DESIGN OF
HIGH-PERFORMANCE AND LOW-COST
PARALLEL LINKS

A DISSERTATION

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL ENGINEERING

AND THE COMMITTEE ON GRADUATE STUDIES

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Evelina Fai-YeeYeung

January 2002

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Mark A. Horowitz (Principal Adviser)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Bruce A. Wooley

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Thomas H. Lee

Approved for the University Committee on Graduate Studies:

ABSTRACT

The need for high-bandwidth and low-latency inter-chip data transfer in short-distance applications has led to widespread use of point-to-point parallel links. For these links, the design goal is not only to increase the bit rate per I/O, but also to integrate a large number of I/Os in the system. As a result, the cost per I/O has to be kept low as performance improves.

Voltage and timing error sources limit the performance of a link and affect its robustness. The voltage and timing noise sources unique in parallel links, such as inter-signal timing skew and inter-signal cross-talk, impose greater challenges as the performance increases. The use of low-cost solutions, such as using cheaper electrical components, single-ended signalling, and simultaneous bidirectional signalling, further increases the voltage and timing noise. Therefore, overcoming the voltage noise, recovering timing at the receiver, and keeping the cost per I/O low form the three fundamental challenges in high-speed parallel link design.

In this research, we characterize, both analytically and experimentally, the voltage and timing noise sources in high-speed single-ended and simultaneous bidirectional links. We built an 8-bit single-ended, simultaneous bidirectional parallel link transceiver test chip in a 0.35μm CMOS process which allows full-range per pin skew compensation. The links achieve a bidirectional data rate of 2.4Gbps/pin with a BER less than 8 x 10^{-15}. The chip dissipates less than 1W total power from a 3.3V supply, and occupies a die area of 1.7 x 3.8mm^2.

We demonstrate that per pin skew compensation improves receiver timing margins in high-performance parallel links, and the cost overhead depends largely on the range and accuracy of the compensation desired. We compare different receiver timing recovery clock generation strategies, and the results show that the receiver clock generation delay makes tracking the high-frequency jitter of a source-synchronous reference clock difficult, and hence using a stable clock source is the best strategy. Low-frequency phase drifts in the interface signals can be compensated by a periodic calibration in a system capable of skew compensation, making the source-synchronous reference clock signal unnecessary.

We also demonstrate that single-ended and simultaneous bidirectional links are viable alternatives to the traditional differential and unidirectional systems. They allow significant savings in wires and pins for the same bandwidth, and the additional voltage noise sources, while significant, can be managed by careful design in circuits and in packaging.

Acknowledgments

My years in graduate school have been marked by inspiring teachers and encouraging friends. I thank, from the bottom of my heart, everyone who has enriched my Stanford experience and contributed to my professional and personal growth.

I sincerely thank Prof. Mark Horowitz for being the best advisor one can ever hope for. Technically, his wisdom and knowledge, both in depth and in breadth; his enthusiasm for teaching and research; the opportunities he provides for all his students; and the time he is willing to spend with us on our work have made these graduate school years a great learning experience. Personally, he always encourages me to do what I want to do, and tries his best to help. I appreciate his guidance, his kindness, and his patience with me, and I will forever cherish him as the most respected and admired teacher.

I gratefully acknowledge Prof. Tom Lee for his advice on my research and for many helpful discussions and ideas, as well as for serving on both my orals and reading committees. He also has enlightened me on many interesting topics because of his vast talents; I appreciate his kindness to me and his sharp insights on just about everything.

I would also like to express my gratitude to my associate advisor, Prof. Bruce Wooley, for his advice and kindness all these years, and for serving on both my orals and reading committees. He has impressed me as a great professor who is also nice to everybody and who cares very much about the concerns of all students.

I chose Stanford for the opportunity to work with Prof. Jim Harris, my first advisor here. I am exceedingly grateful for his kindness to me, which has extended far beyond my short stay in his group, and for his encouragements throughout all these years.

