EE273 Digital Systems Engineering

Final Project Report

Piyush Keshri (piyushk),George Rossin (gar7), Laura Sharpless (ljs62)

Thursday, March 11th, 2010

1 Introduction

In this project we are asked to design a router with 16 line card modules that maintain a bandwidth of 40Gbps/card as well as two bitsliced crossbar connectors (one primary, one redundant). The goal of our design was to create a functional, cost-effective, and high performance router.

2 System Design Description

2.1 Design Discussion

Our design goal was to design a system that connects 16 line modules in a high speed router to a pair of central crossbar switches. Minimum design criteria was to design line cards that communicate with other lines cards at a bandwidth of 40 Gbps using a router with crossbar architecture.

2.1.1 Goals

2.1.1.1 Functionality

Functionality was the key concern in this project. Above all else (cost, performance, etc.) the design needs to meet the required timing margin and noise margin constraints (BER < $10e^{-14}$) and have adequate RMS noise and clock jitter immunity.

2.1.1.2 Minimize Cost

Many different tactics were used to drive down cost. First, we wanted to minimize the number of hardware system components while maintaining functionality. This not only decreases board area, but also decreases the number of chips, driving cost down significantly. Secondly, we wanted to decrease the routing complexity of the system. Reduced routing complexity facilitates manufacturing thinner boards, which not only drives down cost but also any headaches due to routing. Lastly, we desired to use the most inexpensive components possible to produce a working system. Such decisions to be made were whether or not to use bond-wire packages over BGA, cheaper connectors over more expensive connectors (TinMan vs. Xcede), and cheaper vs. much more expensive dielectric material. The general technique was to start with the cheapest components possible then upgrade to either improve performance or maintain functionality.

2.1.1.3 Optimize Performance

Performance could be optimized in many ways. Unfortunately, many times it means adding complexity or cost. We used a few tactics to increase performance without increasing the complexity or cost significantly. The easiest decision was to use differential signaling. This added to the routing complexity but produces much cleaner signals at the receiving end. We also used striplines over microstriplines so we didn’t need to worry about varying E-fields in mediums with different dielectric constants (which affects speed of transmission). Some other ideas to improve performance include reducing crosstalk. Depending on the routing of traces in a signal layer, the impact of crosstalk can be mitigated significantly. Lastly, we conducted some research on offset cancellation techniques to reduce receiver offsets, which we will discuss in detail later.

2.1.2 Challenges

The biggest challenge was maintaining the inherent cost / performance tradeoffs while maintaining functionality. Some examples of tradeoffs that needed to be managed are mentioned below.

2.1.2.1 Differential Signaling

Advantage: Easier to create a higher performance module because of better noise immunity.

Disadvantage: Requires more signals and therefore more space. Since more board space is required this also drives up cost.

2.1.2.2 Increasing BW

Advantage: Better resource utilization and also drives down the cost and space of system. Drives performance up dramatically.

Disadvantage: Higher crosstalk coupled with less noise immunity which results in a higher BER. Additionally, there will be less skew and jitter tolerance, which exerts pressure on the CDR to perform well.

2.1.2.3 Inexpensive Components

Advantage: Cost will go down significantly.

Disadvantage: Performance will degrade, requiring a more efficient design.

Given all these tradeoffs, we needed to balance them in such a way to reduce cost yet optimize performance. This was very challenging. For example, running at a higher BW put a lot of pressure on the noise and timing margins. In order to account for this fact, it was necessary to research reducing the receiver margin, as well as designing an effective equalizer in order to produce a clearer signal on the receiving end. Furthermore, the constraint of the routing length and kind of connector also created issues. These were solved mainly by taking a hit on cost and increasing spacing to facilitate functionality. This was just one challenge that we faced.

After careful simulation and consideration, we decided to boost the per-line BW from 3.125Gbps to 8.4Gbps. This meant that we needed only 6 differential pairs of transmitting lines / line card in order to meet the 40Gbps BW requirement. This per-line BW was decided upon because it was the best performance that could be achieved without putting the system in a dysfunctional state.

2.2 Block Diagrams

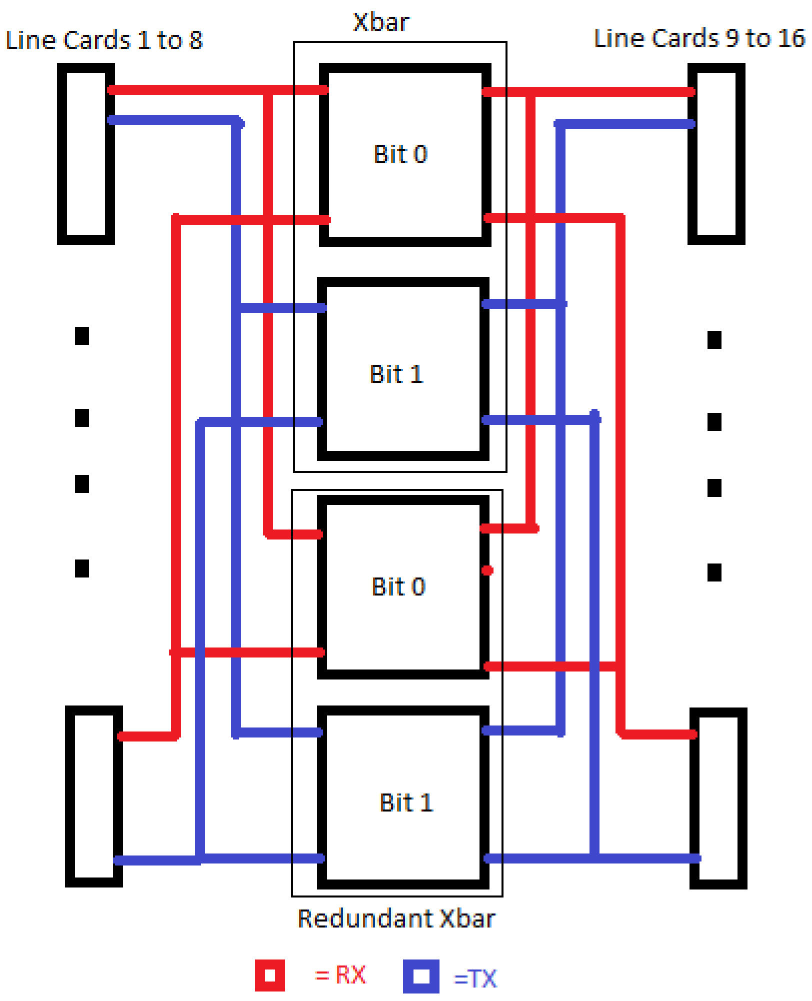


Figure 1:Block diagram of system architecture.

1. The per-linecard BW requirement is 40 Gbps.
2. Frequency of operation per diff pair is **8.4 Gbps with 2 signaling levels**.
   a. Since this is the case, only 6 differential pairs / line card / Xbar are needed to meet BW requirement.
   b. Each blue and red line are bidirectional with 6 diff pairs (RX/TX): 3 incoming diff pairs to each bitslice in the Xbar and redundant Xbar.
   c. The total number of TX/RX lines out of each line card is **48 lines**. Hence, for each bitsliced Xbar, a total of **12 lines / line card** are connected.
   d. There are a total of 16 line cards meaning that there are **192 lines** connecting to each bitsliced crossbar.

Note: 1/3 of the total connections are to power and ground, which are not routed in the routing studies, but have been included in the connector diagrams as red pins.

