EE273 DIGITAL SYSTEMS ENGINEERING
PROJECT

By

Emilio Antunez
Njuguna Njoroge
Yves Ngu

Winter 2002
Stanford University

TABLE OF CONTENTS

1. DESIGN METHODOLOGY

1.1 DESIGN APPROACH

The system is a router using two types of cards (a switch card and a line card) in a 16-slot backplane. The sixteen line cards each output and receive 2Gbps to and from two switch cards for a total data transfer of 4Gbps.

1.2 OVERVIEW OF SIGNALING APPROACH

The signals are sent over a simultaneous bi-directional transmission line, using a bipolar differential current-mode signaling scheme. A 20cm transmission line has been designed to link the cards to the backplane connectors. The primary objective in designing the trace was to match the impedance of the connector-backplane block while keeping crosstalk to a minimum. The crosstalk minimization objective was accomplished through the insertion of a ground line between our traces to isolate differential pairs from each other.

1.3 OVERVIEW OF TIMING APPROACH

To ensure that our data could be accurately sampled, bundled closed-loop timing was implemented. This timing method enabled our system to cancel multiple sources of jitter and skew. Our data needed to be sent together with its clock since we are using a bi-directional scheme.

1.4 SUMMARY OF PERFORMANCE

1.4.1 Summary of Noise Budget Analysis

The bulk of our noise came from attenuation and reverse crosstalk, to which a simultaneous bi-directional system is particularly sensitive. With various noise cancellation techniques, we were able to remain with a 100mV gross noise margin, which resulted in a BER of 4E-25. With this BER, we were below the target BER specification of 1E-20.

1.4.2 Summary of Timing Budget Analysis

Running our system at 1GHz gave us a large timing margin (500ps); however, this translated into a considerable VCO jitter (100ps). Our timing scheme enabled us to cancel multiple sources of skew and jitter. Our system ended with a 190ps timing noise and a 310ps net timing margin, which ensured that our signal would be sampled properly.

2 DESCRIPTION OF 20-CM TRACE

2.1 BOARD SPECIFICATION

2.1.1 Trace Dimensions

The IC package and the backplane connector size limit the maximum width of our transmission lines. Table 2-1 shows the dimensions of the two latter components.

Component	Dimension
BGA Flip-Chip	1mm x 1mm
Teradyne Connector (VHDM HS)	2mm x 2.25mm

Table 2-1: Component Dimensions

The BGA Flip-Chip has the smallest dimensions hence it will constrain the width of our transmission lines. Figure 2-1 shows the dimensions associated with our BGA Flip-Chip.

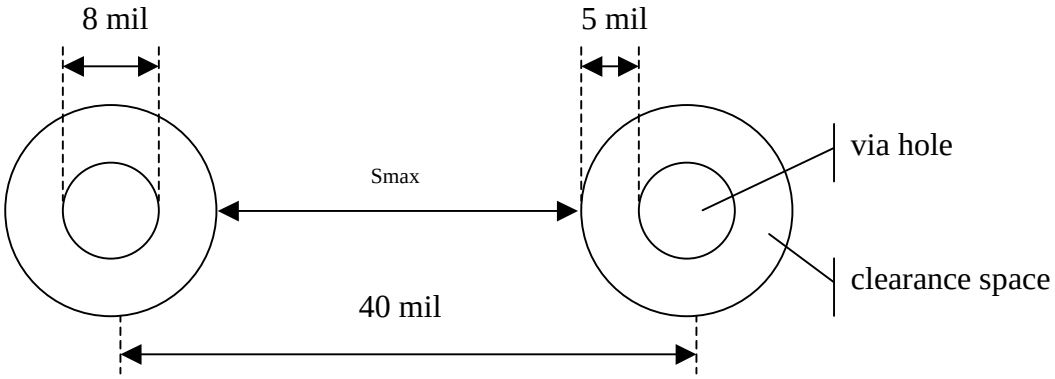

Figure 2-1: BGA Flip-Chip Connector Dimensions

The aspect ratio of our via hole is 1:10. Given that our board thickness is 78.1mil (refer to Table 2-3), our connector requires a via hole of approximately 8mil. A clearance space of 5 mils must be left on either side of the via hole. Two adjacent via holes are separated by 40 mils; hence the space (Smax) that can be occupied by our transmission lines is:

Smax = total space – (2*1/2via hole + 2*Clearance)
⇔ Smax = 40 mil – (8mil + 2*5mil)
⇔ Smax = 22 mil

We decided to run only two lines between each package ball. Each such pair of adjacent lines constitutes a differential pair. To maximize the coupling between our differential pairs, they were placed as close as possible (4mil). This separates our aggressor and victim lines by 18 mils, thus decreasing crosstalk. The width of the lines were determined as follow:

Wmax = (Smax – min_distance)/2
⇔ Wmax = (22 mil – 4 mil)/2
⇔ Wmax = 9 mils

We implemented our lines using a width of 9mils. The routing scheme can be seen in Figure 2-2.

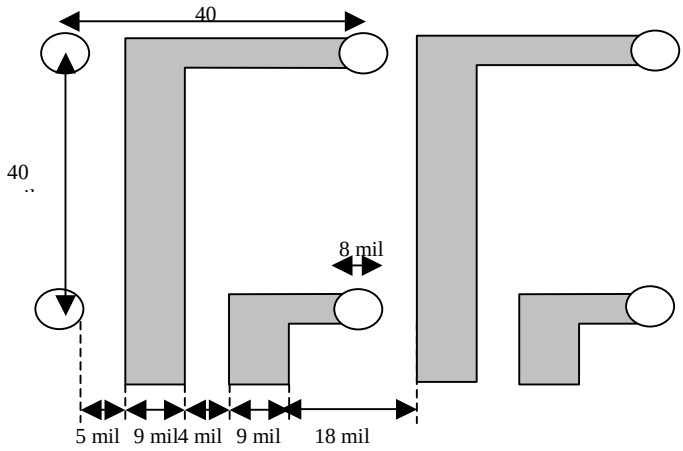

Figure 2-2: Routing

The thickness of our line was obtained through LinPar simulation. Through Spice simulations, we determined that the Teradyne connector and backplane combination presented an odd mode impedance of 53Ω to our trace. Hence, during our LINPAR simulations, we tried to generate a trace with odd mode impedance of 53Ω. This was done because our system uses a differential signaling scheme. Another goal was to make our trace as thick as possible to minimize skin effects. The thickness obtained through simulations was:

$$T_{max} = 4\text{mils}$$

The dimensions of our trace are summarized in Table 2-2.

Trace	**Dimensions**
Wmax	9 mils
Tmax	4 mils

Table 2-2: Trace Dimensions

2.1.2 Board Stackup

Pins are used four-deep on the switch card chip, to access all the necessary lines. The package balls in the third and fourth row route their signal in a similar fashion to that seen in Figure 2-2. However, these traces are routed on different layers. The package balls on the fifth and sixth row are not used for our wiring scheme. The board is then stacked-up as shown on Figure 2-3. The black line between our signal lines is an additional trace that was included to take care of crosstalk (refer to section 2.2.1). The layer description and dimensions are summarized in Table 2-3.

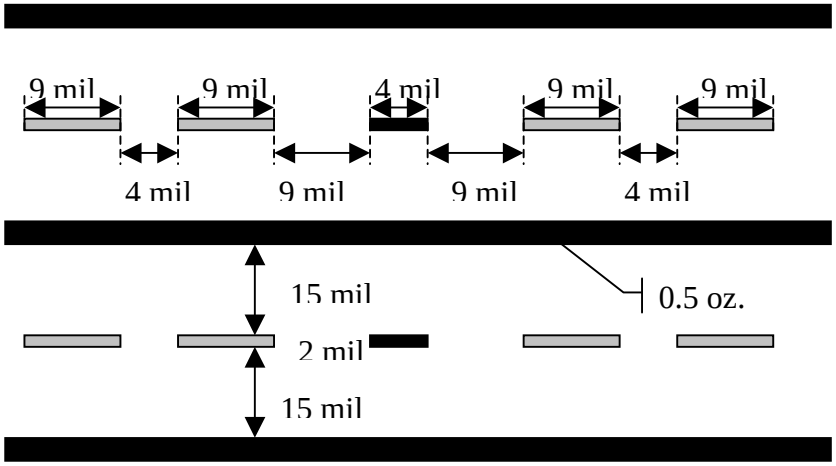

Figure 2-3: Board Stackup

Layer	Description	Dimension (mil)
Top	Prepreg (fiberglass-Epoxy)	4
L1	Ground (Cu)	0.7
Core	Dielectric (fiberglass-Epoxy)	15
L2	Signal (Cu)	4
Core	Dielectric (fiberglass-Epoxy)	15
L3	Power (Cu)	0.7
Core	Dielectric (fiberglass-Epoxy)	15
L4	Signal (Cu)	4
Core	Dielectric (fiberglass-Epoxy)	15
L5	Ground (Cu)	0.7
Bottom	Prepreg (fiberglass-Epoxy)	4

Table 2-3: Layer Specifications

Note that Table 2-3 yields a total board thickness of 78.1mils.

