A HIGH SPEED CHIP TO CHIP INTERCONNECTION CIRCUIT FOR FPGA EMULATION SYSTEMS

by

Brent Alan Beacham

A thesis submitted in conformity with the requirements for the degree of Master of Applied Science
Graduate Department of Electrical and Computer Engineering
University of Toronto

© Copyright by Brent Alan Beacham 2001
The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L’auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L’auteur conserve la propriété du droit d’auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-58788-6
A High Speed Chip to Chip Interconnection Circuit for FPGA Emulation Systems

Brent Alan Beacham
Master of Applied Science
Graduate Department of Electrical and Computer Engineering
University of Toronto
2001

Abstract

In partitioned FPGA designs such as those in FPGA emulation systems, the limited number of I/O pins typically restricts the amount of on-chip logic that can be used. This thesis describes a system that increases the amount of data transferred from chip to chip by multiplexing sixteen 8-bit wide on-chip signals at 100MHz into a 12-bit wide off-chip signal that operates at 1.6Gbps per pin. The transmitted data is encoded so that each set of transmitted bits is dc-balanced and contains guaranteed transitions which the receiver uses to track changes in the timing of the data. The sampler timing in each of the receivers is dynamically adjusted to account for any pin-to-pin timing skew.

The encoder, transmitter and receiver were implemented in a 0.35μm CMOS process. Simulations show that will transmit and receive data at a rate of 1.6Gbps per pin.
This thesis would not have been possible without the help of many people and organizations.

I would like to thank my advisor Professor Dave Lewis for his support and guidance.

I greatly appreciate the friendship and help of my colleagues in the Electronics and Computer groups here at the university. I would like to thank Cameron and Nirmal for their help in many matters. Thanks also to all my friends in LP476 and the Electronics and Computer groups including Tor, Sean, Franklin, and Edward.

I would also like to thank Jaro Pristupa for his expert help in solving many CAD problems.

Chip fabrication support from CMC is greatly appreciated.

Most of all, I am deeply grateful to my family and especially grateful to Jennifer for their love, support and encouragement.
Contents

1 Introduction ........................................ 1
  1.1 Current Focus .................................. 3
  1.2 Thesis Overview ................................ 3

2 Background .......................................... 5
  2.1 Problem Description ............................ 5
  2.2 Chip Interconnection Circuit Design Choices .... 13
    2.2.1 Timing Recovery ............................ 13
    2.2.2 Number of Bits per Transmitted Symbol ....... 15
    2.2.3 Single-Ended/Fully Differential Signaling ...... 16
    2.2.4 Global or Individual Line Phase Tracking ....... 17
    2.2.5 Unidirectional or Bidirectional Signaling ....... 18
  2.3 Previous Work ..................................... 18
    2.3.1 Oversampling Receivers ..................... 18
    2.3.2 Multilevel Signaling ...................... 21
    2.3.3 Bidirectional Transceiver ................... 22
  2.4 Thesis Goals ...................................... 25
  2.5 Phase Locked Loop (PLL) Overview .............. 25
    2.5.1 Voltage Controlled Oscillator (VCO) .......... 26
    2.5.2 Phase/Frequency Detector (PFD) .............. 27
    2.5.3 Charge Pump ............................... 27
    2.5.4 Loop Filter ............................... 28
    2.5.5 Digital Divider ............................. 28
  2.6 Delay-Locked Loop (DLL) Overview .............. 30

3 System and Logic Design ......................... 33
  3.1 Parallel Transceiver Circuit Overview .......... 33
    3.1.1 Receiver Overview .......................... 34
CONTENTS

3.1.2 Encoder Overview ........................................ 36
3.1.3 Transmitter Overview .................................... 38
3.2 DLL Architecture of the Receiver ......................... 38
3.3 Receiver Design ............................................ 40
  3.3.1 Timing Recovery ......................................... 40
  3.3.2 Clock Generator PLL .................................... 42
  3.3.3 Phase Interpolator ....................................... 43
  3.3.4 Receiver State Updating ................................ 44
  3.3.5 Transition Counter ....................................... 45
  3.3.6 Global Clock Synchronization ......................... 45
3.4 Encoder ..................................................... 46
  3.4.1 Reasons for Encoding .................................... 46
  3.4.2 Implementation of Encoder .............................. 48
  3.4.3 Transitions on Output Lines ......................... 51
3.5 Decoder ..................................................... 52
3.6 Transmitter ................................................ 53

4 VLSI Implementation .................................. 55
  4.1 Symmetric Load Delay Cell ............................... 55
  4.2 Phase Locked Loop ......................................... 59
    4.2.1 Voltage Controlled Oscillator (VCO) .............. 60
    4.2.2 Phase Detector ....................................... 61
    4.2.3 Charge Pump .......................................... 63
    4.2.4 Divider ................................................. 63
    4.2.5 Loop Filter Parameters ............................... 65
  4.3 Receiver .................................................. 71
    4.3.1 Sampling Comparator .................................. 71
    4.3.2 Phase (Delay) Interpolator .......................... 73
    4.3.3 Global Clock Synchronizer ............................ 78
    4.3.4 Layout of a Slice of the Slave Receiver ............ 79
    4.3.5 Layout of the Master Slice .......................... 79
  4.4 Transmitter .............................................. 82
  4.5 Encoder ................................................... 85
  4.6 Chip Layout ................................................ 85
  4.7 Test Circuitry ............................................. 86
## CONTENTS

5 Simulation Results .......................... 87  
5.1 Transmitter ............................... 87  
5.2 Receiver .................................. 92  
5.2.1 Master Receiver ......................... 92  
5.2.2 Clock Generator ......................... 94  

6 Conclusions ................................. 97  
6.1 Future Work ............................... 98  

A Package Models ............................. 99  
B Simulation Circuits .......................... 101
List of Tables

2.1 Summary of Chip Described in Paper [9]................................. 21
2.2 Summary of Chip Described in Paper [4]................................. 22
2.3 Summary of Features and Performance for Chip Described in [10]. 22
2.4 Power Consumption @ 3.6V for Chip Described in [10]............. 23
2.5 Summary of Chip Described in Paper [6]................................. 24
2.6 Summary of Chip Described in Paper [8]................................. 24

4.1 Transistor Sizes for figure 4.1............................................. 57
4.2 Transistor Sizes for figure 4.9............................................ 62
4.3 Transistor Sizes for figure 4.11........................................... 65
4.4 Transistor Sizes for figure 4.12........................................... 67
4.5 Transistor Sizes for figure 4.14........................................... 68
4.6 Transistor Sizes for figure 4.15........................................... 69
4.7 Transistor Sizes for figure 4.16........................................... 75
4.8 Capacitor Sizes for figure 4.16............................................ 75
4.9 Transistor Sizes for figure 4.25........................................... 83
4.10 Transistor Sizes for figure 4.26........................................... 84

5.1 PLL Current Consumption................................................... 96
5.2 Component Current Consumption......................................... 96

A.1 Component sizes for figure A.1......................................... 100
B.1 Component values in figures B.1 and B.2............................ 101
List of Figures

2.1 Parallel to Serial Conversion ........................................... 7
2.2 Chip Overview .......................................................... 8
2.3 Partitioned Signals Example ........................................... 11
2.4 Transmitted Signals Example .......................................... 12
2.5 Location of Data Transitions in 4-PAM .............................. 16
2.6 Transmitter with External Resistor ................................... 20
2.7 VCO Loop Architecture ................................................. 26
2.8 Simple PFD Architecture .............................................. 28
2.9 PFD Average Output .................................................... 29
2.10 Charge Pump ............................................................ 30

3.1 System Overview ....................................................... 34
3.2 Receiver Overview ...................................................... 35
3.3 DCW and Clocks in Receiver ......................................... 37
3.4 DLL Architecture of the Receiver ................................... 39
3.5 DLL Phase Step Response .............................................. 39
3.6 Phase Error in Locked Slave Receiver ............................... 40
3.7 Receiver Top Level ....................................................... 41
3.8 Early/Late Timing Decision .......................................... 42
3.9 Comparator Detail ....................................................... 42
3.10 Comparators ............................................................ 43
3.11 Clock Generator VCO .................................................. 43
3.12 Global Clock Synchronization ....................................... 46
3.13 Encoding Scheme Overview .......................................... 51
3.14 Transmitter ............................................................ 54

4.1 Delay Cell ............................................................... 56
4.2 Symmetric Load ......................................................... 57
4.3 Ideal Symmetric Load I-V Characteristic ............................ 58
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4</td>
<td>Simulated load I-V Characteristic</td>
<td>59</td>
</tr>
<tr>
<td>4.5</td>
<td>VCO Loop Architecture</td>
<td>59</td>
</tr>
<tr>
<td>4.6</td>
<td>PLL Circuit Schematic</td>
<td>60</td>
</tr>
<tr>
<td>4.7</td>
<td>Voltage Controlled Oscillator</td>
<td>60</td>
</tr>
<tr>
<td>4.8</td>
<td>Voltage Controlled Oscillator Layout</td>
<td>61</td>
</tr>
<tr>
<td>4.9</td>
<td>Bias Circuit for the VCO Delay Cell</td>
<td>62</td>
</tr>
<tr>
<td>4.10</td>
<td>VCO Frequency vs. Control Voltage</td>
<td>63</td>
</tr>
<tr>
<td>4.11</td>
<td>Phase Detector Schematic</td>
<td>64</td>
</tr>
<tr>
<td>4.12</td>
<td>Charge Pump Schematic</td>
<td>66</td>
</tr>
<tr>
<td>4.13</td>
<td>Digital Divider Schematic</td>
<td>67</td>
</tr>
<tr>
<td>4.14</td>
<td>Dynamic Flip-Flop Schematic</td>
<td>68</td>
</tr>
<tr>
<td>4.15</td>
<td>Differential to Single Ended Converter</td>
<td>69</td>
</tr>
<tr>
<td>4.16</td>
<td>Sampling Comparator</td>
<td>74</td>
</tr>
<tr>
<td>4.17</td>
<td>Sampling Comparator Clocks</td>
<td>74</td>
</tr>
<tr>
<td>4.18</td>
<td>Phase Selector and Blender</td>
<td>76</td>
</tr>
<tr>
<td>4.19</td>
<td>Delay Cell Multiplexer</td>
<td>76</td>
</tr>
<tr>
<td>4.20</td>
<td>Delay Cell Multiplexer Interpolator</td>
<td>77</td>
</tr>
<tr>
<td>4.21</td>
<td>Synchronizer Delay Cell</td>
<td>78</td>
</tr>
<tr>
<td>4.22</td>
<td>Delay Cell Control Voltage Generator</td>
<td>79</td>
</tr>
<tr>
<td>4.23</td>
<td>Layout of Slave Slice</td>
<td>80</td>
</tr>
<tr>
<td>4.24</td>
<td>Layout of Master Slice</td>
<td>81</td>
</tr>
<tr>
<td>4.25</td>
<td>Transmitter Schematic</td>
<td>82</td>
</tr>
<tr>
<td>4.26</td>
<td>Transmitter Bias</td>
<td>83</td>
</tr>
<tr>
<td>4.27</td>
<td>Layout of Chip</td>
<td>85</td>
</tr>
<tr>
<td>5.1</td>
<td>Transmitter Signals viewed at Receiver at 1GHz</td>
<td>89</td>
</tr>
<tr>
<td>5.2</td>
<td>Normalized Received Transmitter Signals at 1GHz</td>
<td>89</td>
</tr>
<tr>
<td>5.3</td>
<td>Normalized Received Transmitter Signals at 1.6GHz</td>
<td>90</td>
</tr>
<tr>
<td>5.4</td>
<td>Transmitter Output</td>
<td>91</td>
</tr>
<tr>
<td>5.5</td>
<td>Master Receiver Unlocked State</td>
<td>93</td>
</tr>
<tr>
<td>5.6</td>
<td>Master Receiver Locked State</td>
<td>94</td>
</tr>
<tr>
<td>5.7</td>
<td>State Changes in the Master Receiver</td>
<td>95</td>
</tr>
<tr>
<td>5.8</td>
<td>VCO output at 800MHz</td>
<td>95</td>
</tr>
<tr>
<td>A.1</td>
<td>Package Model</td>
<td>99</td>
</tr>
<tr>
<td>B.1</td>
<td>Test Circuit For Transmitter</td>
<td>102</td>
</tr>
<tr>
<td>B.2</td>
<td>Test Circuit For Receiver</td>
<td>103</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Improvements in processing technology and in computer aided design (CAD) tools have allowed the complexity of digital integrated circuits to constantly increase. Because of the high costs that are incurred if a design is defective, it is important that chips are fully verified before they are fabricated. However, as the complexity of circuits increases it becomes more difficult to verify that a design functions as desired, both on its own and in the system that it is intended for.

