ABSTRACT

YAN, ZHUO. S-Parameter Based Binary Multimode Interconnect Design Methodology and Implementation. (Under the direction of Dr. Paul D. Franzon).

As higher density of interconnects and packages are demanded, crosstalk noise is becoming more important in input/output (I/O) design. Multimode signaling (or modal signaling) offers the ability to improve wiring density with almost zero crosstalk. To make best use of crosstalk cancelling by multimode signaling in real world applications, practical channels including packages, sockets, and printed circuit boards (PCBs) need to be considered. However, optimum implementation of a realistic multimode signaling subsystem requires a systematic approach to co-design the circuits and the interconnect channel.

This dissertation describes a design methodology for multimode signaling circuit/channel co-design. It is based on a new transceiver design, referred to as “binary multimode interconnect” because the coder/decoder (CODEC) is implemented only in the transmitter to produce conventional binary signals at receiver input with no further decoding needed. It was simulated to operate at an aggregate bandwidth of 16 Gbps and at a power efficiency of 5.6 mW/Gbps in 130 nm CMOS (which would scale with process technology). This was combined with an S-Parameter based CODEC derivation method. A fully tunable multimode signaling transceiver is designed to implement the CODEC. Both the CODEC coefficients and individual channel timing parameters are tunable. The methodology is as follows; first the set of baseline CODEC coefficients and timing parameters are derived from the S-parameters extracted from the baseline channel design. Then the different components in the channel are tuned for high density and impedance characteristics so as to minimize root mean square (RMS) jitter and bit error rate. Thus the circuit and channel are co-optimized, with the
goal of maximizing channel density. The channel can potentially incorporate different elements, such as both a package and a PCB, which are co-optimized together.

Several benchmarks have been used to validate the S-Parameter generated CODEC. Applying the CODEC to an existing channel resulted in around a 50% RMS jitter reduction over traditional single-ended signaling. Permitting an optimized channel to be designed from scratch led to a greater jitter reduction and ability to recover the signals with tighter routing pitch. The optimized channel showed over 75% root mean square (RMS) jitter reduction compared with single-ended signaling. Compared with a practical benchmark channel, the optimized channel has a density improvement of 300% and 97% for the printed circuit board (PCB) and package routing respectively, and 300% for package vertical pitch (µvia, PTH, socket, etc.) with signal to ground pin ratio 2:1. Although the optimized channel has much higher density, it still shows 46% jitter reduction over the benchmark channel with the same crosstalk mitigation circuit and method. This new approach significantly improves the ability to use high density multimode interconnect sub-systems in practical scenarios.

In the work described above, the channel was assumed to be bundled four wires at a time. Preliminary investigation of an eight-trace PCB channel shows about 55% RMS jitter reduction is achieved with a simple CODEC implementation.
S-Parameter Based Binary Multimode Interconnect
Design Methodology and Implementation

by
Zhuo Yan

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Electrical Engineering

Raleigh, North Carolina

2015

APPROVED BY:

Dr. Paul D. Franzon
Committee Chair

Dr. William R. Davis

Dr. John Wilson

Dr. Min Kang
DEDICATION

To my beloved wife Anqi, our children Mia and Jason
BIOGRAPHY

Zhuo Yan received his B.S. degree in electrical engineering from Beijing University of Aeronautics and Astronautics, Beijing (2007), and his M.S. degree from North Carolina State University in electrical engineering (2010). He started his Ph.D. research work with Dr. Paul D. Franzon in summer 2010. He has held a summer internship at Rambus Inc. in 2013 and developed an automatic power delivery network (PDN) characterization flow for their validation platform. His research interests include high performance signaling systems, signal integrity analysis, PDN characterization, and 3D integrated circuits. He is a member of the IEEE.
ACKNOWLEDGMENTS

First and foremost, I would also like to thank my advisor, Dr. Paul Franzon for his endless support and guidance without which this Ph.D work wouldn’t be possible. Whenever I felt lost and puzzled during this journey, he always shed lights on my path and show me the right way to keep carrying on. I’ll always be grateful for his help, advice and encouragement.

I would like to thank Dr. W. Rhett Davis, Dr. John Wilson and Dr. Min Kang who serve as my committee members for all the great advices they gave on my research works.

Special thanks to Dr. Steve Lipa for his great help on wire-bonding, measurement support, and many hours debugging my test board with me even nights and weekends.

I want to thank Dr. Henning Braunisch and Dr. Kemal Aygün from Intel for their mentorship and many helpful suggestions and guidance through teleconferences and in-person meetings.

I would like to thank all my colleagues and friends at North Carolina State University for all the informative and constructive discussion and good time together. They are listed without any meaningful order: Jianchen Hu, Yi Lou, Fan Zhang, Yue Ma, Zhenqian Zhang, Wenxu Zhao, Weiyi Qi, Weifu Li, Peter Gadfort, Gary Charles, Ataul Karim, Eric Wyers, Jong Beom Park, Joshua Schabel, etc.

Last, and most of all, I would like to thank my wife, Anqi for her endless love, support and encouragement through this incredible journey we had together.
LIST OF TABLES

Table 4.1 Design Parameters For PCB and Package Routing Traces .......................................47
Table 4.2. RLGC model of PCB only channel........................................................................48
Table 4.3. Rounded CODEC matrices for PCB only channel...............................................50
Table 4.4 Design Parameters For Package Vertical Components ........................................59
Table 4.5 Detail Comparison between Benchmark Channel and New Channel ....................68
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1.1</td>
<td>ITRS predicted signaling speed and number of I/Os</td>
<td>1</td>
</tr>
<tr>
<td>Figure 2.1</td>
<td>(a) Even propagation mode (b) Odd propagation mode</td>
<td>5</td>
</tr>
<tr>
<td>Figure 2.2</td>
<td>Different arrival time between modes</td>
<td>6</td>
</tr>
<tr>
<td>Figure 2.3</td>
<td>Stub-alternated microstrip line for memory interface</td>
<td>7</td>
</tr>
<tr>
<td>Figure 2.4</td>
<td>Serpentine guard trace structure</td>
<td>8</td>
</tr>
<tr>
<td>Figure 2.5</td>
<td>Added mutual capacitance between signal lanes</td>
<td>8</td>
</tr>
<tr>
<td>Figure 2.6</td>
<td>Active Xtalk Elimination (AXE) on realistic channel models</td>
<td>9</td>
</tr>
<tr>
<td>Figure 2.7</td>
<td>Proposed FEXT channel and XTCR analog front-end block diagram</td>
<td>10</td>
</tr>
<tr>
<td>Figure 2.8</td>
<td>CDMA-like signaling technique block diagram</td>
<td>11</td>
</tr>
<tr>
<td>Figure 2.9</td>
<td>Modal signaling system block diagram</td>
<td>13</td>
</tr>
<tr>
<td>Figure 2.10</td>
<td>Optimal termination network for modal signaling</td>
<td>14</td>
</tr>
<tr>
<td>Figure 3.1</td>
<td>Conventional implementation of multimode signaling</td>
<td>18</td>
</tr>
<tr>
<td>Figure 3.2</td>
<td>Eight-port channel</td>
<td>23</td>
</tr>
<tr>
<td>Figure 3.3</td>
<td>Block diagram of transmitter only MMI</td>
<td>26</td>
</tr>
<tr>
<td>Figure 3.4</td>
<td>Voltage control delay line schematic</td>
<td>27</td>
</tr>
<tr>
<td>Figure 3.5</td>
<td>Tunable CODEC driver</td>
<td>28</td>
</tr>
<tr>
<td>Figure 3.6</td>
<td>Digital controlled vernier delay circuit</td>
<td>29</td>
</tr>
<tr>
<td>Figure 3.7</td>
<td>Channel components in the benchmark problem and the corresponding signaling path</td>
<td>31</td>
</tr>
<tr>
<td>Figure 3.8</td>
<td>Overall Signal to Noise Ratio with decomposition eigenvector matrix derived at each frequency point</td>
<td>33</td>
</tr>
<tr>
<td>Figure 3.9</td>
<td>S-parameters for PCB only: (a) Original S-parameters and (b) modified S-parameters</td>
<td>34</td>
</tr>
<tr>
<td>Figure 3.10</td>
<td>S-parameters for complete channel: (a) Original S-parameters and (b) modified S-parameters</td>
<td>35</td>
</tr>
<tr>
<td>Figure 3.11</td>
<td>Traditional single ended current mode driver</td>
<td>36</td>
</tr>
<tr>
<td>Figure 3.12</td>
<td>(a) Benchmark channel with traditional single ended current mode driver (b) Benchmark channel with transmitter only multimode signaling</td>
<td>37</td>
</tr>
<tr>
<td>Figure 3.13</td>
<td>Binary multimode transceiver layout</td>
<td>38</td>
</tr>
<tr>
<td>Figure 3.14</td>
<td>(a) Fabricated test board with test chips wire bonded (b) Vernier caliper to tune the phases</td>
<td>40</td>
</tr>
<tr>
<td>Figure 3.15</td>
<td>Probe station setup</td>
<td>41</td>
</tr>
<tr>
<td>Figure 3.16</td>
<td>Output pin signal at 3 GHz</td>
<td>42</td>
</tr>
<tr>
<td>Figure 3.17</td>
<td>Output pin signal at 400 MHz</td>
<td>43</td>
</tr>
<tr>
<td>Figure 4.1</td>
<td>Design method illustration</td>
<td>46</td>
</tr>
<tr>
<td>Figure 4.2</td>
<td>Cross-section of routing traces</td>
<td>47</td>
</tr>
<tr>
<td>Figure 4.3</td>
<td>Eye diagram for PCB channel without CODEC</td>
<td>49</td>
</tr>
<tr>
<td>Figure 4.4</td>
<td>Crosstalk free eye diagram of PCB channel</td>
<td>49</td>
</tr>
<tr>
<td>Figure 4.5</td>
<td>S-parameters for PCB only channel: (a) Original S-parameters and (b) modified S-parameters</td>
<td>51</td>
</tr>
<tr>
<td>Figure 4.6</td>
<td>Eye diagram for PCB channel with multimode signaling</td>
<td>52</td>
</tr>
<tr>
<td>Figure 4.7</td>
<td>Full channel without multimode signaling</td>
<td>54</td>
</tr>
<tr>
<td>Figure 4.8</td>
<td>Full channel with multimode signaling</td>
<td>54</td>
</tr>
<tr>
<td>Figure 4.9</td>
<td>Full matched channel without multimode signaling</td>
<td>55</td>
</tr>
<tr>
<td>Figure 4.10</td>
<td>Full matched channel with multimode signaling</td>
<td>56</td>
</tr>
<tr>
<td>Figure 4.11</td>
<td>Full channel with optimum package for multimode signaling</td>
<td>57</td>
</tr>
<tr>
<td>Figure 4.12</td>
<td>Eye diagram for crosstalk free full channel</td>
<td>57</td>
</tr>
</tbody>
</table>
Figure 4.13 Cross-section view of package vertical components
Figure 4.14 Package Vertical Components Top View.
Figure 4.15 (a) Impedance profile of package vertical components with small voids. (b) Impedance profile of package vertical components with same cutout area on each layer.
Figure 4.16 Capacitive coupling due to insufficient void in package verticals.
Figure 4.17 No capacitive coupling due to increased void in package vertical.
Figure 4.18 (a) 1 mm pitch package vertical full channel with multimode. (b) 1 mm pitch package vertical full channel with traditional single ended driver.
Figure 4.19 (a) 0.25 mm pitch package vertical full channel with multimode. (b) 0.25 mm pitch package vertical full channel with traditional single ended driver.
Figure 4.20 (a) Benchmark problem full channel impedance profile. (b) Optimized full channel impedance profile.
Figure 4.21 (a) Insertion loss and power sum of FEXT of Benchmark channel. (b) Insertion loss and power sum of FEXT of optimized channel.
Figure 4.22 Benchmark channel waveforms (a) Aggressor pulse signal at channel 1 input. (b) Crosstalk victim at channel 2 output. (c) Crosstalk victim at channel 3 output. (d) Crosstalk victim at channel 4 output.
Figure 4.23 Optimized channel waveforms (a) Aggressor pulse signal at channel 1 input. (b) Crosstalk victim at channel 2 output. (c) Crosstalk victim at channel 3 output. (d) Crosstalk victim at channel 4 output.
Figure 5.1 Optimized full channel with reduced MMI CODEC
Figure 5.2 Cross-section of 8 channel PCB model.
Figure 5.3 Large bundle example 8 channel PCB model (a) Eye diagram with conventional single ended driver (b) with reduced CODEC binary MMI driver.
Chapter 1

Introduction

1.1 Motivation

Due to aggressive integrated circuit scaling and shifts in computer architectures from single-core to future many-core systems, the on-chip aggregate bandwidth is rapidly scaled into Tb/s range [33]. This trend calls for an increase in off-chip communication bandwidth in order to not limit the overall system performance. The two conventional ways to increase the off-chip communication bandwidth include increasing the signaling data rate at each channel and the Input/Output (I/O) pin number as predicted by International Technology Roadmap of Semiconductor (ITRS) [1] as shown in fig. 1.1.