2.2 ELECTRICAL MODEL OF LINE

The high level interconnection model is illustrated below in Figure 2.2.1.

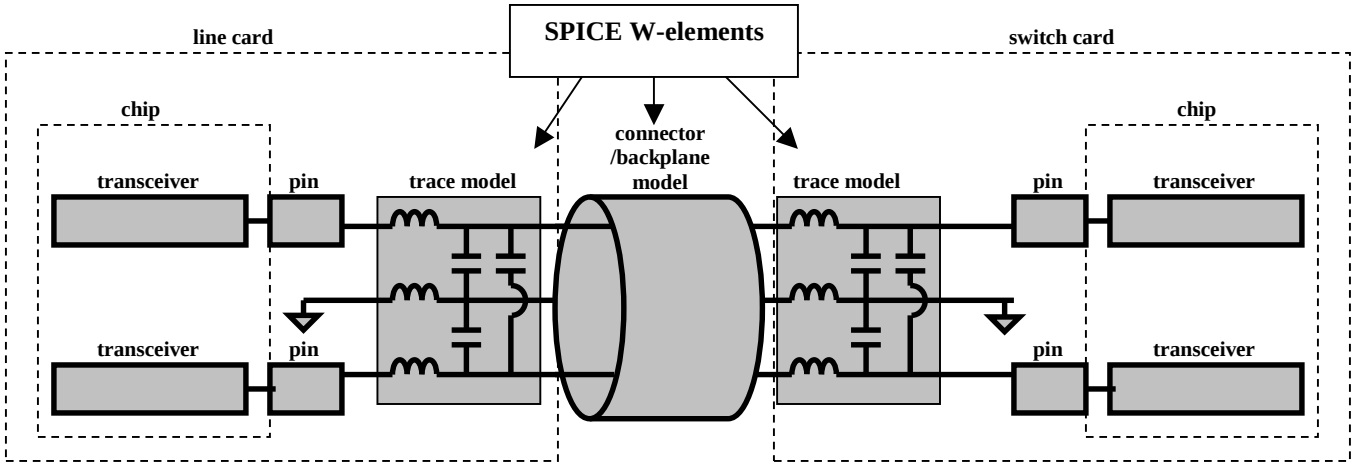

Figure 2-4: System Electrical Model

2.2.1 Three-line Model

We chose the dimensions of our trace wire with specific goals in mind. First, we wanted to make the lines as big (width and thickness) as possible and also keep the separation between differential pairs as large as possible. We were constrained by the dimensions of the BGA layout and the Teradyne connector.

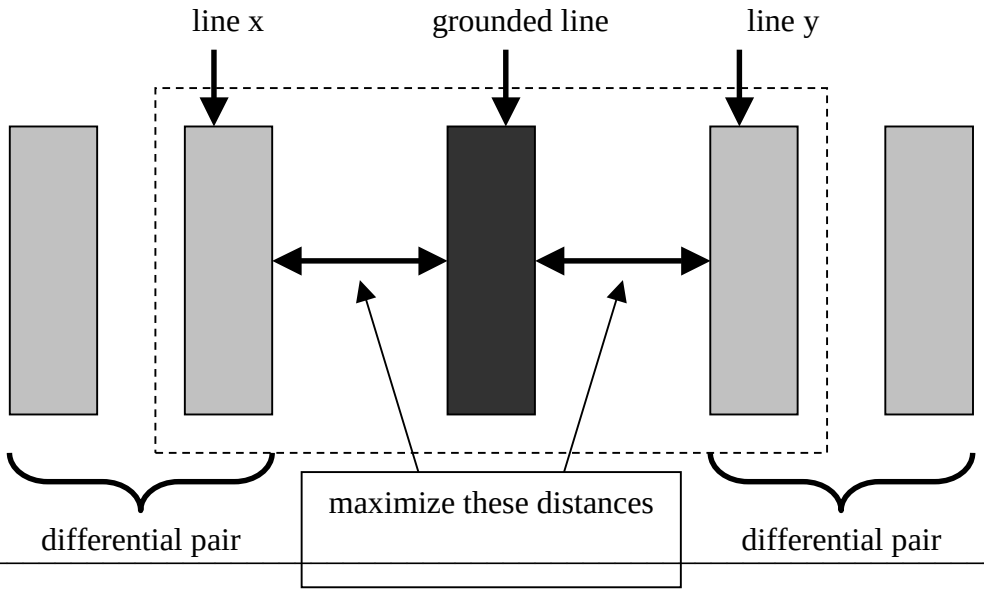

Figure 2-5: Line model constructed in LinPar

As described in section 2.1.1, we used LinPar simulations to achieve the odd-mode characteristic impedance of the differential pair to be 53 Ω. We chose 53 Ω because this is the characteristic impedance of the connector model that we measured. We measured it by injecting a pulse into the connector. By dividing the input voltage by the resulting current, we obtained the impedance. Below is the matrix that LinPar generated. This is the matrix we used in our RLGC file trace3line.rlc file (refer to Appendix E).

Matrix [L] (H/m)			
Line 1	3.995E-07	1.241E-07	4.243E-08
Line 2 (ground line)	1.241E-07	4.762E-07	1.242E-07
Line 3	4.239e-8	1.241E-07	3.995E-07

Table 2-4: Three-line wire Inductance Matrix for RLGC file

Matrix [C] (F/m)			
Line 1	1.425E-10	-3.614E-11	-3.8947E-12
Line 2 (ground line)	-3.614E-11	1.287E-10	-3.614E-11
Line 3	-3.894E-12	-3.614E-11	1.425E-10

Table 2-5: Three-line wire Capacitive Matrix for RLGC file

$$Z_{odd} = \sqrt{\frac{L_{11} - L_{13}}{C_{11} + 2C_{13}}} = 52.5\,\Omega$$

Equation 2-1: Odd Mode Impedance

We used the above equation to compute the odd mode impedance of the lines. L_{11} is the self-inductance and L_{13} the mutual inductance between Line 1 and Line 3. This also applies to the capacitances. We had two other matrices in the RLCG file, the skin-effect matrix and Dielectric loss matrix.

Resistance[Rs] (Ω/m)			
Line 1	0.0037	0	0
Line 2 (ground line)	0	0.0037	0
Line 3	0	0	0.0037

Table 2-6: Three-line wire Skin-effect Matrix for RLGC file

$$R_s = R_{DC}\left(\frac{f}{fs}\right)^{\frac{1}{2}}$$

$$f_s = \frac{\rho}{\mu A}$$

Equation 2-2: Skin-effect resistance and skin-depth frequency

Equation 2-2 represents the equations used for our skin-effect calculations, where R_{DC} is 4.9 Ω/m for our 9-mil copper stripline, A is the cross-sectional area of the stripline ($18mil^2$) and fs = 1.75 MHz. With this calculation, R_s = 116 Ω/m at f = 1GHz. The skin-matrix does not include f component of Rs, which leads to $R_{s,ij}$ = 0.0037.

The dielectric material that we used in our design was the standard FR-4 (fiberglass-Epoxy).

Dielectric	Dielectric Constant	Loss Tangent
FR4	4.7	0.035

Table 2-7: Dielectric Characteristics

We used the dielectric properties to compute our dielectric loss matrix, $G_{d,ij}$.

$$G_{d,ij} = \tan(\delta_D)2\pi C_{ij}$$

Equation 2-3: Dielectric loss matrix

Matrix [Gd] (1/Ωm)			
Line 1	1.425E-10	-3.614E-11	-3.8947E-12
Line 2 (ground line)	-3.614E-11	1.287E-10	-3.614E-11
Line 3	-3.894E-12	-3.614E-11	1.425E-10

Table 2-8: Three-line wire Dielectric loss matrix for RLGC file

2.2.2 Original 2-line Model (not used in our final design)

As mentioned in section 1.2, our 20 cm trace was modeled with three lines. We chose to insert the ground line in the hope that it would absorb some of the capacitive and inductive field lines of the nearby active wires. We anticipated that cross talk would be a prominent factor in our system, so we used this method to reduce it. The method in fact did work because the 2-line model (without the ground line) had larger mutual inductances and capacitances.