There are many ways to verify the operation of a circuit. One is by circuit simulation at the transistor level. Simulations can also be done at the logic gate level or at a higher level where the circuit is described using a hardware description language such as VHDL or Verilog. As the level of abstraction gets lower the simulations become more accurate but take more time to complete. For very large circuits the simulations can take a very long time. Also when creating the test vectors for these simulations it is not always possible to foresee all the possible
Chapter 1. Introduction

cases that the real circuit will face.

Another way that a design can be verified is by emulating it on one or more programmable chips. A field programmable gate array (FPGA) is a chip that contains programmable logic elements and interconnect which can be programmed by readily available software using the proper hardware interface. A digital design can be programmed onto an FPGA and then tested by applying the same signals to it that would be applied to the completed chip. The design implemented on the FPGA would not likely run as fast as the completed chip would but it would run much faster than any computer simulation could.

There are limitations to how large of a design can be implemented on a single FPGA due to the limited number of logic elements available. When the logic on one chip is insufficient, the design can be partitioned and implemented on two or more FPGAs. The disadvantage of this is that as a result of the partitioning signals that previously were only internal must subsequently pass through the I/O pads from one chip to another. Although the signals were not originally output signals, after the partitioning they could be needed on more than one of the chips or the signals could be generated in logic on one chip and used in another. Typically the number of these signals exceeds the number of I/O pins available on contemporary chips. This shortage of pins causes inefficient use of the FPGAs because available logic is not able to be completely used. This effect gets worse as the number of chips that the design is partitioned across is increased.

Some companies([1] [2]) produce emulation systems that have many FPGAs on them connected together to allow simulation of large digital systems. The
1.1 Current Focus

capabilities of these simulation systems as well as any partitioned FPGA design could be increased with a solution for the I/O pin limitations.

1.1 Current Focus

The objective of this thesis was to increase the effective number of signal pins on a chip (while keeping the number of physical pins unchanged.) FPGAs manufactured in a 0.35\textmu{}m CMOS process typically operate with a clock rate of 50MHz to 100MHz. When used in an FPGA emulation system the total interconnect length from chip to chip is expected to be less than 40cm. As well, all of the signal traces from one chip to another should be routed together as a group and will have similar lengths and geometry.

Since the on-chip clock frequencies are of moderate speed, it is possible to increase the frequency of the off-chip interconnect and then multiplex many of the low frequency on-chip signals onto a single off-chip interconnect. To be useful in the system, a chip using the interconnect circuit must use fewer pins than a chip not using the circuit.

1.2 Thesis Overview

Chapter 2 provides a more detailed description of the high speed interconnection circuit and discusses some work that has already been done on the subject. It also contains the goals of the thesis and a brief tutorial on phase locked loops and delay
locked loops. Chapter 3 focuses on the design of the circuit implemented in this thesis. Chapter 4 presents the transistor level design and layout issues. Chapter 5 shows results from HSPICE [3] simulations of the circuit. Chapter 6 presents the conclusions of the thesis and discusses possible future work.
Chapter 2

Background

This chapter contains background information regarding the motivation and goals of this thesis. Section 2.1 describes the motivation behind the implemented chip. Section 2.2 discusses issues with chip to chip interconnection. Section 2.3 describes the previous work done in the area of chip interconnection. The goals of this thesis are presented in section 2.4 followed by a brief overview on phase-locked loops (PLLs) and delay locked loops (DLLs) in sections 2.5 and 2.6 respectively.

2.1 Problem Description

The parallel transceiver circuit (PTC) built for this thesis is intended to increase the amount of data transmitted through chip I/O pads to increase the effective number of I/O pads available. By “effective” it is meant that to the on-chip core
design it looks like there are more I/Os than actually exist physically on the package. The interconnection circuit multiplexes a number of on-chip signals into a smaller number of physical output pads at the transmitter and does the equivalent demultiplexing of the signals at the receiver.

For the same amount of data to be transmitted in the same time on a fewer number of physical I/O pins then either the data symbols must be sent at a faster rate and/or there must be more bits transmitted per symbol. (In a conventional digital system the symbols are either a logical "1" or "0".)

In conventional digital synchronous systems data is synchronized to a system clock. Typically at the rising clock edge input data is latched into internal flip-flops. The clock period is long enough that the I/O data can propagate and settle within that time. If data symbols are to be transmitted and received at a rate that is faster than the system clock then the system clock can no longer be used to drive the transmitter and latch in received data. Either a higher speed clock, or many clocks with the proper delays in their rising edges compared to the system clock must be generated on-chip. These clocks must be generated precisely because as symbol period decreases any errors in the timing of the clock become a larger percentage of the period and can more easily cause errors in the data reception.

Figure 2.1 illustrates how many lower speed data lines can be combined into a single high speed line. In the figure, it can be seen that $N$ lower speed lines with periods of $T_{\text{system}}$ can be transmitted on a single line if the period of the transmitted bits ($T_{I/O}$) is $\frac{T_{\text{system}}}{N}$.

Figure 2.2 shows a typical system used in high speed chip interconnection.
2.1 Problem Description

Data from \( N \) input lines are serialized, encoded, and then transmitted. The bit-rate at the I/O is \( N \) times that of the inputs.

![Diagram of parallel to serial conversion](image)

Figure 2.1: Parallel to Serial Conversion

When the chips are connected together on the printed circuit board, each set of the high speed interconnects will be routed together as a group. They will follow similar paths, be close to the other traces in the group, and will be almost the same length. Because of this, there should be some correlation in the timing jitter that affects the signals because noise that affects one trace should affect the others similarly. In addition to this, timing jitter caused by fluctuations in the power supply on the transmitting chip will be correlated in the output lines from
that chip. To save silicon area, the receiver in the design takes advantage of the assumed correlation and uses a master/slave configuration where changes in the timing on one of the received lines are tracked more quickly than on the other lines. These timing changes are propagated to the other receivers by the clock signals that the master receiver generates and distributes. Because each slave receiver uses the distributed clocks to generate their sampling clocks, timing jitter that is correlated across all the input lines should still be tracked quickly. Each of the slave receivers also determines a timing offset from the distributed clock signals (from the master receiver) to use for sampling the data on their input lines. By changing the offset, the slave receivers will adjust their timing to account for any pin to pin timing variations. The pin to pin timing is not expected to change rapidly and so the slave receivers do not need to update their timing offsets as rapidly as the master receiver adjusts its timing clocks. Because of this, the slave
receivers can be made simpler than the master receiver and can share some logic circuitry in order to save area.

In figure 2.3a an example of two chips connected with conventional interconnect is shown. After partitioning of the design, the output of block Logic1 on Chip1 could be an input for block Logic2 on Chip2, which in turn could be an input for block Logic3 on Chip1, and so forth for the rest of the logic blocks until the output of Logic5 is available for use on Chip1. The signal propagation through the entire chain of logic may be required within one on-chip clock period. The conventional interconnect could be replaced by the high speed interconnect shown in figure 2.3b. However, the same signal propagation through logic blocks Logic1 through Logic5 is still required within one on-chip clock period. It is therefore desired that the interconnect circuit does not add unnecessary delay to the signals or the maximum on-chip clock rate will be reduced.

The software programming the FPGA can calculate the delays in the logic on the chips and will also know when the data from any particular logical output will be transmitted. With this knowledge the software can instead connect the on-chip signals to the output that will transmit the data as early as possible after the data is valid. This is illustrated in figure 2.4 where the low speed on-chip signals A through G could be converted into the high speed off-chip signal H. (Note that a full clock period of the low speed signals is not shown.) Each of the signals is transmitted when they are known to be valid, but before the end of the current clock period.

With this method, it is not necessary to wait until the end of the period before
the signals are sent to the output and the overall latency of the system is reduced.
2.1 Problem Description

a) Conventional Interconnect

b) High Speed Multiplexed Interconnect

Figure 2.3: Partitioned Signals Example
Figure 2.4: Transmitted Signals Example
2.2 Chip Interconnection Circuit Design Choices

This section discusses in more detail some of the main differences and design choices among current chip interconnection circuits.

2.2.1 Timing Recovery

To recover the data from the input lines the receiver must sample the line when the data is valid. In an ideal system where square pulses are transmitted, this would be any time other than at the signal transition. Any real system is band-limited which causes a signal to have finite rise time, reducing the signal strength for some time near the transition. As well, noise causes jitter in the internal clock signals generated on each chip. Clock jitter in the transmitter adds some uncertainty in when the transitions occur in the data, while clock jitter in the receiver adds uncertainty to the time at which the data is sampled. For these reasons it is best to sample the data as far as possible from the transitions, which is in the centre of the data period.

One way to obtain a sample in the centre of the input bit is to sample the input twice in each input period ($T_{bit\_time}$) with the sample times $\frac{T_{bit\_time}}{2}$ apart. In normal circuit operation one sample is done in the middle of the data and the other is done at where the transition in the data (ie. from a logical “1” to “0”) could occur. If there is a transition and the sample at the transition is the same as the preceding data sample, then the sampling clocks are too early. If the transition sample is the same as the following data, then the sampling clocks are too late.
(This is described in more detail in section 3.3.1.) Based on the early/late signals, the sampling clocks are adjusted until they are sampling at the proper times.

Another way to recover the timing at the receiver is to oversample the input $N$ times during each bit time (where $N \geq 3$) [4] [5]. Digital logic is used on the oversampled data to detect where the transitions in the data have occurred and then to select the sample the farthest from the data transition. In the case of $N \times$ oversampling, the spacing in time between the samples is $\frac{\text{bit-time}}{N}$. Unlike the phase tracking approach mentioned previously, the timing of the sampling clocks are never adjusted.

The advantage of the oversampling method is that it can potentially track changes more quickly because the samples chosen can be changed after every bit received. In the phase tracking approach, modifying the clocks typically requires some kind of negative-feedback loop whose bandwidth has to be limited because of stability concerns. This limits the speed at which the timing of the clocks can be changed.

The disadvantages of the oversampling method are that each input is loaded by $N$ samplers, which increases the area and reduces the top speed of the receiver. Also the digital circuitry that determines when transitions are and selects the middle one must operate quickly, which can be challenging depending on the digital algorithm used.
2.2 Chip Interconnection Circuit Design Choices

2.2.2 Number of Bits per Transmitted Symbol

In chip to chip interconnections there are usually only two (binary) logic levels transmitted between chips. Instead of transmitting only two logic levels per pin, it is possible to transmit multiple levels, allowing more than one bit to be transmitted per symbol. In the case of 4-PAM [6], 2 bits are transmitted per symbol.

