be specified in the .temp xx and .param temp=xx statements. The user may also select typical (TT), fast (FF) or slow (SS) process libraries.

TESTBENCH LIBRARY FILE DESCRIPTIONS

PATTERN GENERATION MODELS

The digital vector file feature of HSpice is used to generate data pattern stimulus to the TX buffer models. In addition to the serial 1-0 pattern, each .dvf file sets the data bit rate of the simulation. The data bit period is set in the "period" line near the beginning of each file. The number in this line should be set to the desired data rate bit period (in pS). The following models are provided in the library:

pats_prbs7.dvf

This is an Hspice digital vector file designed to interface the TX buffer model tx_pe.cir. This file provides the industry standard PRBS7 pattern (polynomial X^7+X^6+1). This 127 bit pattern is repeated twice in the file.

pats_prbs31.dvf

This is an Hspice digital vector file designed to interface the TX buffer model tx_pe.cir. This file provides a segment of the PRBS31 (polynomial X^31+X^28+1). A variety of pattern characteristics can be seen in different regions of the overall sequence. A segment with longer run lengths and greater dc imbalance is provided. The segment is 512 bits long.

pats_cjpat.dvf

This is an Hspice digital vector file designed to interface the TX buffer circuits tx_pe.cir. This file provides a shortened version of the CJPAT test pattern used in XAUI standards testing. It contains each of the repeating patterns found in the full length CJPAT. The total length is 320 bits.

BUFFER MODELS

The LVDS TX buffer model provided is described below.

tx_pe.cir

This model was derived from the IC layout description of the LVDS driver section of the SC FPGA SYSio cell. It is a full transistor level model. It includes extracted devices and metal layer connection parasitic capacitances. Termination resistors are not provided in this model, since they are not normally turned on at the TX buffer output.

This file can be modified to select 1 of 4 different amplitude levels by "uncommenting" the appropriate line in the file. This is done by removing the asterisk (*) at the beginning of the appropriate line in the Output Amplitude Section of the file. The nominal output current levels are 2, 3.5, 4 or 6 mA, as indicated in each line. This model is driven by a one of the pattern generation files described in the previous selection.

Typical Usage

Output Type	Current Level Setting
LVDS	3.5 mA
HyperTransport	6 mA
RSDS	2 mA
Mini-LVDS	2 mA

The RX LVDS Buffer models provided are described below.

rx_ideal.cir

This model is simply a 120-ohm ideal resistor, differential termination between the P and N inputs. This model provides no buffer output nodes.

rx_pe.cir

This model contains a partial circuit extraction section, taken from the SC FPGA SYSio cell, and a functionally derived section. The LVDS differential amplifier is the circuit extraction section. It consists of a full transistor level circuit representation, including metal layer parasitic capacitance elements. The internal termination section is functionally modeled. This model should provide accurate receiver amplifier response and operation and low to moderate simulation run time.

The input termination impedance mode can be change between 4 different modes. This is done by uncommenting 1 of 4 lines in the Termination Impedance Selection section of the file. The file provides some additional bias circuitry to operate the extracted model in the correct mode and calls the detailed model files (ios_ibuf.pex.netist and ios_ibuf.pxi). The models single-ended signal output node is designated rout. Power supply voltages are set in this file and may be modified from the initial nominal levels.

Typical Usage

Output Type	Internal Termination
LVDS	120 ohm or 120 ohm CT
HyperTransport	120 ohm
RSDS	120 ohm
Mini-LVDS	120 ohm or 120 ohm CT
LVPECL	external

IC Package

Only the P and N signal leads of a single signal pair and ground are included in the model. The model includes package substrate and wire bond or ball bond portions of the interconnection.

pkg1_null.cir
This is the zero order TX output signal connection path model that essentially eliminates the element from the simulation signal path. It provides a zero delay, zero parasitic impedance connection between adjacent elements.

pkg1_twb.cir
This model represents the TX output signal connection path through a typical wire-bond PBGA package. This model is appropriate for the SC product 256, 484, 672 and 900 ball packages.

pkg1_tfc.cir
This model represents the TX output signal connection path through a typical flip-chip PBGA package. This model is appropriate for the SC product 672, 896, 1152, 1517 and 1704 ball packages.

Pkg2_null.cir
This is the zero order RX output signal connection path model that essentially eliminates the element from the simulation signal path. It provides a zero delay, zero parasitic impedance connection between adjacent elements.

Pkg2_twb.cir
This model represents the RX output signal connection path through a typical wire-bond PBGA package. This model is appropriate for the SC product 256, 484, 672 and 900 ball packages.

Pkg2_tfc.cir
This model represents the RX output signal connection path through a typical flip-chip PBGA package. This model is appropriate for the SC product 672, 896, 1152, 1517 and 1704 ball packages.