Matrix [L] (H/m)		
Line 1	3.563E-7	4.300E-8
Line 2	4.300E-8	3.563E-7

Table 2-9: Three-line wire Skin-effect Matrix for RLGC file

Matrix [C] (F/m)		
Line 1	1.489E-10	-1.797E-11
Line 2	-1.797E-11	1.489E-10

Table 2-10: Two-line wire Capacitance matrix for RLGC file

Note the capacitance between Line 1 and Line 2 is about 5 times larger in this 2-line model than Line 1 and Line 3 in the 3-line model. The complete RLGC file is in Appendix D.

3 SIGNALING

3.1 BIDIRECTIONAL SIGNALING

We opted for bi-directional signaling in our design. The advantage of sending signals in both directions over the same lines is that we can double our original unidirectional transmission rate while maintaining the same clock speed. This has two big benefits: 1) with slower rise times, we can avoid more signal distortion over discontinuities in our line, and 2) by lowering the signaling density on the line, we reduce inter-symbol interference.

The penalty for this technique is an increase in noise. Since each transceiver sees two signals (transmitted and received) on the line simultaneously, decoding the received signal involves not only canceling the transmitted signal, but any reflections associated with the transmitted signal. Near-end crosstalk also now becomes an issue. Fortunately, much of this noise can be cancelled, as is discussed in subsequent sections.

3.2 DIFFERENTIAL SIGNALING

Our design keeps the differential signaling of the original implementation. Despite the expense of having to distribute the signal over two lines, this method enjoys reduced noise from ground bounce, receiver reference error, and crosstalk. Since noise is exceptionally disruptive in a bi-directional signaling system, our team decided that differential signaling was worth the expense.

3.3 BIPOLAR SIGNALING

Our team elected to use bipolar signaling. Together with differential current-mode transmission, bipolar signaling generates a robust signaling system, with easily detectable signals, relatively low noise sensitivity, and high immunity to variations in rail voltages.

3.4 SIGNALING RATE

Our implementation achieves a 4Gbps-signaling rate at the original clock speed (1GHz) through the use of bi-directional signaling.

3.5 CURRENT-MODE SIGNALING

Optimal operation of the transceiver circuit requires the reference source to be very closely matched to the transmission source (otherwise, we get imperfect cancellation of the sent signal when decoding the received signal). Since it is more difficult to build and match a stable voltage source, we chose current-mode signaling to improve the performance of our transceiver.

3.6 EQUALIZATION

Equalization Coefficients	
W1	1.000
W2	-0.376
W3	-0.184
W4	-0.114

Table 3-1: Equalization Coefficients

Signal Strength	
DC	±9.3mA
Isolated	±48mA

Table 3-2: Signal Strength

Our design implements equalization to reduce inter-symbol interference. We used a 4-tap filter to achieve the adjustment, where the input to each tap was scaled by the factor shown in the Table 3-1. (W1 corresponds to the current symbol, while W4 affects the fourth-most-recent bit.) The transmitted signal strengths corresponding to DC operation and an isolated high/low bit are shown in Table 3-2.

3.7 CROSSTALK CONTROL

We included near-end crosstalk control in our system, since this was the biggest source of noise in our system. Specifically, adjacent lines would introduce a significant amount of noise into one another through the backplane connector, which linked adjacent lines with a large capacitance. This meant that a big voltage change on an aggressor line would result in a proportional spike on the victim line.

Since we could not change the connector, there was no way to avoid putting this noise on the line. Instead, we had to find a way to cancel it during decoding at the transceiver. This meant that each transceiver would have to know what signals were being sent on adjacent lines. To do this, we took advantage of the fact that each transceiver already generated a reference output voltage (normally subtracted from the voltage on the line to decode the received signal). We took the reference voltage from each transceiver and put it through a network that simulated the delay and distortion it would experience in going from one transceiver to another (aggressor pin, aggressor via, connector, victim via, victim pin). This near-end crosstalk reference could then be subtracted from the outputs of adjacent transceivers, resulting in a cleaner signal.

3.8 SIGNALING DETAILS

Refer to Table 3-3 for signal specifics.

Signaling Parameters	
Rise Time	50ps
Aperture Time	450ps
I_{HI}	+48mA
I_{LO}	-48mA
V_{HI}	+1.2V
V_{LO}	-1.2V

Table 3-3: Signaling Properties

3.9 HARDWARE/DEVICE DETAILS

3.9.1 Transceiver (Transmission)

The transmission element of our transceiver is a simple current source, generating currents between –48mA and +48mA (corresponding to the maximum allowable voltage swing of 2.4V). To achieve the current necessary for this voltage swing, however, required up to three on-chip current sources in parallel, since the project specifications state that the default sources can only generate 20mA each.

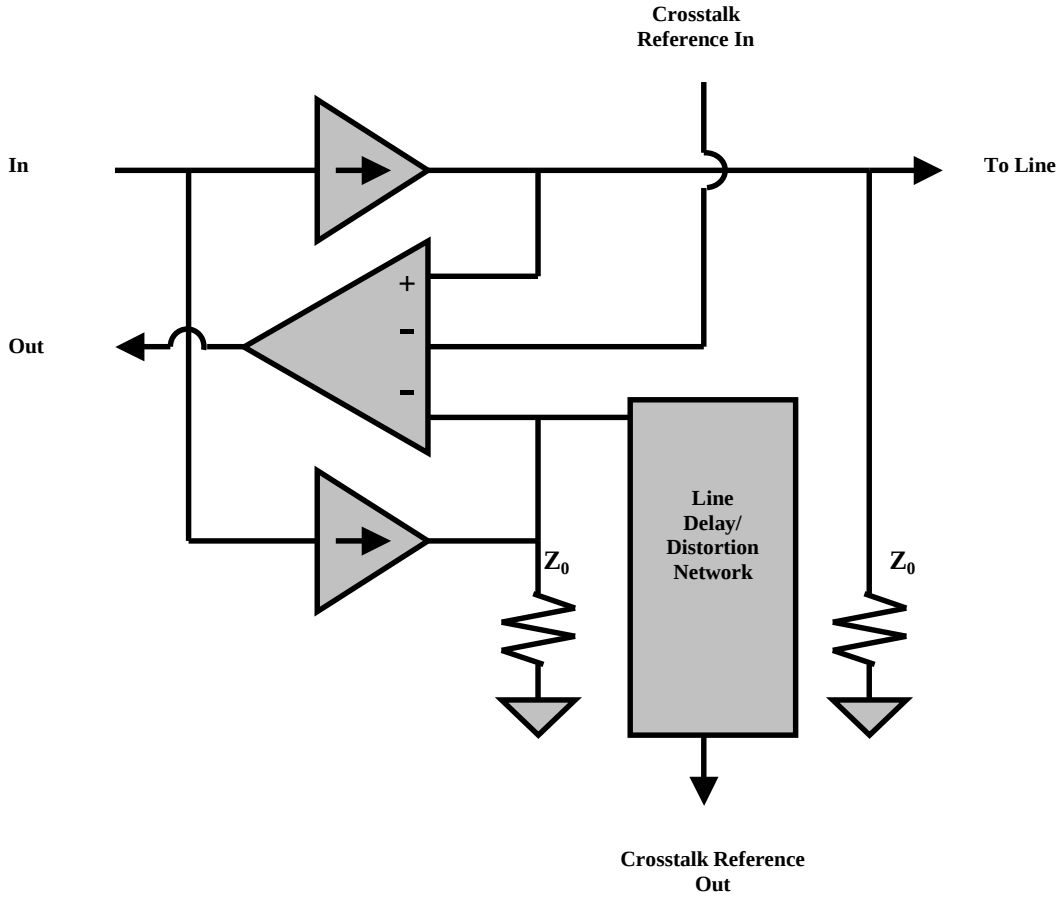

Figure 3-1: Transceiver Block Diagram

3.9.2 Transceiver (Receiving)

Our choice of bi-directional signaling forced us to use a more complex receiver circuit. The incoming bits at each transceiver had to be decoded by taking the signal on the line and subtracting two interfering signals: the outgoing bits and the near end crosstalk from neighboring transceivers.

Subtracting the outgoing bits was straightforward. A reference transmission network generated a signal identical to the one being sent on the line. This reference voltage could then be subtracted from the line voltage, thus eliminating the effect of the transmitted bits on the output.

Eliminating near-end crosstalk originating from the connectors was a little more involved, since it required neighboring transceivers to interface with one another.

To accomplish this, we created a network similar to the pin-via-connector-via-pin path taken by the crosstalk signal between transceivers, and connected it to the transceiver's reference point. This resulted in a fairly accurate reproduction of the crosstalk, which we could then subtract from the output of neighboring transceivers.