Multi-level signaling has the advantages that the bandwidth is less than for binary transmission if the same number of bits are transmitted. This is an advantage when the data is transmitted over a band-limited channel.

The disadvantages of multi-level signaling is that for \( N \) transmitted voltage levels, there must be \( N - 1 \) comparators in a Nyquist-rate analog to digital (A/D) converter. Multi-level signaling is also more susceptible to noise, since the voltage levels are closer together if the same maximum signal swings are used. One additional challenge with multi-level signaling is to recover the timing of the data. With binary levels and symmetric waveforms only two kinds of transmissions are possible - low to high and high to low. They both cross \( \frac{V_{\text{max}} + V_{\text{min}}}{2} \) in the centre of the transition, which is not the case with multi-level signaling. Two possible transitions are shown in figure 2.5 for 4-PAM signaling. It can be seen in the figure that the point in time where the output crosses \( \frac{V_{\text{max}} + V_{\text{min}}}{2} \) is not the same for all possible transitions as it is for binary signaling. Additional care must be taken to ensure that the proper transitions for timing recovery are present and than they can be detected at the receiver [6].
2.2.3 **Single-Ended/Fully Differential Signaling**

One of the most common questions in transceivers is whether to use single-ended or fully-differential signaling. In fully-differential signaling (such as in the low voltage differential signaling (LVDS) standard [7]) two physical traces are used per output. The data is recovered by measuring the difference in voltage between the two lines at the receiver. In single-ended signaling only one pin is used and the threshold for deciding between a logical "1" or "0" is either transmitted along with the data pins, or is some known value (such as $\frac{V_{max} + V_{min}}{2}$).

Fully-differential signaling uses twice as many chip signal pins as single-ended signaling. It also has better performance than single-ended signaling when common-mode noise is present. This is because if the same noise is injected on both the positive and negative signal lines (which is the case of effects such as power supply noise), the noise does not affect the difference measured between the lines. This becomes more important as lower voltages are used in the signaling. Also the power consumed in the transmitter is the same regardless of what data is transmitted, which results in less switching noise on the power supply rails.
in the transmitter. The drawback of fully differential signaling is that it uses twice as many signal pins.

The design presented in this thesis implements a compromise between these two signaling methods. The data is encoded so that the current in the transmitter across all transmitted lines is constant just as with fully differential signaling, but the number of pins used is not doubled as with differential signaling. The reference voltage is generated from the received data, but the noise performance is not as good as with fully differential signaling. (The encoding is discussed in more detail in section 3.4.)

2.2.4 Global or Individual Line Phase Tracking

When information is transmitted on many pins in parallel the receiver will either independently recover the phase from each line or it will assume the phase on all lines is the same.

When the same phase is assumed over all pins, a sampling clock could be transmitted along with the data.

Assuming all the lines have the same phase reduces the complexity of the receiver but it can cause errors if there is a variation in phase from one line to another, as may be the case if there was a difference in drive strength from one driver to another or if the lengths of the signal traces were different from one line to another.
2.2.5 Unidirectional or Bidirectional Signaling

Data may be sent in only one direction on each line, or it may be sent in both directions on the line simultaneously.

In bidirectional signaling (such as in [8]) the same line is used to both transmit and receive simultaneously.

Bidirectional signaling halves the number of pins required if signaling is required both ways across the interconnection. However, the receiver is more complicated since the effect of the transmitted data must be removed from the received data.

2.3 Previous Work

In this section some approaches to chip to chip interconnection in various papers are summarized.

2.3.1 Oversampling Receivers

In [9] a chip implementing a single serial link is described and a summary of the performance of the chip is shown in table 2.1. It is fabricated in a 0.8μm process and operates at 2.5Gb/s. The chip uses 3× oversampling, so the line is sampled 3 times per bit. It also uses a 1:8 multiplexing so the on-chip clock rate is $\frac{1}{8}$ of the transmit/receive rate. Because of the multiplexing and oversampling there are 24 samplers on the input requiring 24 evenly spaced sampling clock edges. The
2.3 Previous Work

clocks are generated by a 6-stage differential VCO which runs at 311MHz. The 24 desired clocks are generated from the 12 from the VCO by interpolating an additional clock between each of the adjacent clocks from the VCO.

The chip uses a pseudo-differential transmitter which is shown in figure 2.6. Using overlapping pulses, a shorter output pulse can be generated than using a conventional inverter type output [9].

In [4] the algorithm for recovering the data from the oversampled received bits is described. A summary of the performance of the chip is shown in table 2.2. To reconstruct the data from the eight samplers the algorithm looks at the previous eight and the subsequent eight samples. Across those three bytes it is determined between which samples the transitions occur and selects data from the sample farthest from the transition. (The data is delayed to allow comparison to both the previous and subsequent byte.)

The chips in [9] and [4] transmit and receive data at high speeds. However the 8:1 multiplexing and the 3× oversampling that were used increase the area of the receiver because 24 comparators are required on the input line. Also the pseudo differential transmitter (figure 2.6) uses external resistors which would clutter up a board with many chips and many outputs per chip.

In [10] a chip is described which implements four fully duplex single ended serial links. A summary of the performance of the chip is found in table 2.3 and the power consumption is shown in table 2.4. The circuit described is intended to relieve inter-chip congestion in an ATM switch system. (It replaces a 72 wire parallel interface with an eight line serial interface that is transparent to the user.)
There are 4 fully-duplex single ended links on each port. The interconnection uses 50Ω coaxial cables. The data on each line is encoded by an 8B10B code, which converts each 8 bit output into 10 serially transmitted bits, where there are guaranteed transitions and the DC component in the output is reduced. 1:9 multiplexing is done on the signals, where the 9th bit is used for parity and flag purposes.

The package used is a custom designed ceramic BGA with 11 metal layers and impedance matched strip-line connections.

Once again, the size of the circuitry in the described chip is not practical for this application. Each of the receivers have their own PLL and all of the transmitters share another PLL. The encoding used also slows the rate of data transferred because 8 bits are encoded into 10 bits that are transmitted in series.
2.3 Previous Work

<table>
<thead>
<tr>
<th>Supply range @ 2.5Gbps</th>
<th>4.4 - 5.5V</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLL frequency range</td>
<td>80 - 385MHz</td>
</tr>
<tr>
<td>Data rate range</td>
<td>320Mbps - 2.7Gbps</td>
</tr>
<tr>
<td>Min. input amplitude</td>
<td>&lt; 100mV</td>
</tr>
<tr>
<td>Input time constant</td>
<td>~150ps</td>
</tr>
<tr>
<td>Input setup and hold</td>
<td>40ps ± 5ps</td>
</tr>
<tr>
<td>Supply sensitivity of VCO (closed loop)</td>
<td>0.4ps/mV</td>
</tr>
<tr>
<td>Power - (Samplers + PLL)</td>
<td>1W</td>
</tr>
<tr>
<td>Die Size</td>
<td>3mm × 3mm</td>
</tr>
<tr>
<td>(Samplers + PLL)</td>
<td>(0.5mm × 1.5mm)</td>
</tr>
<tr>
<td>Technology</td>
<td>0.8μm MOSIS</td>
</tr>
</tbody>
</table>

Table 2.1: Summary of Chip Described in Paper [9].

2.3.2 Multilevel Signaling

In [11] a single line serial link transmitter that uses 4-PAM differential signaling over co-axial cable is described. The chip uses 5:1 multiplexing and uses a three-tap finite impulse response (FIR) pre-emphasis filter at the transmitter to boost the high frequency components of the transmitted signal to compensate for attenuation in the co-axial cable. The receiver is described in [6] and a summary of its performance is found in table 2.5. The chip was implemented in a 0.3μm CMOS process and operates at 8Gb/s.

A receiver for 4-PAM signaling needs three times as many comparators as one for binary signaling. The area overhead of the multilevel signaling makes it unsuitable for a chip interconnection circuit that transmits data over many pins and which is desired to use the smallest amount of area possible.
### Chapter 2. Background

<table>
<thead>
<tr>
<th>Supply Voltage</th>
<th>2.7 - 4.0V</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. Transmit Rate @3.3V</td>
<td>4.8Gb/s</td>
</tr>
<tr>
<td>Max Receive Rate @3.3V</td>
<td>4.3Gb/s</td>
</tr>
<tr>
<td>Max. Frequency Difference</td>
<td>1MHz (&lt; 10^-9 BER)</td>
</tr>
<tr>
<td>Power @ 4Gb/s</td>
<td>500mA</td>
</tr>
<tr>
<td>Analog (2PLL)</td>
<td>150mA</td>
</tr>
<tr>
<td>Input Samp,Rcv Logic</td>
<td>25mA,10mA</td>
</tr>
<tr>
<td>Transmitter,Xmt Logic</td>
<td>50mA, 10mA</td>
</tr>
<tr>
<td>Parallel Data Driver</td>
<td>90mA</td>
</tr>
<tr>
<td>Min. Input Amplitude</td>
<td>170mV &lt; 10^-9 BER</td>
</tr>
</tbody>
</table>

Table 2.2: Summary of Chip Described in Paper [4].

| Data Rate | 4 × 500MBd (4×50 MBytes/s) Full Duplex |
| Technology | CMOS: 0.8μm (0.5μm L_eff), 4 Metal |
| Die Size | 9.7×9.7mm² (100k Transistors) |
| No. of PLL | 6 |
| Features | ac Coupling, On-Chip Control, Back-Pressure Support, Programmable Interface Timing |
| Test Modes | LSSD, Boundary Scan, Self-Test, Local and Remote Wrap |
| Bit Error Rate | < 10^-15 (for -7dB Link Loss @ 500MHz) |
| Power Dissipation | 3.5W @ 3.6V |

Table 2.3: Summary of Features and Performance for Chip Described in [10].

#### 2.3.3 Bidirectional Transceiver

In [8] a chip is described with 100 bidirectional pins transmitting data at 550Mbps/pin. The chip is manufactured in 0.25μm CMOS. It uses a 1595-pin package out of which 571 pins are used for power and ground. A summary of the characteristics of this chip is shown in table 2.6.

The chip uses a single PLL to generate the on-chip clock. (All of the inputs
Table 2.4: Power Consumption @ 3.6V for Chip Described in [10].

are sampled with the same clock so there is no individual adjustment per pin.)

This chip uses a special package with a large number of pins which would not be practical for the application described in this thesis. As well, the bit transmission rate per pin is slower than what is desired for the application described in this thesis.
### Transmitter Performance

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum transmitter rate</td>
<td>10Gb/s @3.3V, 8Gbps @3V</td>
</tr>
<tr>
<td>Max. eye opening @ 8Gbps</td>
<td>350mV, 110ps (10m cable)</td>
</tr>
<tr>
<td>Max. eye opening @ 10Gbps</td>
<td>200mV, 90ps (10m cable)</td>
</tr>
</tbody>
</table>

### Receiver Performance

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum receive rate</td>
<td>9Gbps @ 3.3V, 8Gbps @ 3V</td>
</tr>
<tr>
<td>Data PLL jitter @ 8Gbps</td>
<td>28ps(p-p), 4ps(rms)</td>
</tr>
<tr>
<td>Data PLL capture range</td>
<td>&gt; 200MHz</td>
</tr>
<tr>
<td>Min. swing to capture lock</td>
<td>±400mV</td>
</tr>
<tr>
<td>Min. swing to maintain lock</td>
<td>±300mV</td>
</tr>
<tr>
<td>Data PLL dynamic</td>
<td>BW ~35MHz, Ph.m. ~50°</td>
</tr>
</tbody>
</table>

### Power Dissipation @ 8Gbps, 3V

<table>
<thead>
<tr>
<th>Component</th>
<th>Power (mW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output driver</td>
<td>220mW</td>
</tr>
<tr>
<td>Analog (2 PLLs)</td>
<td>130mW</td>
</tr>
<tr>
<td>Input samplers and logic</td>
<td>130mW</td>
</tr>
<tr>
<td>Total</td>
<td>1100mW</td>
</tr>
</tbody>
</table>

Table 2.5: Summary of Chip Described in Paper [6].