2.3 Placement Drawing

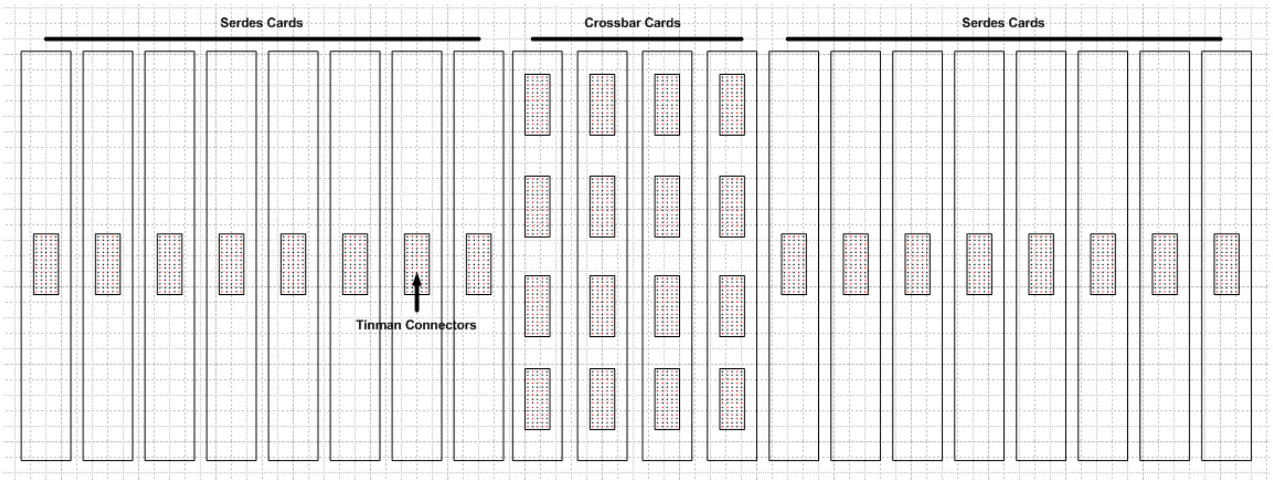
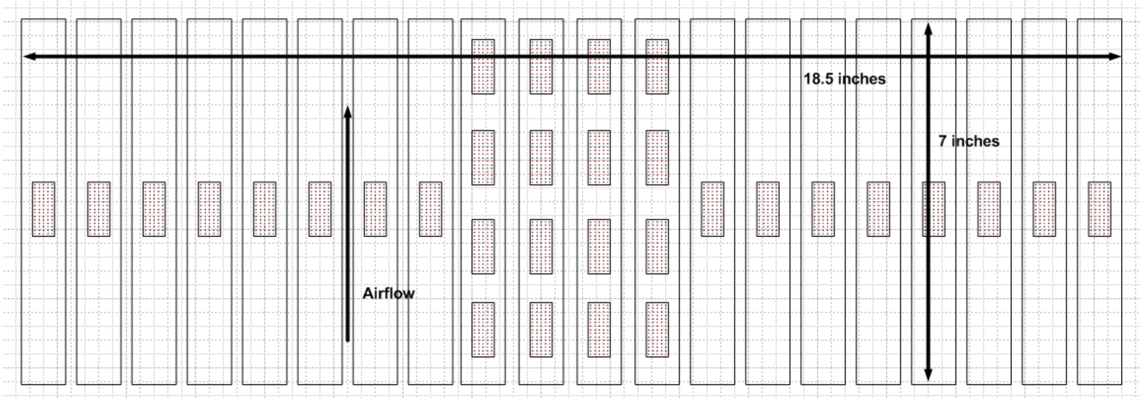
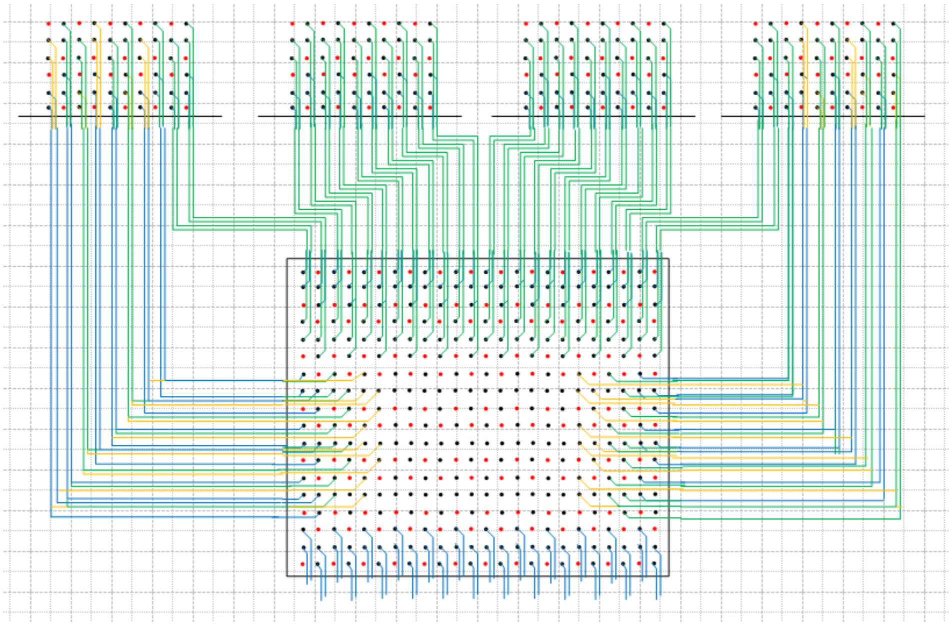


Figure 2: Placement Drawing

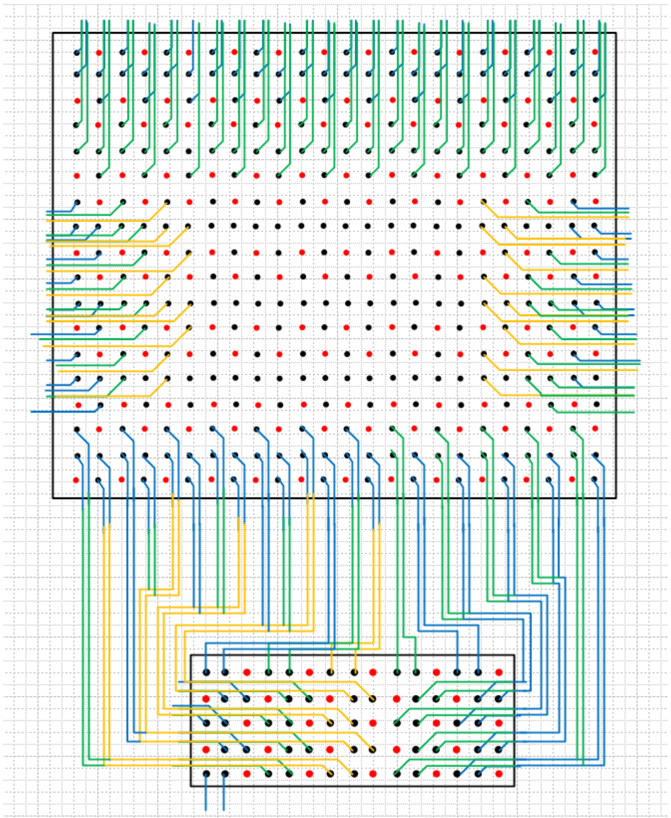
The backplane design was chosen for multiple reasons. As shown in the drawing below, it facilitates cooling. Only one unidirectional fan is needed to cool effectively. For other designs, such as midplane and orthogonal architectures, cooling as well as scalability becomes an issue. Overall the backplane design is much easier to implement in large systems given its scalability and portability, as well as ease of access and thus maintenance.