3.9.3 Line Termination

To minimize error in our terminations (and thus minimize reflections) we will make use of controlled-output-impedance drivers instead of static resistors. These controlled resistances operate by turning on/off an array of PFETs in parallel, thus changing the effective resistance by intervals of 5%. Unfortunately, we cannot statically set the number of active transistors and still expect the desired resistance, as the resistance for a given setting can vary as much as 50% from chip to chip due to production variations. Fortunately, between such devices on the same chip, we can assume a variation as small as 2%. The answer is to set up a circuit to calibrate one of these variable resistors against a single externally provided resistor, measured to precisely the desired value (50Ω, in this case). Once we have figured out the optimal setting for that one device, we can apply that to every other variable resistor on the chip and expect <7% variation from nominal (5% accuracy of variable resistor, plus 2% variation from device to device).

4 TIMING

4.1 TIMING CONCEPT

In our system, the timing is generated by a Voltage Control Crystal Oscillators (VCCO) with nominal frequency between 1MHz and 300MHz and is distributed through clock trees to all parts of our card. Once the clocks reach the card, it is multiply via a Phase Lock Loop (PLL) to generate 1GHz clock signals. A PLL consists of a voltage-controlled oscillator (VCO) that is tuned. The VCO is initially tuned to a frequency closed to the desired transmitting or receiving frequency. The phase comparator causes the VCO to seek and lock onto the desired frequency. The system is based on the output of a voltage control crystal reference oscillator.
A model of a PLL is shown in Figure 4-1. The signal coming from the VCCO is represented by Ref Osc while fo represents the 1GHz signal used in our data-clocking scheme.

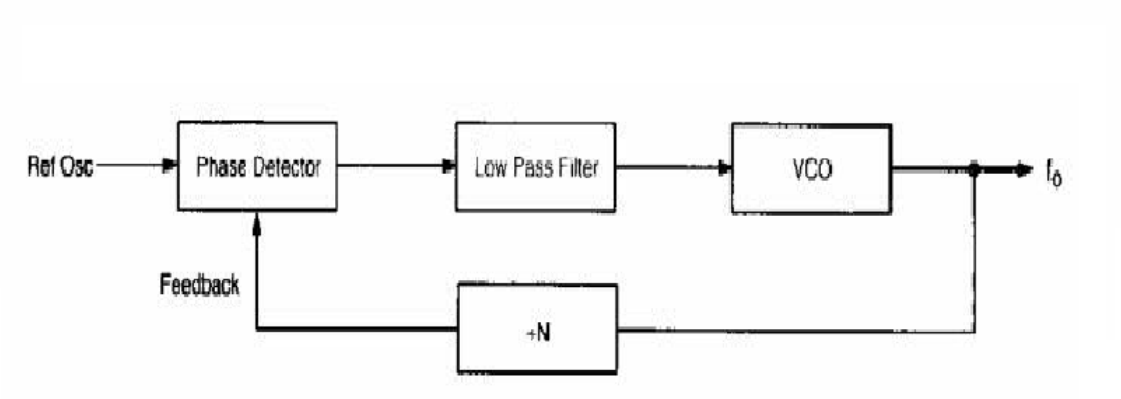

Figure 4-1: Phase-Locked Loop Block Diagram

4.2 INDEPENDENT OSCILLATORS

Our timing scheme requires that each card have one Voltage Control Crystal Oscillators. Each of the oscillators generates a clock frequency within 100-300MHz.

4.3 TIMING CONVENTION – CLOSE LOOP TIMING

The f_o clocks generated by our PLLs are fed to ϕ_{tx} and ϕ_{rcv} in Figure 4-2. The ϕ_{rcv} signal is then skew corrected against the incoming data line to yield the sampling clock. A 90-degree phase shift and a phase comparator are used to compensate for aperture offset and place the sampling window in the center of the eye.

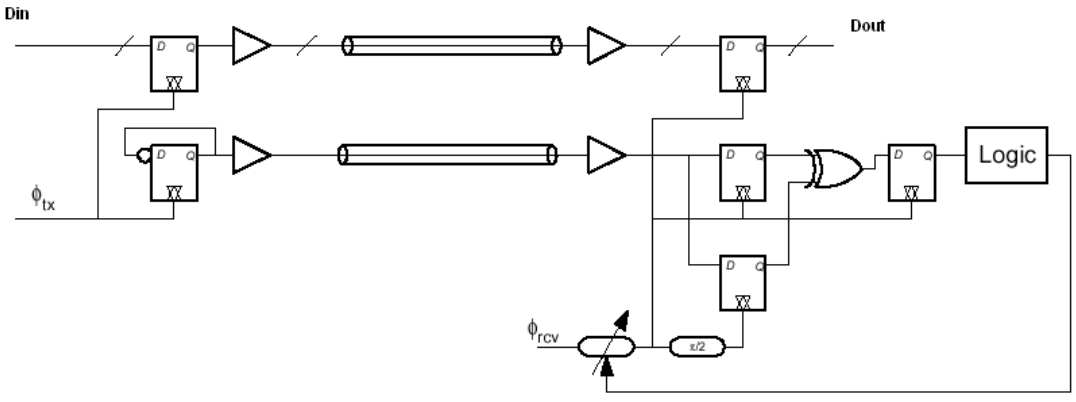

Figure 4-2: Link with Bundled Closed-Loop Timing

5 NOISE BUDGET ANALYSIS

Simultaneous bi-directional systems are particular sensitive to noise, especially noise from cross talk. Below is a noise budget for the noise sources in the system.

System Specs	
System Drive Current: Differential pair.	48 mA to –48 mA
Signal Swing	2.4 V (from 1.2 V to –1.2V)
Gross Margin	1.2 V

Proportional Noises	**Fraction of Signal Swing**
Attenuation	0.61 (new Gross Margin= 468 mV)
Crosstalk	0.15
ISI	0
Total	0.76

Gaussian Noises	**RMS**
Backplane	5 mV
Resistors	8 mV
Total RMS	9.43 mV

Fixed Noise sources	
Transmitter Offset	10mV
Receiver Offset	0
Total Fixed Noises	10mV

Fixed Noise sources	
Net Margin	100mV (eye diagram is 116mV)
V_{SNR}	10.6
BER	4E-25

Figure 5-1: Noise Budget

5.1 TRANSMITTER OFFSET

Our current drivers operate within +/-5% of the specified value. In the worst case, our drivers will be down to 45.6mA. Our termination resistors also operate with 5% of the 50 Ω. With the source resistance down to 47.5 Ω, this will cause a variation of 10 mV in our system.

5.2 RECEIVER OFFSET AND SENSITIVITY

We researched methods of eliminating the +/- 40mV offset. We discovered that if we use a receiver (in our transceiver) that uses a "StrongArm sense amplifier with capacitively trimmed offset voltages", we can cancel the receiver offset.

5.3 CROSS TALK

Simultaneous bi-directional systems are very sensitive to crosstalk, especially since they have to accommodate reverse crosstalk. The 20 cm trace in our design was designed to attempt to minimize forward and reverse crosstalk. We used a homogeneous stripline-wiring scheme that cancels out the forward crosstalk. We used an additional wire, which was placed in between adjacent differential pairs (refer to section 2.1), to help absorb some of the crosstalk. This method was effective and kept our reverse crosstalk co-efficient down to 0.033. We measured the cross talk using the equations below.

$$k_{lx} = \frac{M}{L} \qquad kcx = \frac{Cd}{Co + Cd} \qquad k_{fx} = \frac{k_{lx} - k_{cx}}{2} \qquad k_{rx} = \frac{k_{lx} + k_{cx}}{4}$$

Equation 5-1: Crosstalk-coupling equations

Note that in a homogeneous material forward crosstalk is negligible. Even though our trace had little crosstalk, the bulk of the cross talk came from the connector, which has 0.25 reverse cross talk, which is very significant. The backplane reverse crosstalk coefficient is 0.05, which is significant.

Crosstalk Element	Factor
20 cm Trace	0.033
Connector	0.25
Backplane	0.05
Overall (measured in spice)	0.30
Overall with cancellation techniques	**0.15**

Table 5-2: Crosstalk factors

All this crosstalk essentially used the remainder of our noise margin and left us with no eye opening. After employing our crosstalk cancellation techniques, we were able to obtain an eye opening. Based on our eye opening, we estimated that our final crosstalk in the system is 0.15. We had essentially halved the amount of crosstalk in our system.