### Chip Specifications

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effective Gate Length</td>
<td>0.25μm</td>
</tr>
<tr>
<td>Number of Metal layers</td>
<td>5</td>
</tr>
<tr>
<td>Supply Voltage</td>
<td>2.5V(VDD), 1.5V(VDDQ)</td>
</tr>
<tr>
<td>Die size</td>
<td>17.24 × 17.34 mm²</td>
</tr>
<tr>
<td>Total Raw Gates</td>
<td>2.5MG (SOG)</td>
</tr>
<tr>
<td>Number of I/O</td>
<td>1024</td>
</tr>
<tr>
<td>Number of Solder Bumps</td>
<td>3727</td>
</tr>
<tr>
<td>Gate Delay</td>
<td>0.11ns (2-input NAND, Av. Load)</td>
</tr>
</tbody>
</table>

Table 2.6: Summary of Chip Described in Paper [8].
2.4 Thesis Goals

The goal of the chip to chip interconnection circuit is to increase the number of effective pins on a chip. By “effective” it is meant that to the internal on-chip circuit it appears that there are more I/O pins available than are physically on the chip package. The operation of the circuit must be such that to the on-chip circuitry each of the effective I/O pins appear to function as regular I/O pins.

The interconnection circuit must transmit the data accurately (a bit error rate of less that $2 \times 10^{-22}$ is desired for an FPGA emulation system containing 100 chips to transmit data on 100 pins per chip at a rate of 1.6Gbps per pin without error for ten years of continuous operation.) As well, the circuit should be compact so that it can be implemented on an FPGA without adding a large area overhead and the circuit should add as little delay as possible to the transmitted data.

2.5 Phase Locked Loop (PLL) Overview

A PLL is used in the circuit to synthesize the internal clock signals. This section contains a brief summary of how the equations were obtained that were used in section 4.2 and it is by no means complete. A more detailed description of PLLs can be found in [12].

Figure 2.7 shows a top-level model of the PLL. From the figure, the input-output transfer function is found to be:
$$\frac{\phi_{VCO}(s)}{\phi_{reference}(s)} = \frac{K_v K_p Z_F(s)}{s} \frac{s}{1 + \frac{K_v K_p Z_F(s)}{N s}} \quad (2.1)$$

Figure 2.7: VCO Loop Architecture

### 2.5.1 Voltage Controlled Oscillator (VCO)

An ideal VCO generates a periodic output whose frequency ($\omega_{VCO}$) is proportional to the input control voltage:

$$\omega_{VCO} = K_{VCO} v_{control} \quad (2.2)$$

Since frequency is the derivative with respect to time of phase:

$$\phi_{VCO}(t) = \phi_{VCO}(0) + K_{VCO} \int v_{control}(t) dt \quad (2.3)$$
Taking the Laplace transform of both sides, we get:

\[
\frac{\Phi_{\text{VCO}}(s)}{V_{\text{control}}(s)} = \frac{K_{\text{VCO}}}{s}
\]  

(2.4)

### 2.5.2 Phase/Frequency Detector (PFD)

The purpose of the phase detector is to generate a signal whose dc value is proportional to the difference in phase between the two input signals.

In this design a phase-frequency detector (PFD) was used. It has the added benefit over a phase detector that it can lock on to input signals at any frequency. A simple schematic of the PFD is shown in figure 2.8. If examined, it can be seen that the average of “up - down” is proportional to the difference in phase between the reference signal and the feedback signal. The average of the “up - down” signals from the phase detector plotted versus the phase difference in the inputs is shown in figure 2.9.

### 2.5.3 Charge Pump

The charge pump is shown in figure 2.10. It converts the “up-down” output from the PFD into an output current \(I_{\text{CPout}}\). If the amount of current in the current sources is \(I_{\text{CP}}\) then output of the charge pump will be:

\[
I_{\text{CPout}}(t) = \frac{I_{\text{CP}}}{2\pi} (up(t) - down(t))
\]

(2.5)

\[
= K_{P}(up(t) - down(t))
\]
2.5.4 Loop Filter

If the loop filter consists of an impedance of $Z_{LF}(s)$, then the voltage at its output ($V_{control}$) will be:

$$V_{control}(s) = Z_{LF}(s)I_{CPout}(s)$$

(2.6)

2.5.5 Digital Divider

If the divider reduces the output frequency of the VCO ($\omega_{VCO}$) by $N$, then its output frequency ($\omega_{feedback}$) will be:

$$\omega_{feedback} = \frac{\omega_{VCO}}{N}$$

(2.7)
2.5 Phase Locked Loop (PLL) Overview

Figure 2.9: PFD Average Output

Using:

\[ \omega = \frac{d\phi}{dt} \]  \hspace{1cm} (2.8)

and taking the Laplace transform of both sides of equation 2.7 the following is obtained:

\[ \frac{\Phi_{feedback}(s)}{\Phi_{VCO}(s)} = \frac{1}{N} \]  \hspace{1cm} (2.9)
2.6 Delay-Locked Loop (DLL) Overview

The main difference between a DLL and a PLL is that the DLL contains a voltage controlled delay line\(^1\) (VCDL) instead of a VCO. The DLL does not generate a signal on its own, it instead delays an incoming periodic signal until the phase of the feedback signal matches that of the reference signal.

A DLL could be created with all the components in section 2.5 except for the VCDL. Since the output phase of the VCDL is proportional to the control voltage:

\[
\phi(t) = K_{VCDL} v_{control}
\]

Taking the Laplace transform of both sides and rearranging:

\(^1\)The delay line is not necessarily controlled by voltage. Current or a digital control word could be used instead.
\[ \frac{\Phi(s)}{V_{control}(s)} = K_{VCDL} \] (2.11)

Substituting \( K_{VCDL} \) for \( \frac{K_v}{s} \) in the VCO block in figure 2.7 and removing the divider block, we get the following as the input-output transfer function.

\[ \frac{O(s)}{I(s)} = \frac{K_{VCDL}K_pZ_F(s)}{1 + K_{VCDL}K_pZ_F(s)} \] (2.12)
Chapter 3

System and Logic Design

This chapter discusses the system design of the parallel transceiver chip. Section 3.1 contains an overview of the circuit. Section 3.2 discusses the DLL architecture of the receiver. The design of the receiver is discussed in section 3.3. Sections 3.4 and 3.5 are on the encoder and decoder, respectively. The design of the transmitter is contained in section 3.6.

3.1 Parallel Transceiver Circuit Overview

The parallel transceiver circuit (PTC) is designed for high speed data transfer between two separate chips. A general overview of the system is shown in figure 3.1. In the figure, the transmitter, encoder, and receiver enclosed by the dotted line were fabricated on the test chip. The FPGA core on the left in the figure has sixteen 8-bit outputs that are updated at a rate of 100MHz. Those outputs are mul-
tiplexed into two 8-bit wide signals that are clocked into the encoder at a rate of 800MHz. The data is then encoded and transmitted on twelve signal pins between the chips at a rate of 1.6Gbps per pin. In the second chip the timing of the data on each line is tracked and the data is recovered. The data is then resynchronized to the on-chip clock, decoded, and demultiplexed back into sixteen 8-bit wide outputs for use in the second chip core.

Figure 3.1: System Overview

3.1.1 Receiver Overview

The data and clock recovery circuitry contains twelve receivers operating in parallel. One receiver is called the master receiver and the others are slave receivers. A top-level overview of the slave receiver (without details that will be discussed in later chapters) is shown in figure 3.2.

The master/slave configuration was used to save area while trying to maintain phase tracking performance. The master receiver tracks the timing changes in
the input faster than the slave receivers and it distributes the timing changes in the clock signals that it generates. Since each of the slave receivers use those clock signals to generate their sampling clocks, the sampler timing in the slave receivers will track changes in the master receiver. If the timing variation across all the inputs is correlated then the master/slave configuration will allow the slave lines to track timing changes quickly. Each of the slave receivers are simpler than the master receiver and share some circuitry, so they use less area. The clock
generators in each of the receivers generates an internal offset from the clocks that come from the master receiver to account for any pin to pin timing offset between the master line and the slave lines.

One set of inputs to each slave receiver is a four phase fully differential clock which is generated by the master receiver. A delay control word (DCW) is used by the comparator clock generator (see figure 3.2) to select two adjacent phases of the input clock and interpolate between them to generate the clock driving the comparator. The clock generator attempts to adjust the sampling clock so that the input is sampled in the middle of the data eye.

The master receiver has its own circuitry to update its DCW. To save area, the slave receivers all share the circuitry that updates their DCWs. Since timing changes in the master receiver are propagated to the slave receivers, the receivers will still track timing changes quickly that affect all the inputs. The connections between the receivers and DCW updating circuitry is shown in figure 3.3.

### 3.1.2 Encoder Overview

The encoding serves two purposes. The first is to dc-balance each transmitted symbol. This is done by transmitting an equal number of high outputs as low outputs (or logic ones as logic zeros.) This also helps reduce the variation in the current consumption of the current source transmitter. As well, the comparator in the receiver can use the average received voltage as the reference to make a decision which reduces the effect of common-mode noise on the circuit.

The second purpose of the encoder is to add data transitions in the transmitted
3.1 Parallel Transceiver Circuit Overview

In the implemented chip, each 8-bit on chip data word is encoded into a 12-bit word where six output bits are always "1" and six output bits are always "0". In the encoding scheme, each encoded output and its bitwise complement will decode to the same data so a guaranteed transition can be inserted on any output line at any possible signal transition by inverting all of the output bits.

Another feature of the encoding scheme is that if half of the received bits are not "1" it is known that an error has occurred during transition. (An even number of bit errors could occur in such a way that six “1”s and six “0”s are still received.)
3.1.3 Transmitter Overview

The transmitter drives a transmission line that is terminated with a 50Ω resistor. The twelve transmitters are current sources that will either source or sink 8mA of current on the line resulting in a signal swing of 400mV. Because of the encoding, half of the lines will be sourcing current and the other half will be sinking current on each set of transmitted bits. A common mode feed-back (CMFB) circuit is used to keep the average of the voltages over all the transmitted lines at $V_{dc} \over 2$.

3.2 DLL Architecture of the Receiver

Each of the master and slave receivers include digital DLLs that adjust the phase of their generated clocks to match that of the input. They are illustrated in figure 3.4. The receiver periodically makes a decision whether to increase, decrease, or leave unchanged the phase of the clock for its samplers. Because only the sign (positive or negative) of the output and not the magnitude of the phase difference is used in the decision to increment or decrement the timing, the loop is nonlinear. Also, in the locked state the output will “jitter” about the ideal output phase because of the minimum phase step ($K_M$) of the output and because the output phase is updated every clock period.

The rate at which the master loop is clocked ($clk_M$ in figure 3.4) is much faster than the rate each slave counter is clocked at ($clk_L$ in figure 3.4) because each of the slave receivers shares the same slave DCW logic.

The response to a phase step of the master loop is shown in figure 3.5. The
3.2 DLL Architecture of the Receiver

The loop updates its output phase in increments ($K_M$ and $K_L$ in figure 3.4). The update time for the master loop ($T_M$ in figure 3.5) is determined by the minimum decision time of the logic that chooses the next phase blender state.