Figure 1.1 ITRS predicted signaling speed and number of I/Os
However, due to manufacturing and technology limitations, the physical resources (pins and interconnect routing area) available for off-chip signaling remain almost constant. This implies that inter-chip communication data rates must increase dramatically to keep up with the advancements of the CMOS technology nodes with additional power consumption and area costs. Also high edge rates causes high crosstalk noise due to capacitive and inductive coupling between channels, which would further limit the pin counts and routing pitch. Therefore a new solution to solve the two problems at one time is needed which are high data transmission rates and high density interconnects. The multimode signaling is proved to be able to fully eliminate the crosstalk in high density uniform channel [4, 26] which could be our solution for this problem.

1.2 Contribution

The research aim of this work is to explore the implementation methodology and difficulties of applying multimode signaling to real life practical channel. Previous work of multimode signaling mainly focused on using this scheme in uniform channels and channels with small discontinuities [2, 9, 16, 29-32]. To make best use of crosstalk cancelling by multimode signaling in real world applications, practical channels including packages, sockets and PCBs need to be co-designed with the circuits in a systematic approach [7]. To solve this problem, the resultant contributions of the research are:

- S-parameter based multimode CODEC derivation method [6] is combined with transmitter only binary MMI circuit implementation [10] to complete a robust chip/channel co-design flow. This design flow involves model build/extraction,
CODEC generation, implementation in fully tunable circuit, and evaluation in eye diagram.

- The transmitter only binary MMI circuit is optimized for full tunability including CODEC signs, magnitude and timing adjustments. Also the circuit is improved for power efficiency at 5.6 mW/Gb/s in 130 nm CMOS technology which would scale with process technology nodes.

- Multimode signaling is applied to a real-life complexity benchmark channel with the new co-design methodology and circuit implementation. The results shows about 50% root mean square (RMS) jitter reduction over traditional single-ended signaling.

- A new channel consisting of PCB, package routings, package vertical components is designed from scratch, optimized to be working with multimode signaling. Comparing with the benchmark channel, the resulting optimized channel has a density improvement of 300% and 97% for the PCB and package routing respectively, and 300% for package vertical pitch (μvia, PTH, socket, etc) with signal to ground pin ratio 2:1. Although the optimized channel has much higher density, it still shows 46% jitter reduction over the benchmark channel with the same crosstalk mitigation circuit and method. This new approach significantly improves the ability to use high density multimode interconnect sub-systems in practical scenarios.

- A new CODEC implementation method is introduced, it only required inputs from adjacent signal lanes which in turn reduce the CODEC complexity and save power consumption. An eight trace PCB channel is used as an example to validate this
simple CODEC implementation method, preliminary results shows about 55% RMS jitter reduction is achieved comparing with traditional single ended signaling.

1.3 Organization

This dissertation is organized as follows. Chapter 3 describes an S-parameter based binary multimode signaling design methodology, including the mathematical derivation, CODEC generation method, and fully tunable circuit implementation. Also a real-life complexity benchmark channel is introduced to validate the CODEC derivation and circuit design.

In chapter 4, an optimized channel is designed from scratch consisting of PCB, package routing, and package vertical components to reach the goal of higher channel density and low jitter. The optimized channel and benchmark channel are compared and discussed in detail to analyze the signaling improvement, and generalize the channel design rules for multimode signaling.

Chapter 5 introduces a design approach for large bundle multimode implementation with low power consumption and area penalty. With the new approach we could further increase the channel density by employing larger bundle size.

Finally the paper concludes by summarizing the contributions and discussing future efforts in chapter 6.
Chapter 2

Related Works

2.1 Introduction

Due to increased signal speed, faster rise/fall edge and high density interconnect design, the crosstalk especially far end crosstalk (FEXT) has become a major problem for the signal integrity at receiver end. To illustrate the crosstalk noise, we could use a simple example as a two microstrip channel as shown in Fig. 2.1. The electric wave velocity is different between the even mode and odd mode caused by different electrical and magnetic fields between those two modes.

![Figure 2.1](image-url)  
(a) Even propagation mode  
(b) Odd propagation mode

Figure 2.1 (a) Even propagation mode (b) Odd propagation mode
The different velocities will cause different arrival times between different modes, as for two line system, it could be illustrated as in fig. 2.2. For two independent signals, the system will have all modes present which will cause uncertainty in signal arrival time called jitter. We refer this kind of jitter as crosstalk induced jitter (CIJ). Besides the timing impact, the crosstalk noise will also cause signal magnitude noise in adjacent channels, normally of the same polarity for next end crosstalk (NEXT) and opposite polarity for FEXT comparing with the aggressor signal. This thesis is concerned with crosstalk mitigation solutions for tightly coupled practical channels with reasonable discontinuities. Many crosstalk mitigation methods have been investigated in past literature, we will go through several of them in this chapter to reveal the necessarily of this research work.

Figure 2.2 Different arrival time between modes
2.2 Passive Crosstalk Cancellation Scheme

There have been several passive solutions proposed to mitigate the crosstalk noise. S.K. Lee has proposed a sub-alternated microstrip line structure as shown in fig. 2.3 [11]. The uniformly distributed stub structures reduce the FEXT and CIJ by increasing the mutual capacitance with little change in the mutual inductance, so as to reduce the velocity mismatch between odd and even mode.

![Figure 2.3 Stub-alternated microstrip line for memory interface taken from [11]](image)

K. Lee proposed a serpentine guard trace [14] between signal lines to increase the mutual capacitance and further improve the FEXT and CIJ. With the proposed structure as shown in fig. 2.4, jitter is reduced to 40% comparing with no guard.
J. Lee also proposed a similar approach to add capacitance directly between adjacent DQ lanes for DDR3 memory system [15] as shown in fig. 2.5. These passive solutions are hard to be applied in general tightly coupled channels, and the required resources (area, capacitors) are not always viable options in real systems.
2.3 Active Crosstalk Cancellation Scheme

Many active solutions are investigated to mitigate crosstalk noise under different circumstances or with different channels. John Wilson has proposed a crosstalk elimination technique [24] by injecting an equal size but opposite polarity pulse of aggressor signal onto the victim line as depicted in fig. 2.6. The AXE capacitance could be swept to find optimum value for different channels, and prove to be effective in reducing FEXT noise and CIJ to improve eye opening.

R. Harjani suggested a crosstalk cancellation and signal reutilization (XTCR) analog front-end implementation [19] [21] [23] as shown in fig. 2.7. This crosstalk mitigation scheme is implemented in 65 nm CMOS technology operating at 12 Gb/s, and prove to be effective in improving eye opening by 37.5 % and 26.4 % horizontal and vertical respectively.
Figure 2.7 Proposed FEXT channel and XTCR analog front-end block diagram taken from [23]
The XTCR shows very promising results eliminating crosstalk noise in a multi-lane channel of any size at a low power consumption of 0.96 pJ/bit/lane, but the implementation is difficult to be incorporated into current I/O circuit structure, and didn't include the impact of discontinuities of practical channel such as socket, vias, different routing sections, etc.

Another CDMA-like crosstalk cancellation scheme [25] is proposed by T. Hsueh. Fig. 2.8 shows the block diagram of the proposed scheme, \(D_1(t), D_2(t)\) and \(C(t)\) are encoded by \(S(t)\) and transmitted over four wires. Due to the orthogonal relationship between \(S(t)\) and \(C(t), D(t)\), the crosstalk between \(C(t)\) and \(D(t)\) is suppressed. But it requires complicated circuit implementation, and the scheme’s power consumption is about 28 mW/Gb/s for 90 nm CMOS technology.

![Figure 2.8 CDMA-like signaling technique block diagram taken from [25]](image)
There are several other FEXT cancellation schemes proposed to eliminate the noise caused by crosstalk and inject the opposite signals that will cancel the FEXT at transitions [17, 18]. K. Sham implemented the discrete-time (DT-FIR) crosstalk equalizers at the transmitter side [19], while Nazari proposed a continuous time (CTLE) [20] method to cancel out crosstalk at receiver side. The main issues of these techniques over tightly coupled buses are that it’s difficult to generate an accurate replica of the crosstalk signal. Another technique proposed is to shift the coupled FEXT away from the data transitions [21], and then cancel out this voltage noise with a glitch canceler [12], but this increase the design complexity and power consumption.

2.4 Crosstalk Cancellation with Modal Approach

Last two sub-chapters summarize a few techniques to cancel or mitigate crosstalk noise with passive or active circuit implementation. But as the coupling gets stronger with tighter routing pitch, the methods could become very complicated to implement or more power hungry.

An alternative approach called multimode signaling [2, 6, 7, 9, 10, 16] is based on multi-conductor transmission line theory (MTL) [5, 27]. It’s also referred as modal signaling investigated by several other researchers. Nguyen [22, 28] proposed to transmit signals through orthogonal modes on a multiple parallel lines system. Due to orthogonal property of the modes, crosstalk could be fully eliminated. Broyde and Clavelier [4, 26] shows very
promising behavioral simulation results for lossless MTL, the practical implementation issues for real life channel need to be further investigated.

Pavle Milosevic applied modal signaling to channel with small discontinuities [3, 28-31], the block diagram is shown in fig. 2.9. The encoder block maps data bits to be transmitted onto modal signals, and the decoder translate modal signals back to received data bits. To prevent reflections and mode conversion, a full resistive grid termination network [32] has been investigated for bundle of 6 as shown in fig. 2.10.