5.4 INTER-SYMBOL INTERFERENCE

With our source and receiver terminations well matched, ISI from them is not very significant. With a 5% variance in the termination resistances, we had reflection coefficients of 2.4%, which do not affect our signal. However, attenuation represents significant sources of ISI. We used the equation below to determine each stage of attenuation:

$$\alpha = \frac{R_{DC}}{2Z_0}\sqrt{\frac{f}{f_s}}$$

Equation 5-2: Attenuation across a line.

R_{DC} is measured to be 4.4Ω/m, Z_0 is 50 Ω/m and f_s is 1.75MHz.

Line	Attenuation
20 cm Trace	0.22
Connector	0.027*2
Backplane	0.76
Overall	0.86 (includes 2 traces and 2 connectors)
Overall with equalization	**0.61**

Table 5-3: Attenuation in System

We therefore had to equalize our circuit, which reduced the level of attenuation to 0.615, which we measured using HSPICE simulations. We also more or less removed the ISI in our system after we equalized it using a four-tap filter.

5.5 GAUSSIAN NOISE

Besides the 5mV RMS Gaussian noise from the backplane, another source of Gaussian noise are the resistors. The RMS amplitude associated with the resistors is determined from the equation below.

$$V_R = \sqrt{4k_B TRB}$$

Equation 5-3: Gaussian Noise from a Resistor

In this equation, k_B is the Boltzmann's constant= $1.38*10^{-23}$ Joules/Kelvin, T is the temperature, R is the resistance and B is the bandwidth of the signal. Our system uses R = 50 Ω, B = 1GHz and T = 300K. For a given V_R, the probability that the noise exceeds a certain voltage Vx, is constrained by the equation 5-4.

$$P(|V| > Vx) = e^{-\frac{Vx^2}{2V_R^2}}$$

Equation 5-4: Probability that Noise exceeds Vx.

For our system, with a probability of E-20, we could reach up to 8.0 mV RMS. Lastly, perpendicular crosstalk does not play a role in our design because our line routing does not have wires routed in a perpendicular manner.

5.6 POWER SUPPLY NOISE

We use differential signaling, which allows each differential pair to form its own reference. This allows us to avoid sharing return lines.

6 TIMING BUDGET ANALYSIS

6.1 BOUNDED SOURCE OF JITTER

The greatest source of jitter is produce by the VCO's, which have a jitter of +/- 10% of the cycle time. Our system operates at 1GHz, hence our cycle time is 1ns and the VCO jitter is 100ps.

6.2 STATISTICAL SOURCE OF JITTER

The possible statistical sources of jitter in our system have been cancelled by our clocking scheme by the fact that our reference clock is being sent at the same time as our data. Our clock is sent with our data, which cancels all source of common mode noise.

6.3 SOURCES OF SKEW

The timing loop in our clocking scheme cancels the following sources of repeated timing skew:

- Transmitter and Receiver flip-flop delay skew
- Clock line and Data line skew
- Receiver flip-flop aperture skew
- 90-degree delay line skew

The sources of skew not canceled out are tabulated in Table 6-1.

Timing Components	Time (ps)
Gross Margin	500
Rise Time	50
Receiver Clock Aperture	20
DLL	10
Phase Comparator offset	10
Jitter	
PLL (10% cycle time)	100
Total Timing Noise	190
Net Margin	310

Table 6-1: Timing Budget

Our net margin is large enough to ensure that our data will be transmitted and received accurately.

7 SPICE SIMULATIONS

7.1 EYE DIAGRAM AND LONE ONE PULSE

Below is our eye diagram. Our noise margin is measured at 116mV.

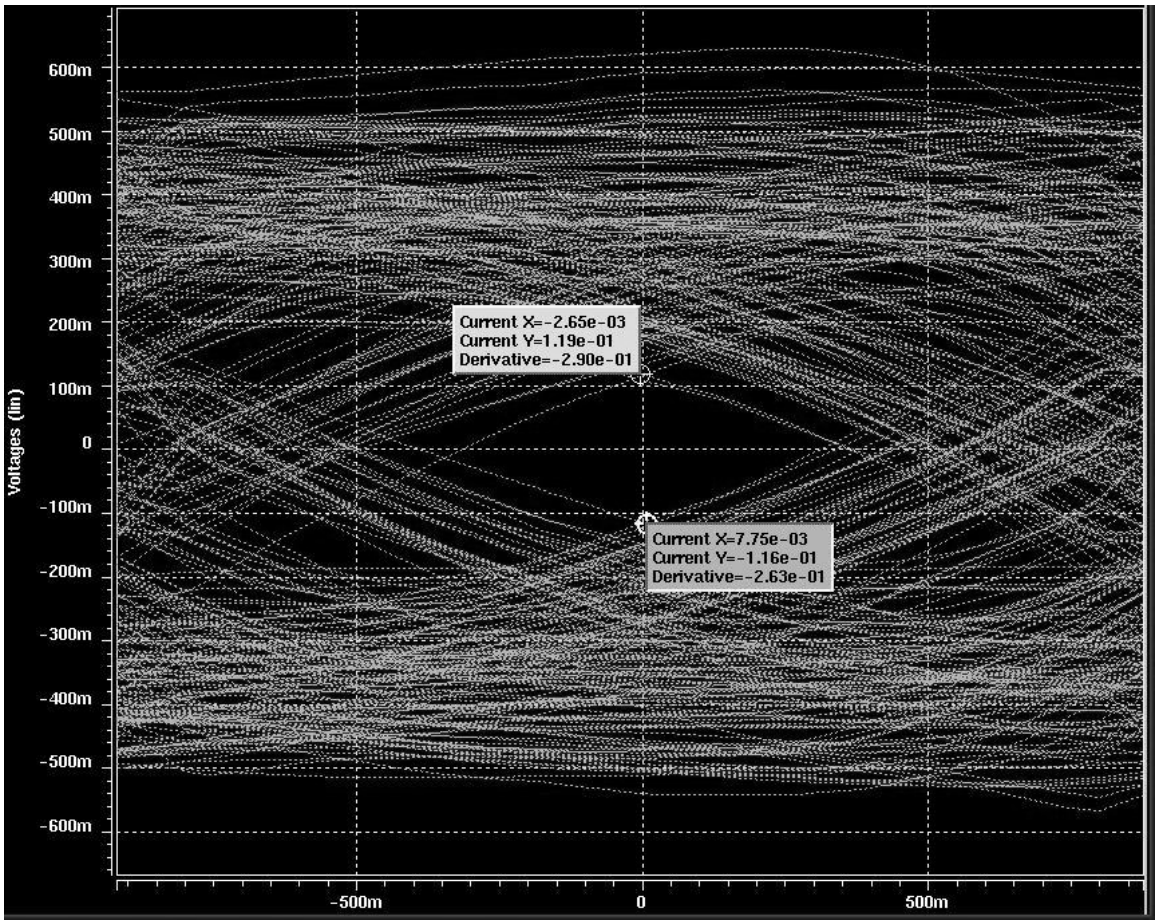

Figure 7-1: Eye diagram of our received signal

The lone-pulse simulation. Note that ISI is removed.

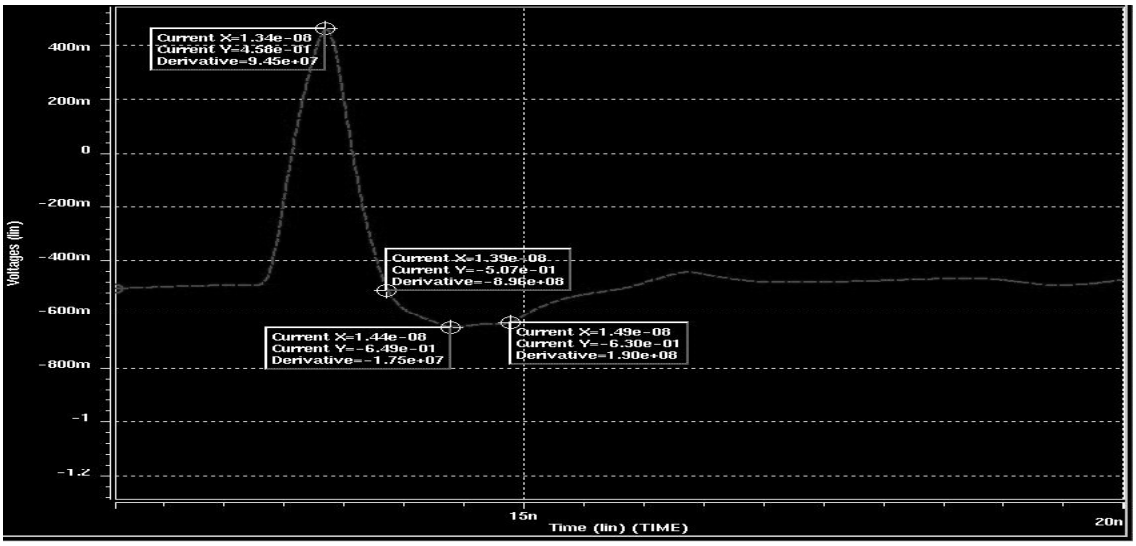

Figure 7-2: Lone One Pulse with Equalization

Lone one without equalization is shown below. Note that "one" never reaches a positive value. Each maker denotes a new bit time.