The locked behavior of the slave lines is shown in figure 3.6. The master loop alternates around the two states closest to the input phase with a period of $2T_M$. With a period of $T_L$, the slave loops update their output phase. A change in the master loop will also affect the slave loop. In the worst case, this could make the slave loop output phase error to be $K_M + K_L$. 
Chapter 3. System and Logic Design

3.3 Receiver Design

The goals of the design were for high speed and efficient area use. The solution for this was the receiver design shown in figure 3.7. High speed was attained by using comparators in parallel with different comparators sampling adjacent bits in the middle of the data eyes and at the signal transitions. To save area, all of the slave receivers share the logic that updates the settings for the phase selector and interpolator. The state flip-flops in figure 3.7 are updated by the DCW updating logic as shown in figure 3.3.

3.3.1 Timing Recovery

The timing on each data line into the chip is recovered separately by each receiver. The receiver recovers the timing in the channel by sampling the data once in the middle of the eye and once at what is expected to be the transition. If there was a transition and the sampling is early (see figure 3.8.) then the “edge” sample will be the same as the previous sample. If the sampling is late the “edge” sample
will be the same as the next sample. If the “edge” sampler in the receiver were sampling exactly in the middle of the data transition, the probability of early or late transitions should both be $\frac{1}{2}$. The sampling is done by sampling comparators that are shown in figure 3.9.

Greater detail of each of the receivers is shown in figure 3.10. Each one consists of four comparators in parallel, each sampling at a rate of $\frac{1}{2T_{bit-time}}$ with samples spaced $\frac{T_{bit-time}}{2}$ apart. The parallelism reduces the rate at which the comparators must run at the cost of increased area and complexity. However, the requirements of the input stage are unchanged in that they must be able to acquire a sample of the input signal in $\frac{T_{bit-time}}{2}$. 
3.3.2 Clock Generator PLL

The 4 phase differential clocks are initially generated by a PLL in which the VCO consists of 4 fully differential delay elements. (The VCO is shown in figure 3.11.) Because of the parallel comparators at the input, the frequency of oscillation of the VCO is $\frac{1}{2T_{ba-time}}$. The VCO generates 4 phase differential clocks, so the phases at the output are separated by $\frac{2\pi}{8}$. 
3.3 Receiver Design

3.3.3 Phase Interpolator

The phase selector and interpolator (seen in figure 3.7) operates by selecting two of the adjacent differential clocks (spaced $\frac{2\pi}{8}$ radians apart) and then interpolating between the two clocks to generate clocks for the comparators whose phases lie between that of the input signals. The interpolator has 32 possible settings so with
the eight possible available phases the minimum phase separation is \( \frac{2\pi}{256} \) radians. Since the period of the on-chip clock is \( 2 \times T_{\text{bit-time}} \), the minimum resolution of the clock generator is \( \frac{T_{\text{bit-time}}}{128} \). Because the maximum phase error in the slave receivers is twice the minimum phase resolution (see figure 3.6), the maximum timing error in the slave receivers will be \( \frac{T_{\text{bit-time}}}{64} \).

With an increased number of settings the output phase separation decreases which reduces the timing jitter in the receiver. However the complexity of the circuit increases and the time that the receiver takes to adjust to a change in the input timing will increase.

### 3.3.4 Receiver State Updating

The state of the tap settings on the phase selector and interpolator is stored in flip-flops in each receiver (see figure 3.7.) In each receiver, the number of early and late samplings is monitored and an up or down signal is generated to indicate whether the sampler should sample earlier, later, or remain at the same setting. The master receiver has its own dedicated logic to update the settings of the interpolator (referred to as the DCW in figure 3.2) so it is updated at a frequency limited only by the delay of the updating circuitry. All of the slave receivers share a single set of logic to update their delay settings. The updating logic reads the current delay settings and the state of the up and down bits, and generates new settings for the interpolator. The frequency of DCW updating is determined by the number of slave receivers and by the circuitry. Since there are 11 slave receivers, if the updating period per receiver is \( T_{\text{US}} \), then the time between updates
3.3 Receiver Design

is $11 \times T_{US}$.

3.3.5 Transition Counter

In both the master and the slave lines transitions are guaranteed to be in the input data at certain points in time. The transition counter in figure 3.2 is used to count the data transitions at both of the “edge” samplers when the guaranteed transitions should be occurring. This is done because the receivers will lock on to either edge and the transition counter sets a multiplexor on each receiver to make sure that the two output bits are sent to the output on the proper lines.

3.3.6 Global Clock Synchronization

Each of the receivers adjust their sampling timing to compensate for any pin-to-pin timing differences. To be useful on-chip the incoming data signals must be synchronized to an on-chip system clock. The way that this could be done is illustrated in figure 3.12. In the figure, $clk_{\text{global}}$ is the system clock that the data is desired to be synchronized with. The inputs to the synchronization circuit are the data ($data_s$) and clock ($clk_s$) outputs from the receiver. The timing of the clock signal in relation to the data signal is the same as that in a regular synchronous system - the data signal could be latched into a flip-flop using the clock signal. If the clock synchronizer adjusts the delay of the clock signal from the receiver so that it is in phase with the global clock and also delays the data by the same amount, the data will be synchronized to the global clock. In figure 3.12,
the delay through the voltage controlled delay line (VCDL) is adjusted so that \texttt{clk\_resync} is in phase with \texttt{clk\_global}. If \texttt{data\_s} passes through a delay line with the same delay, then \texttt{data\_resync} will be synchronized to \texttt{clk\_global}.

The phase detector in the figure (labeled \textit{PD}) can be implemented with a dynamic flip-flop which has low set-up time. In a similar manner to the way the delay control word is updated periodically in the various slave receivers, the delay control setting for the voltage-controlled delay line can be updated periodically by a logic which is shared between all of the synchronizers. The delay settings of the delay line can be more coarse that those of the clock generator because the resynchronizer only has to make sure that the data is retimed so that the setup and hold times of flip-flops it feeds into are met.

![Figure 3.12: Global Clock Synchronization](image)

### 3.4 Encoder

#### 3.4.1 Reasons for Encoding

As discussed in section 3.1.2 the encoder has the following requirements:
1. Each symbol is dc-balanced (so the common-mode of all the signals can be used as the reference voltage at the receiver.)

2. Transitions can be inserted at any pin between any desired bits (so that there are enough transitions for the receiver to be able to track the input timing properly.) This also allows the master line to have a higher frequency of transitions.

3. Latency is kept as low as possible.

Requirement 1 can be satisfied by transmitting on an even number of pins with one half of the bits transmitting a logical “1” and the other half transmitting a logical “0”.

Requirement 2 can be satisfied by having two possible transmitted outputs for each different input, where each of the encodings are bitwise complements of each other (for example “1111100000” and “0000011111”). In this case, a transition could be inserted at any pin at any transition time by merely inverting all the output bits if there was not previously a transition. A simple way to implement the encoder for this is to use one bit (which will be called the “flip” bit) as an indicator of which encoding is used. Therefore instead of encoding the input into an \( x \) bit output where \( \frac{x}{2} \) bits are “1”, the input is encoded into an \( x - 1 \) bit output where \( \frac{x}{2} - 1 \) bits are “1”. Then the outputs are left unchanged and the flip bit with a value of “1” is added, or the outputs are all inverted and a “0” flip bit is added.

Since the input to the encoder is an 8-bit word, there are \( 2^8 = 256 \) possible words to be encoded. For \( x \) signal bits, the number of combinations where \( y \) of
the bits are "1" and \( x - y \) are "0" is given by:

\[
\binom{x}{y} = \frac{x!}{y!(x-y)!}
\]  

(3.1)

For a nine bit output with five "1"s and an eleven bit output with 5 "1"s we get:

\[
\binom{9}{4} = 126
\]

(3.2)

\[
\binom{11}{5} = 462
\]

(3.3)

Equation 3.2 shows that nine output bits are not sufficient to encode all of the \( 2^8 \) possible input combinations, however from equation 3.3 it is seen that eleven bits are sufficient.

Latency is kept low (requirement 3) because the encoder output is created by combinational logic from the current inputs, with only a possible inverting of all the bits being necessary depending on the previous outputs.

\subsection{3.4.2 Implementation of Encoder}

The encoding could be done using a monolithic 8-bit to 12-bit encoder. This would result in a large, complex encoder. Instead the design was be broken up into smaller encoders, which were easier to implement. This was done by looking at the top bits of the input and then selecting the appropriate output. (This is summarized in figure 3.13 and is described in more detail below.) Each encoded word contains eleven bits, where five of the bits are "1" and six are "0". The
output is converted to a 12-bit word by appending the flip bit. As explained in section 3.4.1, each transmitted 12-bit word and its bitwise complement decode to the same number. The appended bit can be used at the receiver to determine if all the bits should be inverted before they are decoded.

In the following description "X" represents a don’t-care, bits are numbered with the least significant bit as “0” and each more significant bit number incremented by one (for example, in “110”, bit 0 is “0” and bits 1 and 2 are both “1”.)

Also, (x:y) refers to an x bit number where y bits are “1” and x – y bits are “0”.

A. **Input is “1XXXXXXXX”**

There are 128 codes where bit 7 is “1”. The 16 possible numbers from bits 6 through 3 can be represented by a (6:3) code (which has 20 combinations) and the 8 combinations of bits 2 through 0 can be represented by a (5:2) code (which has 10 possible combinations.) This is summarized in figure 3.13.

B. **Input is “01XXXXXXXX”**

There are 64 codes that start with “01”. The 8 combinations of bits 5 through 3 can be represented by a (6:2) code and the 8 combinations of input bits 2 through 0 can be represented in a (5:3) code (which has 10 possible combinations.) (see figure 3.13)

C. **Input is “001111XX”**

The 4 codes that start with “001111” are represented by setting output bits 5 through 1 to “00000” and encoding the four bits into a (6:3) code. (see figure 3.13)
D. **Input is "00XXXXXXXX (but not "001111XX")**

There are 60 codes that start with "00" and where bits 5 through 2 are not "1111". The 15 allowed combinations of bits 5 through 2 are encoded into a (6:4) code (which has 15 possible combinations) and the remaining 2 low input bits are encoded into a (5:1) code (see figure 3.13.)

Several simplifications can be made with the above encoding when implementing it in hardware. The (5:2) encoding in part A and the (5:3) encoding in part B can be implemented with the same encoder since the outputs can be simply inverted in part B and the inputs are the same.

The (6:2) encoder in part B was implemented with a copy of the (5:2) encoder mentioned above with a "0" added to make the output 6 bits. (The 10 output combinations from the (5:2) code is sufficient to encode 3 input bits.)

The (6:3) encoder in part C and the (5:1) encoder in part D can be implemented with the same (4:1) encoder. In part C two "1" bits are appended to make the (4:1) code a (6:3) code. In part D a "0" is appended to make the code a (5:1).

With these simplifications, the encoder can be created with one 4-bit to (6:3) encoder, one 4-bit to (6:4) encoder, two identical 3-bit to (5:2) encoders and one 2-bit to (4:1) encoder. In addition a 12-bit wide 4:1 multiplexor and a small amount of glue logic is required. As well, a state machine is required to keep track of which output line should have guaranteed transitions. Twelve flip-flops are required to save the previous output and if a transition is required then all the output bits can be inverted.

An algorithm for generating balanced codes with parallel encoding and de-
3.4 Encoder

coding can be found in [13]. This algorithm was not known to the author of this thesis when the encoder was created and was not used.