Line Card and Backplane PCB traces

Printed circuit board transmission line connections are modeled with the Hspice W-element lossy transmission line element. This model is well accepted in the industry and includes provisions for frequency dependent copper and dielectric loss. The line card and backplane model file described below use RLC form models. The “wele” named files represent two matched 50 ohm lines with no coupling provision between the two (P and N signal) lines. The loss parameters used are intended to approximate the losses of a FR4 stripline structure, but they have not verified for accuracy. The “ms” named files represent micro-strip, FR4 structures which are comparable in loss to a real structure with 8 mil edge-coupled P and N lines that was measured in our lab. The “sl” named files represents stripline, FR4 structures, which are comparable in loss to a real structure with 6 mil edge, coupled P and N lines.

Customers wishing to achieve accurate simulation results for their PCB applications will need to develop their own models to reflect the specific line characteristics.

lc1_wele_3in.cir

This model represents the line card signal interconnection at the TX side. It provides 3-inch long segments (P and N lines) of the W-element lossy line described above. The L parameter value defined at the end of the WP1 and WN1 lines may be changed to vary the PCB trace length. L is the length of each line, in meters.

lc1_ms_5in.cir

This model represents the line card signal interconnection at the TX side. It provides 5-inch long segments (P and N lines) of the “ms” type, described above. The L parameter value defined at the end of the WP1 and WN1 lines may be changed to vary the PCB trace length. L is the length of each line, in meters.

lc2_wele_3in.cir

This model represents the line card signal interconnection at the RX side. It provides a 3 inch long segment of the W-element line described above. The L parameter value defined at the end of the WP1 and WN1 lines may be changed to vary the 3-inch length. L is the length of each line, in meters.

Lc2_ms_5in.cir

This model represents the line card signal interconnection at the RX side. It provides 5-inch long segments (P and N lines) of the “ms” type, described above. The L parameter value defined at the end of the WP1 and WN1 lines may be changed to vary the PCB trace length. L is the length of each line, in meters.

bp_wele_6in.cir

This model represents the backplane PCB interconnection signal traces. It provides a 6-inch long segment of the W-element line pair as described above. The L parameter value defined at the end of the WP2 and WN2 lines may be changed to vary the PCB trace length. L is the length of each line, in meters.

bp_wele_18in.cir

This model represents the backplane PCB interconnection signal traces. It provides an 18-inch long segment of the W-element line pair as described above.

bp_wele_30in.cir

This model represents the backplane PCB interconnection signal traces. It provides a 30-inch long segment of the W-element line pair as described above.

bp_ms_20in.cir

This model represents the backplane PCB interconnection signal traces. It provides a 20-inch long segment of the “ms” type line pair as described above.

bp_ms_30in.cir

This model represents the backplane PCB interconnection signal traces. It provides a 30-inch long segment of the “ms” type line pair as described above.

bp_sl_30in.cir

This model represents the backplane PCB interconnection signal traces. It provides a 30-inch long segment of the “sl” type line pair as described above.

bp_ms_60in.cir

This model represents the backplane PCB interconnection signal traces. It provides a 60-inch long segment of the “ms” type line pair as described above.

bp_null.cir

This is the zero order model, which essentially eliminates the element from the simulation signal path. It provides a zero delay, zero parasitic impedance connection between adjacent elements.

Several controlled impedance backplane connector models were provided by vendors and are included in the library. A brief description of each follows:

conn1_gbx_ab.cir

This model represents backplane connector at the TX side. It uses the Teradyne GBX 8-row connector, row a/b (shortest length) pin pair. This file calls the Teradyne pin model in file gbxab.cir.

conn1_gbx_jk.cir

This model represents backplane connector at the TX side. It uses the Teradyne GBX 8-row connector, row j/k (longest length) pin pair. This file calls the Teradyne pin model in file gbxjk.cir.

conn1_null.cir

This is the zero order model, which essentially eliminates the element from the simulation signal path. It provides a zero delay, zero parasitic impedance connection between adjacent elements.

conn2_gbx_ab.cir

This model represents backplane connector at the RX side. It uses the Teradyne GBX 8-row connector, row a/b (shortest length) pin pair. This file calls the Teradyne pin model in file gbxab.cir. Note: if conn1_gbx_ab.cir is selected in the testbench file, the .include line in this file must be commented out (* added as the first character of the line), to avoid a compiling error.

conn2_gbx_jk.cir

This model represents backplane connector at the RX side. It uses the Teradyne GBX 8-row connector, row j/k (longest length) pin pair. This file calls the Teradyne pin model in file gbxjk.cir. Note: if conn1_gbx_jk.cir is selected in the testbench file, the .include line in this file must be commented out (* added as the first character of the line), to avoid a compiling error.

conn2_null.cir

This is the zero order model, which essentially eliminates the element from the simulation signal path. It provides a zero delay, zero parasitic impedance connection between adjacent elements.