Figure 2.9 Modal signaling system block diagram taken from [30]
2.5 Summary

In this chapter, we discussed the sources and impact of crosstalk noise in high speed signaling environment. Several previously proposed crosstalk mitigation techniques are reviewed, summarized for the ideas, pros and cons of each. Both passive and active methods have been reviewed, and the research works of multimode or modal signaling are discussed and compared. We have noticed that although multimode signaling has been applied to non-uniform channels or channels with small discontinuities, it still haven’t been applied to real-life complexity channel. Also, to better take advantage of multimode, the channel needs to be co-designed with circuit in a systematic approach.
Chapter 3

S-Parameter Based Binary Multimode Design

Methodology and Circuit Implementation

3.1 Conventional Multimode Implementation

Multimode signaling is based on Multi-conductor Transmission Line (MTL) theory [5]. MTL refers to (n+1) parallel conductors consisting of n conductors and a reference conductor. The n conductors serve as signal path between transmitter and receiver, and the reference conductor is used as current return path. If we assume the n conductors have uniform cross section along the z axis and Transverse Electromagnetic (TEM) mode, where E and H fields are perpendicular to the direction of signal propagation. The voltages and currents on the signal lines are governed by two differential equations referred as telegrapher’s equations in time domain as shown in (3.1)

\[- \frac{dN}{dz} = ZI \]

\[- \frac{dI}{dz} = YV \]

(3.1)
The 2\textsuperscript{nd} order of this differential equation can be rewritten as

\[- \frac{d^2 V}{dz^2} = (ZY)V\]

\[- \frac{d^2 I}{dz^2} = (YZ)I\]  
(3.2)

A general method for canceling cross-coupled relations among links is to convert natural voltage or current $V$ or $I$ to modal voltage or current $V_m$ or $I_m$ by multiplying a converting matrix $T$ or $S$ for voltage mode or current mode, respectively (3.3).

\[V = TV_m\]

\[I = SI_m\]  
(3.3)

Here we could consider $V_m$ or $I_m$ as the data to be sent from the transmitter. By substitute (3.3) into (3.2), the resulting differential equations are
\[- \frac{d^2 V_m}{dz^2} = (T^{-1} ZYT)V_m\]

\[- \frac{d^2 I_m}{dz^2} = (S^{-1} YZS)I_m\]  \hspace{1cm} (3.4)

The equations above were implemented as circuits by adding encoding (\(T\) or \(S\)) and decoding (\(T^{-1}\) or \(S^{-1}\)) matrices at each end of the channel, outside of the transmission line bundle. \(TV_m\) and \(SI_m\) are the physically transmitted natural voltages and natural currents for binary type modal signals, \(V_m\) and \(I_m\), respectively.

The key to crosstalk cancellation is the selection of \(T\) and \(S\), so that \(T^{-1}ZYT\) and \(S^{-1}YZS\) could be formed as diagonal matrices, then there are no off-diagonal terms and thus no crosstalk. Due to the relationship of \(S^t = T^{-1}\), it suffices to diagonalize either \(T^{-1}ZYT\) and \(S^{-1}YZS\). Consider diagonalize \(T^{-1}ZYT\) in (3.5),

\[T^{-1}ZYT = \gamma^2\]  \hspace{1cm} (3.5)

Where \(\gamma^2\) is a \(n \times n\) diagonal matrix
The equation (3.5) could be rewritten as $\mathbf{ZYT} = \gamma \mathbf{T}$, in this form the $\mathbf{T}$ is the eigenvector of $\mathbf{ZY}$, and $\gamma^2$ would be the eigenvalue. This multimode signaling scheme could be implemented in circuit as shown in Fig. 3.1.

There are several limitations for the conventional MMI. First of all, it requires the channel to be uniform which is not the case for most of practical channels. Secondly, the implementation would require an encoder circuitry at transmitter and a decoder circuitry at the receiver which makes it difficult to change the CODEC during run time or in simulations. To conquer these limitations and develop a design flow to optimize the channel design with
multimode signaling, instead of using RLGC of a uniform channel to derive the CODEC, an S-parameter based binary multimode signaling scheme is employed.

### 3.2 S-Parameter Based Systematic Design Approach

For an arbitrary N channel network (2N ports), the relationship between incident voltages and reflected voltages of each port could be depicted with S-parameter of the network [8], as in (3.7):

\[
\begin{bmatrix}
V_1^- \\
V_2^- \\
\vdots \\
V_{2N}^-
\end{bmatrix} =
\begin{bmatrix}
S_{11} & S_{12} & \cdots & S_{12N} \\
S_{21} & S_{22} & \cdots & S_{22N} \\
\vdots & \vdots & \ddots & \vdots \\
S_{2N1} & S_{2N2} & \cdots & S_{2N2N}
\end{bmatrix}
\begin{bmatrix}
V_1^+ \\
V_2^+ \\
\vdots \\
V_{2N}^+
\end{bmatrix}
\]

(3.7)

Here \(V_j^+\) is the voltage of the incident wave at port \(j\) and \(V_i^-\) is the voltage of the reflected wave at port \(i\), and \(S_{ij}\) is the ratio of reflected wave voltage \(V_i^-\) and incident wave voltage \(V_j^+\), with all ports other than port \(j\) terminated with matched loads as shown in (3.8).
With the assumption of perfect termination and no reflections, a direct relationship between transmitted voltages and received voltages can be established with a reduced S-parameter matrix as shown in (3.9), and we abbreviated the expression as (3.10).

\[
S_{ij} = \frac{V^-_i}{V^+_j}, \quad V^+_k = 0 \text{ for } k \neq j
\]  

(3.8)

\[
\begin{bmatrix}
V^-_{N+1} \\
V^-_{N+2} \\
\vdots \\
V^-_{2N}
\end{bmatrix} =
\begin{bmatrix}
S_{(N+1)1} & S_{(N+1)2} & \cdots & S_{(N+1)N} \\
S_{(N+2)1} & S_{(N+2)2} & \cdots & S_{(N+2)N} \\
\vdots & \vdots & \ddots & \vdots \\
S_{2N1} & S_{2N2} & \cdots & S_{2NN}
\end{bmatrix}
\begin{bmatrix}
V^+_1 \\
V^+_2 \\
\vdots \\
V^+_N
\end{bmatrix}
\]  

(3.9)

\[
V_{\text{out}} = S \cdot V_{\text{in}}
\]  

(3.10)
In this reduced $S$-parameter matrix $S$, the magnitude of diagonal entries depict the insertion loss of each line and the off-diagonal entries represent the far end crosstalk noise (FEXT) between lines. So we could have a crosstalk free channel if $S$ matrix is diagonalized. Similar to what happened in conventional MMI, the diagonalization could be implemented as (3.11), in which matrix is the eigenvector matrix of $S$ matrix. Then the modified transfer function matrix $T^{-1}ST$ would be diagonal, indicating FEXT have been cancelled out.

$$V_{out} = T^{-1}ST \ V_{in}$$

(3.11)

Since the entries in the $S$-parameter matrix $S$ of a practical channel are complex numbers, to fully diagonalize $S$, the CODEC matrixes $T$ and $T^{-1}$ have to be complex as well. Phases of entries in matrix $T$ and $T^{-1}$ represent input voltage controlled phase shift. However, it is difficult to implement this input voltage controlled phase shift for each coding entry in the transceiver circuits. Thus, in this study we derived the CODEC using the absolute values of the entries of $S$ with the assumption that this will still provide satisfactory performance.

Also, $S$-parameter matrix is frequency dependent, each frequency point corresponds to an eigenvector matrix. To find out the optimal setting, we need to determine the CODEC matrix that gives the highest overall Signal to Noise Ratio (SNR) performance. A figure of merit (FOM) in introduced to represent the overall SNR as shown in (3.12), knee frequency ($f_{knee}$) is the highest frequency content within a particular digital signal which relates to the rise/fall time (3.13). The eigenvector matrix of $S$-parameter matrix at each frequency will be examine
with (3.12), whichever gives the highest FOM will provide best overall SNR and will be chosen as CODEC for multimode signaling.

\[
FOM = \sum_{freq=0}^{f_{knee}} \min \left\{ \frac{|(T^{-1}ST)_{\text{diagonal}}|}{| (T^{-1}ST)_{\text{off-diagonal}} |} \right\}
\]

(3.12)

\[
f_{knee} \approx \frac{0.35}{t_{10-90\%}}
\]

(3.13)

### 3.3 Transmitter only Binary MMI

#### 3.3.1 Matrix Transform and Derivation

To show how the S-parameters can be used in multimode signaling design in a detailed example, an eight-port channel is studied. The channel is composed of four appropriately referenced signal lines as shown in Fig. 3.2. The signal lines are characterized by general models; they can include transmission lines on a package and a PCB, and discontinuities such as vias, sockets, etc. With proper terminations, we can apply (3.9) to this channel model by setting \( N = 4 \) as shown in (3.14).
As discussed in chapter 3.2, with proper selection of CODEC matrix $T$ we could diagonalize the reduced S parameter matrix which eliminate the FEXT and improve the overall SNR in form of $T^{-1}ST$. Assuming the CODEC can fully diagonalize the S-parameter matrix, we have

$$
\begin{bmatrix}
V_5^- \\
V_6^- \\
V_7^- \\
V_8^-
\end{bmatrix} =
\begin{bmatrix}
S_{51} & S_{52} & S_{53} & S_{54} \\
S_{61} & S_{62} & S_{63} & S_{64} \\
S_{71} & S_{72} & S_{73} & S_{74} \\
S_{81} & S_{82} & S_{83} & S_{84}
\end{bmatrix}
\begin{bmatrix}
V_1^+ \\
V_2^+ \\
V_3^+ \\
V_4^+
\end{bmatrix}
\tag{3.14}
$$
Where $\gamma^2$ is a diagonal matrix. Now, let’s consider a coding scheme at the transmitter side that implements $T(\gamma^2)^{-1}T^{-1}$. Then the signal transfer function becomes

$$T^{-1}ST = \begin{bmatrix} \gamma_1^2 & 0 & 0 & 0 \\ 0 & \gamma_2^2 & 0 & 0 \\ 0 & 0 & \gamma_3^2 & 0 \\ 0 & 0 & 0 & \gamma_4^2 \end{bmatrix} = \gamma^2$$

(3.15)

Substituting (3.15) into (3.16) we get

$$V_{out} = S \cdot T(\gamma^2)^{-1}T^{-1}V_{in}$$

(3.16)

Substituting (3.15) into (3.16) we get

$$V_{out} = S \cdot T(T^{-1}ST)^{-1}T^{-1}V_{in}$$

$$= S \cdot TT^{-1}S^{-1}TT^{-1}V_{in} = S \cdot S^{-1}V_{in} = V_{in}$$

(3.17)

Thus, by employing $T(\gamma^2)^{-1}T^{-1}$ at the transmitter side, the input signal can be recovered at the receiver side without any decoding. Let $(\gamma^2)^{-1} = \delta$ the entries of the diagonal matrix represent the delay of each mode which can be mapped to a delay on each line by multiplying with $T^{-1}$. Let $T^{-1} = U$ and expand the coding scheme $T\delta U$ as
By fully expanding the right-hand side of (3.18) and moving the delay factor $\delta$ to $V_{in}$, we obtain

$$\begin{align*}
\begin{bmatrix}
\text{Driver 1} \\
\text{Driver 2} \\
\text{Driver 3} \\
\text{Driver 4}
\end{bmatrix}
&=
\begin{bmatrix}
T_{11} & T_{12} & T_{13} & T_{14} \\
T_{21} & T_{22} & T_{23} & T_{24} \\
T_{31} & T_{32} & T_{33} & T_{34} \\
T_{41} & T_{42} & T_{43} & T_{44}
\end{bmatrix}
\begin{bmatrix}
\delta_1 & 0 & 0 & 0 \\
0 & \delta_2 & 0 & 0 \\
0 & 0 & \delta_3 & 0 \\
0 & 0 & 0 & \delta_4
\end{bmatrix}
\begin{bmatrix}
U_{11} & U_{12} & U_{13} & U_{14} \\
U_{21} & U_{22} & U_{23} & U_{24} \\
U_{31} & U_{32} & U_{33} & U_{34} \\
U_{41} & U_{42} & U_{43} & U_{44}
\end{bmatrix}
\begin{bmatrix}
V_{in1} \\
V_{in2} \\
V_{in3} \\
V_{in4}
\end{bmatrix}
\end{align*}$$

(3.18)

Each row of (3.19) is a coded driving voltage and there is a delay time factor after each signal voltage. This can be implemented as delay blocks followed by CODEC drivers. In this case, we no longer need any decoding circuit at the receiver end, it would be binary signal when it reaches the receiver.
3.3.2 Circuit Implementation

To implement the right hand transformation of (3.19) in transmitter circuit, we have employed circuit structure as shown in Fig. 3.3.