I was extremely fortunate to be part of a close and highly cooperative research group. I sincerely thank all the members for their help on my research and for the friendships that last beyond graduation. I am especially grateful to Stefanos Sidiropoulos, who served as a mentor to me when I first started working on links, and who has always been remarkably generous with his advice and help. I benefited tremendously from the kindness of Ron Ho, whom I shared office with and partnered with on the FLASH project, and from whom I learned a lot of different things that would benefit me lifelong. Ken Yang also played a mentor role in my early years of research, and I appreciate his help and his friendship. I am extremely thankful to Jaeha Kim and Dean Liu for helping in various tasks in the test chip; their contribution speeded up my progress considerably. Bill Ellersick was a great resource to me: he provided me with a lot of practical ideas on my test setup and an abundant supply of snacks. Hema Kapadia is my best friend in the group; her friendship and laughter have made all these long years seem somewhat shorter. The favorable comments of Gu-Yeon Wei, my first friend in the group, about everyone and everything helped attract me to this wonderful group. I also thank him for showing me what friends are for at times when I needed help the most. I am grateful to Birdy Amrutur, whom I was fortunate to work with during my first year, for his help and kindness throughout all these years. I am also thankful to Ken Chang, Dave Harris (who was my other officemate), Ken Mai, and Dan Weinlader for their help and many useful discussions; to Bennett Wilburn and Ricardo Gonzalez for their friendship; and to everyone else for his/her role in helping to create and maintain a supportive and fun environment.

The love and support of my family are what make everything possible in my life. I express my deepest appreciation for their patience with my seemingly never-ending schooling. I dedicate this thesis to my parents, Mr. and Mrs. Nin-Sing Yeung, for their lifelong efforts in trying their best for me and my sisters, and for their strong beliefs in giving us the opportunities to receive the best education and to be independent. They support me in everything I choose to do, even though that often is a decision that they do not consider the best or one that they need to make sacrifice for. The opportunities they have made possible for me have allowed me to broaden my scope and open up new opportunities. They supported my decision to go to Berkeley for college (and turn down the prestigious UK/HK scholarship to attend LSE), which was a turning point in my life. That would have been an easy financial decision for the affluent but certainly was not for a single-income family with three daughters studying abroad, but they had not asked me even once to reconsider the alternative. As I come to realize later that some parents are not even willing to pay for a good education for their daughters, I also realize how lucky I am to have parents who would do everything they can to give me a slightly better education, or in general a slightly better choice in everything. My Ph.D. means much more to me because it means a lot to my parents.

I sincerely thank my lovely younger sister, Athena, who is also my best friend in life, for her unfailing support, and for sharing my joy at good times and my sorrow at bad times. She is the smartest person I know, and despite the fact that she is a lot younger, she advises and guides me on everything like an older sister. I also sincerely thank my older sister, Minerva, who has guided me and greatly influenced what I decide to pursue all through my life. For one thing, I probably would not have chosen electrical engineering and pursued a Ph.D. if it was not because of her example. I am also extremely thankful to my brother-in-law, Boon-LockYeo, whom I am as close to as a real brother.

I also sincerely thank Kok-Wui Cheong for his love and support, and for his efforts in helping me to finish the thesis. If my family is the reward for me for being a good person in a past life (even though I honestly do not believe there was one), he is the ultimate reward for me for being a good person until now in this present life and for all the principles I hold fast to. He gives me every reason to be optimistic about the future.

I have been equally rewarded personally as academically during my years at Stanford (and Berkeley as well), for all the interesting people I met who broadened my horizons, and for all the friendships I have established and kept. In particular, I thank Samuel Liu for all the insights and discussions about my ever-changing problems.

The highly academic environment at Berkeley fostered in every undergraduate the desire to go to graduate school for a Ph.D.; I sincerely thank all my Berkeley professors who gave me the first glimpse of what true education is about, and who encouraged me and gave me opportunities to explore my research interests.