SIMULATION EYE-DIAGRAM DISPLAY

A recursive time variable parameter (teye) is defined in the .print statement of the testbench file. When one of the 4 versions of this parameter (teye0, teye1, teye2 or teye3) is selected as the x variable in an Awaves display panel, an eye-diagram waveform of any output variable may be created. To obtain a clean eye-diagram display, the following must also be done:

1. The TX buffer clock rate period (set in tx_XXX.cir) must equal the eye-diagram period (set in testbench.sp).

2. The eye-diagram display initialization time (set in testbench.sp) must be set to allow waveform stablization from circuit transient start up anomalies. This portion of the simulation waveform will appear to the left of the eye-diagram section of the display.

3. The 4 teye parameter versions allow 4 different horizontal positionings of the eye-diagram display. The one which best centers the eye is usually best.

4. The output variable in the display should be selected in Awaves; then right-click in the display and toggle the "continuous display" to the "monotonic plot" mode. This will eliminate the retrace lines in the eye-diagram display.

5. The X-zoom feature in Awaves may be used to show only the eye-diagram portion of the initial display.

SIMULATION PARAMETER CONTROL

The testbench.sp file and the library model files control the parameters of the transient simulation normally performed. The following list identifies the file locations for the controllable simulation parameters.

1. Simulation time interval and step size -- testbench.sp (.tran line)
2. Buffer supply voltages -- are set in the tx_* and rx_* model files. The voltage parameters are identified with comments in those files.
3. Extracted buffer model temperature -- testbench.sp (.temp XX line and .param temp=xx)
4. Extracted buffer process model selection -- testbench.sp (uncomment appropriate .include line(s)
5. Eye-Diagram waveform display time period -- testbench.sp (per parameter)
6. Eye-Diagram display initialization time -- testbench.sp (dly parameter)
7. Setting TX buffer clock (data) rate -- pats_XXX.dvf (set 'per' parameter in the data pattern file being used)
8. TX buffer output amplitude -- Uncomment appropriate line in tx_pe.cir model file being used. Available options are explained with comments in the files.
9. RX buffer internal termination impedance -- Uncomment appropriate line in rx_pe.cir model file. Available options are explained with comments in the file.
10. TX line card PCB trace length can be changed in any of the lc1_XXX.cir files (length L is designated in meters).
11. RX line card PCB trace length can be changed in any of the lc2_XXX.cir files (length L is designated in meters).
12. Backplane PCB trace length can be changed in any of the bp_XXX.cir files (length L is designated in meters).

OTHER LIBRARY MODELS

Several additional Hspice model files are provided in the libr directory that are not supported in the testbench. These are more complex package and connector models that may be useful for application crosstalk

simulations. The user must create a new testbench or modify the provided testbench to make use of these models.

gbx_xt.cir

Teradyne GBX connector model - 6 adjacent signal conductors with cross coupling. Provided courtesy of Teradyne. (Note this model is not set up for use with the provided testbench file.)

***

Four low-level models files, ios_lvdsobuf.pex.netist, ios_lvdsobuf.pxi, ios_ibuf.pex.netlist, and ios_ibuf.pxi, are the detailed buffer circuit extraction files from the modeled product devices. These files are partially encrypted to protect Lattice proprietary design information. The low level files are called (included) by higher-level files tx_pe and rx_pe.

Two low-level model files, gbx_ab and gbx_jk, are present in the LIBR directory. These files are subroutines used by connector model files already described.

There are 3 library device files, defining key device elements for the extracted circuit models. These files are partially encrypted to protect Lattice and Fujitsu proprietary design information. The file names are cs100a_g, cs100a_ll_rev1.03, and cs100a_models. These files must be present in the libr/ directory when extracted buffer models are called by the testbench.