Figure 3.3 Block diagram of transmitter only MMI
The phase adjustment blocks generate 4 different phases of each binary signal; these 16 bit signals along with their inverse signals are sent into each CODEC driver. For the delay control block, we used a voltage controlled delay line (VCDL) to control the phase delay as shown in fig. 3.4. By lowering the control voltage, the signal rise/fall edge will be slower and therefore the delay will be longer. So by setting the DC control voltage, the phase delay could be adjusted as needed. In the driver, we use a DAC style line driver as shown in Fig. 3.5. The NMOS transistor gates at the bottom are connected to control bias signals, and the transistors at the top are connected to pre-delayed signals. By setting control bias signals we can choose which signals are in the summation equation for programmability. Because the transistor’s current is proportional to its width, the magnitude of the CODEC can be controlled by weighting the width of the transistors. The sign of the CODEC is controlled by choosing whether to turn on the path of the signal or inverted signal. Thus, both magnitude and sign of the CODEC can be fully programmed and easily implemented.

Figure 3.4 Voltage control delay line schematic
First, let’s examine the VCDL circuit, by lowering the control voltage from supply voltage 1.2 V to 0.6 V which is the threshold of the buffer, we could have phase delay from 70 ps to 234 ps which gives us the tuning range about 164 ps. This range could be extended by adding more buffer stages under control voltage with sacrifice of more area and power consumption. This control voltage could be supplied by either direct DC voltage supply or digital controlled DAC. The DC voltage supply has the advantage of simple implementation, but depending on the packaging option (C4 or wire-bond), the large inductance of the supply path and insufficient on-chip decoupling capacitance could cause large noise on the control voltage, and further introduce jitter to the transmitting signals. The DAC implementation is less impacted by the packaging choice, but it requires more power consumption, and chip area depending upon how much control precision is desired, for example a 4 bit DAC would give 16 tuning points while an 8 bit DAC would have 256 tuning points, but 8 bit DAC would consume twice the power and area than a 4 bit DAC.
Another delay control circuit has been explored as well, it employs a digital controlled vernier circuitry as shown in fig. 3.6. The signal propagation path will be controlled by digital control signals SW0-SWn in n individual control cells. Only one control is on at one time to make sure there is no race condition, the resolution is controlled by the phase difference between the two outputs in one cell. In one delay cell as shown in (b), the phase difference could be controlled by sizing the load capacitors C1 and C2. The larger difference between C1 and C2 values, the larger phase difference between the phases with same sized.
buffer before the load capacitors. In simulation, we could get fine resolution as 3 ps by sizing C1 as 0.1 fF and C2 as 1 fF, and also could get larger step resolution as 100 ps by sizing C1 as 0.1 fF and C2 as 10 fF, or even higher with larger capacitor values. To have larger tuning range, we would need more delay cells and more control bits from input scan chain meaning more area and power consumption. Due to it is pure digital circuit, the power consumption of this implementation would be less than the DAC method.

3.4 Benchmark Channel Application

To test the S-parameter based multimode signaling and binary multimode circuit design, we crafted a benchmark problem that includes all the components of a practical real-life complexity I/O channel. The architecture of this channel is depicted in Fig. 3.7. It is a symmetrical channel comprised of two central processing unit (CPU) dies flip-chip assembled in two packages that reside on the opposite ends of the channel. Both packages are mounted on a land-grid-array (LGA) socket. Each socket is assembled to a PCB through ball-grid-array (BGA) connections. As shown in Fig. 3.7, the signaling path for the I/O communication starts with the horizontal routing on one of the packages, continues with the vertical transition through the package vias and the socket and the horizontal routing on the board before repeating a similar set of transitions on the second package. In this case, the routing on the PCB is assumed to be microstrip routing; hence there are no PCB vias in the path of the signals.
The stack-up technology design rules (DRs) for the package and the PCB (dielectric and conductor thicknesses for each layer, via dimensions, etc.) and the material set are selected to be within the envelope of today’s high volume manufacturing capability. Similarly, a readily available LGA socket technology is used in this design.

Next, we concentrate on the physical and electrical design of the multimode bundles. For this, we select bundle size $N$ to be 4 which is consistent with the formulation presented in chapter 3.3. The width of each package and PCB routing trace is selected to match the channel characteristic impedance target. Exceptions to this are the break-out sections on the package and the PCB where narrower trace widths are used to escape signals out of the die and the package shadow, respectively. The pitch for each routing section on both the package and the PCB is selected such that the density of typical CPU packages can be achieved. A similar strategy is also applied in the design of the bump-out pattern between the CPU die and the package, and the pin-out pattern between the package and the PCB.
The electrical models for the different components are constructed separately for sake of flexibility. This allows investigation of an S-parameter based multimode signaling design not only for the complete channel but also for various sub-channel configurations (PCB only and so on). The routing length of on PCB is 10.8 cm and on package is 2.5 cm.

We applied the S-parameter based multimode signaling design method to the benchmark problem for two distinct scenarios: PCB only and PCB with packages and sockets. The results are evaluated using two metrics, suppression of off-diagonal entries in the S-parameter matrix and signal jitter improvement in the time domain.

To apply the crosstalk mitigation method, first we need to extract the reduced S-parameter that only contains the insertion loss and FEXT information from the original matrices as shown in (3.14). Then, to find the optimal CODEC matrix, eigenvector matrix is derived from the absolute value of the reduced S-parameter matrix for every frequency point. Each eigenvector matrix is examined with SNR figure of merit (3.12), fig. 3.8 shows the resulting plot of the FOM versus the frequency at which the eigenvector matrix is derived for the benchmark full channel. For the benchmark problem, the eigenvector matrix derived at frequency around 9.6 GHz gives the highest FOM value, will be used as the CODEC matrix for the full channel. The same method is done for the PCB only channel.
We plot the original S-parameter matrix and the modified S-parameter matrix in dB to evaluate the effectiveness of the CODEC. To make the plots concise, only results for ports 5 and 6 are shown to illustrate the crosstalk suppression. The figure of merit is the difference in magnitude between diagonal entries and off-diagonal entries in dB. This corresponds to signal-to-noise ratio (SNR) in the system. The frequency of interest is chosen to be 4 GHz.

Fig. 3.9 shows that the off-diagonal entries of the S-parameter matrix for the PCB only channel are suppressed by 29 dB compared to the original S-parameter entries at 4 GHz; and suppressed by 25 dB at 10 GHz. The CODEC generated by the S-parameter method performs well in this case where only few discontinuities are present in the channel. Ideally, with the assumptions that a channel has uniform cross-section and transverse electromagnetic mode (TEM) wave propagation, the multimode signaling scheme can eliminate crosstalk noise.
entirely. Though the PCB has break-out and break-in sections, most of the main routing has a uniform cross-section. Thus, this scheme works well improving the SNR for the PCB only channel.

Figure 3.9 S-parameters for PCB only: (a) Original S-parameters and (b) modified S-parameters.
Fig. 3.10 shows the S-parameters for the complete channel including the PCB, the two packages and the two sockets. The SNR was improved by 9 dB at 4 GHz and 5 dB at 10 GHz. This improvement, even though less than for the PCB only case due to the additional discontinuities in the channel, still provides an appreciable increase in the SNR of the complete channel.

Figure 3.10 S-parameters for complete channel: (a) Original S-parameters and (b) modified S-parameters.
Next, we will examine the multimode signaling improvement in time domain by implementing the derived CODEC to the transmitter circuit as shown in fig. 3.3. The transmitter circuit is simulated at 4 GT/s with 50 Ω termination resistors at both ends of the channel and each line has parasitic capacitive loadings of 0.5 pF at both ends. Comparing with traditional single ended current mode driver as in Fig. 3.11, the RMS jitter of the benchmark channel is improved by 52% from 90 ps to 43 ps with direct application of multimode signaling as shown in fig. 3.12. This CODEC generation method and circuit implementation for multimode signaling is proved to be effective in reducing the RMS jitter and improve signal quality. The power consumption of the multimode transmitter is around 5.6 mW/Gb/s, comparing with the traditional single ended implementation which is 2.2 mW/Gb/s, the add-on power consumption is 3.3 mW/Gb/s which is using 130 nm technology. If we scale to 21 nm technology node [34], the add-on power consumption would be 0.56 mW/Gb/s.

Figure 3.11 Traditional single ended current mode driver
Figure 3.12. (a) Benchmark channel with traditional single ended current mode driver (b) Benchmark channel with transmitter only multimode signaling
3.5 Chip Test Setup, Debug and Discussion

To validate the binary multimode circuit, we have tape-out a test chip employing the tunable binary multimode transceiver. All the CODEC circuit is implemented in transmitter side, only termination circuits are implemented at receiver side. For better tunability, we used four input clocks with tunable phases to sample the input signals, as well as implemented the VCDLs controlled by four DC supplies CVDD1-4. The layout floor plan is shown below.

Figure 3.13 Binary multimode transceiver layout
Fig. 3.14 (a) shows the fabricated test board with test chips wire-bonded on it. The test needs 4 pseudo random data signals, and 4 clock signals with different phases. But we only got 1 pulse generator HP8133, so we use a Bit Error Rate Tester (BERT) to generate additional signal and clock. We need four data signals aligned and four clock signals with tunable phases. Below is how we set it up:

1). HP8133 trigger -> BERT trigger -> Sampling scope TDS 8008

2). Align all data signals from 8133 and BERT to the same phase

3). Align BERT clk_bar to good phase (from simulation as start point) with data phase.

4). Connect BERT clk to a vernier caliper module as shown in fig. 3.14 (b) and align to BERT clk_bar with good phase (if not enough, add a small length wire to adjust the phase)

5). Adjust HP 8133 clk into good phase with BERT clk_bar by adjusting the delays on the equipment

6) Connect 8133 clk_bar to a vernier caliper module to align the phase by either tuning the module or add a small wire to the routing if needed

With above steps, we could gain full control of the phases of the four clocks.
Figure 3.14 (a) Fabricated test board with test chips wire bonded (b) Vernier caliper to tune the phases
Two DC power supply equipment are used to provide supply voltage for the chip as well as four control DC voltage and supply for pre-driver stage. Figure 3.15 shows the probe station setup.
With multimode CODEC scanned in using digital pattern generator HFS9009, we expect to see the crosstalk cancelled binary signals at receiver, but instead only clock noise fed through the supply shows as shown in Fig. 3.16. To debug this problem, first we slow down the signal from 3 GHz to 400 MHz as shown in fig. 3.17, but still only clock signals, not data signals seen. CODEC have been changed but only affect the DC level as more ‘1’’s in the CODEC, lower the DC level since more branches are active pulling. All these signs shows a stuck at ‘1’ error, since it’s behaving all data signals are stuck to high.