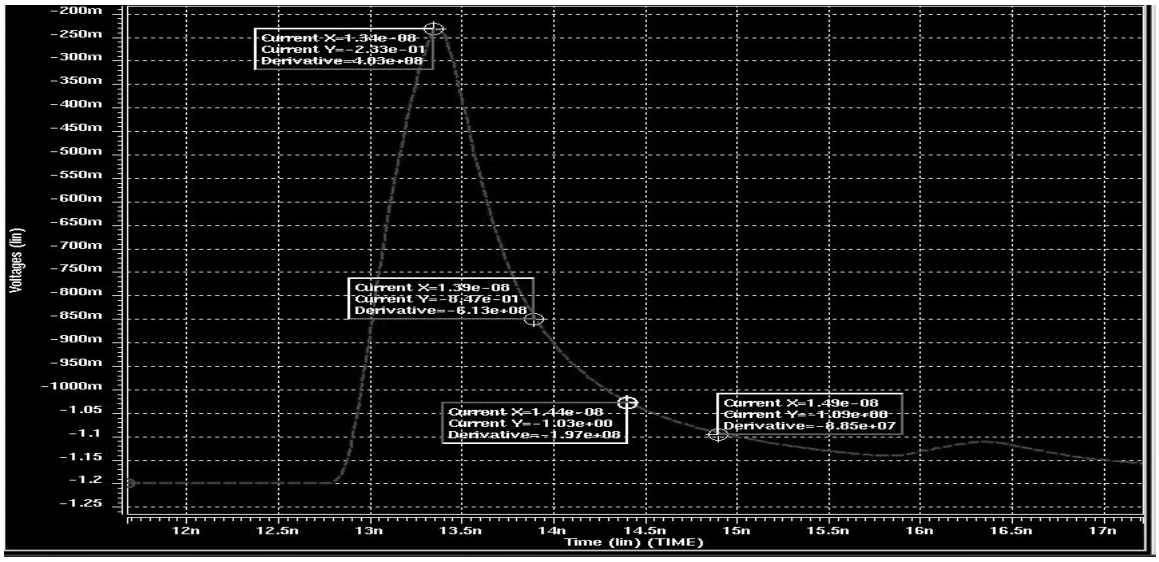

Figure 7-3: Lone One Pulse without Equalization

7.2 TDR AND TDT

TDR simulation is depicted below.

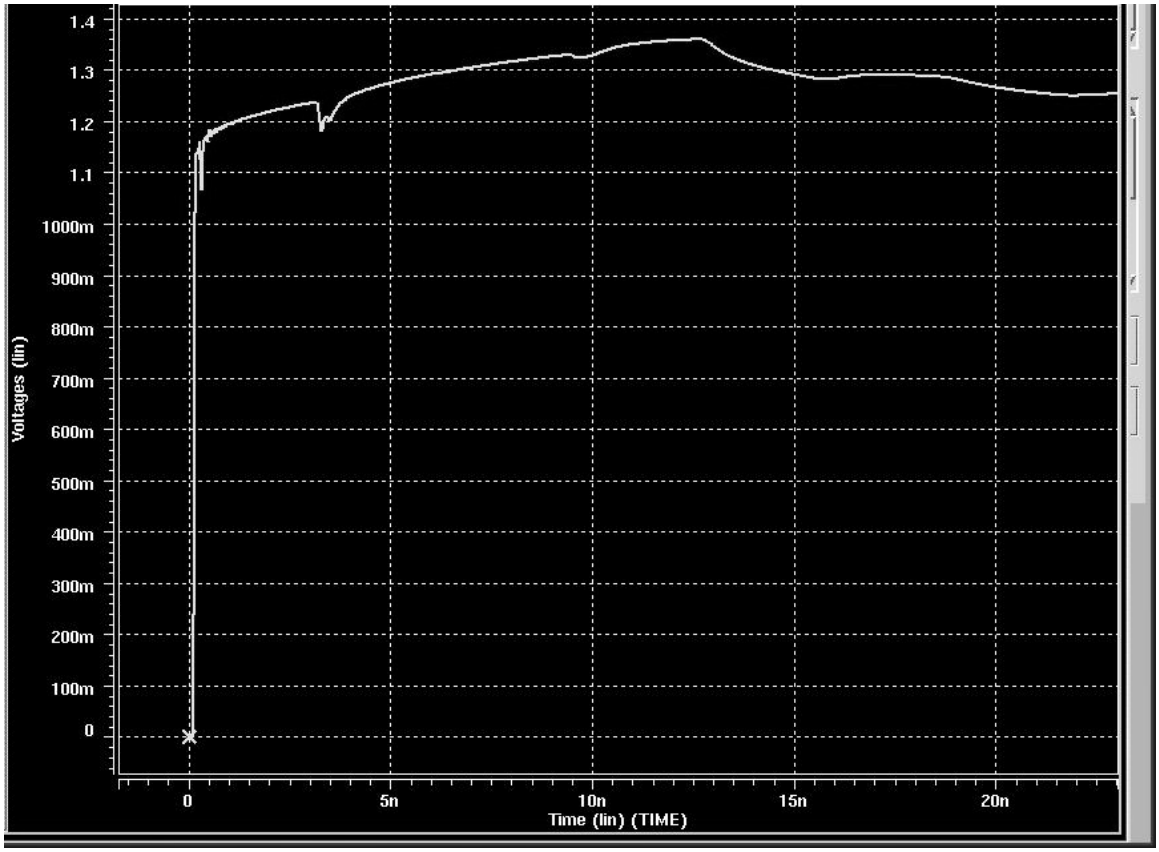

Figure 7-4: TDR Simulation

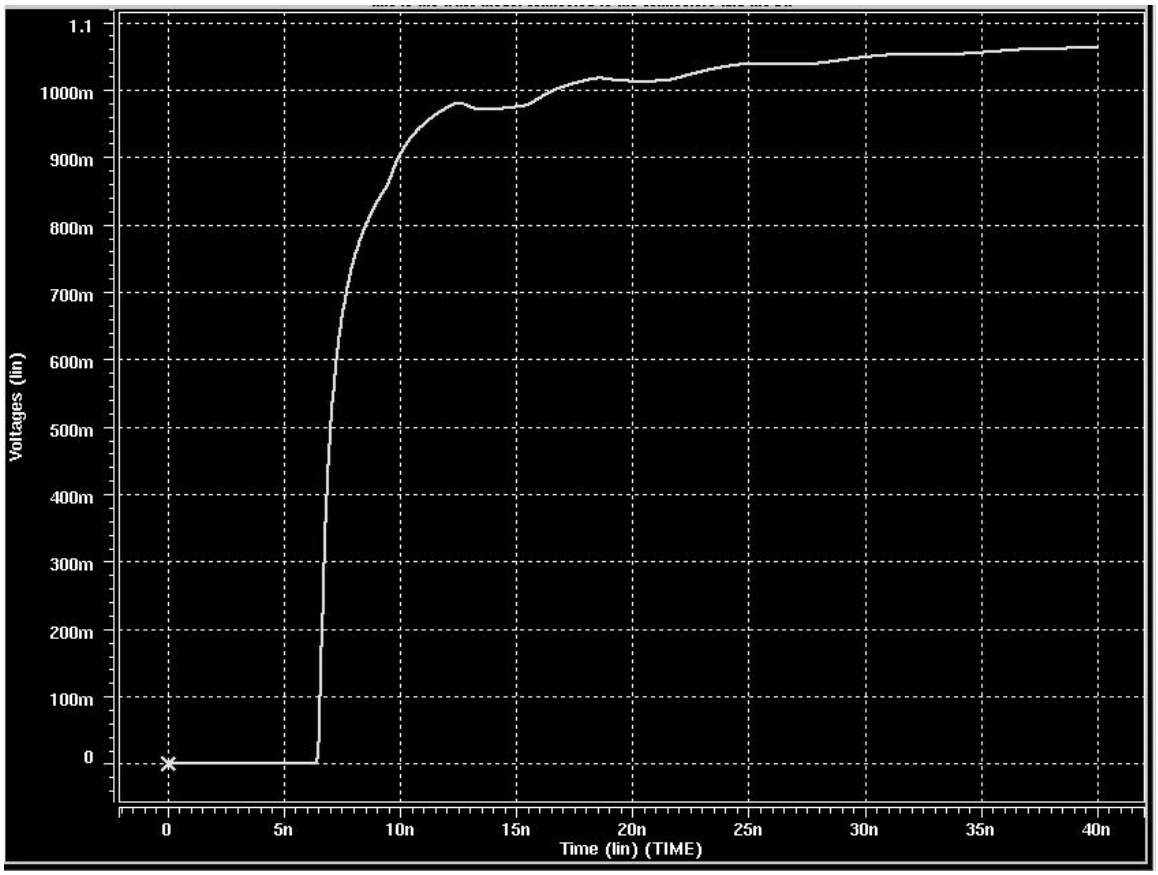

Figure 7-5: TDT Simulation

8. REFERENCES

[1] Lee, Ming-Ju Edward and et al., “A 90mW 4Gb/s Equalized I/O Circuit with Input Offset Cancellation”, IEEE International Solid-State Circuits Conference, 2000