SIMULATION EXAMPLES

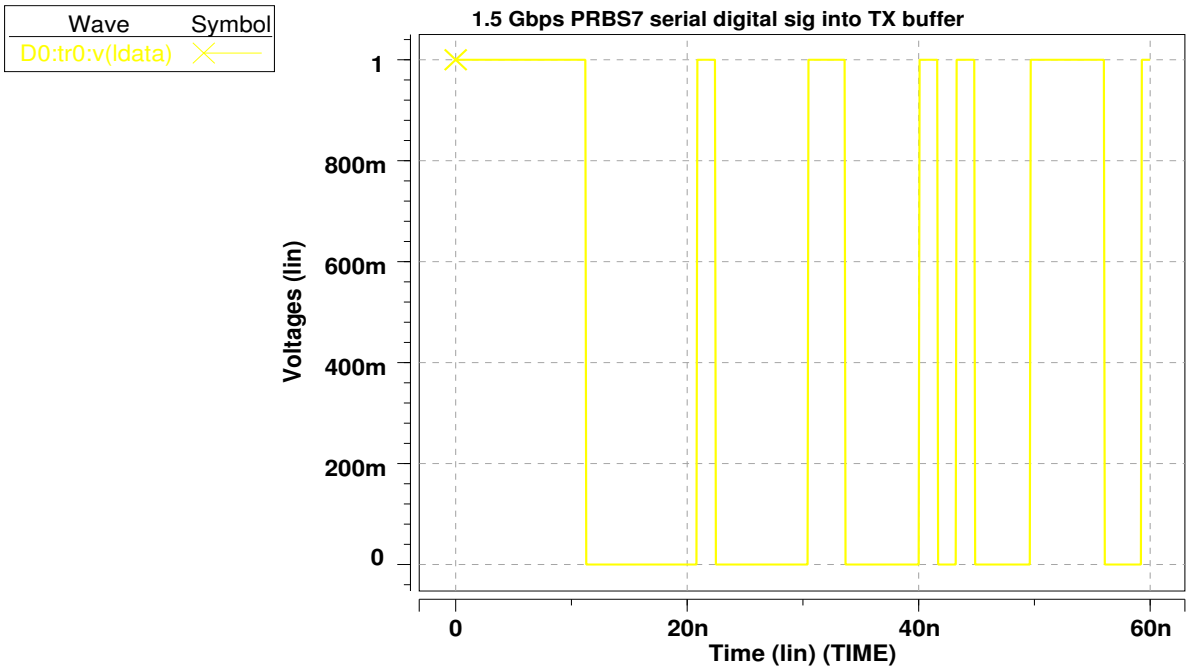
The first 3 examples described in this section show a progression of test system complexity. The first model uses the simplest function TX buffer model only and a single package and transmission line segment interconnection to an ideal. The second and third examples add the RX buffer model, demonstrating logical signal generation and recovery across a data link.

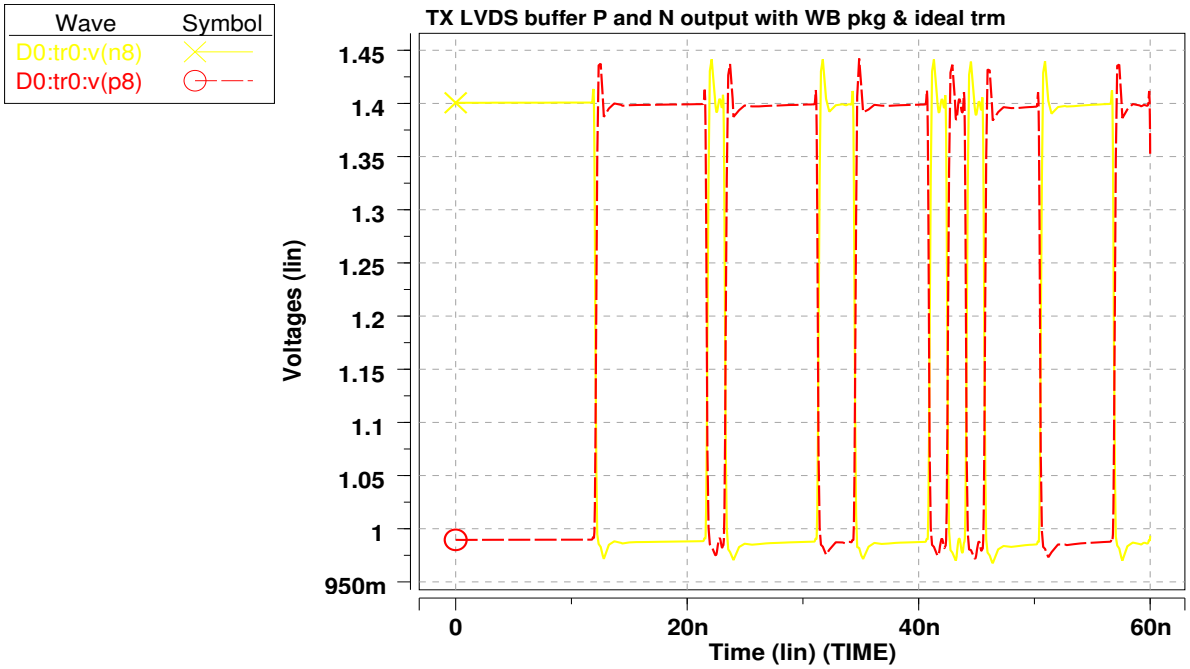
The .sp top-level files used for these simulations are included in the IO Kit, to provide a reference for users wishing to verify operation of their simulation environment or verify their understanding of the IO Kit testbench use.

EXAMPLE 1

This example uses a minimal test bench configuration. A PRBS7 pattern generation element drives the TX buffer. The buffer output is passed through a package model element and through a 6 inch lossy line to an ideal resistor termination element. The digital signal input waveform is shown as well as buffer P and N signal output waveforms, at the end of the transmission line segments.