![Image of oscilloscope showing output signal at 3 GHz](image)

Figure 3.16 Output pin signal at 3 GHz
To find out which stage the stuck at 1 error comes from, we tried to do a voltage contrast test with help of Analytical Instrument Facility in NCSU, but due to the technology used is 8RF 130nm with 8 layers of metal stack. The metal layers are too thick for the scanning microscopy to work. After tried everything we could, still unable to find out the cause. One important lesson learned here is always keep intermediate stages available for probe to debug, even if the pad area is too small to probe, at least bring the signals to the surface metal where we could use voltage contrast method to read the voltage and debug any stuck at 1 or 0 errors.
3.6 Summary

In this chapter, we have discussed the S-parameter based binary multimode signaling from mathematical derivation, circuit implementation, and application to a real-life benchmark channel example. The method is effective in mitigating crosstalk noise even for channels not optimized for MMI with lots of discontinuities, shows RMS jitter reduction from 90 ps to 43 ps, a 53% improvement. The benchmark channel is designed without multimode in mind, so not optimized for it, in next chapter we will design a channel from scratch including PCB routing, package routing and socket vertical components with the same material and basic design rules as benchmark channel to explore the MMI optimized channel design method.
Chapter 4

Chip/Channel Co-Design Methodology and Implementation

4.1 Introduction

There are three major steps in the proposed design methodology as shown in Fig. 4.1. First, parameterized models of PCB or package are built in 2D or 3D electromagnetic (EM) simulators. In order to achieve the maximum density, the baseline channel is constructed at highest density within manufacture design rules. Then, S-parameters are extracted from each model, cascaded to form the model of full channel which is used to generate the CODEC for multimode signaling with a MATLAB routine. Once the CODEC is generated, it will be implemented in a fully programmable multimode transceiver we designed to evaluate the channel and the CODEC. Based on the resulting eye diagram, we determine whether larger spacing, wider trace or any other modifications are needed for the channel design. By iterating through the above steps, we can maximize the channel density while having control over signal quality.
4.2 MMI Optimized Channel PCB Design

In this work, we consider an embedded microstrip bundle consisting of four lines as shown in Fig. 4.2 for both board routing and package routing with different size and spacing. To have a solid comparison, the basic parameters of channel model are the same as of the benchmark channel which is defined by manufacture. For the PCB design, the thickness of solder mask, dielectric substrate, signal trace and reference plane are 10 µm, 70 µm, 50 µm, and 50 µm, respectively. The minimum signal trace width and spacing are both 100 µm. For package design, the thickness of solder mask, dielectric substrate, signal trace and reference plane are 20 µm, 25 µm, 15 µm, and 15 µm, respectively. The minimum signal trace width and spacing are both 15 µm. To maximize the routing density, the channel pitch of our baseline model is chosen to be the minimum pitch defined by manufacture as described above. Thus, our baseline model has signal trace width and spacing of 100 µm for the PCB
and 15 µm for the package routing. The basic design parameters are presented in Table. 4.1, the trace width and space will be adjusted to find the optimum values, while other parameters will stay the same.

![Cross-section of routing traces](image)

**Figure 4.2 Cross-section of routing traces**

<table>
<thead>
<tr>
<th>Table 4.1 Design Parameters For PCB and Package Routing Traces</th>
</tr>
</thead>
<tbody>
<tr>
<td>**</td>
</tr>
<tr>
<td>Minimum Trace Width</td>
</tr>
<tr>
<td>Minimum Trace Space</td>
</tr>
<tr>
<td>Signal Trace Thickness</td>
</tr>
<tr>
<td>Dielectric Thickness</td>
</tr>
<tr>
<td>Solder Mask Thickness</td>
</tr>
<tr>
<td>Reference Thickness</td>
</tr>
</tbody>
</table>

We begin the channel design with the PCB only baseline channel with minimum signal trace width and spacing of 100 µm. We target a signaling rate of 4 GT/s and a wiring length of 4.25 inches. To examine the coupling between the signal lines, we extract the RLGC matrix of the 2-D model as shown in Table. 4.2. We can see that the mutual capacitances and
inductances are comparable to the self-capacitances and self-inductances, which is expected since the channel is highly coupled.

Table 4.2. RLGC model of PCB only channel.

<table>
<thead>
<tr>
<th>Lo</th>
<th>(H/m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3e-07</td>
<td></td>
</tr>
<tr>
<td>6.8e-08</td>
<td>2.9e-07</td>
</tr>
<tr>
<td>2.4e-08</td>
<td>6.5e-08</td>
</tr>
<tr>
<td>1.4e-08</td>
<td>2.4e-08</td>
</tr>
<tr>
<td>2.9e-07</td>
<td></td>
</tr>
<tr>
<td>6.8e-08</td>
<td></td>
</tr>
<tr>
<td>3e-07</td>
<td></td>
</tr>
<tr>
<td>Co</td>
<td>(F/m)</td>
</tr>
<tr>
<td>10.8e-11</td>
<td></td>
</tr>
<tr>
<td>-1.4e-11</td>
<td></td>
</tr>
<tr>
<td>-1.1e-12</td>
<td></td>
</tr>
<tr>
<td>-0.8e-12</td>
<td></td>
</tr>
<tr>
<td>11.3e-11</td>
<td></td>
</tr>
<tr>
<td>11.3e-11</td>
<td></td>
</tr>
<tr>
<td>10.8e-11</td>
<td></td>
</tr>
<tr>
<td>Ro</td>
<td>(Ω/m)</td>
</tr>
<tr>
<td>118</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
</tr>
<tr>
<td>117</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
</tr>
<tr>
<td>118</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
</tr>
<tr>
<td>118</td>
<td></td>
</tr>
<tr>
<td>Go</td>
<td>(S/m)</td>
</tr>
<tr>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>-0.003</td>
<td></td>
</tr>
<tr>
<td>-1.9e-6</td>
<td></td>
</tr>
<tr>
<td>-0.003</td>
<td></td>
</tr>
<tr>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>-5e-9</td>
<td></td>
</tr>
<tr>
<td>-1.8e-6</td>
<td></td>
</tr>
<tr>
<td>-0.003</td>
<td></td>
</tr>
<tr>
<td>0.05</td>
<td></td>
</tr>
</tbody>
</table>

To further evaluate the coupling effect of this baseline channel, we examine the eye diagram with traditional single ended drivers without any coding in Fig. 4.3. As comparison, we would like to see the impact of inter-symbol interference (ISI) on the channel by exciting only one signal trace instead of four. Signals are transmitted at 4 GT/s. Comparing Fig. 4.3 and 4.4, we can see that with strong coupling, the jitter has increased from 23 ps to 70 ps due to crosstalk-induced jitter (CIJ). Thus, our goal here is to minimize the CIJ as much as possible with multimode signaling.
Figure 4.3 Eye diagram for PCB channel without CODEC.

Figure 4.4 Crosstalk free eye diagram of PCB channel.
After time domain analysis of the baseline channel, we extract the S-parameter file with the ANSYS Q3D Extractor 2-D field solver to generate CODEC matrices $T$ and $T^{-1}$ for multimode signaling as shown in Table 4.3. The numbers in the CODEC matrices given are rounded approximations for implementation. To evaluate the frequency domain performance of the baseline channel and the effectiveness of the CODEC derived, we plot the original reduced S-parameter matrix $S$ and the modified reduced S-parameter matrix $T^{-1}ST$ in dB. To make the plot concise, only two traces out of four are plotted here. The figure of merit is magnitude difference between diagonal entries and off-diagonal entries in dB representing overall SNR. The good performance in crosstalk noise suppression is expected since we use a uniform channel without any discontinuities. Fig. 4.5 shows that the modified S-parameter entries have 16 dB crosstalk noise suppression compared to the original S-parameter entries at 4 GHz, and 17 dB at 10 GHz. This confirms that the generated CODEC provides a significant SNR improvement in the PCB only channel. This CODEC is implemented in the multimode transceiver to evaluate the jitter reduction performance in the next step.

Table 4.3. Rounded CODEC matrices for PCB only channel.

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>T^-1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.6</td>
<td>0.6</td>
</tr>
<tr>
<td></td>
<td>-0.5</td>
<td>0.3</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>-0.4</td>
</tr>
<tr>
<td></td>
<td>-0.3</td>
<td>-0.5</td>
</tr>
<tr>
<td></td>
<td>-0.5</td>
<td>-0.6</td>
</tr>
<tr>
<td></td>
<td>-0.5</td>
<td>-0.6</td>
</tr>
<tr>
<td></td>
<td>0.6</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>-0.3</td>
<td>0.6</td>
</tr>
<tr>
<td></td>
<td>-0.6</td>
<td>-0.3</td>
</tr>
<tr>
<td></td>
<td>-0.5</td>
<td>-0.5</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>-0.4</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>-0.3</td>
<td>0.6</td>
</tr>
<tr>
<td></td>
<td>-0.6</td>
<td>0.3</td>
</tr>
</tbody>
</table>
Figure 4.5 S-parameters for PCB only channel: (a) Original S-parameters and (b) modified S-parameters.
By setting the control signal sequence, we implement the CODEC in Table 4.3 in the multimode transmitter circuit. With the particular trace width selection, the characteristic impedance of the traces is around 50 Ω. We set the termination resistance to be 50 Ω at both ends of the channels, and assume 0.5 pF parasitic capacitive loading at both the transmitter and receiver for the simulation. By tuning the timing adjustments $\delta_1$–$\delta_4$ to compensate the delay of each mode, we minimize jitter at the receiver side. Fig. 4.6 shows that while the eye height is about the same as for traditional single ended drivers, the RMS jitter is reduced from 69 ps as shown in fig. 4.3 to 26 ps, a 62% improvement. We can also see that this jitter value is very close to the ISI jitter value of 23 ps in fig. 4.4 which does not have any crosstalk noise. Thus, we conclude that the minimum trace width and spacing might be used in a practical PCB channel with multimode signaling.

![Eye diagram for PCB channel with multimode signaling.](image)

Figure 4.6 Eye diagram for PCB channel with multimode signaling.
4.3 MMI Optimized Package Routing Trace Design

Next, we will focus on the package routing trace design for the full channel, and try to increase the routing density as much as possible with multimode signaling. We assume microstrip for package routing similar to the PCB board as mentioned above, the baseline design rules are listed in Table 4.1. For the high routing density, we construct the baseline package with minimum trace width and spacing of 15 µm. Instead of examining the package itself, we consider a pseudo full channel scenario which is package routing plus PCB board channel plus another package routing. The length of package routing trace is set to be 2.5 cm which is consistent with benchmark channel, and the data rate is again 4 GT/s. Then, the S-parameters of the two packages are extracted and cascaded with the PCB channel for evaluation and generating the CODEC of the full channel.

With the minimum spacing and width design rules, the eye diagram for the traditional single ended signaling is shown in Fig. 4.7. The RMS jitter is 68 ps, roughly the same as for the PCB only case, but the eye height is considerably smaller at around 80 mV. Then, the same procedure as for the PCB only channel is repeated to examine this pseudo full channel scenario. First, the S-parameters of the package model are extracted and cascaded the PCB channel model, then a CODEC is generated from the S-parameters of the cascaded full channel model. After that, by implementing the generated CODEC in the transceiver, the performance of multimode signaling with the full channel is examined. Fig. 4.8 shows with multimode signaling, the RMS jitter is reduced to 29 ps from 68 ps, a 57% reduction compared with single-ended signaling, and eye height is 62 mV.
We can see large floor noise in the simulations which causes the small eye height. Since we used the minimum design rules, it is speculated that there may be a characteristic impedance mismatch between the package and PCB which causes this problem. It turns out
that the characteristic impedance of the package signal line is 67 Ω which is significantly higher than that of the PCB channel at 50 Ω. By increasing the signal trace width to 34 µm and keeping the spacing the same at 15 µm, we match the characteristic impedance to 50 Ω. By going through the same procedure (S-parameter extraction, CODEC generation, and implementation in transceiver circuit) we generate the eye diagram to examine the new channel. Fig. 4.9 shows the matched channel with traditional single ended signaling. The floor noise is reduced with the matched impedance and the eye height is increased to 117 mV. The matched channel also improves multimode signaling as shown in Fig. 4.10, the RMS jitter is 26 ps compared with 29 ps for the minimum width case, and the eye height is 76 mV compared with 62 mV.

Figure 4.9 Full matched channel without multimode signaling.
To further optimize this channel, we tried to increase the spacing while keeping the characteristic impedance constant at 50 Ω. By doing a sweep on the spacing, it is found that as the spacing increases, the eye height gets larger but the RMS jitter stays more or less the same. This trend holds until the spacing reaches 25 µm, beyond which the eye height no longer improves significantly. For 25 µm spacing, the width is 36 µm to keep the characteristic impedance at 50 Ω. With this setting, the eye diagram for the multimode signaling scheme is shown in Fig. 4.11. The eye height is 92 mV, increased by 48% compared with the initial minimum spacing setting that gives 62 mV. The RMS jitter is 26.7 ps, about the same as for the impedance match case with minimum spacing, which gives us a 63% reduction compared with the 73 ps with single-ended signaling in Fig. 4.9.