[2] Dally, William J. and Poulton, John W., *Digital Systems Engineering*, 1998, Cambridge University Press

APPENDIX

A. COMPLETE SPICE MODEL: FINAL.SP

```
*********************************************************
* FINAL SPICE MODEL (final.sp)
*********************************************************
* This file models the complete connection between the line card
*chip and the switch card chip. This includes the pins the traces
*the backplane connectors and the backplane.
*********************************************************
* Authors: Njuguna Njoroge, Yves Ngu, Emilio Antunez
* *********************************************************
* Only 4 lines are needed to add eye diagrams to a circuit:
* Here they are:
     .Param per = 500ps          $ bit rate
     vT T 0 PULSE(-1 1 10ns per per 0 '2*per')
     *eT0 T0 0 Vol='(TIME*1/per)-int((TIME*1/per))'
     *eT  T 0 DELAY T0 0 TD=10ns
     rT T 0 1k

.option post
.print V(outXL)
.print V(outXR)
.print V(outYL)
.print V(outYR)

.include 'proj02model.sp'
.include 'transceiver.sp'
.include 'sourceXL.sp'
.include 'sourceYL.sp'
.include 'sourceXR.sp'
.include 'sourceYR.sp'

* gnd is at near end of line
vg gnd 0 0

XsourceXL inXL gnd sourceXL
XtranXL inXL outXL xa0 xtalkoutYL xtalkoutXL gnd transceiver
Cesd1   xa0  gnd 0.5p
Lesd1   xa0  xain 0.2n
T1 xain gnd xal gnd Z0=50 TD=66.7ps
Lvia1  xal  xa 0.5n
Cvia1  xa gnd 0.3pF

Vza za gnd 0

XsourceYL inYL gnd sourceYL
*VYL inYL gnd 0
XtranYL inYL outYL ya0 xtalkoutXL xtalkoutYL gnd transceiver
Cesd2   ya0  gnd 0.5p
Lesd2   ya0  yain 0.2n
T2 yain gnd yal gnd Z0=50 TD=66.7ps
Lvia2  yal  ya 0.5n
Cvia2  ya gnd 0.3pF

* Pin model for transmitter side
```

```
w1 N=3 xa za ya gnd xb zb yb gb rlgcfile=trace3line.rlc l=0.2
Vzb zb gb 0
xmodel xb yb gb xc yc gc model
Vzc zc gc 0
w2 N=3 xc zc yc gc xd zd yd gd rlgcfile=trace3line.rlc l=0.2

* Pin model for receiver side
Lvia3  xd  xdin 0.5n
cvia3  xd gd   0.3pF
T3 xdin gd xeout gd Z0=50 TD=66.7ps
cesd3   xe  gd 0.5p
Lesd3   xeout  xe 0.2n

Lvia4  yd  ydin 0.5n
cvia4  yd gd   0.3pF
T4 ydin gd yeout gd Z0=50 TD=66.7ps
cesd4  ye gd 0.5p
Lesd4   yeout  ye 0.2n

*termination
XsourceXR inXR gd sourceXR
*VXR inXR gd 0
XtranXR inXR outXR xe XtalkoutYR xtalkoutXR gd transceiver
Vzd zd gd 0
XsourceYR inYR gd sourceYR
*VYR inYR gd 0
XtranYR inYR outYR ye xtalkoutXR xtalkoutYR gd transceiver

rab gnd gb 100k
rcd gc gd 100k

.tran 25ps 160ns

.end
```

B. TRANSCEIVER SPICE MODEL: TRANSCEIVER.SP

```
*********************************************************
* TRANSCEIVER MODEL (transceiver.sp)
*********************************************************
* This file models a transceiver for a bidirectional
* signaling system.  This system takes an input voltage
* "in" and converts it to a current-mode signal to be
* sent over output "line".  At the same time, it filters
* out the "line" signal coming from the opposite direc-
* tion, and sets the corresponding voltage on "out".
*********************************************************
* Authors: Njuguna Njoroge, Yves Ngu, Emilio Antunez
* *******************************************************

.probe I1(Gs)

.param Z0     = 50
.param DCI    = 48mA
.param Vcc    = 1.2
.param VW1    = 1
.param VW2    = -0.39
.param VW3    = -0.181
.param VW4    = -0.113

*************************************************************
**                   TRANSCEIVER MODEL                     **
*************************************************************

.subckt transceiver in out line xtalkin xtalkout gnd

* Voltage controlled current source, to drive "line".
Gs line gnd VCCS PWL(1) in gnd
+ 'Vcc*(-VW1+VW2+VW3+VW4)', '-DCI'
+  'Vcc*(VW1-VW2-VW3-VW4)', 'DCI'

* Voltage controlled current source, to drive "ref".
Gref ref gnd VCCS PWL(1) in gnd
+ 'Vcc*(-VW1+VW2+VW3+VW4)', '-DCI'
+  'Vcc*(VW1-VW2-VW3-VW4)', 'DCI'

* A three-input subtracting filter (out = line - ref - other).
Efilter out gnd VCVS POLY(3) line gnd ref gnd xtalkin gnd 0 1 -1 -1

* Matching resistor
Rmatch line gnd Z0

* ------------------------------------------------------
* The remainder of the code defines the network attached
* to our reference output, to generate a
* prediction of the crosstalk across the connector.
* ------------------------------------------------------
* Note:  We have used an exact copy of the line, up to
* the connector for simulation of what is possible with
* an ideal reference network.
* ------------------------------------------------------
```

```
* aggressor transceiver's matching resistor
Rref ref gnd Z0

* aggressor pin model
Cpin1x ref gnd 0.5pF
Lpin1x ref refxain 0.2nH
Tpinx refxain gnd refxal gnd Z0=50 TD=66.7ps
Lpin2x refxal refxa 0.5nH
Cpin2x refxa gnd .3pF

* agressor/victim trace model
Wtrace N=3 refxa refza refya gnd refxb refzb refyb refgb
rlgcfile=trace3line.rlc l=0.2
Vrefza refza gnd 0
Vrefzb refzb refgb 0
Rrefgb refgb gnd 100k

* connector model
Xconn refxb refyb refgb refxc refyc refgc conn
Rrefxc refxc refgc 53
Rrefyc refyc refgc 53
Rrefgc refgc gnd 100k

* victim pin model
Cpin2y refya gnd .3pF
Lpin2y refyal refya 0.5nH
Tpiny refyain gnd refyal gnd Z0=50 TD=66.7ps
Lpin1y xtalkout refyain 0.2nH
Cpin1y xtalkout gnd 0.5pF

* victim transceiver model
RotherY xtalkout gnd Z0

.ends
```