Table Of Contents

List of Tables

List of Figures

Chapter 1

Introduction

One ‘revolution’ of the past century is the rapid development in information acquisition, processing, and distribution. In almost all digital systems, advances in fabrication technologies allow the number of transistors to grow much more rapidly than the number of inputs and outputs (I/Os). An illustrative observation is that chips started with only one transistor and a few pins for discrete active components [1], [2][1], and have evolved to modern designs with tens of millions transistors but only a thousand pads [3], [4]. This huge discrepancy in the growth rates means that the bandwidth of each data I/O pin becomes more critical as technology scales.

Point-to-point parallel links have shown potential in delivering high-bandwidth and low-latency inter-chip communication, and have been widely used in short-distance applications such as multiprocessor interconnections [5], [6], [7], [8], [9], [10], [11], networking and communication switches [12], [13] , and consumer products with extensive multimedia applications [14], [15]. For these links, the design goal is not only to increase the bit rate per I/O, but also to enable the integration of a large number of I/Os in the system. As a result, the cost per I/O has to be kept low as performance improves.

This thesis characterizes potentially performance-limiting voltage noise and timing error sources in high-speed point-to-point parallel links and explores design trade-offs in low-cost signalling systems. In particular, we explore the voltage and timing noise in single-ended and simultaneous bidirectional links, and the design trade-offs in inter-signal skew compensation.

[1] These earliest single-transistor components, however, were not micro-fabricated and were used almost exclusively for analog applications, e.g. in telephone equipment, hearing aid, and radio.

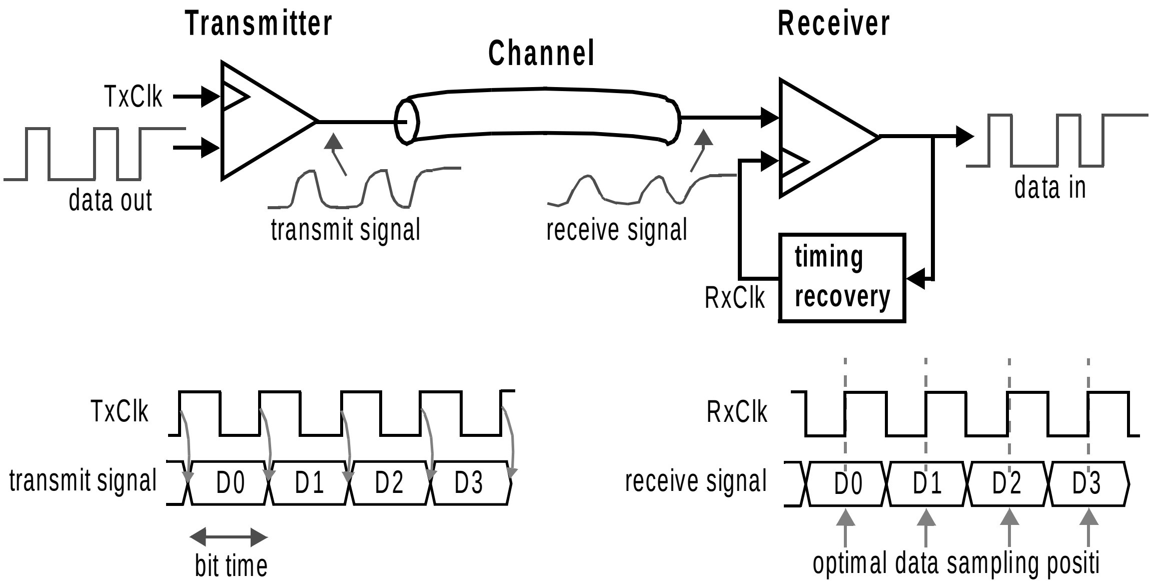


Figure 1.1: A basic link, showing the three primary components: transmitter, channel, and receiver, and their operation.

1.1 Link Basics

There are three primary components in a link: the transmitter, the channel, and the receiver. The transmitter converts a digital data sequence into a n analog signal in the channel; the channel is the entire transmission path along which the signal travels; and finally the receiver converts the received analog signal back to a digital data sequence. Figure 1.1 illustrates these key components in a typical link and their operation, and introduces the notations and terminology used throughout this dissertation.