To evaluate the overall performance of the optimized channel with multimode signaling, we compare it with the crosstalk free case, where only one line has active signals as shown in Figure 4.10.
Fig. 4.12. The RMS jitter is 17% larger and the eye height is 13% lower compared with the absolutely crosstalk free scenario.

Figure 4.11 Full channel with optimum package for multimode signaling.

Figure 4.12 Eye diagram for crosstalk free full channel.
4.4 MMI Optimized Channel Package Vertical Design

To complete the full channel design, vertical components on package need to be designed. Fig. 4.13 shows the stack up of the proposed package vertical, it consists of four major parts: top build-up layers, core layers connected with Plated Through Holes (PTH), bottom build-up layers and socket to mount on PCB. There are five build-up layers both top and bottom, which are connected with micro vias (µvia). For sake of simplicity, we assume µvias, PTHs, and socket pins are all stacked directly on top of each other. We start with signal to ground pins ratio as 2:1 with inline pattern with top view shown in Fig. 4.14. The detailed design parameters are presented in Table. 4.4.

Figure 4.13 Cross-section view of package vertical components.
Table 4.4 Design Parameters For Package Vertical Components

<table>
<thead>
<tr>
<th>Design Parameters</th>
<th>Values (µm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core thickness</td>
<td>700</td>
</tr>
<tr>
<td>Core metal thickness</td>
<td>35</td>
</tr>
<tr>
<td>PTH/Pad diameter</td>
<td>250/400</td>
</tr>
<tr>
<td>Build-up metal thickness</td>
<td>15</td>
</tr>
<tr>
<td>Build-up dielectric thickness</td>
<td>30</td>
</tr>
<tr>
<td>Mvia/Pad diameter</td>
<td>100/150</td>
</tr>
<tr>
<td>Socket/Pad diameter</td>
<td>200/800</td>
</tr>
<tr>
<td>Socket height</td>
<td>2500</td>
</tr>
<tr>
<td>Socket pin pitch</td>
<td>1000</td>
</tr>
</tbody>
</table>

Based on the above design parameters, the pitch for signal and ground pins is set to be 1mm. First we set the void around the socket pads, µvia pads and PTH pads to be 60 µm, 25 µm and 100 µm respectively to separate the signal pins from the ground plane. To examine the impedance profile of the verticals, a Time-domain reflectometer (TDR) simulation is set.
up with rise time 50 ps. From the TDR result in Fig. 4.15. (a), we could see a big drop of the impedance from standard 50 Ω which indicates large capacitive discontinuity. In order to reduce to the reflections due to the impedance mismatch, this discontinuity in the package needs to be identified and fixed in the design. By carefully examining the design, we could see that with current void around the pads, there are capacitive coupling between the build-up layers and PTH pads, as well as the socket pads as shown in Fig. 4.16. To remove this capacitive coupling, the void space around the socket pads, μvia pads and PTH pads are adjusted to 60 μm, 385 μm and 260 μm, so that the cutout area on each build-up layers are the same as in Fig. 4.17. The impedance profile of the adjusted model is shown in Fig. 4.15. (b), the capacitive discontinuities have been removed, and replaced by higher impedance profile. We could try to add partial capacitive coupling back in to smooth out the impedance profile and bring it closer to 50 Ω, but the implementation would be difficult to realize when dealing with more complicated designs and the cancellation also depends on the working signal frequency.
Figure 4.15 (a) Impedance profile of package vertical components with small voids. (b) Impedance profile of package vertical components with same cutout area on each layer.
Figure 4.16 Capacitive coupling due to insufficient void in package verticals.
Figure 4.17  No capacitive coupling due to increased void in package vertical.
Since the package vertical impedance spike at around 64 Ω, which is not too far from the target impedance 50Ω. Instead of keep exploring ways to reduce the impedance, it is better to first examine how it fits the whole channel design with multimode signaling. The full channel scenario is under the same structure as the benchmark channel which is package routing plus package vertical plus PCB routing and go into another package verticals and routings. With the same method, we generate CODEC from the cascaded S parameter of the whole channel and apply to the multimode transmitter circuit. The resulting eye diagram is shown in Fig. 4.18 (a), the RMS jitter is around 24 ps and eye height is around 85 mV which is nearly the same as the case with no package vertical. But without multimode signaling, the full channel with package vertical has much higher jitter and smaller eye height at 109 ps and 58 mV as shown in Fig. 4.18 (b).
Figure 4.18 (a) 1 mm pitch package vertical full channel with multimode. (b) 1 mm pitch package vertical full channel with traditional single ended driver.
The current pitch for signal and ground pins is 1 mm, smaller pitch values are explored to further increase the channel density. To reduce the pitch, the whole design needs to be shrunk proportionally in x-y direction. The package vertical design has been modified accordingly for 0.5 mm, 0.25 mm and 0.125 mm pitch separately. By bring signal pins closer more crosstalk noise would be induced, however, there isn’t too much signal degradation. The reason is when the pin pitch reduces, the ground pins are also brought closer to signal pins which reduces the loop inductance between the signal and ground pins. We choose 0.25 mm as the optimum pitch, which gives us 300% density increase, meanwhile leave enough room for routing and maintain reasonable aspect ratio for manufacturability. The resulting eye diagram with 0.25 mm pitch package vertical is shown in Fig. 4.19 (a) with RMS jitter 25 ps and eye height at 82 mV. Comparing with traditional single ended driver, the multimode signaling reduces RMS jitter by 76% from 107 ps as in Fig. 4.19 (b).
Figure 4.19 (a) 0.25 mm pitch package vertical full channel with multimode. (b) 0.25 mm pitch package vertical full channel with traditional single ended driver.
4.5 Optimized Channel Analysis and Discussion

We have demonstrated that with careful design of the channel, multimode signaling could deliver both high density channel and minimum crosstalk noise. Table 4.5 shows the detailed comparison between the benchmark and optimized channels, the optimized channel gives density increase of 300%, 97% and 300% for PCB routing, package routing and package vertical respectively. With applying multimode signaling to both channels, the optimized channel has RMS jitter 25 ps, 42% less than 43 ps of benchmark channel. To root cause this improvement, TDR simulation is used to examine the impedance profile of both channels as shown in Fig. 4.20. The benchmark channel package start with impedance as high as 65 Ω, but at the junction between package and PCB there appears to be a large capacitive discontinuity which pull the impedance to around 30 Ω; also the PCB channel has characteristic impedance at around 43 Ω not well matched to 50Ω. On the other hand for the optimized channel, despite one impedance peak at 66 Ω on the package it’s well matched to 50 Ω within the range of 5 Ω. The imbalanced impedance profile will cause more reflections as the signals propagate through the benchmark channel which lead to insertion loss degradation.

| Table 4.5 Detail Comparison between Benchmark Channel and New Channel |
|---|---|---|---|---|---|---|---|
| | PCB Pitch | Package Pitch | Socket Pitch | Jitter Without MMI | Height Without MMI | Jitter Binary MMI | Height Binary MMI |
| Benchmark Channel | 800 µm | 120 µm | 1000 µm | 90 ps | 60 mV | 43 ps | 77 mV |
| New Channel | 200 µm | 61 µm | 250 µm | 108 ps | 57 mV | 25 ps | 82 mV |
Figure 4.20 (a) Benchmark problem full channel impedance profile. (b) Optimized full channel impedance profile
To examine the channel difference in frequency domain, the insertion loss and power sum of FEXT of optimized and benchmark channel are presented in Fig. 4.21. At 10 GHz, insertion loss of benchmark channel drops to around -30 dB, 10 dB worse than the optimized channel which could be caused by the impedance mismatch. On the other hand, due to high density routing, the optimized channel suffers more FEXT than benchmark especially at higher frequencies.
Figure 4.21 (a) Insertion loss and power sum of FEXT of Benchmark channel. (b) Insertion loss and power sum of FEXT of optimized channel
To further compare the crosstalk effect of benchmark channel and the optimized channel, we have conducted several time domain simulations. A pulse signal is used as excitation on one side channel, the far end voltage of the rest three channels are examined to see the effect of FEXT (all channels are terminated with 50 Ω at both ends). Fig. 4.22 shows the results of benchmark channel, the FEXT magnitude of adjacent channel is around 100 mV, and drops to 40 mV and 20 mV for the 2\textsuperscript{nd} and 3\textsuperscript{rd} adjacent channels. The optimized channels shows different trend as depicted in Fig. 4.23, the FEXT magnitude stays almost the same level for all victim channels due to strong coupling effect from high density routing.
Figure 4.22 Benchmark channel waveforms (a) Aggressor pulse signal at channel 1 input. (b) Crosstalk victim at channel 2 output. (c) Crosstalk victim at channel 3 output. (d) Crosstalk victim at channel 4 output.
(a) Vin1 (V)

(b) Vout2 (mV)
Figure 4.23 Optimized channel waveforms (a) Aggressor pulse signal at channel 1 input. (b) Crosstalk victim at channel 2 output. (c) Crosstalk victim at channel 3 output. (d) Crosstalk victim at channel 4 output.
From above frequency and time domain analysis, we could see that the optimized channel suffers more crosstalk noise than the benchmark channel, but still shows 42% less RMS jitter with multimode signaling (24.7 ps VS 42.7 ps). To take advantage of the crosstalk free nature of multimode signaling, the impedance profile should be well controlled and very high density channel could be achieved.
Chapter 5

Large Bundle MMI Design

5.1 Introduction

To further increase channel density, earlier research have been done to study inter-bundle crosstalk impact [16], we will try to explore possibility to further increase bundle size without power consumption increase or with acceptable power penalty. Current implementation is focused on bundle size of 4, as shown in (3.19), the implementation would require signal number 4 times the modes number 4, total 16 CODEC to be active at driver stage as shown in fig. 3.4. If we keep the same implementation, to increase bundle size to n, the driver stage would have n$^2$ active branches, which means the power consumption would rise exponentially. With such high power consumption for large bundle, the bundle size would be strictly limited, and hard to further increase. But by expanding the driver implementation equation (3.19), we could notice that for channel 1, there are 4 CODEC entries from channel 4 which is the 3rd adjacent neighbor. If we eliminate these four CODEC entries, we would lose the crosstalk compensation from the 3rd adjacent neighbor which will make (3.19) into (5.1).
\[
\begin{bmatrix}
\text{Driver 1} \\
\text{Driver 2} \\
\text{Driver 3} \\
\text{Driver 4}
\end{bmatrix}
= \sum_{j=1}^{4} \sum_{k=1}^{3} T_{1j} U_{jk} \delta_j V_{\text{ink}} \\
\sum_{j=1}^{4} \sum_{k=1}^{4} T_{2j} U_{jk} \delta_j V_{\text{ink}} \\
\sum_{j=1}^{4} \sum_{k=1}^{4} T_{3j} U_{jk} \delta_j V_{\text{ink}} \\
\sum_{j=1}^{4} \sum_{k=2}^{4} T_{4j} U_{jk} \delta_j V_{\text{ink}}
\]

But given the loose coupling between channel 1 and 4, we would still get nice crosstalk suppression while lower the power consumption. Thus, we tuned off these CODEC entries in the circuit and examine the eye diagram shown in fig. 5.1 for the optimized full channel to compare with the case with all CODEC active as in fig. 4.19 (a).
With partial CODEC the RMS jitter rises from 24.8 ps to 36.4 ps, nearly 50%. But comparing with the case with no MMI case 107.4 ps as in Fig. 4.19 (b), the reduced CODEC implementation still give us about 66% jitter improvement. Also considering the optimized channel is utilizing the highest density routing possible, there is still certain degree of coupling between the 3rd adjacent neighbors. If we utilize this reduced CODEC implementation in a much larger bundle, the pitch could be further fine-tuned to find an optimum setting.
5.2 Eight Channel Bundle Example Design

5.2.1 Mathematical Derivation

To examine the reduced CODEC implementation in a large bundle, we constructed an eight channel PCB with the same setting as the PCB in the optimized channel as shown in Fig. 5.2.