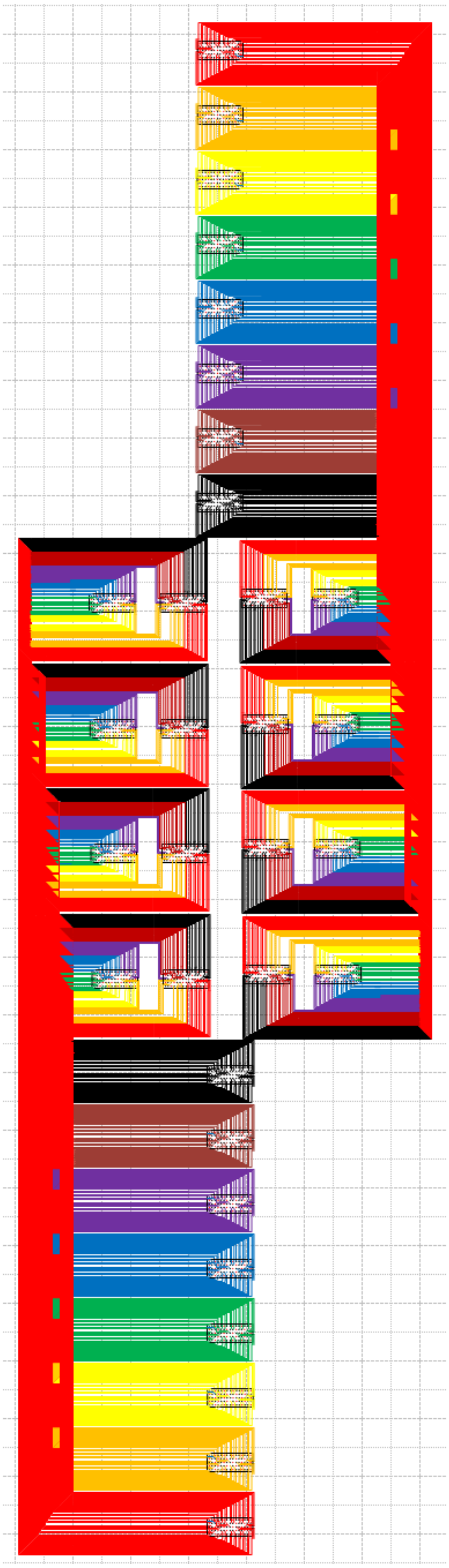
2.4 Routing Study Drawings



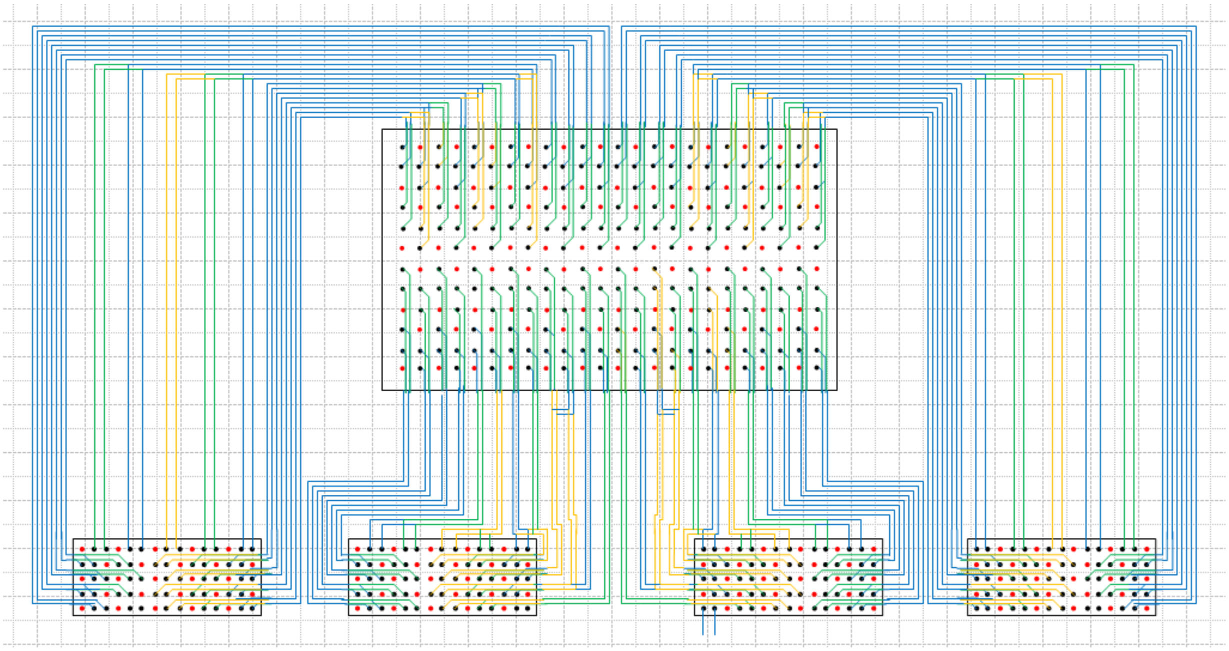
Linecard Routing: serdes to aggregate serdes



Linecard Routing: aggregate serdes to connector



Backplane Routing



Linecard Routing: crossbar to connectors

Figure 3: Routing Studies.

2.4.1 Routing Challenges

1) Placement and Position of line cards respective to Xbars.
   - This automatically constrains the longest and shortest paths.
2) Number of stack layers.
   - Necessary to minimize the number of layers while still producing a functional design. Cost and space increases dramatically with the number of layers.
   - Dual striplines were used to decrease the number of stack layers in the backplane, as well as the height of the PCB stack.
3) Reducing via capacitance.
   - Since back drilling is not used, need to route in such a way to minimize stack layers to reduce via sizes.
4) Routing of differential pairs.
   - In order to reduce crosstalk, differential pairs needed to be routed together.
5) Symmetry
   - By routing in a symmetric fashion, it makes it much easier to determine the longest and shortest paths.
   - Mitigates crosstalk because if path lengths are different, the ISI between differential pairs will be different because of different reflection coefficients due to different line lengths, further degrading performance.

Sections	**Shortest (in)**	**Longest (in)**
Line Card	*0.5*	*1*
Backplane	*6.67*	*15*
Xbar bitslices	*0.5*	*4*
Total Path	*0.5+6.67+0.5*	*1+15+4*

Table 1: Longest and shortest traces on all sections of design.

3 Signaling Description

3.1 Link Block Diagram

The figure below shows a generic drawing of our card-to-card signal links. It includes Tx, Rx, packages, connectors and transmission lines.

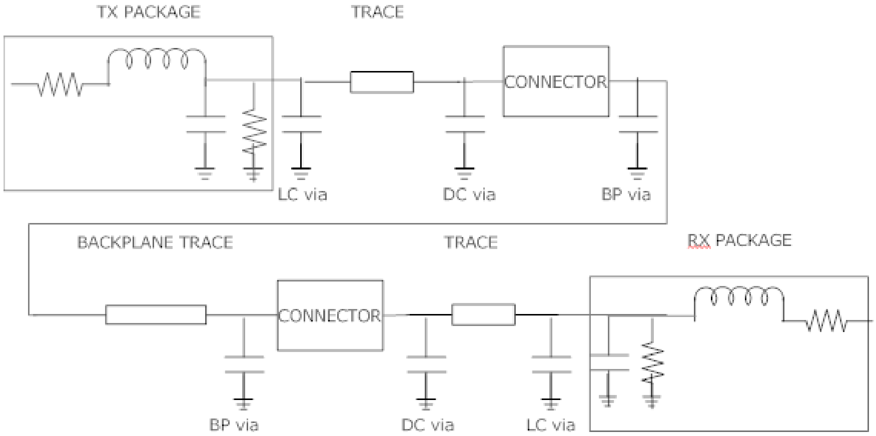


Figure 4: Generate layout of HSPICE simulation.

NOTE: In order to see values for each element please refer to attached spice deck.

3.2 Signaling Convention

3.2.1 Differential, Unidirectional, Current mode driver and receiver

Differential signaling was used within our system. In differential signaling mode, the receiver ignores the two signals voltages with respect to ground. Therefore, small changes in ground potential between transmitter and receiver do not adversely affect the receiver's ability to obtain a clear signal. As far as transmitters and receivers, current mode not only seems to be the more practical option from a design perspective, but also proves to be more immune to receiver Xtalk due to its high output impedance.

3.2.2 Special Circuitry to Reduce Transmitter Offsets

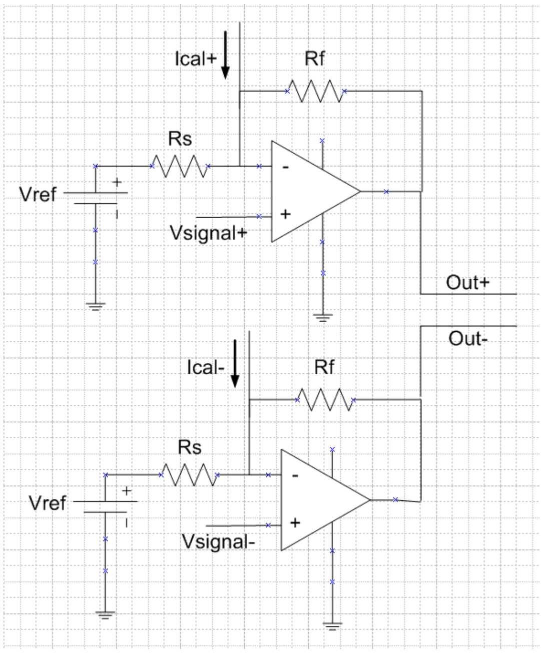


Figure 5: Circuitry to reduce transmitter offset.

We used current steering to reduce the receiver offset to 7.5mV from 100mV. This method was borrowed from Yasuda and Andoh's 'Differential Analog Data Path DC Offset Calibration Methods' paper. The received signals are shown as Vsignal± in Figure 5. After the circuit has been fabricated, Ical± are set to calibrate Out± to be equal, given equal inputs Vsignal+ and Vsignal-. Once Ical± have been set, the differential offset voltage can be calculated as follows:

$$\mathrm{Out}\pm = \frac{\mathrm{R_f+R_s}}{\mathrm{R_s}}\mathrm{V}\pm\ - \mathrm{I_{cal\pm}R_f} - \mathrm{V_r}\frac{\mathrm{R_f}}{\mathrm{R_s}}.$$

$$\mathrm{V}\pm = \frac{\mathrm{R_s}}{\mathrm{R_f+R_s}}(\mathrm{Out}\pm\ - \mathrm{I_{cal\pm}R_f} - \mathrm{V_r}\frac{\mathrm{R_f}}{\mathrm{R_s}}).$$

$$\mathrm{V_{diff\,offset}} = \mathrm{V_+} - \mathrm{V_-} = \frac{\mathrm{R_s}}{\mathrm{R_f+R_s}}(\mathrm{Out_+} + \mathrm{I_{cal+}R_f} - \mathrm{Out_-} - \mathrm{I_{cal-}R_f}).$$

With Out+ being equalized to Out-, this becomes

$$\frac{\mathrm{R_s}}{\mathrm{R_f+R_s}}(\mathrm{I_{cal+}R_f} - \mathrm{I_{cal-}R_f}) = \frac{\mathrm{R_sR_f}}{\mathrm{R_f+R_s}}(\mathrm{I_{cal+}} - \mathrm{I_{cal-}}).$$

Using 100Ω and 300Ω resistors for R_s and R_f, and a maximum current of 500µA, we can cancel 37.5mV of offset. But, if R_s and R_f have ±10% variation, we can cancel between 33.75mV and 41.25mV of offset, leaving us a gap of 7.5mV of offset that might not be cancelable. Thus, our differential clocked receiver is left with a possible 7.5mV maximum offset voltage, reduced from 40mV.