The transmitter generally contains an encoder and a modulator, while the receiver is comprised of a demodulator and a decoder. A great variety of encoding and decoding schemes exists for data communication across a link. A coding scheme determines what bit patterns, or codes, should be used to represent the messages to be communicated[2]. The conversion of a discrete-time digital signal into a continuous-time analog signal is called

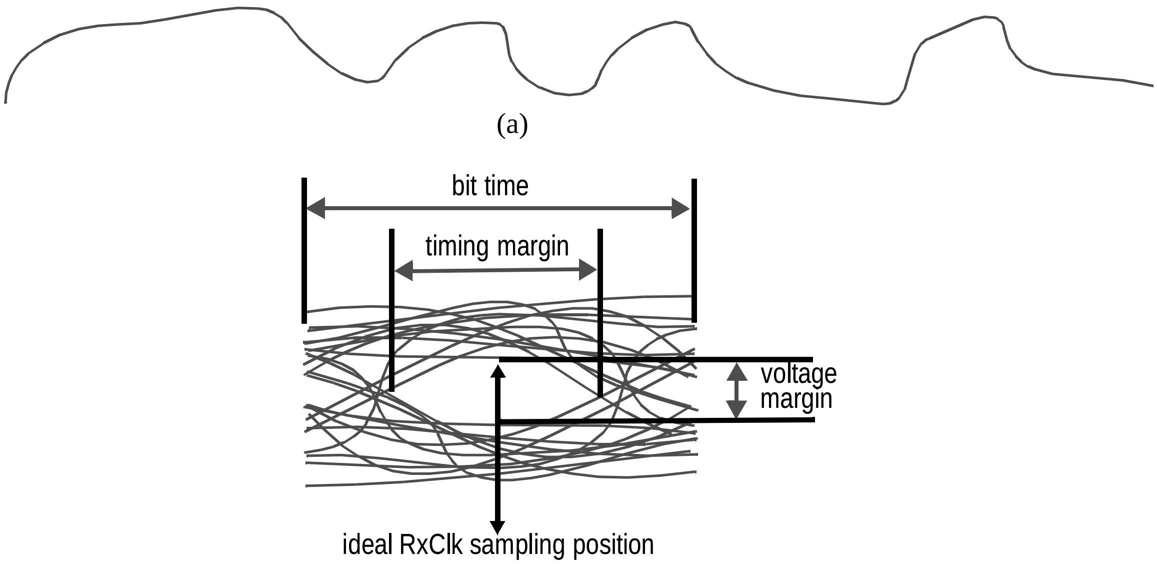
[2] Coding schemes can be as simple as inserting redundant bits to guarantee signal transitions to facilitate receiver timing recover, e.g. 8B10B, or as complex as using bit patterns that maximize receive signal energy or minimize receive signal error probability. These topics are beyond the scope of this dissertation, and are covered in great detail in [16] and [17] from a digital communication perspective and in [18] from a networking perspective.

modulation. This thesis explores electrical links using simple non-return-to-zero (NRZ) format, where the data is sent directly on the channel with no coding, and the signal levels are represented by different electrical voltages. Thus, the transmit signal is a binary signal, synchronized to the transmitter clock, TxClk, and is often low swing to reduce the power consumption in the signalling. The duration between successive signal transitions is called the bit time[3].

The channel is the entire transmission path or the physical media that the signal propagates through from the transmitter output to the receiver input. It consists of all the packaging components on both ends and the cables: bond wires or chip solder balls to connect the chip to the package, any package traces , printed-circuit board (PCB) traces, connectors to cables and to PC boards, and cables such as coaxial cables, ribbon cables, or twisted pairs. The channel is the origin of many voltage noise sources and imposes an increasingly challenging design environment as data rate increases -- often times it dictates the design choices in the transmitter and the receiver. Frequency-dependent attenuation in the board traces and cables distorts the received signal and creates inter-symbol interference (ISI), i.e. a symbol is distorted by noise introduced by earlier symbols. Impedance discontinuities due to packaging components generate reflections of the transmit signal leading to more inter-symbol interference. Therefore, equalizers optimized to the specific channel are often incorporated in the transmitter and the receiver to compensate for the filtering effects [19], [20], [21], [22]. The signal can also pick up cross-talk from nearby signals on its flight down the channel.