3.4.3 Transitions on Output Lines

The encoder inserts transitions on the output lines so that the signal timing can be recovered at the receiver. For fast tracking, the master line has a guaranteed tran-
sition every second bit. On the other possible transition, the guaranteed transition will occur on one of the slave lines. The guaranteed transition will cycle through the slave lines with six transitions occurring every second bit on one line, then six transitions occurring every second bit on the next line.

The receiver also needs to know which slave line is having the guaranteed transitions. To indicate when the first of the slave lines is starting their series of transitions, no transition will occur when it should on the master line.

### 3.5 Decoder

The decoder was not implemented on the fabricated chip. However the encoder is of about the same complexity as the encoder. Just as with the encoder, the decoder can be implemented with smaller decoders than with a single 12-bit to 8-bit decoder. In fact, the reverse of what was explained in section 3.4.2 can be done by simply using the corresponding decoder instead of the encoder. (For example, using a (6:3) to 4 bit decoder in place of the 4 bit to (6:3) encoder.) Therefore the decoder can be created with one (6:3) to 4 bit decoder, one (6:4) to 4 bit decoder, two identical (5:2) to 3 bit decoder and one (4:1) to 2 bit decoder. Decoding the received data can be done by the following algorithm:

1. If bit 0 is "0" then invert all the received bits

2. If bits 5 through 1 are "00000" (can be detected by inverting and ANDing the bits together) then the encoding was done as in case C in figure 3.13.
3. Otherwise, if the exclusive or (XOR) of bits 5 through 1 is zero then the encoding was done as in case A in figure 3.13. (This is the only remaining case with an even number of “1”s)

4. Otherwise, if there is only one “1” in bits 5 through 1, then the encoding was done as in case D in figure 3.13.

5. Otherwise, the encoding was done as in case B in figure 3.13.

3.6 Transmitter

The transmitter performs the multiplexing and transmitting of the encoded bits. A schematic of the transmitter is shown in figure 3.14. In the implemented chip two input bits are multiplexed serially onto a single output. A common-mode feed-back circuit regulates the average of all the transmitted bits. (Six “1”s and six “0”s are always transmitted so the average voltage of all the outputs should always be the same.)
Figure 3.14: Transmitter
Chapter 4

VLSI Implementation

This chapter discusses the circuit implementation of the parallel transceiver chip. Section 4.1 is on the symmetric load delay cell that is used in many of the cells in the circuit. Section 4.2 discusses the phase locked loop. Section 4.3 is about the receiver and its components. Section 4.4 discusses the transmitter. The implementation of the encoder is the subject of section 4.5. The layout view of the implemented chip is shown in section 4.6 and the included test circuitry is discussed in section 4.7.

4.1 Symmetric Load Delay Cell

One of the main components of the circuit is the symmetric load delay cell for which a schematic is shown in figure 4.1. The transistor sizes are shown in table 4.1. (The symmetric load delay cell is described in [14].) Delay cells
with linear loads are the best for rejecting supply noise. However, it is impractical to have poly resistor loads on chip because of their size. MOS transistor loads are generally nonlinear over a wide range of inputs and when diode-connected (with the gate and drain shorted) the varying conductance causes the rise and fall to be asymmetric. Asymmetric loads will convert common-mode noise into differential-mode noise [14] as well as increasing low-frequency noise up-conversion into phase noise [15].

A single symmetric load is shown in figure 4.2. The current versus voltage (and also impedance versus voltage) for the load is symmetric around the centre voltage around which the load voltage oscillates.

![Diagram of Delay Cell]

*Figure 4.1: Delay Cell*

If $V_{\text{ctrl}} > 2V_t$, then the current in the load (with the polarities shown in fig-
4.1 Symmetric Load Delay Cell

<table>
<thead>
<tr>
<th>Transistor(s)</th>
<th>Width ($\mu$m)</th>
<th>Length ($\mu$m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1, M2</td>
<td>5</td>
<td>0.35</td>
</tr>
<tr>
<td>M3, M4, M5, M6</td>
<td>6.4</td>
<td>0.35</td>
</tr>
<tr>
<td>M7</td>
<td>32</td>
<td>0.4</td>
</tr>
</tbody>
</table>

Table 4.1: Transistor Sizes for figure 4.1

Figure 4.2: Symmetric Load

Figure 4.2 is as follows.

For $0 < V_{in} < |V_i|$

$$I_{DS} = \frac{1}{2} \mu C_{ox} \frac{W}{L} [2(V_{ctrl} - |V_i|)V_{in} - V_{in}^2]$$  \hspace{1cm} (4.1)

For $|V_i| < V_{in} < (V_{ctrl} - |V_i|)$

$$I_{DS} = \frac{1}{2} \mu C_{ox} \frac{W}{L} [(V_{in} - |V_i|)^2 + 2(V_{ctrl} - |V_i|)V_{in} - V_{in}^2]$$  \hspace{1cm} (4.2)

For $(V_{ctrl} - |V_i|) < V_{in} < V_{ctrl}$

$$I_{DS} = \frac{1}{2} \mu C_{ox} \frac{W}{L} [(V_{in} - |V_i|)^2 + (V_{ctrl} - |V_i|)^2]$$  \hspace{1cm} (4.3)
The current ($I_{DS}$) versus voltage ($V_{in}$) is plotted in figure 4.3.

The simulated I-V characteristic of the actual device is plotted in figure 4.4. The difference from the ideal plot in figure 4.3 is because of second order effects such as short channel effects.

Figure 4.3: Ideal Symmetric Load I-V Characteristic
4.2 Phase Locked Loop

The loop is shown in figure 4.5. More circuit details of the implementation are shown in figure 4.6.

Figure 4.5: VCO Loop Architecture
4.2.1 Voltage Controlled Oscillator (VCO)

The voltage controlled oscillator is composed of four differential delay cells (shown in figure 4.1) that are connected in the manner shown in figure 4.7. The layout of the VCO is shown in figure 4.8. To reduce the differences in delay because of the capacitive load or process variations on each cell, the layout of the four delay cells was done in a common-centroid arrangement and with the same amount of metal attached to each of the output lines.
4.2 Phase Locked Loop

Figure 4.8: Voltage Controlled Oscillator Layout

The $V_{ctrl}$ and $V_{cs}$ voltages on all the delay cells are supplied by the bias circuit [14] which is shown in figure 4.9. The sizes of the transistors used in the bias circuit are listed in table 4.2. The frequency of oscillation of the VCO is adjusted by changing the voltage of $V_{in}$ in the bias circuit. The frequency of oscillation of the VCO versus $V_{dd} - V_{in}$ is plotted in figure 4.10.

4.2.2 Phase Detector

A schematic of the phase detector is shown in figure 4.11 and the transistor sizes are listed in table 4.3. This phase detector is similar to two dynamic flip-flops and has a minimum frequency of operation. The reset pulses occur at a rate that is the minimum of the two inputs ($\text{ref}$ or $\text{fb}$). The reset pulse is delayed so that
Figure 4.9: Bias Circuit for the VCO Delay Cell

<table>
<thead>
<tr>
<th>Transistor(s)</th>
<th>Width (µm)</th>
<th>Length (µm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1,M5,M9,M10,M14</td>
<td>32</td>
<td>0.4</td>
</tr>
<tr>
<td>M2,M6,M15</td>
<td>5</td>
<td>0.35</td>
</tr>
<tr>
<td>M3,M4,M7,M8</td>
<td>6.4</td>
<td>0.35</td>
</tr>
<tr>
<td>M7,M16</td>
<td>4</td>
<td>0.35</td>
</tr>
<tr>
<td>M11,M12,M13</td>
<td>25.6</td>
<td>0.35</td>
</tr>
<tr>
<td>M16</td>
<td>12.8</td>
<td>0.35</td>
</tr>
<tr>
<td>M17,M18</td>
<td>0.8</td>
<td>10</td>
</tr>
<tr>
<td>M19</td>
<td>1</td>
<td>0.35</td>
</tr>
<tr>
<td>M20</td>
<td>6.4</td>
<td>0.35</td>
</tr>
</tbody>
</table>

Table 4.2: Transistor Sizes for figure 4.9.

even under near-lock conditions both the *up* and *down* signals will be both high for a short period of time. This helps to reduce the "dead zone" of some PLLs, where small differences in phase are not reacted to because the output of the phase detector occurs for too short a time to cause a change in the charge pump.
4.2 Phase Locked Loop

4.2.3 Charge Pump

The schematic of the charge pump is shown in figure 4.12 and the transistor sizes are in table 4.4. In the schematic, the purpose of transistors $M3$ and $M4$ and $M9$ through $M13$ are to keep the voltage at the output of the current source that is not connected to the output close to the voltage of the output $cpout$. If this is not done, the voltage at the current source not connected to the output will go to near the supply voltage, causing error because of the injected charge when the output is switched to be connected to that current source.

4.2.4 Divider

The divider used is a digital divider that is preceded by a differential-to-single-ended signal converter. An overview of the divider is shown in figure 4.13. The
differential-to-single ended converter is shown in figure 4.15 and the transistor sizes are in table 4.6. Following the differential-to-single ended converter is a digital divider whose output frequency is $\frac{1}{8}$ of the its input frequency. The flip-flops used in the divider are shown in figure 4.14 and the transistor sizes are in table 4.5.
4.2 Phase Locked Loop

<table>
<thead>
<tr>
<th>Transistor(s)</th>
<th>Width (μm)</th>
<th>Length (μm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1,M10</td>
<td>2.00</td>
<td>0.35</td>
</tr>
<tr>
<td>M2,M3,M11,M12</td>
<td>4.50</td>
<td>0.35</td>
</tr>
<tr>
<td>M4,M5,M13,M14</td>
<td>2.85</td>
<td>0.35</td>
</tr>
<tr>
<td>M6,M15</td>
<td>3.20</td>
<td>0.35</td>
</tr>
<tr>
<td>M7,M16</td>
<td>4.00</td>
<td>0.35</td>
</tr>
<tr>
<td>M8,M17</td>
<td>6.40</td>
<td>0.35</td>
</tr>
<tr>
<td>M9,M18</td>
<td>0.80</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 4.3: Transistor Sizes for figure 4.11.

4.2.5 Loop Filter Parameters

From the loop shown in figure 4.5, the closed-loop transfer function is:

\[
H(s) = \frac{s}{K_v K_p Z_F(s)} \left(1 + \frac{s}{K_v K_p Z_F(s)}\right) \quad (4.4)
\]

Where \(K_v\) and \(K_p\) are the VCO and charge pump gains, respectively.

If \(Z_F(s) = R + \frac{1}{sC}\) a (simple low-pass RC filter) then:

\[
H(s) = \frac{K_v K_p (R + \frac{1}{sC})}{1 + \frac{K_v K_p (R + \frac{1}{sC})}{sN}}
\]

\[
= N \cdot \frac{s K_v K_p R + K_v K_p}{s^2 + s \left(\frac{K_v K_p R}{N} + \frac{K_v K_p}{CN}\right)}
\]

\[
= N \cdot \frac{2\zeta \omega_n s + \omega_n^2}{s^2 + 2\zeta \omega_n s + \omega_n^2} \quad (4.5)
\]
In equation 4.5, the following substitutions were made:

$$\omega_n = \sqrt{\frac{K_V K_{CP}}{CN}}$$  \hspace{2cm} (4.6)

$$\zeta = \frac{R}{2} \sqrt{\frac{K_V K_{CP}}{N}}$$  \hspace{2cm} (4.7)

In a differential VCO containing $n$ stages, the oscillation frequency is:

$$f_{oscillation} = \frac{1}{2n \times T_{stagedelay}}$$  \hspace{2cm} (4.8)
### 4.2 Phase Locked Loop

<table>
<thead>
<tr>
<th>transistor(s)</th>
<th>width ((\mu m))</th>
<th>length ((\mu m))</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>3.6</td>
<td>0.5</td>
</tr>
<tr>
<td>M2, M4</td>
<td>3.6</td>
<td>0.35</td>
</tr>
<tr>
<td>M3, M5</td>
<td>2</td>
<td>0.35</td>
</tr>
<tr>
<td>M6</td>
<td>9</td>
<td>0.5</td>
</tr>
<tr>
<td>M7</td>
<td>4.3</td>
<td>0.5</td>
</tr>
<tr>
<td>M8</td>
<td>4.3</td>
<td>0.35</td>
</tr>
<tr>
<td>M9</td>
<td>1</td>
<td>0.35</td>
</tr>
<tr>
<td>M10</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>M11</td>
<td>3.6</td>
<td>0.5</td>
</tr>
<tr>
<td>M12</td>
<td>9</td>
<td>0.5</td>
</tr>
<tr>
<td>M13</td>
<td>4</td>
<td>0.4</td>
</tr>
<tr>
<td>M14</td>
<td>9</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Table 4.4: Transistor Sizes for figure 4.12.