Note: The calibration current can be set using a digital to analog converter with the number of bits used determining the sensitivity of the resulting current.

3.2.3 Equalization

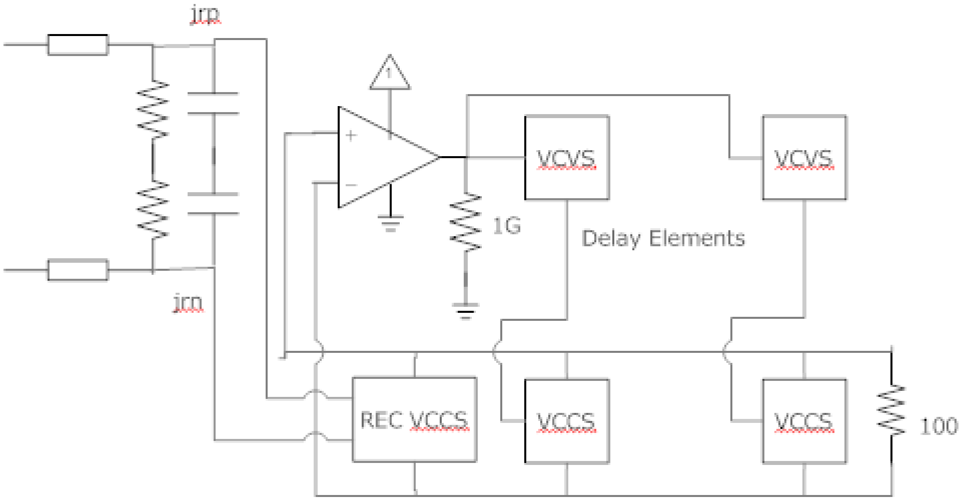


Figure 6: DFE Equalizer circuit

The equalization procedure that we used was a tandem effort between the FFE that was provided to us in the initial design template, as well as our custom designed DFE. First, it is necessary to understand how the DFE operates. The REC_VCCS is the voltage controlled current source receiver whose output current is controlled by differential signals (jrn, jrp). The receiver has a gain of 0.01 and termination impedance of 100Ω. The output of the receiver feeds back to a comparator which outputs 0 or 1, dependent on whether the signal is degraded or not. The comparator then drives a series of delay units which effectively sample the bit pattern at different time intervals, which are in turn used to drive the VCCS sources and output current based on the user provided tap settings on each of the delay elements. When given the correct tap settings the output will eventually converge.

As far as procedure on setting the taps, it was deemed that the DFE cannot tune the pre-cursor that well since it only looks at delayed versions of the bitpattern. Therefore, we used the FFE pre-cursor tap to reduce ISI with the cursor, and the DFE to moderate the post-cursors. Tuning the DFE portion was an iterative process. Initially the first post-cursor was tuned to an optimum value, then the second post-cursor, and so on until a desirable eye was obtained.

We would like to thank Pete Stevenson helping us set up our DFE.

3.2.4 Line Termination

We used source and receiving termination in order to match transmitter and receiver to the line. The terminations were placed **on-package**. Observing Figure 4, you can see how the on-die termination matches the entire system (line card trace + connector + backplane trace + connector + Xbar trace). If the terminations were placed on the line card trace or backplane trace, this would generate more ISI. We also performed the 4 corner test to account for termination tolerances. The worst case found was 45Ω terminating resistors on both the transmitting and receiving ends.

3.3 Timing Conventions

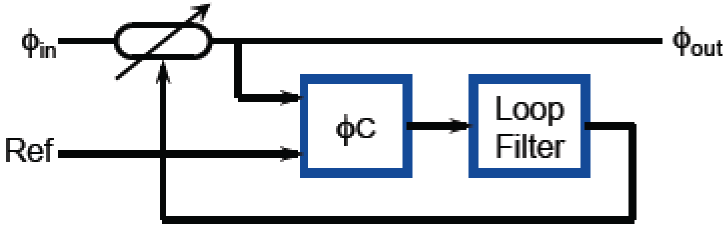
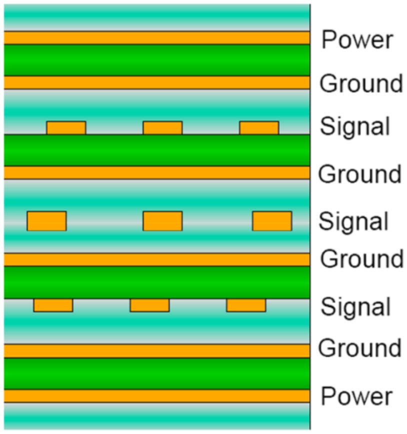


Figure 7: Clock Data Recovery Unit

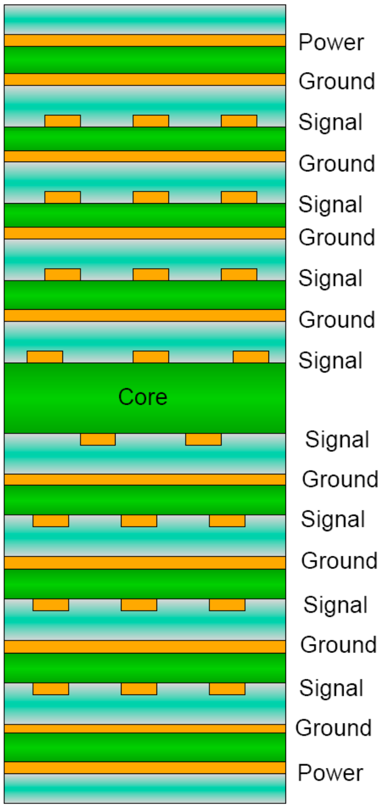
In our design we use per-bit timing using the PRBS7 encoding. The CDR is shown in the figure above. Ref is the reference clock that is generated per board. Board reference clocks are distributed over the boards as necessary using an H-tree. There are three important elements in the CDR. The variable delay unit, the phase comparator, and the loop filter. The incoming clock first runs through the variable delay unit and into the phase comparator (comparing to reference clock generated on board). This phase difference is then fed into a loop filter to reduce any high frequency noise, and then this signal is used as feedback to adjust the variable delay element. Eventually, the phases will converge and be matched.

4 Interconnect Description

4.1 PCB Stack-ups



Linecard PCB Stackup



Backplane PCB Stackup

Figure 8: PCB Stackups

4.1.1 Striplines

The electromagnetic wave carried by a microstripline exists partly in the dielectric substrate, and partly in the air above it, which means the medium in which an EM wave travels is inhomogeneous. In consequence, in cannot transmit a true TEM wave, and is more susceptible to Xtalk. Although striplines are easier to manufacture and ultimately occupy less space, microstriplines were chosen for our design due to the fact that they allow EM waves to travel in a homogeneous medium with the added benefit of reducing Xtalk.

4.1.2 FR4 (tand=0.3)

We decided to go with the standard FR-4 dielectric material with a tand of 0.3 to keep costs down. We still achieved our desired performance constraint of 8.4 Gbps / line with the lower cost material.

4.2 Signal Trace Descriptions

All of our PCB traces are 6mil wide, with 6mil of spacing between them, resulting in 50Ω trace impedance. The resulting Xtalk coefficients can be found from running the RLGC file.

$$K_b = \frac{k_c + k_m}{4} = \frac{\frac{L_{12}}{L_{11}} + \frac{C_{12}}{C_{11}}}{4} = \frac{\frac{3.754512\mathrm{e}-08}{3.346937\mathrm{e}-07} + \frac{-1.510695\mathrm{e}-11}{1.346700\mathrm{e}-10}}{4} = -1.03e-4$$

Since we are using striplines, the forwards Xtalk coefficient will be zero. Because the spacing between all of our lines will be greater than or equal to 6mils, the backwards Xtalk coefficient will be, at most, -.0103%. This is included in our spice channel simulations. We are not using back drilling on our backplane.