Selected Model Elements in Simulation		
Pat Gen: prbs7, 1.5 GBPS	**Conn 1:** none	**Ln Crd 2:** none l
TX Buf: partial extracted, 3.5 mA	**Bck Pln:** 6 inch loss pcb	**Pkg 2:** none l
Pkg 1: typical wire-bond	**Conn 2:** none	**Rx Buf:** 120 ohm ideal
Ln Crd 1: none	**Comment:**	





14:54:03 EDT, 09/12/2005

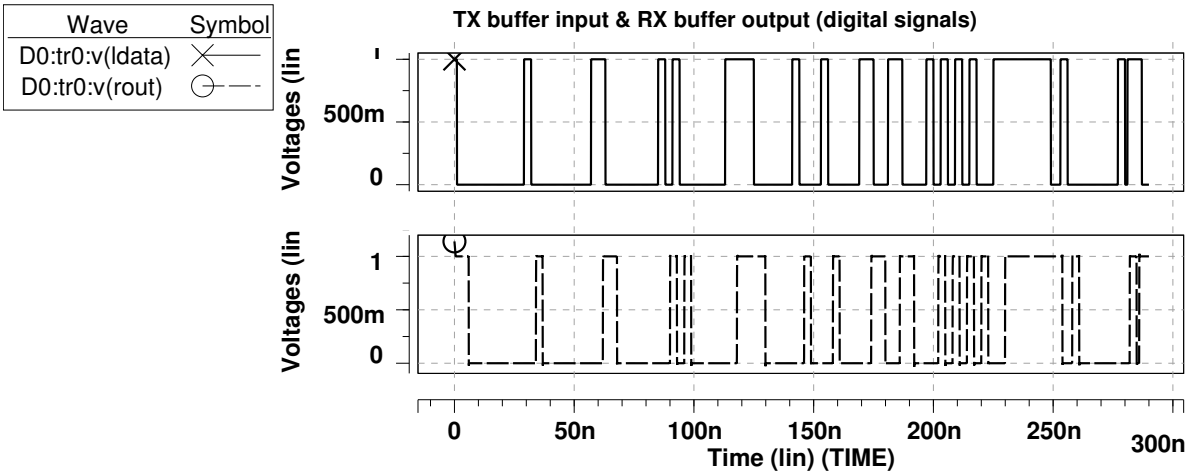
Example1.sp *top-level simulation file*

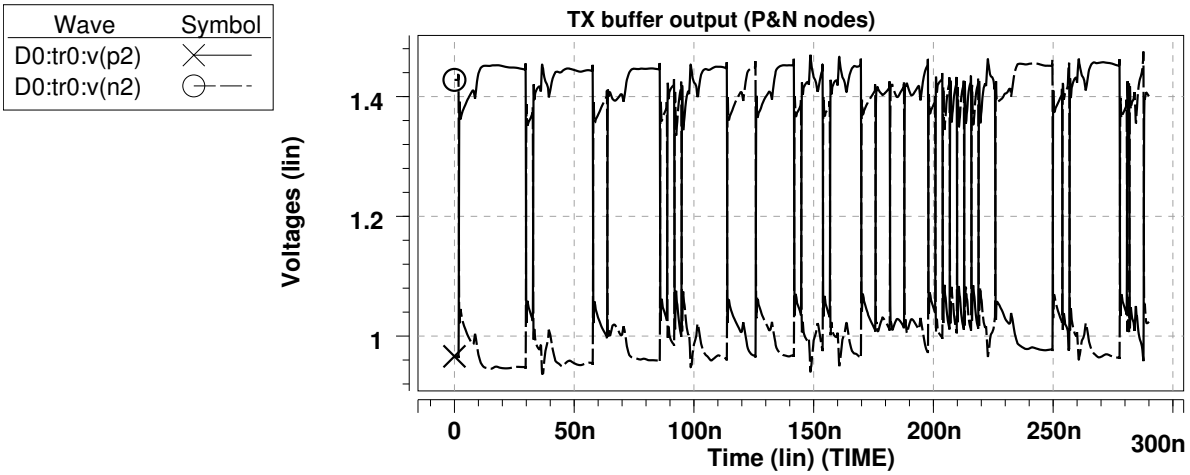
```
* Lattice SC FPGA LVDS buffer example 1 simulation

*****************************************************************
************* LATTICE SEMICONDUCTOR CORPORATION **************
**** CONFIDENTIAL AND SUBJECT TO NON-DISCLOSURE AGREEMENT ****
*****************************************************************
.include tx_pe.cir
.vec pats_prbs7.dvf
.include ../libru/pkg1_tfc.cir
.include ../libru/lc1_null.cir
.include ../libru/conn1_null.cir
.include ../libru/bp_null.cir
.include ../libru/conn2_null.cir
.include ../libru/lc2_null.cir
.include ../libru/pkg2_null.cir
.include ../libru/rx_ideal.cir

****************** library model selections *********************
.lib ../libru/cs100a_models TT

***************** simulation analysis & control *****************
.options method=gear
.options probe post=2 ingold=2 brief
.param temp=70
.temp 70
.tran .3ns 60ns
.PARAM per=1.6ns
.PARAM dly=10ns

.print tran
+          v(ldata)
+          v(rout)       $ RX output
+          V(p8)   V(n8) $ RX input at chip
+          v(p7)     v(n7) $ RX package input
+          v(p6)     v(n6) $ RX line card PCB trace input
+          v(p5)     v(n5) $ backplane PCB trace end (at RX end)
+          v(p4)     v(n4) $ backplane PCB trace end (at TX end)
+          v(p3)     v(n3) $ TX line card PCB trace output
+          v(p2)     v(n2) $ TX package output
+          v(p1)   v(n1) $ TX output at chip
* RX input differential voltage
+   dif_rxin=par('v(p8)-v(n8)')
* Eye Diagram display time variables
+   teye0=par('TIME-int((TIME-dly)/per)*per-dly')
+   teye1=par('TIME-int((TIME-dly-per/4)/per)*per-dly-per/4')
+   teye2=par('TIME-int((TIME-dly-per/2)/per)*per-dly-per/2')
+   teye3=par('TIME-int((TIME-dly-per*3/4)/per)*per-dly-per*3/4')

.end
```