![Figure 4.13: Digital Divider Schematic](image)

Since the delay cell behaviour is similar to an RC delay [16] and defining as in [16]:

\[
t_{\text{delay}} = R_{\text{eff}} C_{\text{eff}}
\]

\[
= \frac{C_{\text{eff}}}{g_m}
\]  \hspace{1cm} (4.9)
Figure 4.14: Dynamic Flip-Flop Schematic

<table>
<thead>
<tr>
<th>transistor(s)</th>
<th>width (µm)</th>
<th>length (µm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1, M10</td>
<td>2</td>
<td>0.35</td>
</tr>
<tr>
<td>M2, M3</td>
<td>4.5</td>
<td>0.35</td>
</tr>
<tr>
<td>M4, M5, M7, M8</td>
<td>2.85</td>
<td>0.35</td>
</tr>
<tr>
<td>M6, M9, M11</td>
<td>3.2</td>
<td>0.35</td>
</tr>
</tbody>
</table>

Table 4.5: Transistor Sizes for figure 4.14.

In each delay cell the current through the current source is:

\[
I_{DS} = \frac{1}{2} \mu C_{ox} \frac{W}{L} (V_{ctrl} - V_t)^2
\]

\[
= \frac{1}{2} \beta (V_{ctrl} - V_t)^2
\]

(4.10)

The transconductance is:

\[
g_m = \frac{\partial I_{DS}}{\partial V_{ctrl}}
\]

\[
= \beta (V_{ctrl} - V_t)
\]

\[
= \sqrt{2\beta I_{DS}}
\]

(4.11)
The oscillation frequency is:

\[
F = \frac{1}{2n t_{\text{delay}}} = \frac{\beta (V_{\text{ctrl}} - V_t)}{2n C_{\text{eff}}} = \frac{\beta (V_{\text{ctrl}} - V_t)}{C_B} = \sqrt{2\beta I_{\text{DS}} C_B} \quad (4.12)
\]

The VCO gain, \(K_V\) is the derivative of the oscillation frequency with respect
Chapter 4. VLSI Implementation

to $V_{ctrl}$:

$$K_V = \left| \frac{dF}{dV_{ctrl}} \right|$$

$$= \frac{\beta}{C_B} \quad (4.13)$$

From equation 4.13, the VCO gain, $K_V$ is constant and does not depend on the current in the delay cells.

To avoid the use of an additional bias circuit, the same bias voltage was used for the charge pump. From equation 4.6 and using equation 4.12 we get:

$$\frac{\omega_n}{\omega_{ref}} = \frac{1}{2\pi F_{ref}} \sqrt{\frac{K_{CP}K_V}{NC}}$$

$$= \frac{1}{2\pi} \sqrt{\frac{C_B}{2\beta I_{DS}}} \sqrt{\frac{K_{CP}K_V}{NC}}$$

$$= \frac{1}{2\pi} \sqrt{\frac{C_B}{2\beta I_{DS}}} \sqrt{\frac{x_2 I_{DS} K_V}{\sqrt{2\pi}}}$$

$$= \frac{1}{2\pi} \sqrt{\frac{C_B x K_V}{NC 2\pi}} \quad (4.14)$$

In equation 4.14 it is seen that the ratio of $\frac{\omega_n}{\omega_{ref}}$ depends only on parameters that do not change with the operating frequency of the loop. However, from equation 4.7 it is seen that the damping ratio, $\zeta$ varies with $\sqrt{I_{CP}}$. Therefore care must be taken to make sure the loop is not too under-damped (causing ringing) or over-damped (causing slow tracking of changes in frequency) over all the ranges in charge pump current. This shows that the bandwidth relative to the frequency of
4.3 Receiver

the loop is constant. However the damping frequency $\zeta$ will vary and this must be accounted for in the filter design.

From figure 4.10, the gain of the VCO ($K_V$) found from simulation using a netlist extracted (including parasitics) from the layout is about $3.35 \times 10^9 \text{rad/sV}$.

4.3 Receiver

4.3.1 Sampling Comparator

There are four comparators in parallel, each sampling the channel at $\frac{T_{bit-time}}{2}$. Because of the de-multiplexing by two of the channel, each comparator may run at half of the channel speed reducing the requirements of the comparator. However, the fundamental frequency of the input is unchanged so they must be able to track the input. There are four samplers in parallel, each sampling at a rate that is $\frac{1}{2}$ of the input frequency so comparator must make a decision within that time. Because of these points, a sampling comparator like that used in figure 4.16 was used.

If the bit-rate of the input is 1.6 Gb/s, then the bit period on the input is 625ps. Because each bit is sampled in the centre of the eye and at the transition, the time per sampler is only 312.5ps. If the single-pole time constant is $\tau$, it is chosen that

$$6\tau = 312.25 \text{ps} \quad (4.15)$$
Which gives:

$$\tau = 52\, ps$$ (4.16)

If the equivalent impedance of the input transistors (M1 and M9 in figure 4.16) is:

$$R \approx \frac{1}{\mu C_{ox} \frac{W}{L} (V_{gs} - V_t)}$$ (4.17)

The highest impedance occurs when the input voltage is at the high end of its range (2.05V). By equation 4.17 approximate impedance is then 227Ω.

Substituting values for $R$ and $\tau$, we get:

$$C = \frac{\tau}{R} = \frac{52\, ps}{227\, \Omega} = 229\, fF$$ (4.18)

Transistor sizes for figure 4.16 are shown in table 4.7 and the capacitor sizes are shown in table 4.8. Transistors M9 and M10 are to cancel out the charge injection from transistors M1 and M9.

An approximation of the channel charge held in transistors M1 and M9 is given by equation 4.19.
4.3 Receiver

\[ Q_{CH} = W L C_{ox}(V_{gs} - V_t) \]  \hspace{1cm} (4.19)

Assuming that half of the charge held in the channel goes to each of the source and the drain when the transistor turns off, the perturbation on each side is given by equation 4.20.

\[ \Delta V = \frac{Q_{CH}}{2C} \]  \hspace{1cm} (4.20)

On the comparator side of the switch there is a transistor that is half the width of the switch transistor which is meant to try to cancel out the injected charge. By equation 4.19 the charge absorbed by the channel in the charge injection compensating transistor is \( \frac{Q_{cm}}{2} \) which is equal to the injected charge. On the other side of the switch, there is the pad and inputs to other samplers, which has a relatively large capacitance and should reduce the effects of the injected charge. Also, the next sample is not made for \( \frac{T_{hit-time}}{2} \) so the output should settle by then.

4.3.2 Phase (Delay) Interpolator

A general overview of the phase interpolator is shown in figure 4.18. The MUX and BLENDER cells in the figure are shown in more detail in figures 4.19 and 4.20 respectively.
The interpolator first selects which two adjacent phases to interpolate between and which polarity of phases to select. Then the blender interpolates between the two signals. There are 32 different blender settings, equivalent to a 5 bits of a binary weighted word. However, in the blender the largest weight of current source that is switched is $\frac{4}{32}$ of the total current source width. This is done to minimize the perturbation of the output during the time that the inputs of the current source are changing.
4.3 Receiver

<table>
<thead>
<tr>
<th>Transistors</th>
<th>Width (μm)</th>
<th>Length (μm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1,M9</td>
<td>20</td>
<td>0.35</td>
</tr>
<tr>
<td>M2,M10</td>
<td>10</td>
<td>0.35</td>
</tr>
<tr>
<td>M3,M11</td>
<td>4</td>
<td>0.35</td>
</tr>
<tr>
<td>M4,M12</td>
<td>8</td>
<td>0.35</td>
</tr>
<tr>
<td>M5,M13</td>
<td>16</td>
<td>0.35</td>
</tr>
<tr>
<td>M6,M14</td>
<td>16</td>
<td>0.35</td>
</tr>
<tr>
<td>M7,M8</td>
<td>24</td>
<td>0.35</td>
</tr>
</tbody>
</table>

Table 4.7: Transistor Sizes for figure 4.16.

<table>
<thead>
<tr>
<th>Capacitors</th>
<th>Capacitance Value (fF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1,C2</td>
<td>229</td>
</tr>
</tbody>
</table>

Table 4.8: Capacitor Sizes for figure 4.16.

The settings to the multiplexors are only changed when the interpolator settings on the output of those multiplexors have a weighting of zero. Therefore changing the multiplexor settings have no negative effect on the output.

There are 32 different settings in the blender and 8 different phases that can be selected, making a total of 256 different phases, or a minimum phase resolution of $\frac{2\pi}{256}$. 
Figure 4.18: Phase Selector and Blender

Figure 4.19: Delay Cell Multiplexer
Figure 4.20: Delay Cell Multiplexer Interpolator
4.3.3 Global Clock Synchronizer

Although the clock synchronization was not implemented in the fabricated chip, this section describes a possible circuit implementation of the components in found in figure 3.12.

The digitally controlled delay line could be composed of elements such as that shown in figure 4.21. If many of these cells are connected together in series then the total delay through the delay line can be adjusted by changing $V_{delayControl}$. A circuit such as that shown in figure 4.22 could be used to generate the delay control voltage from the digital input. A dynamic flip-flop, such as that shown in figure 4.14 could be used as the phase detector.

![Figure 4.21: Synchronizer Delay Cell](image-url)
4.3 Receiver

4.3.4 Layout of a Slice of the Slave Receiver

The layout of a single slice of a slave receiver is shown in figure 4.23. The layout is designed so that it can be arrayed with other slices. At the bottom are the enabling signals and busses for updating the state of the phase blender in the receiver. The high speed clock is buffered before it is fed through to the next slice and at the input of the phase blender.

4.3.5 Layout of the Master Slice

The layout of the master slice is shown in figure 4.24. It contains a receiver similar to that in the slave slices. It also contains the PLL that locks on to the external reference clock signal and generates the 8 phase internal clock signals. After the PLL is a phase interpolator that generates the 8 phase clock signals that are distributed to all of the slave receivers and are used by the master receiver.
Figure 4.23: Layout of Slave Slice
Figure 4.24: Layout of Master Slice
4.4 Transmitter

The transmitter schematic is shown in figure 4.25 and the transistor sizes are shown in figure 4.9. Transistors $M1, M2$ and $M5, M6$ are current sources that will either source 8mA of current into the output, or draw it out from the output. The transmitter also serves as a 2-to-1 multiplexor. There is a common-mode feedback circuit to keep the common mode of all the outputs constant. The bias circuit (from [17]) is shown in figure 4.26 and the transistor sizes for the figure are in table 4.10.