For 8 channel model, the reduced S-parameter matrix would be 8 by 8, and so will the CODEC matrix $T$. With the same CODEC derivation and transformation method discussed in chapter 3, we could have a driver equation which has 64 (8x8) CODEC entries for each line, comparing with 16 entries for a 4 channel model. By applying the CODEC simplification method we presented in last sub-chapter which removes entries from 3$^{rd}$ adjacent neighbor and further neighbors, we could have the driver CODEC expression as shown in (5.2). With the new method which only consider the signal itself and two adjacent neighbors, we have successfully reduced the entry number from 64 (8x8) to 40 (5x8), since we still need the 8 timing adjustment for each mode on any line, this entry number would rise linearly as we expand the bundle size. Ideally we would want a fixed implementation which
would allow us to expand the bundle size to infinite, but so far the research work shows us we still need all 8 timing adjustments for 8 channel model. Since as the bundle size increases, the tuning work in simulation gets larger and larger which makes the validation work more difficult, a surrogate model or machine learning algorithm would help to accelerate the developing and exploring cycle for a better solution.

\[
\begin{align*}
\text{Driver 1} & : \sum_{j=1}^{8} \sum_{k=1}^{3} T_{1,j} U_{jk} \delta_j V_{ink} \\
\text{Driver 2} & : \sum_{j=1}^{8} \sum_{k=1}^{4} T_{2,j} U_{jk} \delta_j V_{ink} \\
\text{Driver 3} & : \sum_{j=1}^{8} \sum_{k=1}^{5} T_{3,j} U_{jk} \delta_j V_{ink} \\
\text{Driver 4} & : \sum_{j=1}^{8} \sum_{k=2}^{6} T_{4,j} U_{jk} \delta_j V_{ink} \\
\text{Driver 5} & : \sum_{j=1}^{8} \sum_{k=3}^{7} T_{5,j} U_{jk} \delta_j V_{ink} \\
\text{Driver 6} & : \sum_{j=1}^{8} \sum_{k=4}^{8} T_{6,j} U_{jk} \delta_j V_{ink} \\
\text{Driver 7} & : \sum_{j=1}^{8} \sum_{k=5}^{8} T_{7,j} U_{jk} \delta_j V_{ink} \\
\text{Driver 8} & : \sum_{j=1}^{8} \sum_{k=6}^{8} T_{8,j} U_{jk} \delta_j V_{ink}
\end{align*}
\] (5.2)
5.2.2 Results Analysis

To implement the reduced CODEC (5.2) in binary MMI circuit and validate the method, we have to make modifications to the current circuit implementation for the 8 channel example and reduced CODEC. In 4 channel example, each input signal will be adjusted for 4 phases, now instead of 4, each input signal will be adjusted for 8 different phases. At the driver stage, each driver will only take inputs from the two adjacent neighbors and itself, which gives us total 40 sub-driver entries (5x8) to implement the CODEC shown in (5.2). The updated transmitter circuit is simulated at 4 GT/s with 50 Ω termination resistors at both ends of the channel and each line has parasitic capacitive loadings of 0.5 pF at both ends. Fig. 5.3 shows the simulation results of the new 8 channel with reduced CODEC multimode signaling and with conventional single ended driver. The reduced CODEC implementation gives us 55% RMS jitter reduction from 67 ps to 30 ps, and due to we have more sub-drivers implemented in the circuit with same sizing, it gives us larger eye opening for 220 mV than 4 channel example channel. For the same reason, it also burns more power at 9.2 mW/Gb/s, near twice the 4 channel case at 5.6 mW/Gb/s. But if we size down the transistors to half the size, it would reduce the eye height to half, but also reduce the power consumption to about half the value. Thus, by carefully sizing the transistors at driver stage, we could utilize the multimode signaling for much larger bundle size with nearly no power efficiency loss.
Figure 5.3 Large bundle example 8 channel PCB model (a) Eye diagram with conventional single ended driver
(b) with reduced CODEC binary MMI driver
5.3 Summary

In this chapter, we introduced a new approach to apply binary MMI to large bundle size channel with nearly no power efficiency penalty. Since the CODEC is reduced to only include two adjacent neighbors’ crosstalk noise, we won’t be able to eliminate any crosstalk coupling from 3rd adjacent neighbor or further. The chip/channel co-design methodology we have been using is to routing the signals as close as possible, but if we stop cancelling crosstalk from far neighbors, this rule could be changed when we’re dealing with very tight channel parts such as package routing or vertical. So for large bundle design, co-design methodology should be revisited to explore more on the impacts from far neighbors’ crosstalk both in frequency and time domain to reveal more insights.
Chapter 6

Conclusion and Future Work

6.1 Conclusion

This research work presents a design methodology for multimode signaling circuit/channel co-design. It is based on an S-parameter based multimode CODEC derivation methods, and a fully tunable transmitter only multimode circuit implementation. The circuit is simulated to operate at 4 GT/s at a power efficiency of 5.6 mW/Gbps in 130 nm CMOS technology. A benchmark channel with real-life complexity has been used to validate the S-Parameter generated CODEC. Applying the CODEC to the existing channel resulted in around a 50% RMS jitter reduction over traditional single-ended signaling. Permitting an optimized channel to be designed from scratch led to a greater jitter reduction and ability to recover the signals with tighter routing pitch. The optimized channel showed over 75% RMS jitter reduction compared with single-ended signaling. Compared with a practical benchmark channel, the optimized channel has a density improvement of 300% and 97% for the PCB and package routing respectively, and 300% for package vertical pitch (μvia, PTH, socket, etc.) with signal to ground pin ratio 2:1. Although the optimized channel has much higher density, it still shows 42% jitter reduction over the benchmark channel with the same
crosstalk mitigation circuit and method. This new approach significantly improves the ability to use high density multimode interconnect sub-systems in practical scenarios.

A new multimode design approach has been proposed to be used for large bundle scenarios, with reduced CODEC implementation the required power consumption and design complexity is greatly improved for large bundle channel. An 8 channel example is used to validate the new method, with the reduced CODEC implementation RMS jitter is improved by 55% compared with traditional single ended driver, and the power consumption is 9.2 mW/Gb/s but gives twice eye height as the benefit of more power burning. The power efficiency could be improved by further sizing the driver transistor strength.

### 6.2 Future Work

There are several aspects of the multimode signaling research could be done to expand the current work.

- **Large Bundle MMI Design:**

  The reduced CODEC implementation for large bundle is proved preliminarily for a PCB channel. Due to the increased CODEC complexity comparing with 4 channel case, the number of tuning knobs of the timing adjustment and CODEC driver is getting larger. To find the optimum setting for certain channel, it is better to employ a machine learning based algorithm to simulate for optimal results and possibly employ in-situ calibration algorithm to train the CODEC on-chip after tape-out.

  Since this scheme eliminate the inputs for 3rd adjacent neighbor and further, the crosstalk noise is not fully removed but largely reduced. Further research could be
done to review the MMI optimized channel including package routing and vertical, to ensure there is enough spacing between signals and their 3rd adjacent neighbors. Instead of placing the signal routings as close as possible as presented in this work, the channel design rules would be the tradeoffs between routing density and tolerable crosstalk noise from far neighbors. For the large bundle design, package vertical could be further explored with different pin-out patterns and signal to ground ratio in order to be further optimized.

- **Test Chip Reliability and Testability Improvement:**
  
  From the chip test results, we noticed there is substantial supply noise for the transmitter only binary MMI scheme. For future tape-out, the power supply needs to be carefully designed to ensure the power integrity of the correct functionality of binary MMI. Simulations should be made to estimate the necessary on-chip decoupling capacitance needed to maintain noise in power rail under certain range. Also power regulator could be designed to maintain power supply stability. And to ensure the testability, it is recommended to bring as many as possible intermediate signals to the top metal (via buffers instead of direct wiring to protect inner logic). Even if the top pad is too small to probe, we could always use scanning electron microscopes (SEM) to do voltage contrast for debug purpose.

- **Circuit Implementation Improvement:**
  
  Equalization circuits are not added in this scheme on purpose to show the signal improvement is done by suppression crosstalk instead of compensating high frequency loss by equalization. So far, we have proven the crosstalk could be greatly
suppressed if not totally removed, equalizing circuits could be added to the current scheme to explore how feed forward equalization (FFE) at transmitter end, decision feedback equalization (DFE) and continuous-time linear equalization (CTLE) could be added to the scheme in order to reach the optimal signal integrity potential for the whole link and reach for higher signal rates (possibly with better CMOS technology nodes) with large eye opening.
REFERENCES


Appendix A

Binary MMI CODEC Generation Matlab Code

clc
clear all
close all

% low limit for s-parameter plots
lowdb=-70;
fsise=20; % Plot font size
lwidth=3; % Plot line width

% S-parameter read and Encoding/Decoding process
% data = read(rfdata.data,'MS4_8milw_5mils_5mmlength.s8p');
% data = read(rfdata.data,'Sp_ms4_10cm_sp_30G.s8p');
% data = read(rfdata.data,'MS4_8milw_5mils_10cmlength.s8p');
% data1 = read(rfdata.data,'PKG_match_MIN_2_5CM.s8p');
% data2 = read(rfdata.data, 'PCB_minimum_425INCH.s8p');
% data = read(rfdata.data,'ExtractedSparam_from_MS4_20cm_20GHz.s8p');
% data3 = read(rfdata.data, 'Design_Package_V11_2G2.s8p');

data1 = read(rfdata.data,'PKG.s8p');
data2 = read(rfdata.data, 'PCB.s8p');

freq = data1.Freq;

s_params1 = extract(data1,'S_PARAMETERS',50);
s_params2 = extract(data2,'S_PARAMETERS',50);
% s_params3 = extract(data3,'S_PARAMETERS',50);

%s_params = cascadesparams(s_params1, s_params3, s_params2, s_params3, s_params1, 4);
s_params = cascadesparams(s_params1, s_params2, s_params1, 4);

% data transfer from touchstone file to s-parameter matrix
%s_params = extract(data,'S_PARAMETERS',50);
s11 = s_params(1,1,:); 
s12 = s_params(1,2,:); 
s13 = s_params(1,3,:); 
s14 = s_params(1,4,:); 
s21 = s_params(2,1,:); 
s22 = s_params(2,2,:); 
s23 = s_params(2,3,:); 
s24 = s_params(2,4,:);
s31 = s_params(3,1,:);
s32 = s_params(3,2,:);
s33 = s_params(3,3,:);
s34 = s_params(3,4,:);
s41 = s_params(4,1,:);
s42 = s_params(4,2,:);
s43 = s_params(4,3,:);
s44 = s_params(4,4,:);
s51 = s_params(5,1,:);
s52 = s_params(5,2,:);
s53 = s_params(5,3,:);
s54 = s_params(5,4,:);
s61 = s_params(6,1,:);
s62 = s_params(6,2,:);
s63 = s_params(6,3,:);
s64 = s_params(6,4,:);
s71 = s_params(7,1,:);
s72 = s_params(7,2,:);
s73 = s_params(7,3,:);
s74 = s_params(7,4,:);
s81 = s_params(8,1,:);
s82 = s_params(8,2,:);
s83 = s_params(8,3,:);
s84 = s_params(8,4,:);

% Original reduced S-parameter matrix
s_m=[
    s51  s52  s53  s54 ;
    s61  s62  s63  s64 ;
    s71  s72  s73  s74 ;
    s81  s82  s83  s84 ];

% total samling number
num_freq=length(s51)