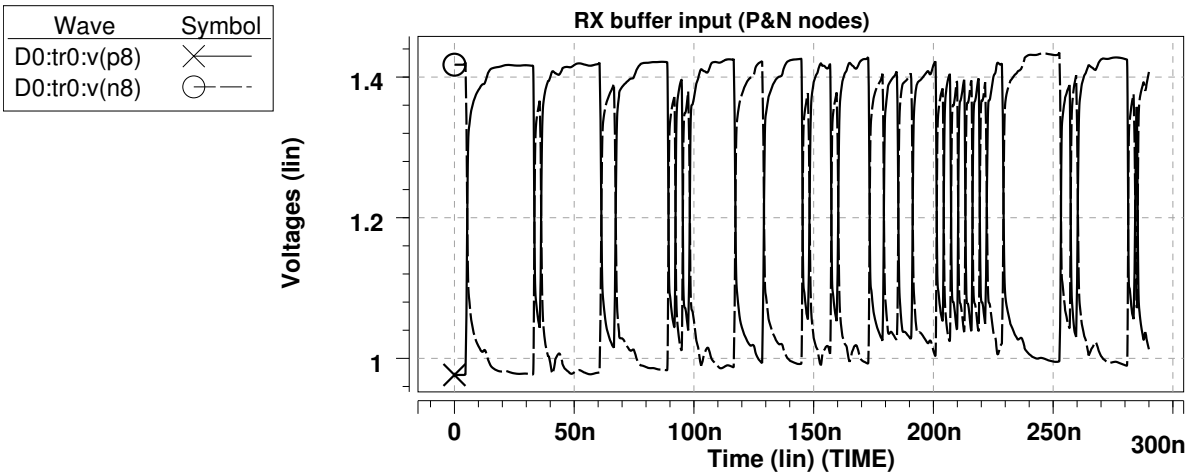
EXAMPLE 2

This example demonstrates a 625 Mbps serial data link interconnection across an 18-inch lossy line segment. Both the TX buffer and RX buffer are included in the simulation. Digital signals in and out of the data link are observed, as well as the differential buffer P and N signals at the transmission line ends.

Selected Model Elements in Simulation		
Pat Gen: cjpat, 625 Mbps	**Conn 1:** none	**Ln Crd 2:** none
TX Buf: partial extraction, 4 mA	**Bck Pln:** 18" lossy line	**Pkg 2:** typical, wire-bond
Pkg 1: typical, flip-chip	**Conn 2:** none	**Rx Buf:** partial extraction
Ln Crd : none	**Comment:**	