4.3 Connectors and Cables

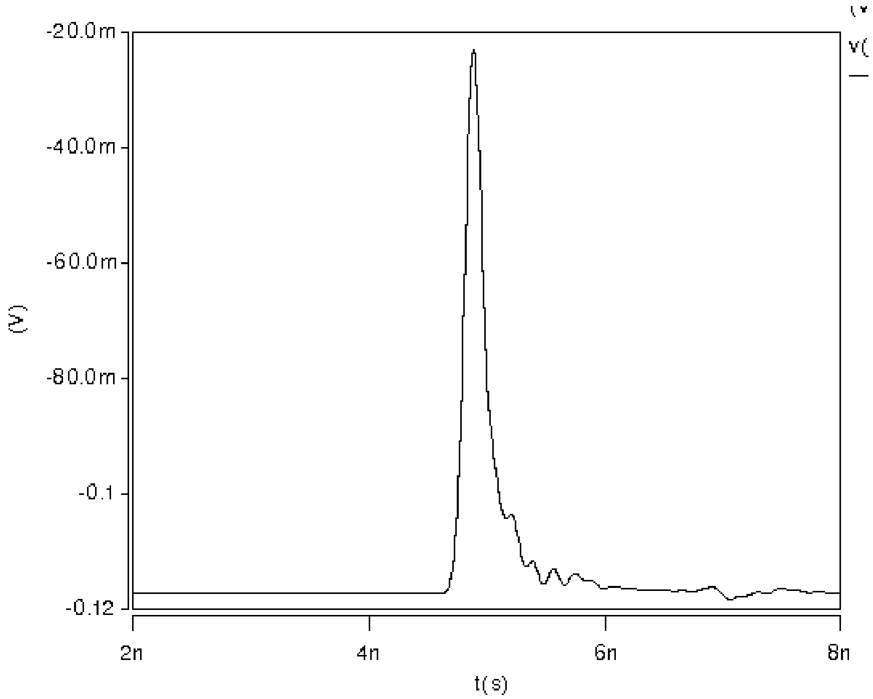
Connector Manufacturer:	Tyco Electronics
Model:	TinMan
Size:	15x5 pins, 1” x .4”
Impedance:	100Ω
Forward Xtalk:	0.5%
Backward Xtalk:	1.6%

4.4 Integrated Circuit Packages

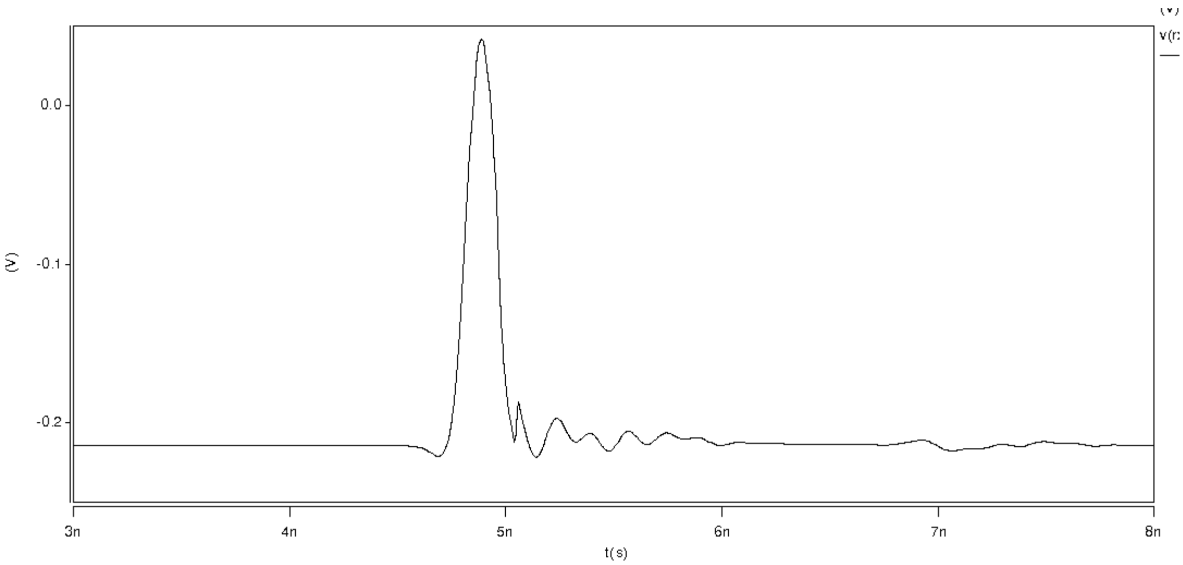
Package Types:	Bond wire
SERDES Dimensions:	18x24 pins, 7” x 9.5”
Crossbar chip Dimensions:	12x24 pins, 4.7” x 9.5”

5 Design Analysis

5.1 SPICE Simulation

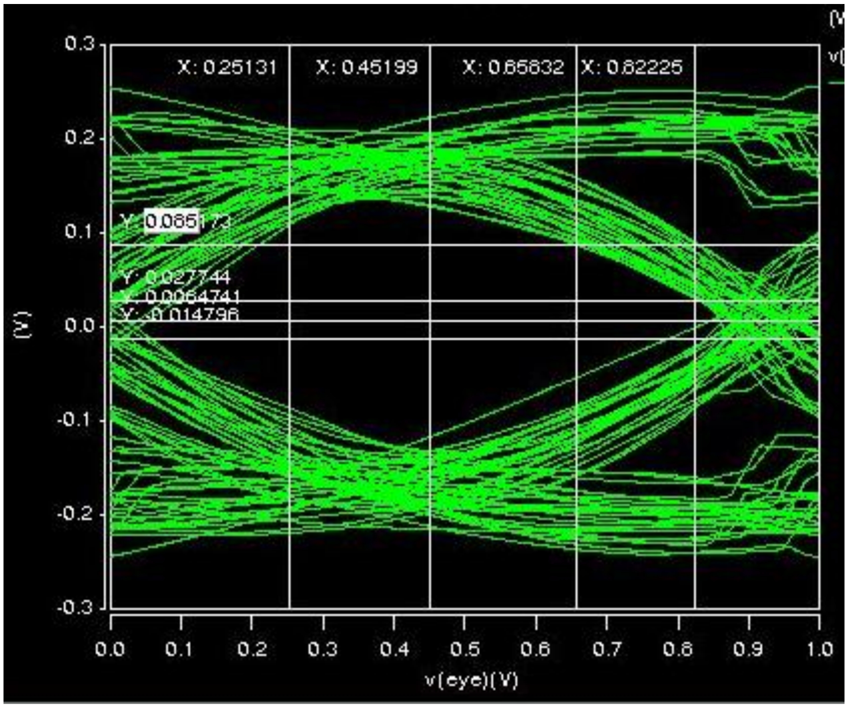


Unequalized Channel Single Bit Response

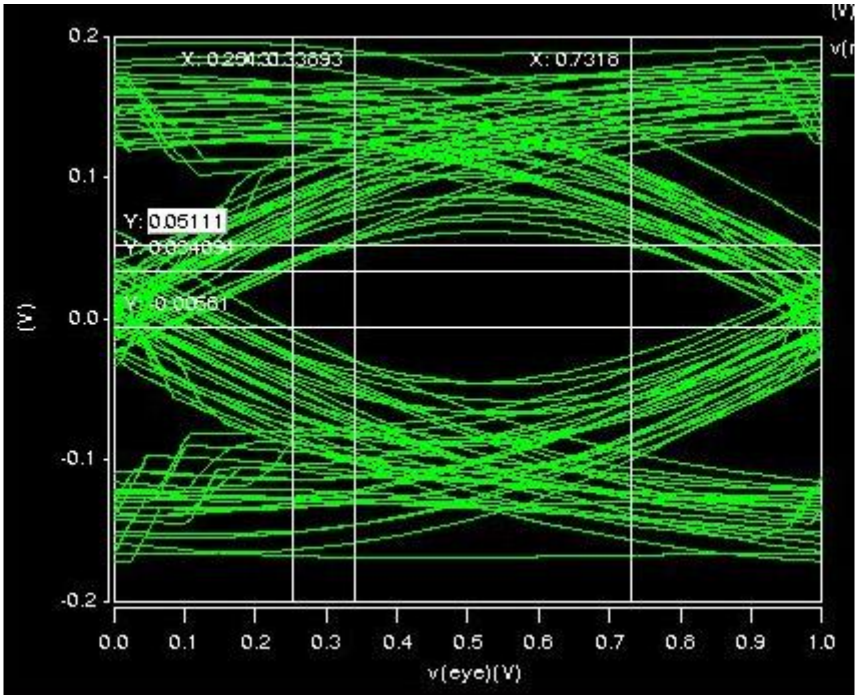


Equalized channel Single Bit Response

Figure 9: Unequalized and equalized channel single bit responses



Shortest Channel Path



Longest Channel Path

Figure 10: Shortest channel path and longest channel path

5.1.1 Simulation Procedure

5.1.1.1 Estimated longest and shortest paths

These were estimated based on a basic idea of routing design, connector dimensions, and estimated backplane and Xbar dimensions. TinMan connector widths were estimated to be 1" for line cards and Xbars. We estimated that there would need to be at least 2" of spacing between line cards in our backplane design to ensure that the connectors and any necessary routing could be realizable. Since the Xbar cards required more complex routing, we estimated that the spacing between Xbar cards should be 3", including the connector width, to ensure that routing was possible. All together, the longest backplane path was estimated to be 36" and the shortest 9". Longest and shortest line card and Xbar card routes were estimated to be 3" and 1" respectively.