![Transmitter Schematic](image)

Figure 4.25: Transmitter Schematic
Table 4.9: Transistor Sizes for figure 4.25.

<table>
<thead>
<tr>
<th>Transistor(s)</th>
<th>Width (μm)</th>
<th>Length (μm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>161</td>
<td>0.35</td>
</tr>
<tr>
<td>M2</td>
<td>161</td>
<td>0.40</td>
</tr>
<tr>
<td>M3</td>
<td>100</td>
<td>0.35</td>
</tr>
<tr>
<td>M4</td>
<td>250</td>
<td>0.35</td>
</tr>
<tr>
<td>M5</td>
<td>505</td>
<td>0.40</td>
</tr>
<tr>
<td>M6</td>
<td>505</td>
<td>0.35</td>
</tr>
<tr>
<td>M7</td>
<td>2</td>
<td>0.35</td>
</tr>
<tr>
<td>M8</td>
<td>2</td>
<td>0.40</td>
</tr>
<tr>
<td>M9,M12</td>
<td>16</td>
<td>0.35</td>
</tr>
<tr>
<td>M10,M13</td>
<td>3.3</td>
<td>0.40</td>
</tr>
<tr>
<td>M11,M14</td>
<td>3.3</td>
<td>0.35</td>
</tr>
</tbody>
</table>

Figure 4.26: Transmitter Bias
### Table 4.10: Transistor Sizes for figure 4.26.

<table>
<thead>
<tr>
<th>Transistor(s)</th>
<th>Width (µm)</th>
<th>Length (µm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>40</td>
<td>0.35</td>
</tr>
<tr>
<td>M2,M6,M13</td>
<td>10</td>
<td>0.40</td>
</tr>
<tr>
<td>M3,M7,M10</td>
<td>32.5</td>
<td>0.40</td>
</tr>
<tr>
<td>M4,M8,M11</td>
<td>32.5</td>
<td>0.35</td>
</tr>
<tr>
<td>M5,M12</td>
<td>10</td>
<td>0.35</td>
</tr>
<tr>
<td>M9</td>
<td>4</td>
<td>0.40</td>
</tr>
<tr>
<td>M14</td>
<td>6.5</td>
<td>0.40</td>
</tr>
<tr>
<td>M15,M16</td>
<td>1.5</td>
<td>1.00</td>
</tr>
<tr>
<td>M17</td>
<td>20</td>
<td>0.35</td>
</tr>
<tr>
<td>M18</td>
<td>1.5</td>
<td>20.0</td>
</tr>
</tbody>
</table>
4.5 Encoder

The encoder is composed of many smaller encoders as discussed in section 3.4. The layout can be seen in figure 4.27. In the figure, the encoder appears quite large. However, its layout is not as dense as the other elements on the chip and it could be smaller.

4.6 Chip Layout

The chip layout (with metal and capacitive fill removed) is shown in figure 4.27.
4.7 Test Circuitry

The chip contains a pseudo-number generator to generate data that is fed into the encoder and then transmitted.

At the receiver, sixteen consecutive received bits are clocked into a shift register and then sent to output pins in parallel at a rate that is $\frac{1}{16}$ that of the input. A multiplexor selects which lines are to be sent to the output.
Chapter 5

Simulation Results

This chapter shows the simulated behavior of the implemented chip. Simulations were done using HSPICE [3] using netlists that were generated using Cadence [18] CAD tools. The netlists were generated from the layout views of the circuits and included parasitic capacitances added by the extractor of the CAD tool.

Because of constraints of computer memory and time it was not possible to simulate the entire circuit at once. Because of this, simulations were done on the transmitter and the receiver separately. Simulations of the transmitter are discussed in section 5.1 and simulations of the receiver are discussed in section 5.2.

5.1 Transmitter

This section contains results from simulations involving the transmitter. Figure 5.1 shows the voltages seen at the input of the receiver when data is transmitted at a
rate of 1Gbps. Figure 5.2 shows the same signals that have been normalized by subtracting the reference signal voltage from them. (The reference signal is the one in the middle of the plot in figure 5.1.) The normalized waveforms for data transmitted at a rate of 1.6Gbps is shown in figure 5.3. The circuit used for the simulation is shown in figure B.1 which is found in appendix B.

Figure 5.4 shows the transmitted data across several lines. The bottom trace shows the signals on the reference line which has guaranteed transitions every second bit. (The guaranteed transitions are marked with a dotted oval.) As explained in section 3.4, every second bit has a guaranteed transition in the master line. Between the other bits the transitions cycle through the other lines so that the slave receivers have guaranteed transitions that they can use for phase recovery and bit alignment. In the implemented chip, there are six of these transitions on one line, then the transitions occur on the next line. These transitions are also marked with dotted ovals on the lines that they occur in figure 5.4.
Figure 5.1: Transmitter Signals viewed at Receiver at 1GHz

Figure 5.2: Normalized Received Transmitter Signals at 1GHz
Figure 5.3: Normalized Received Transmitter Signals at 1.6GHz
5.1 Transmitter

Figure 5.4: Transmitter Output
5.2 Receiver

5.2.1 Master Receiver

In this section, waveforms in the master receiver are shown as the circuit starts from an unlocked initial state and then moves to a locked state. By locked state, it is meant that the receiver circuit finds where the transitions occur in the input and samples the data in the proper locations which are in the middle of the data eye and in the middle of the transition. Figure 5.5 shows the circuit near the beginning of the simulation when it has not properly recovered the data timing yet. (The circuit used for the simulation is shown in figure B.2 in appendix B.) Shown in the figure are waveforms for the input to the sampling comparators, which is the wire connected to the pad where the bondwire connects to the circuit. Waveforms that are also plotted in the figure are for the voltages on the sampling capacitors on of the two the sampling comparators on the line. (The capacitors can be seen in the schematic of the comparator in figure 4.16. The function of the comparators is described in more detail in sections 3.3 and 4.3.1.)

There are four sampling comparators on each input, two of which operate on one input bit and the other two on the next input bit. Under normal operation one sampler will sample the input in the middle of the data eye and the other will sample the data at the data transition. In figure 5.5 it is seen that although the data samples appear to occur near the middle of the data eye, the edge samples are always away from the middle of the transition and are always the same as the data sample, indicating that the sampling is occurring too early and so the sampling
5.2 Receiver

clocks should be delayed.

Figure 5.6 shows the same waveforms after the circuit has modified the sampling clocks so that the edge sample occurs closer to the data transition.

The settings on the phase blender (the blender schematic is shown in figure 4.20) during the simulation are shown in figure 5.7. The strength of the drive of one branch of the blender is shown in the column on the left side. The figure illustrates how the tap settings of the phase blender are updated as the circuit modifies the timing of the output clocks.

Figure 5.5: Master Receiver Unlocked State
5.2.2 Clock Generator

The output from the VCO running showing all of the evenly spaced output phases while running at a frequency of 800MHz is shown in figure 5.8.

The PLL current consumption at various frequencies is shown in table 5.1. The numbers include the currents in the phase detector, frequency divider, VCO bias circuit, VCO, and two output buffers for the VCO.
5.2 Receiver

Figure 5.7: State Changes in the Master Receiver

Figure 5.8: VCO output at 800MHz
Table 5.1: PLL Current Consumption

<table>
<thead>
<tr>
<th>Frequency (MHz)</th>
<th>Current (mA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>9.18</td>
</tr>
<tr>
<td>500</td>
<td>13.07</td>
</tr>
<tr>
<td>600</td>
<td>17.74</td>
</tr>
<tr>
<td>700</td>
<td>23.41</td>
</tr>
<tr>
<td>800</td>
<td>30.12</td>
</tr>
</tbody>
</table>

Table 5.2: Component Current Consumption

<table>
<thead>
<tr>
<th>Component</th>
<th>Current Consumption (@3.3V, 1Gbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master Receiver (includes PLL and DCW logic for both the master and slave receivers)</td>
<td>54.4mA</td>
</tr>
<tr>
<td>Slave Receiver (each)</td>
<td>30.4mA</td>
</tr>
<tr>
<td>Transmitter (includes encoder and bias circuit)</td>
<td>146mA</td>
</tr>
<tr>
<td>Total (includes master receiver, 12 slave receivers, and transmitter)</td>
<td>565mA</td>
</tr>
</tbody>
</table>
Chapter 6

Conclusions

This thesis describes a parallel transceiver circuit that is used to increase the number of effective pins on a chip. The implemented design converts two 8-bit inputs into a 12-bit output that can be scaled to any number of inputs. The circuit can track the timing of the input and can adapt to process, voltage and temperature variations.

The design was implemented in a 0.35μm CMOS process. Simulations show that the circuit can operate at a maximum bit rate per pin of 1.6Gbps. The circuit contains a master receiver and 11 more identical slave receivers. In the 0.35μm technology, the dimensions of the master block are 835μm × 700μm. The dimensions of each slave block are 218μm × 700μm. If the design were to be used with more inputs, only the number of slave circuits need be increased.
6.1 Future Work

Rather than quickly tracking the timing on one line and propagating that information to the other lines, it may be preferable to do the slow adjustment of each line individually and use the average of all the up/down signals from each receiver to do the quick adjustment of the master clocks.
Appendix A

Package Models

In figure A.1 the package model (supplied by CMC) is shown.

Figure A.1: Package Model

The model of the package is shown in figure A.1. The sizes of the components are shown in table A.1.
<table>
<thead>
<tr>
<th>Resistor Name</th>
<th>Resistance Value (Ω)</th>
<th>Capacitor Name</th>
<th>Capacitance Value (F)</th>
<th>Inductor Name</th>
<th>Inductance Value (H)</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>16.85m</td>
<td>C1</td>
<td>1.246p</td>
<td>L1</td>
<td>1.098n</td>
</tr>
<tr>
<td>R2</td>
<td>15.32m</td>
<td>C2</td>
<td>119.0f</td>
<td>L2</td>
<td>1.098n</td>
</tr>
<tr>
<td>R3</td>
<td>15.89m</td>
<td>C3</td>
<td>1.064p</td>
<td>L3</td>
<td>1.098n</td>
</tr>
<tr>
<td>R4</td>
<td>14.50m</td>
<td>C4</td>
<td>1.321p</td>
<td>L4</td>
<td>1.098n</td>
</tr>
<tr>
<td>R5</td>
<td>25.42m</td>
<td>C5</td>
<td>1.388p</td>
<td>L5</td>
<td>1.098n</td>
</tr>
<tr>
<td>R6</td>
<td>18.13m</td>
<td>C6</td>
<td>1.261p</td>
<td>L6</td>
<td>1.098n</td>
</tr>
<tr>
<td>R7</td>
<td>25.95m</td>
<td>C7</td>
<td>1.264p</td>
<td>L7</td>
<td>1.098n</td>
</tr>
<tr>
<td>R8</td>
<td>18.26m</td>
<td>C8, C9</td>
<td>1.047p, 587.3f</td>
<td>L8</td>
<td>1.098n</td>
</tr>
</tbody>
</table>

Table A.1: Component sizes for figure A.1.
Appendix B

Simulation Circuits

Figure B.1 contains a schematic of the circuit for the simulations done in section 5.1.

<table>
<thead>
<tr>
<th>Component Name</th>
<th>Component Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>50Ω</td>
</tr>
<tr>
<td>$C_p$</td>
<td>1.2pF</td>
</tr>
<tr>
<td>$C_e$</td>
<td>1.2pF</td>
</tr>
</tbody>
</table>

Table B.1: Component values in figures B.1 and B.2

Figure B.2 contains a schematic of the circuit used for the simulations done in section 5.2.1.
Figure B.1: Test Circuit For Transmitter
Figure B.2: Test Circuit For Receiver
Bibliography