19:36:55 EST, 12/20/2005

Example2.sp *top-level simulation file*

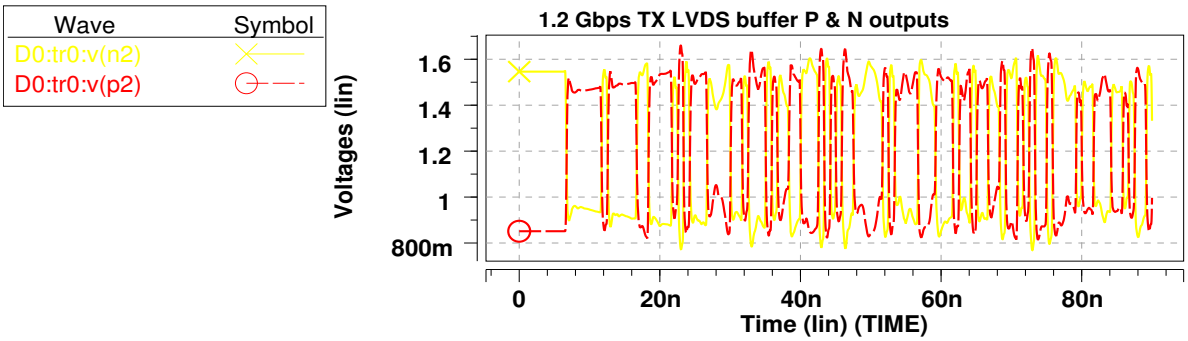
```
* Lattice SC FPGA LVDS buffer example 2 simulation

*****************************************************************
************* LATTICE SEMICONDUCTOR CORPORATION **************
**** CONFIDENTIAL AND SUBJECT TO NON-DISCLOSURE AGREEMENT ****
*****************************************************************
.include tx_pe.cir
.vec pats_cjpat.dvf
.include ../libru/pkg1_tfc.cir
.include ../libru/lc1_null.cir
.include ../libru/conn1_null.cir
.include ../libru/bp_wele_18in.cir
.include ../libru/conn2_null.cir
.include ../libru/lc2_null.cir
.include ../libru/pkg2_twb.cir
.include rx_pe.cir

****************** library model selections ***************************
.lib ../libru/cs100a_models TT


***************** simulation analysis & control ******************
.options method=gear
.options probe post=2 ingold=2 brief
.param temp=70
.temp 70
.tran .3ns 90ns
.param per=1.6n dly=8n

.print tran
+          v(ldata)
+          v(rout)       $ RX output
+          V(p8)   V(n8) $ RX input at chip
+          v(p7)     v(n7) $ RX package input
+          v(p6)     v(n6) $ RX line card PCB trace input
+          v(p5)     v(n5) $ backplane PCB trace end (at RX end)
+          v(p4)     v(n4) $ backplane PCB trace end (at TX end)
+          v(p3)     v(n3) $ TX line card PCB trace output
+          v(p2)     v(n2) $ TX package output
+          v(p1)   v(n1) $ TX output at chip
* RX input differential voltage
+   dif_rxin=par('v(p8)-v(n8)')
* Eye Diagram display time variables
+   teye0=par('TIME-int((TIME-dly)/per)*per-dly')
+   teye1=par('TIME-int((TIME-dly-per/4)/per)*per-dly-per/4')
+   teye2=par('TIME-int((TIME-dly-per/2)/per)*per-dly-per/2')
+   teye3=par('TIME-int((TIME-dly-per*3/4)/per)*per-dly-per*3/4')

.end
```

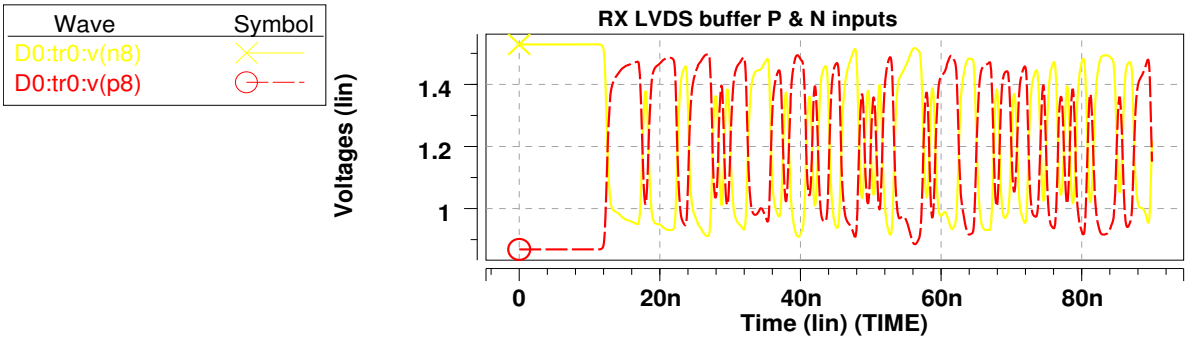
EXAMPLE 3

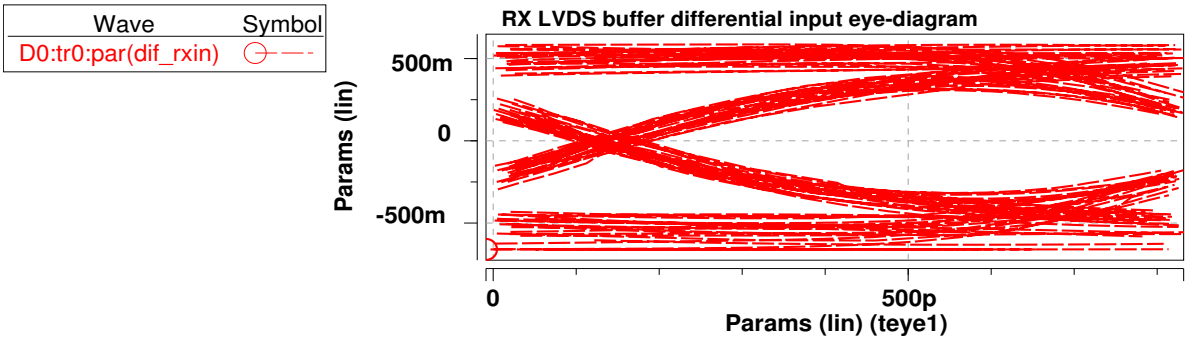
This example demonstrates the operation of a serial data link across a backplane interconnection. A 1.2 Gbps PRBS7 data pattern is applied to the TX buffer. 5 inch lines between the TX and RX buffers and the backplane connectors are assumed. A 20 inch lossy coupled -stripline pair is assumed on the backplane. P and N differential buffer waveforms at the IC package terminals are shown. The RX buffer input and output differential eye-diagram waveforms are also shown.