5.1.1.2 Modeling the system in SPICE

Figure 4 shows the general layout of our system in spice. This included modeling the TX/RX packages, adding on-die terminations (w/ tolerances), modeling daughter card and backplane via capacitances, adding connectors, and using above estimations of shortest and longest paths as parameters for line card, backplane, and Xbar trace lengths between connector vias.

5.1.1.3 Equalization

The very first step was to determine the single pulse response without equalization. By looking at the relative sizes of the front porch and back porch of the response, we estimated the relative importance of the tap weights. It was clear that signal at the first post-cursor tap was producing the most ISI at the receiver; therefore, after finding a suitable value for the pre-cursor, the main concern was setting the post-cursor tap of the FFE. However, our DFE allowed us to tune the post-cursor tap using a feedback loop, which proved to be much more effective than using the FFE alone. How we set our DFE tap settings was discussed in Section 3.2.3 of the report. We did not move on to the next step until a suitable eye was generated.

5.1.1.4 Parameter Changing

Once we had a suitable eye, we initially attempted to find ways in which it could be degraded. We tested BGA vs. bond-wire packaging, different connectors (TinMan, Xcede, HM-ZD), different dielectric material, and added crosstalk. We also performed the 4 corner test to take into account any worst-case termination tolerances. Overall, we decided upon using **bond wire packaging, TinMan connector, FR4 with tand=0.3, and 45 Ohm TX end and 45 Ohm RX end terminations.** Once we felt that we had adequately accounted for proportional noise sources we starting increasing our bit rate to improve performance, and ultimately settled on **8.4 Gbps.**

5.1.1.5 Constraints

Our constraints basically acted as our feedback loop for re-tuning and re-simulating in order to ensure we met the project specifications as well as our design goals.

5.1.1.5.1 Noise Margin

Based on the given receiver sensitivity and offset, it was deemed that the minimum required eye height (from the edge of the CDR tolerance) needed to be 100mV assuming no noise margin. However, in section 3.2.2 we discuss how we were able to reduce the necessary margin to 35mV, which gave us more room to increase performance. Nonetheless, this 35mV was our sanity bound.

5.1.1.5.2 Timing Margin

For reliable recovery of the signal given the CDR tolerance, which amounts to 40% of the total bit time, the eye width (from the edge of the receiver offset margins) must be at least 40% to have any margin at all. Once again, if any simulations performed failed this criteria, parameters were readjusted to ensure that it was met.

5.1.1.5.3 Actual Routing

This was probably the most onerous task to account for, considering that our eye looked perfect for our estimated (albeit completely wrong) values of shortest and longest paths. Our actual routing diagrams clamped the shortest and longest paths, and at that point it was back to step 1. Unfortunately, there was no way around this except to deal with it.

5.2 Noise and Jitter

5.2.1 Noise Sources

5.2.1.1 Included in SPICE

Transmitter offset:	5%
Crosstalk:	Modeled by dual-stripline in SPICE
Termination Tolerances:	10%
Receiver/Offset Sensitivity:	35mV
ISI:	Accounted for in model
Differential current mode signaling:	No power supply noise to RX Xtalk

5.2.1.2 Not included in SPICE

All proportional noise sources are accounted for in SPICE. We cannot account for any RMS noise that may be introduced into system. We can only budget for how much RMS noise we can handle.

5.2.2 Timing noise Sources

5.2.2.1 Included in SPICE

We did not include any models of timing noise sources in SPICE.

5.2.2.2 Not included in SPICE

Accounted for CDR tolerance by running simulations, observing the eye, and using CDR tolerance as a check on whether or not the specific design under test was sufficient. Clock jitter is a type of randomly generated noise that could not be accounted for in SPICE. Like RMS voltage noise, jitter could only be budgeted for in our BER calculation for timing margin.

Jitter Sources:

- Transmission line length mismatch between differential pairs.
- Asymmetric dual stripline design.

NOTE: Since we did not pipeline the clock we did not need to worry about synchronization problems. All we needed to worry about was the CDR tolerances and possible sources of jitter. As for proportional noise sources, all possible sources that we could surmise were included in our SPICE simulations. RMS or random noise tolerance is determined by our resulting noise margin discussed below.

5.3 Bit Error Rates

The bit error rate is calculated from 2xQ(SNR), with the SNR defined to be the effective voltage margin / RMS noise. To get a BER of 10^{-14}, we need an SNR of 5.68 (from online tables).

Random Noise: V_{nm} = 37mV (from simulation of longest channel)

σ = 37mV / 5.68 = **6.51mV**

Jitter: V_{nm} = 9% of the bit-time (from simulation of longest channel)

σ = 9% / 5.68 = 1.5% -> **1.78pS**

5.4 Power Dissipation and Cost

5.4.1 Bill of Materials

Backplane: 18.5" x 7" = 835 cm^2, at 18 layers, 45 cents per cm^2	$376
Linecards: 20cm x 40cm, 9 layers, 25 cents per cm2	20 linecards x $200
Tinman connectors: 15x5 pins, 15 cents per pin	32 connectors x $11.25
Chips: $10 per chip	20 chips x $10
Xbar: 12 x 24 pins = 288 pins at 5 cents per pin	4 chips x $14.40
Aggregate serdes: 18 x 24 pins = 432 pins at 5 cents per pin	16 chips x $21.60
I/O power: $20 per watt, at 11.52W (see below)	$230.40
Total System Cost:	**$5,569.60**

5.4.2 Power Dissipation

Power = N*25mA*1.2V, with N being the number of Tx differential signal pairs in the design. So each linecard has 12 Tx differential pairs and each Xbar bitslice has 48 Tx differential pairs. So overall the router has 384 Tx differential pairs, giving a power dissipation of 11.52W.

6 Conclusion

Overall we were able to design a cost-effective, working system. However, our obtained noise and timing margins were a little on the small side. We didn't realize until the end just how bad our shortest and longest path length approximations were until we had to re-iterate over simulation procedure outlined in 5.1.1 multiple times to achieve a suitable eye. Nonetheless, we did set the bar high trying to run the system at 8.4 Gbps instead of the required >3.125 Gbps. A much easier solution would have been to go with 6.25 Gbps, and we certainly could have improved our noise and timing margins by going on that route.