%Initial coefficient calculation
%To calculate initial coefficients,

Total_SNR=0;
Freq_chosen=0;

for samplepoint = 1:1:num_freq

    s_s=[
        s51(1,1,samplepoint)  s52(1,1,samplepoint)  s53(1,1,samplepoint)  s54(1,1,samplepoint) ];

s61(1,1,samplepoint)  s62(1,1,samplepoint)  s63(1,1,samplepoint)  s64(1,1,samplepoint) ;
s71(1,1,samplepoint)  s72(1,1,samplepoint)  s73(1,1,samplepoint)  s74(1,1,samplepoint) ;
s81(1,1,samplepoint)  s82(1,1,samplepoint)  s83(1,1,samplepoint)  s84(1,1,samplepoint) ];

% The absolute value of S matrix is used for calculating Eigenvalue.
[cmxT_tmp, cmxD_tmp] = eig(s_s);
invcmxT_tmp=inv(cmxT_tmp);

s_matrix=abs(s_s);
[T_tmp, D_tmp] = eig(s_matrix);
invT_tmp=inv(T_tmp);

total_SNR_tmp=0;
for test_point = 1:1:num_freq

    s_s=[
    s51(1,1,test_point)  s52(1,1,test_point)  s53(1,1,test_point)  s54(1,1,test_point) ;
    s61(1,1,test_point)  s62(1,1,test_point)  s63(1,1,test_point)  s64(1,1,test_point) ;
    s71(1,1,test_point)  s72(1,1,test_point)  s73(1,1,test_point)  s74(1,1,test_point) ;
    s81(1,1,test_point)  s82(1,1,test_point)  s83(1,1,test_point)  s84(1,1,test_point) ];

    New_s_s_tmp=invT_tmp*s_s*T_tmp;
    Abs_New_s=abs(New_s_s_tmp);
    total_SNR_tmp=total_SNR_tmp+(Abs_New_s(1,1)/(Abs_New_s(1,2)+Abs_New_s(1,3)+Abs_New_s(1,4)))+(Abs_New_s(2,2)/(Abs_New_s(2,1)+Abs_New_s(2,3)+Abs_New_s(2,4)))+(Abs_New_s(3,3)/(Abs_New_s(3,1)+Abs_New_s(3,2)+Abs_New_s(3,4)))+(Abs_New_s(4,4)/(Abs_New_s(4,1)+Abs_New_s(4,2)+Abs_New_s(4,3)));
end

SNR(samplepoint)=total_SNR_tmp;

if total_SNR_tmp>Total_SNR
    Total_SNR=total_SNR_tmp;
    T=T_tmp;
    D=D_tmp;
    invT=invT tmp;
    Freq_chosen=samplepoint;

    cmxT=cmxT_tmp;
    invcmxT=invcmxT_tmp;
end
end

Total_SNR
Freq_chosen

%Calculate the CODEC for tx only

matrix=T

InvMatrix=invT

D

CODEC = zeros(4,16);

for i = 1:1:4
    for j = 1:1:4
        for k = 1:1:4
            CODEC(i,(j-1)*4+k)=matrix(i,j)*InvMatrix(j,k);
        end
    end
end

CODEC

CODEC_MINUS=-CODEC

% For-loop to calculate modified s-parameters
for m=1:num_freq
    new_s1(:,:,m)=invT*s_m(:,:,m)*T;
    new_s(:,:,m)=new_s1(:,:,m);
end

ns51 = new_s(1,1,:);
s52 = new_s(2,1,:);
s53 = new_s(3,1,:);
s54 = new_s(4,1,:);
s61 = new_s(1,2,:);
s62 = new_s(2,2,:);
s63 = new_s(3,2,:);
s64 = new_s(4,2,:);
s71 = new_s(1,3,:);
s72 = new_s(2,3,:);
s73 = new_s(3,3,:);
s74 = new_s(4,3,:);
s81 = new_s(1,4,:);
s82 = new_s(2,4,:);
s83 = new_s(3,4,:);
s84 = new_s(4,4,:);
for m=1:num_freq
    new_s1(:,m)=invcmxT*s_m(:,m)*cmxT;
    new_s(:,m)=new_s1(:,:);
end
cmxns51 = new_s(1,1,:);
cmxns52 = new_s(2,1,:);
cmxns53 = new_s(3,1,:);
cmxns54 = new_s(4,1,:);
cmxns61 = new_s(1,2,:);
cmxns62 = new_s(2,2,:);
cmxns63 = new_s(3,2,:);
cmxns64 = new_s(4,2,:);
cmxns71 = new_s(1,3,:);
cmxns72 = new_s(2,3,:);
cmxns73 = new_s(3,3,:);
cmxns74 = new_s(4,3,:);
cmxns81 = new_s(1,4,:);
cmxns82 = new_s(2,4,:);
cmxns83 = new_s(3,4,:);
cmxns84 = new_s(4,4,:);

%GHz frequency
freq=freq./1e9;

%Plotting for magnitude of original s-parameters
%==================================================
figure(1)
h1 = plot( freq, 20*log10(abs(s51(:))) ); hold on;
h2 = plot( freq, 20*log10(abs(s52(:))) ); hold on;
h3 = plot( freq, 20*log10(abs(s53(:))) ); hold on;
h4 = plot( freq, 20*log10(abs(s54(:))) ); hold on;
h5 = plot( freq, 20*log10(abs(s61(:))) ); hold on;
h6 = plot( freq, 20*log10(abs(s62(:))) ); hold on;
h7 = plot( freq, 20*log10(abs(s63(:))) ); hold on;
h8 = plot( freq, 20*log10(abs(s64(:))) ); hold on;
hold off;

h = legend('
'S51','S52','S53','S54','S61','S62','S63','S64',2);
set(h,'Interpreter','none','fontsize',fsize,'fontweight','b');

set(h1,'color','black','Linewidth',lwidth);
s(h2,'color','blue','Linewidth',lwidth);
s(h3,'color','yellow','Linewidth',lwidth);
s(h4,'color','green','Linewidth',lwidth);

set(h5,'color','yellow','Linewidth',lwidth,'LineStyle','--');
s(h6,'color','cyan','Linewidth',lwidth,'LineStyle','--');
s(h7,'color','magenta','Linewidth',lwidth,'LineStyle','--');
s(h8,'color','red','Linewidth',lwidth,'LineStyle','--');
% Plotting for the magnitude of modified s-parameters

% Plotting for the magnitude of modified s-parameters
figure(2)
fig2 = figure;

h1 = plot( freq, 20*log10(abs(ns51(:))) ); hold on;
h2 = plot( freq, 20*log10(abs(ns52(:))) ); hold on;
h3 = plot( freq, 20*log10(abs(ns53(:))) ); hold on;
h4 = plot( freq, 20*log10(abs(ns54(:))) ); hold on;
h5 = plot( freq, 20*log10(abs(ns61(:))) ); hold on;
h6 = plot( freq, 20*log10(abs(ns62(:))) ); hold on;
h7 = plot( freq, 20*log10(abs(ns63(:))) ); hold on;
h8 = plot( freq, 20*log10(abs(ns64(:))) ); hold on;
hold off;

h = legend('S51', 'S52', 'S53', 'S54', 'S61', 'S62', 'S63', 'S64', 2);
set(h, 'Interpreter', 'none', 'FontSize', fsize, 'FontWeight', 'b');
set(h1, 'Color', 'Black', 'LineWidth', lwidth);
set(h2, 'Color', 'Blue', 'LineWidth', lwidth);
set(h3, 'Color', 'Yellow', 'LineWidth', lwidth);
set(h4, 'Color', 'Green', 'LineWidth', lwidth);
set(h5, 'Color', 'Yellow', 'LineWidth', lwidth, 'LineStyle', '--');
set(h6, 'Color', 'Cyan', 'LineWidth', lwidth, 'LineStyle', '--');
set(h7, 'Color', 'Magenta', 'LineWidth', lwidth, 'LineStyle', '--');
set(h8, 'Color', 'Red', 'LineWidth', lwidth, 'LineStyle', '--');

Title('Package+PCB+Package S-parameters with CODEC', 'FontSize', fsize, 'FontWeight', 'b');
xlabel('Frequency (GHz)', 'FontSize', fsize, 'FontWeight', 'b');
ylabel('Magnitude (dB)', 'FontSize', fsize, 'FontWeight', 'b');
grids minor;
xlim([0.1, 10]); ylim([lowdb, 0]);

% Plotting for the magnitude of the complex modified s-parameters

figure(3)
h1 = plot( freq, 20*log10(abs(cmxns51(:))) ); hold on;
\begin{verbatim}
    h2 = plot( freq, 20*log10(abs(cmxns52(:))) ); hold on;
    h3 = plot( freq, 20*log10(abs(cmxns53(:))) ); hold on;
    h4 = plot( freq, 20*log10(abs(cmxns54(:))) ); hold on;
    h5 = plot( freq, 20*log10(abs(cmxns61(:))) ); hold on;
    h6 = plot( freq, 20*log10(abs(cmxns62(:))) ); hold on;
    h7 = plot( freq, 20*log10(abs(cmxns63(:))) ); hold on;
    h8 = plot( freq, 20*log10(abs(cmxns64(:))) ); hold on;
    hold off;

    h = legend('S51','S52','S53','S54','S61','S62','S63','S64',2);
    set(h,'Interpreter','none','fontsize',fsize,'fontweight','b');

    set(h1,'color','black','Linewidth',lwidth,'LineStyle','--');
    set(h2,'color','blue','Linewidth',lwidth,'LineStyle','--');
    set(h3,'color','yellow','Linewidth',lwidth,'LineStyle','--');
    set(h4,'color','green','Linewidth',lwidth,'LineStyle','--');

    set(h5,'color','yellow','Linewidth',lwidth);
    set(h6,'color','cyan','Linewidth',lwidth);
    set(h7,'color','magenta','Linewidth',lwidth);
    set(h8,'color','red','Linewidth',lwidth);

    Title('Package+PCB+Package S-parameters with complex CODEC',
    'fontsize',fsize,'fontweight','b');
    xlabel('Frequency (GHz)', 'fontsize',fsize,'fontweight','b');
    ylabel('Magnitude (dB)', 'fontsize',fsize,'fontweight','b');
    grid minor;
    xlim([0.1,10]);
    ylim([lowdb,0]);

    figure(4)
    h1 = plot( freq, 180/pi*angle(s51(:)) ); hold on;
    h2 = plot( freq, 180/pi*angle(s62(:)) ); hold on;
    h3 = plot( freq, 180/pi*angle(s73(:)) ); hold on;
    h4 = plot( freq, 180/pi*angle(s84(:)) ); hold on;
    hold off;

    h = legend('S51','S62','S73','S84',2);
    set(h,'Interpreter','none','fontsize',fsize,'fontweight','b');

    set(h1,'color','black','Linewidth',lwidth);
    set(h2,'color','blue','Linewidth',lwidth);
    set(h3,'color','red','Linewidth',lwidth);
    set(h4,'color','green','Linewidth',lwidth);
\end{verbatim}
```matlab
Title('S-parameters without CODEC','fontsize',fsize,'fontweight','b');
xlabel('Frequency [GHz]','fontsize',fsize,'fontweight','b');
ylabel('Phase [dB]','fontsize',fsize,'fontweight','b');
grid minor;
xlim([0.1,10]);
ylim([-200,200]);

figure(5)
h1 = plot( freq, 20*log10(abs(SNR(:))) ); hold on;
hold off;

h = legend('SNR',2);
set(h,'Interpreter','none','fontsize',fsize,'fontweight','b');
set(h1,'color','black','Linewidth',lwidth);
Title('SNR with decomposition matrix T','fontsize',fsize,'fontweight','b');
xlabel('Frequency [GHz]','fontsize',fsize,'fontweight','b');
ylabel('SNR (dB)','fontsize',fsize,'fontweight','b');
grid minor;
xlim([0.1,10]);
ylim([80,90]);```