C. RANDOM BIT GENERATOR: SOURCEGEN.PL

```
#!/user/bin/perl
require 5.002;

# Default inputs
use constant SOURCEID_D => "";
use constant NSAMPLES_D => "10";
use constant VCC_D      => "1.2V";
use constant PERIOD_D   => "500ps";
use constant RISETIME_D => "50ps";
use constant EPSILON    => "16ps";

use constant PIN_DELAY  => "67ps";
use constant C_REFLECTION => "0";

use constant PAST_SIZE  => 3;      # Total W's - 1
use constant W1         => 1;
use constant W2         => -0.376;
use constant W3         => -0.184;
use constant W4         => -0.114;

use constant SWITCHING_PROBABILITY => .35;

if($ARGV[0]) {

  # Set up array with default values
  @inputs = (SOURCEID_D, NSAMPLES_D, VCC_D, PERIOD_D, RISETIME_D);

  for ($i = 0; $i < scalar(@ARGV); $i++) {
    $inputs[$i] = $ARGV[$i];
  }

  $sourceID = $inputs[0];
  $nsamples = $inputs[1];
  $vcc      = $inputs[2];
  $period   = $inputs[3];
  $risetime = $inputs[4];

  $filename = join("", ">source", $sourceID, ".sp");
}
else {
  die "Usage: perl sourcegen.pl SOURCEID [NSAMPLES Vcc PERIOD
RISETIME]\n";
}

print "SOURCEID: source", $sourceID,"\n";
print "FILENAME: ", $filename, "\n";
print "NSAMPLES: ", $nsamples, "\n";
print "Vcc:      ", $vcc, "\n";
print "PERIOD:   ", $period, "\n";
print "RISETIME: ", $risetime, "\n";

$value = rand;
```

```
for($i = 0; $i < PAST_SIZE; $i++) {
    $output[$i] = -$vcc;
}

$current_output = -$vcc;

# Generate bitstream
for($i = PAST_SIZE; $i < $nsamples+PAST_SIZE; $i++) {
  $value = rand;

  if ($value < SWITCHING_PROBABILITY) {
    $current_output = -$current_output;
  }

  $output[$i] = $current_output;

  if ($output[$i] == $vcc) {
    $binary_symbol = "1";
  } else {
    $binary_symbol = "0";
  }

  print "(", $i - PAST_SIZE, ") ", $binary_symbol, "\n";
}

###OPEN
FILE#########################################################
open (FILE, $filename);
###START
CONTENT######################################################
print FILE "*******************************************************
\n";
print FILE "* RANDOMLY-GENERATED VOLTAGE SOURCE (source", $sourceID,
".sp) \n";
print FILE "*******************************************************
\n";
print FILE "* The binary voltage output by this source was generated
\n";
print FILE "* by script 'sourcegen.pl' using the following
parameters:\n";
print FILE "*
\n";
print FILE "* COMPONENT NAME: source", $sourceID, "\n";
print FILE "* NUM SAMPLES:    ", $nsamples, "\n";
print FILE "* RAIL VOLTAGE:   ", $vcc, "\n";
print FILE "* PERIOD TIME:    ", $period, "\n";
print FILE "* RISE TIME:      ", $risetime, "\n";
print FILE
"*******************************************************\n";
print FILE "* Authors: Yves Ngu, Njuguna Njoroge, Emilio Antunez
\n";
print FILE
"*******************************************************\n";
print FILE "\n";
print FILE ".param per = ", $period,      "    * bit period\n";
print FILE ".param tr  = ", $risetime,    "    * signal rise
time\n";
```

```
print FILE "\n";
print FILE "* Bit pattern\n";
print FILE ".param init     = -1\n";

for ($i = 0; $i < $nsamples; $i++) {
  $b1 = $output[$i + PAST_SIZE];
  $b2 = $output[$i + PAST_SIZE - 1];
  $b3 = $output[$i + PAST_SIZE - 2];
  $b4 = $output[$i + PAST_SIZE - 3];

  $equalizedbit = W1*$b1 + W2*$b2 + W3*$b3 + W4*$b4;
  print FILE ".param eqbit", $i, "_", $sourceID, "   = ",
$equalizedbit;
  print FILE "       * bit", $i, "_", $sourceID, "   = ", $output[$i
+ PAST_SIZE], "\n";
}

print FILE "\n";
print FILE ".subckt source", $sourceID, " pos neg\n\n";
print FILE "Vs pos neg\n";
print FILE   "+ pwl 0                   init\n";
print FILE   "+     tr                  eqbit0", "_",
$sourceID,"\n\n";
for ($i = 1; $i < $nsamples; $i++) {
  print FILE "+     '",$i,"*per'              eqbit",    $i-
1,"_",$sourceID,"\n";
  print FILE "+     '",$i,"*per+tr'           eqbit",    $i,
"_",$sourceID,"\n";
  print FILE "+\n";
}
print FILE "\n";
print FILE ".ends\n";
###END
CONTENT############################################################
close(FILE);
###FILE
CLOSED############################################################
```

D. TWO LINE MODEL: TRACE2LINE.RLC

```
* trace2line.rlc
* This is the two line model of the trace that we originally tried
to implement.

* N = number of lines
*********
2

* Lo = inductance matrix
***************
3.563e-7
4.300e-8 3.563e-7

* Co = Capacitance matrix
**************
1.489e-10
-1.797e-11     1.489e-10

* Ro = Resistance matrix
**************
0
0  0

* Go = conductance matrix
*******************
0
0  0

* Rs = skin effect matrix
***************
0.0034
0  0.0034

* Gd = dielectric loss matrix
***************
3.274e-11
-3.952e-12     3.274e-11
```

E. THREE LINE MODEL: TRACE3LINE.RLC

```
* trace3line.rlc
* This is a 3 line model of our trace.  The middle line is always
*held to ground and serves to aborb some of the fields lines from
*the outer two lines. With the middle line, we hope to reduce
*crosstalk between the two outer lines.
* We used LinPar to generate the parameters.
* Parameters used (in mils):
* w1=9, w2=4, w3=9 t1=t2=t3=2. x1=2, x2=18, x3=29. y1=y2=y3=15.
* Cover height=32, Ground Plane Width=40. er=4.7

* N = number of lines
*********
3

* Lo = inductance matrix
***************
3.995e-7
1.241e-7 4.762e-7
4.239e-8 1.241e-7    3.995e-7

* Co = Capacitance matrix
**************
1.425e-10
-3.614e-11     1.287e-10
-3.894e-12     -3.614e-11  1.425e-10

* Ro = Resistance matrix
**************
0
0  0
0  0     0
* Go = conductance matrix
*******************
0
0  0
0  0     0
* Rs = skin effect matrix
***************
0.0037
0  0.0037
0  0     0.0037

* Gd = dielectric loss matrix
***************
3.133e-11
-7.95e-12      2.83e-11
-8.56e-13      -7.95e-12   3.133e-11
```

F. EYE DAIGRAM EXTRACTOR: EXTRACTEYE.PL

```
#!/user/bin/perl -w
require 5.002;

########################################################################
# extracteye.pl
########################################################################
# This code extracts the a waveform placed in a .lis file through
# a SPICE print statement, and sets up the code necessary to display
# the eye in Matlab.  The name of a .lis file and a target node are
# taken in as command line arguments, and the output file will be of
# the form "eye_[target].m"
########################################################################
# Authors: Emilio Antunez, Njuguna Njoroge, Yves Ngu
########################################################################

use constant SAMPLES_PER_NS => 40;
use constant N_SKIPSAMPLES  => 7*SAMPLES_PER_NS;

# read in arguments
if($ARGV[0]) {

  $source = $ARGV[0];
  $target = $ARGV[1];

  # all output files will have names of form "eye_[target].m"
  $outfilename = join("", ">eye_", $target, ".m");
}
else {
  die "Usage: perl extracteye.pl .LIS_FILE OBSERVED_NODE\n";
}

###OPEN
FILE##########################################################
open (INFILE, $source);
open (OUTFILE, $outfilename);
###START
CONTENT#######################################################
print OUTFILE $target, "= [";


while ($line = <INFILE>) {

  # Looks for a line stating "time  voltage", which
  # indicates the start of a SPICE print stream.
  if ($line =~ /time\s+voltage/) {
    print "Found time/voltage tag\n";

    $line = <INFILE>;

    # Checks whether this print stream matches
    # the one we are currently trying to read.
    if ($line =~ /$target/) {
      print "Found target line.\n";
```

```
      # Skips a certain number of samples, until we believe
      # we have reached a stabler state.  Initial readings may
      # may not reflect overall behavior (i.e. they mess up
      # our eye until equalization kicks in).
      for ($i = 0; $i < N_SKIPSAMPLES; $i++) {
   $line = <INFILE>;
      }

      while ($line = <INFILE>) {
   if ($line =~ /[0123456789munpf\.]+\s+([0123456789munpf\.\-]+)/) {
     $data = $1;

     $data =~ s/m/e-3/;
     $data =~ s/u/e-6/;
     $data =~ s/n/e-9/;
     $data =~ s/p/e-12/;
     $data =~ s/f/e-15/;

     print OUTFILE $data, " ";
   } else {
     last;
   }
      }
    }
  }
}
print OUTFILE "];\n";

print OUTFILE "eyediagram(", $target, ", ", SAMPLES_PER_NS/2,", 500,
", SAMPLES_PER_NS/4,");";

###END
CONTENT############################################################
close(OUTFILE);
close(INFILE);
###FILE
CLOSED############################################################
```