7 Spice Deck

7.1 Project.sp

```
Project Channel

************************************************************************
* *
* Parameter Definitions *
* *
************************************************************************
* Simulation Run Time *
*.PARAM simtime = '160/bps' * Use/adjust for single pulse response
.PARAM simtime = '256/bps' * Use/adjust for and eye diagram

* Parameter Definitions for CLT *
.PARAM z1     = 0.1g           * Equalizer zero, Hz
.PARAM p1     = 10g       * Equalizer pole, Hz

* Driver Pre-emphasis Levels - ALTER AS REQUIRED BY YOUR DESIGN *
.PARAM pre1 = 0.00 * Driver pre-cursor pre-emphasis
.PARAM post1 = 0.00 * Driver 1st post-cursor pre-emphasis
.PARAM post2 = 0.00 * Driver 2nd post-cursor pre-emphasis
.PARAM post3 = 0.00 * Driver 2nd post-cursor pre-emphasis
.PARAM post4 = 0.00 * Driver 2nd post-cursor pre-emphasis
.PARAM post5 = 0.00 * Driver 2nd post-cursor pre-emphasis
.PARAM pre1feed = 0.00 * DFE pre-cursor pre-emphasis
.PARAM post1feed = 0.00 * DFE 1st post-cursor pre-emphasis
.PARAM post2feed = 0.00 * DFe 2nd post-cursor pre-emphasis
.PARAM post3feed = 0.00 * DFE 2nd post-cursor pre-emphasis
.PARAM post4feed = 0.00 * DFE 2nd post-cursor pre-emphasis
.PARAM post5feed = 0.00 * DFE 2nd post-cursor pre-emphasis

* Driver Voltage and Timing - ALTER AS REQUIRED BY YOUR DESIGN *
.PARAM vd = 625m * Driver peak to peak drive voltage, volts
.PARAM trise = 60p * Driver rise time, seconds
.PARAM tfall = 60p * Driver fall time, seconds
.PARAM bps = 8.4g * Bit rate, bits per second

* PCB Line Lengths - ALTER AS REQUIRED BY YOUR DESIGN *
.PARAM len1 = 3 * Line segment 1 length, inches
.PARAM len2 = 30 * Line segment 2 length, inches
.PARAM len3 = 3 * Line segment 3 length, inches

* Backplane Connector Via Parameters - ALTER TO REFLECT YOUR DESIGN *
* If you plan to back-drill, then leave len2via set to 0.03
* If you don''t back-drill, then set len1via & len2via to simulate
* one trace nearest and one farthest from the connector
*
.PARAM len1via = 0.1 * Active via length, inches
.PARAM len2via = 0.03 * Via stub length, inches

* Receiver Parameters - ALTER AS REQUIRED BY YOUR DESIGN *
.PARAM cload = 100f * Receiver input capacitance, farads
.PARAM rsource = 45 * Receiver input resistance, ohms
.PARAM rterm = 45 * Receiver input resistance, ohms

************************************************************************
* *
* Signal Source & Driver *
* *
************************************************************************
* Single Pulse Signal Source *
* Vs inp 0 PULSE (1 0 0 trise tfall '(1/bps)-trise' simtime)

* PRBS7 Signal Source - REPLACE AS NEEDED *
Xs inp inn (bitpattern) dc0=0 dc1=1 baud='1/bps' latency=0 tr=trise

* Behavioral Current Mode Driver with Pre-emphasis - REPLACE AS REQ''D *
* 1 pre-cursore & 2 psot-cursor taps *
```

```
Xf inp in (RCF) TDFLT='0.25*trise'
Xd in ppad npad (tx_4tap_diff) ppo=vd bps=bps a0=pre1 a2=post1 a3=post2 a4=post3 a5=post4
a6=post5

* CLT
*Xclt in ppad npad (rx_eq) az1=z1 ap1=p1 * Instantiate equalizer

*************************************************************************
* *
* Interconnect - ALTER TO REFLECT YOUR CHANNEL *
* *
*************************************************************************
* True Interconnect *
Xpp1 ppad jp1 (package) * Driver package model
R1 jp1 0 'rsource'
Xvp1 jp1 jp2 (via) Cvia=0.3p * Package via
Xlp1 jn2 jp2 jn3 jp3 (stripline6_fr4) length=len1 * Line seg 1
Xvp2 jp3 jp4 (via) Cvia=0.7p * Daughter card via
* Xkp1 0 jp4 jp5 (FPTOP) * Backplane connector
Xvp3 jp5 jp6 (mvia) * Backplane via
Xlp2 jn6 jp6 jn7 jp7 (stripline6_fr4) length=len2 * Line seg 2
Xvp4 jp7 jp8 (mvia) * Backplane via
* Xkp2 0 jp9 jp8 (FPTOP) * Backplane connector
Xvp5 jp9 jp10 (via) Cvia=0.7p * Daughter card via
Xlp3 jn10 jp10 jn11 jp11 (stripline6_fr4) length=len3 * Line seg 3
Xvp7 jp11 jp12 (via) Cvia=0.3p * Package via
Xpp2 jrp jp12 (package) * Recvr package model
R2 jp12 0 'rterm'

Xkpn1 jp4 jn4 jp5 jn5 0 (TM195PRNO)
*****Xkpn1 jp4 jn4 jp5 jn5 0 (TM195PRAB)
Xkpn2 jp9 jn9 jp8 jn8 0 (TM195PRNO)
*****Xkpn2 jp9 jn9 jp8 jn8 0 (TM195PRAB)

* Compliment Interconnect *
Xpn1 npad jn1 (package) * Driver package model
R3 jn1 0 'rsource'
Xvn1 jn1 jn2 (via) Cvia=0.3p * Package via
*Xln1 jn2 jn3 (stripline6_fr4) length=len1 * Line seg 1
Xvn2 jn3 jn4 (via) Cvia=0.7p * Daughter card via
* Xkn1 0 jn4 jn5 (FPTOP) * Backplane connector
Xvn3 jn5 jn6 (mvia) * Backplane via
*Xln2 jn6 jn7 (stripline6_fr4) length=len2 * Line seg 2
Xvn4 jn7 jn8 (mvia) * Backplane via
* Xkn2 0 jn9 jn8 (FPTOP) * Backplane connector
Xvn5 jn9 jn10 (via) Cvia=0.7p * Daughter card via
*Xln3 jn10 jn11 (stripline6_fr4) length=len3 * Line seg 3
Xvn7 jn11 jn12 (via) Cvia=0.3p * Package via
Xpn2 jrn jn12 (package) * Recvr package model
R4 jn12 0 'rterm'

*************************************************************************
* *
* Behavioral Receiver - REPLACE WITH YOUR RECEIVER *
* *
*************************************************************************
Rrp1 jrp 0 rterm
Rrn1 jrn 0 rterm
Crp1 jrp 0 cload
Crn1 jrn 0 cload

* Differential Receive VCVS *
*Ex  rx_diff 0  (jrp,jrn) 1
*Rx  rx_diff 0  1G

* DFE tap *
Xfeed jrp jrn rx_diff 0 (tx_4tap_feed_diff) ppo=vd bps=bps a0=pre1feed a2=post1feed a3=post2feed
a4=post3feed a5=post4feed a6=post5feed

* Eye Diagram Horizontal Source *
Veye1 eye 0 PWL (0,0 '1./bps',1 R TD='edui/bps')
```

```
Reye eye 0 1G

*************************************************************************
* *
* Sub-Circuit Definitions *
* *
*************************************************************************

* Motherboard Via Sub-circuit *
* zvia = via impedance, ohms
* len1via = active via length, inches
* len2via = via stub length, inches
* prop = propagation time, seconds/inch
*
.SUBCKT (mvia) in out zvia=42 len1via=0.09 len2via=0.03 prop=180p
T1 in 0 out 0 Z0=zvia TD='len1via*prop'
T2 out 0 2 0 Z0=zvia TD='len2via*prop'
.ENDS (mvia)

* Daughter Card Via Sub-circuit -- typical values for 0.093"" thick PCBs *
.SUBCKT (via) in out Rvia=1m Lvia = 0.5n Cvia = 0.7p
X1 in 1 (section_t) R_sec='0.5*Rvia' L_sec='0.5*Lvia' C_sec='0.5*Cvia'
X2 1 out (section_t) R_sec='0.5*Rvia' L_sec='0.5*Lvia' C_sec='0.5*Cvia'
.ENDS (via)

* Generic "T" Section *
.SUBCKT (section_t) in out R_sec=1m L_sec=5n C_sec=2p
Rs1 in 1 '0.5*R_sec'
Ls1 1 2 '0.5*L_sec'
Cs1 2 0 C_sec
Ls2 2 3 '0.5*L_sec'
Rs2 3 out '0.5*R_sec'
.ENDS (section_t)

********
* Simple package model - REPLACE WITH YOUR PACKAGE MODEL *
* Package Parameters *
.PARAM GENpkgR = 0.337 * Typ GEN package trace resistance, ohms
.PARAM GENpkgL = 5.675n * Typ GEN package trace induct., henries
.PARAM GENpkgC = 2.27p * Typ GEN package trace capac., farads
*
* Generic 5-section Package Model *
.SUBCKT (gen_pkg) in out R_pkg=GENpkgR L_pkg=GENpkgL C_pkg=GENpkgC
X1 in 1 (section_t) R_sec='0.25*R_pkg' L_sec='0.25*L_pkg' C_sec='0.25*C_pkg'
X2 1 2 (section_t) R_sec='0.25*R_pkg' L_sec='0.25*L_pkg' C_sec='0.25*C_pkg'
X3 2 3 (section_t) R_sec='0.25*R_pkg' L_sec='0.25*L_pkg' C_sec='0.25*C_pkg'
X4 3 out (section_t) R_sec='0.25*R_pkg' L_sec='0.25*L_pkg' C_sec='0.25*C_pkg'
.ENDS (gen_pkg)
********

  **********************************************************
  *                                                        *
  *             Pole/Zero Linear Equalizer                 *
  *                                                        *
  **********************************************************
  *
  *    az1 = zero location in Hz (real axis)
  *    ap1 = pole location in Hz (real axis)
  *
  * Overall gain is normalized to 1 at high-band.
  * Default value for az1 & ap1 are over-ridden at instantiation
  *
  .SUBCKT (rx_eq) in outp outn az1=1k ap1=10k fz1=0 fp1=0 pi=3.14159265359
      E1 outp outn POLE in 0    1   'az1' 'fz1' /
+                           1   'ap1' 'fp1'
  .ENDS (rx_eq)
  **********************************************************

.SUBCKT (package) in out C_pkg_esd=1p L_pkg_bump=3n Trace_len_pkg=1cm L_pkg_ball=0.5n
X1 in 1 (section_t) R_sec=0 L_sec=L_pkg_bump C_sec=C_pkg_esd
T1 1 0 2 0 Z0=50 TD=60p
```

```
X3 2 out (section_t) R_sec=0 L_sec=L_pkg_ball C_sec=0
.ENDS

********
* Simple connector model - REPLACE WITH YOUR CONNECTOR MODEL *
.SUBCKT (conn) ref in out
T1 in ref out ref Z0=48 TD=100p
.ENDS (conn)
********

*************************************************************************
* *
* Included Files - MODIFY TO POINT TO YOUR LOCATIONS *
* *
*************************************************************************
.INCLUDE './stripline6_fr4_new.inc'
.INCLUDE './prbs7.inc'
.INCLUDE './tx_4tap_diff_pc.inc'
.INCLUDE './tx_4tap_feed_diff_pc.inc'
.INCLUDE './filter.inc'
.INCLUDE 'connectors/TinMan/Spice_Models/TM195PRAB.SLM'
.INCLUDE 'connectors/TinMan/Spice_Models/TM195PRNO.SLM'
.INCLUDE 'connectors/Xcede/Spice_Models/xcede_2diffpair_conn.inc'

*************************************************************************
* *
* Simulation Controls and Alters *
* *
*************************************************************************
.OPTIONS post
.TRAN 5p simtime start='160/bps' SWEEP DATA=pdata

.DATA pdata

len1    len2   len3    pre1   post1   post2  post3   post4  post5   edui    pre1feed
        post1feed      post2feed      post3feed      post4feed      post5feed
1          15      4      -0.10   -0.10      -0.10   -0.00  -0.00   -0.00   0.60   0.02
        -0.00          0.00           -0.00          -0.00          -0.00 * Equalized 8.4

*len1 len2 len3 pre1 post1 post2 post3 post4 post5 edui
*3.0 6.0 3.0 -0.1 -0.90 -0.0 -0.0 0 0 0.3 * Equalized 6.25
*3.0 30.0 3.0 0 -0.70 -0.3 -0.0 0 0 0.6 * Equalized 8.4

*len1 len2 len3 pre1 post1 post2 edui
*3.0 30.0 3.0 0 0 -0.00 0.3 * Unequalized
*3.0 30.0 3.0 -0.2 -0.8 -0.00 0.3 * Equalized

.ENDDATA pdata

.END
```