The receiver recovers the data sequence from the received signal stream. (The conversion of the continuous-time analog signal back to the original discrete-time digital signal is called demodulation.) The receiver amplifies and samples the receive signal , using a timing recovery circuit to optimally position the receiver clock, RxClk, to sample the data.

[3] In more sophisticated transmitters that encode multiple bits into a symbol, e.g. encoding two bits into a four-level signal, this quantity is more generally termed the *symbol* time.



(a)

(b)

Figure 1.2: Receive signal eye of a link: (a) a portion of the received data signal, and (b) folded signal eye (enlarged).

1.1.1 Receiver Voltage and Timing Margins

Figure 1.2 shows the noisy receive signal and the eye diagram, formed by wrapping the receive signal waveform around one bit time in the time domain, with the center of the eye at the ideal RxClk sampling position. The voltage margin is the voltage range the receiver can move its decision threshold and can still correctly determine the value of the receive signal when RxClk is optimally positioned at the center of the eye width. The timing margin is the time difference the receiver can shift RxClk and can still accurately detect the signal when the receiver threshold is centered in the eye height. Ideally, the receive signal would have a voltage margin equal to the nominal swing of the transmitted signal, and a timing margin equal to the nominal bit time. However, the various voltage and timing errors from the transmitter and the channel, as well as from the receiver itself, close the eye.

1.1.2 Serial Links vs. Parallel Links

The design of the above link components depends on the link architecture. Point-to-point links can be divided into two classes: serial links and parallel links. Serial links extract the clock from the data, while parallel links use an explicit clock associated with a number of data pins.

The link architecture shown earlier in Figure1.1 is a serial link. Parallel on-chip data signals are serialized into one data sequence. Timing information is embedded in this serial data, which is sent over a single interconnect. As described earlier, the receiver recovers the embedded clock from the signal transitions and aligns its local clock accordingly for optimal data detection. Serial link is the design of choice in any application where the cost of the communication channel is high and duplicating the links in large numbers is uneconomical. Its applications span every sector of the communication and networking markets [23], [24], [25], [26]. The dominant design goal is to maximize the data rate across each link, and in some cases to extend the transmission range. In order to meet the performance requirements, the links are more costly and complex.

Parallel links add an explicit clock signal to simplify the receiver design. Figure1.3 shows a conventional source-synchronous point-to-point parallel link, and the timing of the corresponding interface signals. Transmission of all data signals, data[0-n], and a reference clock signal[4], refClk, is triggered synchronously (hence the name source-synchronous) by TxClk. The receiver timing recovery circuitry generates a *global* receiver clock, RxClk, by delaying the received refClk by half of a bit time. RxClk is then used to sample all incoming data signals in the middle of their transitions to maximize timing margins. The elimination of clock recovery for each individual data pin allows each receiver to be simpler and smaller.

To amortize the cost of the refClk line and the receiver timing recovery circuitry, and to achieve the lowest overall system cost, the width of the parallel interface should be maximized. Each I/O should be fast, and at the same time low-cost to allow mass integration of a large number of the I/Os on the same chip. This is precisely why this dissertation studies parallel link design from a cross optimization point of view between high performance and low cost.

[4] The reference clock line can be viewed as just another signal that transmits alternating zeros and ones forever.