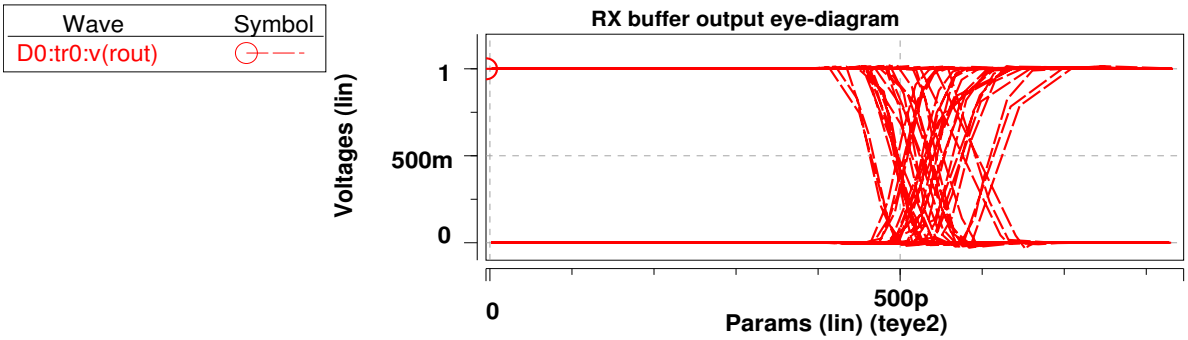
Selected Model Elements in Simulation

Pat Gen:prbs7, 1.2 GBPS	**Conn 1:** Teradyne GBX-ab	**Ln Crd 2:** 5” microstrip
TX Buf: partial extraction, 6 mA	**Bck Pln:** bp_sl_20in	**Pkg 2:** typical, flip-chip
Pkg 1: typical, wire-bond	**Conn 2:** Teradyne GBX-jk	**Rx Buf:** partial extraction
Ln Crd 1: 5” microstrip	**Comment:**	









14:27:24 EDT, 09/14/2005

Example3.sp *top-level simulation file*

```
* Lattice SC FPGA LVDS buffer example 2 simulation

************************************************************
************* LATTICE SEMICONDUCTOR CORPORATION **************
**** CONFIDENTIAL AND SUBJECT TO NON-DISCLOSURE AGREEMENT ****
************************************************************
.include tx_pe.cir
.vec pats_prbs7.dvf
.include ../libru/pkg1_twb.cir
.include ../libru/lc1_ms_5in.cir
.include ../libru/conn1_gbx_ab.cir
.include ../libru/bp_sl_20in.cir
.include ../libru/conn2_gbx_jk.cir
.include ../libru/lc2_ms_5in.cir
.include ../libru/pkg2_tfc.cir
.include rx_pe.cir

***************** library model selections ***************************
.lib ../libru/cs100a_models TT


**************** simulation analysis & control ******************
.options method=gear
.options probe post=2 ingold=2 brief
.param temp=70
.temp 70
.tran .3ns 90ns
.param per=833p dly=8n

.print tran
+          v(ldata)
+          v(rout)       $ RX output
+          V(p8)   V(n8) $ RX input at chip
+          v(p7)     v(n7) $ RX package input
+          v(p6)     v(n6) $ RX line card PCB trace input
+          v(p5)     v(n5) $ backplane PCB trace end (at RX end)
+          v(p4)     v(n4) $ backplane PCB trace end (at TX end)
+          v(p3)     v(n3) $ TX line card PCB trace output
+          v(p2)     v(n2) $ TX package output
+          v(p1)   v(n1) $ TX output at chip
* RX input differential voltage
+   dif_rxin=par('v(p8)-v(n8)')
* Eye Diagram display time variables
+   teye0=par('TIME-int((TIME-dly)/per)*per-dly')
+   teye1=par('TIME-int((TIME-dly-per/4)/per)*per-dly-per/4')
+   teye2=par('TIME-int((TIME-dly-per/2)/per)*per-dly-per/2')
+   teye3=par('TIME-int((TIME-dly-per*3/4)/per)*per-dly-per*3/4')

.end
```