7.2 Stripline6_fr4_new.inc

```
*************************************************************************
*                                                                       *
*                   6 mil Wide 50 ohm Stripline in FR4                  *
*                                                                       *
*************************************************************************

 .SUBCKT (stripline6_fr4) in1 in2 out1 out2 length=1 *inch
    W1 in1 in2 0 out1 out2 0 RLGCMODEL=cond2_sys N=2 l='length*0.0254'
 .ENDS (stripline6_fr4)


*SYSTEM_NAME : cond2_sys
*
*  ------------------------------------ Z = 4.012800e-04
*  //// Top Ground Plane /////////////
*  ------------------------------------ Z = 3.860400e-04
*       diel_1   H = 3.708000e-04
*  ------------------------------------ Z = 1.524000e-05
*  //// Bottom Ground Plane ///////////
*  ------------------------------------ Z = 0

* L(H/m), C(F/m), Ro(Ohm/m), Go(S/m), Rs(Ohm/(m*sqrt(Hz)), Gd(S/(m*Hz))


.MODEL cond2_sys W MODELTYPE=RLGC, N=2
+ Lo = 3.346937e-07
+      3.754512e-08 3.346937e-07
+ Co = 1.346700e-10
+      -1.510695e-11 1.346700e-10
+ Ro = 7.474938e+00
+      0.000000e+00 7.474938e+00
+ Go = 0.000000e+00
+      -0.000000e+00 0.000000e+00
+ Rs = 1.024984e-03
+      -4.980763e-05 1.024984e-03
+ Gd = 2.538470e-11
+      -2.847593e-12 2.538470e-11
```

7.3 tx_4tap_diff_pc.inc

```
*************************************************************************
*                                                                       *
*                       Behavioral Four Tap Equalizer                   *
*                                                                       *
*************************************************************************
*
* This parameterized sub-circuit implements a 4-tap pre-emphasis driver
* with 1 precursor, 1 cursor and 2 postcursor taps.
*
* Parameter Meanings:
*    ppo = Differential peak to peak output voltage into 100 ohms
*    bps = signaling rate in bits per second
*    a0 = pre-cursor as fraction of max full swing
*    a2 = 1st post-cursor as fraction of max full swing
*    a3 = 2nd post-cursor as fraction of max full swing
*
* *** RESTIRCTIONS: ***
* 1. This differential driver must be loaded with 100 ohms to give correct levels
* 2. The input swing must be zero to 1 volt
*
 .SUBCKT (tx_4tap_diff) in outp outn  ppo=700m bps=840p a0=0 a2=0 a3=0 a4=0 a5=0 a6=0
    R1  in 1  1G                                        * Terminate "open" input
    V1  1  0  DC 0.5
    Ed1 2  0  DELAY (in,1) TD='1/bps'                   * One bit delay
    Ed2 3  0  DELAY (in,1) TD='2/bps'                   * Two bit delay
    Ed3 4  0  DELAY (in,1) TD='3/bps'                   * Three bit delay
        Ed4 5  0  DELAY (in,1) TD='4/bps'                       * Four bit delay
        Ed5 6  0  DELAY (in,1) TD='5/bps'                       * Five bit delay
        Ed6 7  0  DELAY (in,1) TD='6/bps'                       * Six bit delay
    R2  2  0  1G                                        * Delay termination
    R3  3  0  1G                                        * Delay termination
    R4  4  0  1G                                        * Delay termination
        R7  5  0  1G                                    * Delay termination
        R8  6  0  1G                                    * Delay termination
        R9  7  0  1G                                    * Delay termination
    *
    G0  outp outn  (in,1) '0.5*ppo*a0/14'               * Pre-cursor source
    G1  outp outn  (2,0)  '0.5*ppo*(1-(a0+a2+a3+a4+a5+a6))/14'  * Cursor source
    G2  outp outn  (3,0)  '0.5*ppo*a2/14'               * 1st post-cursor source
    G3  outp outn  (4,0)  '0.5*ppo*a3/14'               * 2nd post-cursor source
        G4  outp outn  (5,0)  '0.5*ppo*a4/14'           * 3rd post-cursor source
        G5  outp outn  (6,0)  '0.5*ppo*a5/14'           * 4th post-cursor source
        G6  outp outn  (7,0)  '0.5*ppo*a5/14'           * 4th post-cursor source
    R5  outp 0     50                                   * Driver output termination
    R6  outn 0     50                                   * Driver output termination
 .ENDS (tx_4tap_diff)
```

7.4 tx_4tap_feed_diff_pc.inc

```
*************************************************************************
*                                                                       *
*                       Behavioral Four Tap Equalizer                   *
*                                                                       *
*************************************************************************
*
* This parameterized sub-circuit implements a 4-tap pre-emphasis driver
* with 1 precursor, 1 cursor and 2 postcursor taps.
*
* Parameter Meanings:
*    ppo = Differential peak to peak output voltage into 100 ohms
*    bps = signaling rate in bits per second
*    a0 = pre-cursor as fraction of max full swing
*    a2 = 1st post-cursor as fraction of max full swing
*    a3 = 2nd post-cursor as fraction of max full swing
*
* *** RESTIRCTIONS: ***
* 1. This differential driver must be loaded with 100 ohms to give correct levels
* 2. The input swing must be zero to 1 volt
*
 .SUBCKT (tx_4tap_feed_diff) in1 in2 outp outn  ppo=700m bps=840p a0=0 a2=0 a3=0 a4=0 a5=0 a
6=0

    .PARAM cur = 10e-3

    *************************** Differential Receive VCCS ******************************
    Gx  outp outn  (in2,in1)  0.01
    Rx  outp outn  100                                  * Driver output termination
    ************************************************************************************

    *********************************** Comparator ************************************
    Ecomp  outcomp 0  PWL(1)  outp  outn  -1f,-1v  1f,1v
    ***********************************************************************************

    R1  outcomp 0  1G                                   * Terminate "open" input
    Ed1 2  0  DELAY (outcomp,0) TD='1/bps'                      * One bit delay
    Ed2 3  0  DELAY (outcomp,0) TD='2/bps'                      * Two bit delay
    Ed3 4  0  DELAY (outcomp,0) TD='3/bps'                      * Three bit delay
    Ed4 5  0  DELAY (outcomp,0) TD='4/bps'                      * Four bit delay
    Ed5 6  0  DELAY (outcomp,0) TD='5/bps'                      * Five bit delay
    Ed6 7  0  DELAY (outcomp,0) TD='6/bps'                      * Six bit delay
    R2  2  0  1G                                        * Delay termination
    R3  3  0  1G                                        * Delay termination
    R4  4  0  1G                                        * Delay termination
    R7  5  0  1G                                        * Delay termination
    R8  6  0  1G                                        * Delay termination
    R9  7  0  1G                                        * Delay termination
    *

    G1  outp outn  (2,0)  'a0*cur'      * Cursor source
    G2  outp outn  (3,0)  'a2*cur'              * 1st post-cursor source
    G3  outp outn  (4,0)  'a3*cur'              * 2nd post-cursor source
    G4  outp outn  (5,0)  'a4*cur'              * 3rd post-cursor source
    G5  outp outn  (6,0)  'a5*cur'              * 4th post-cursor source
    G6  outp outn  (7,0)  'a6*cur'              * 5th post-cursor source
 .ENDS (tx_4tap_feed_diff)
```