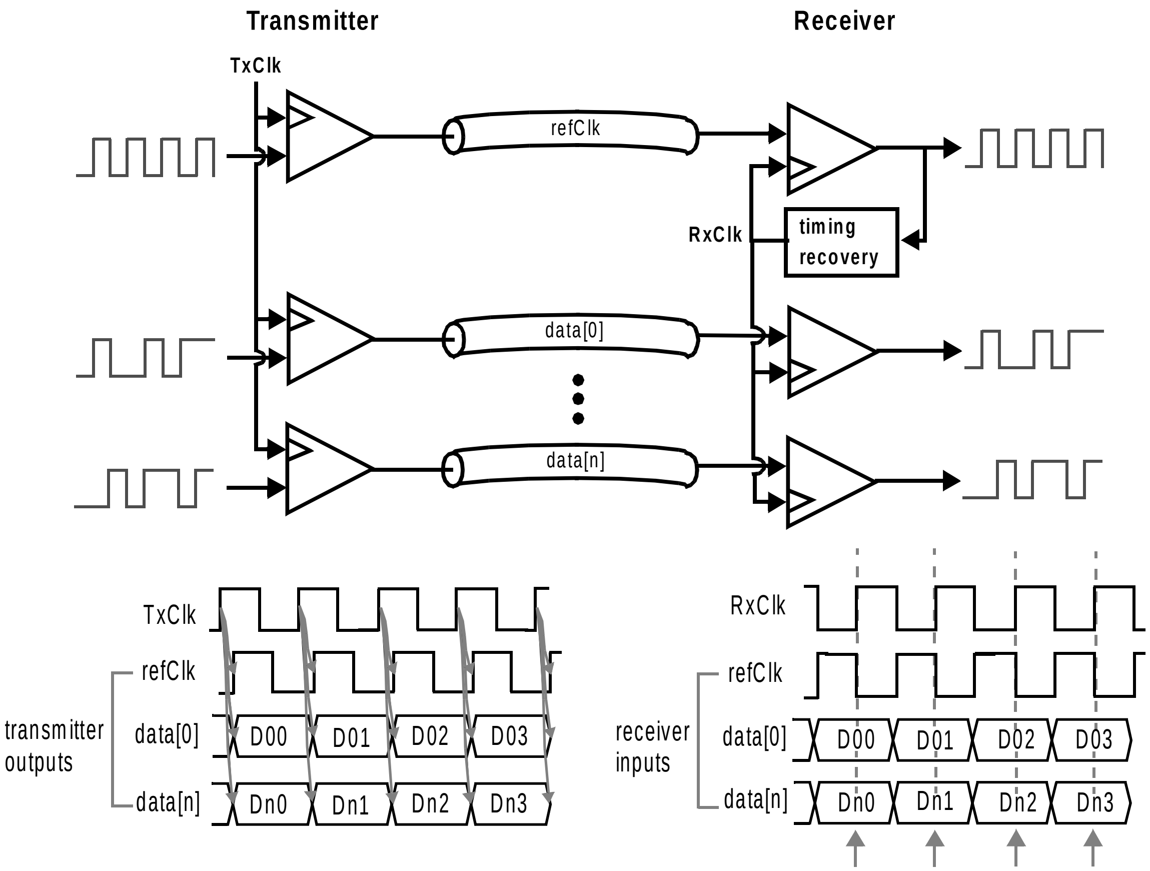


Figure 1.3: A conventional source-synchronous parallel link, where the clock is sent along with the data for easier receiver timing recovery.

As system requirements change over time, the design goals, features, and applications of modern serial links and parallel links move in converging paths. Parallel links employ traditional serial link techniques in the quest for higher speed, while serial links aim for lower cost to allow their integration in large numbers. As a result, the distinctions between serial links and parallel links are blurred. The discussions in this thesis also draw on examples of techniques used in serial link designs as we evaluate how the traditional parallel links can be modified to improve the performance while keeping the system cost low.

1.1.3 Signalling Methods

Conventional point-to-point links are mostly unidirectional and differential. In unidirectional signalling, data flows in one specified direction only in the channel. A differential signalling scheme transmits both the signal and its complement on a differential channel. As we will see in subsequent chapters, unidirectional , differential links have nice noise properties and hence are widely used.

However, the need to integrate a large number of I/Os on the same chip has called for lower cost designs. Since scaling in fabrication technologies decreases the cost of transistors faster than the cost of I/O pins, pin saving becomes an important cost parameter and signalling setups that reduce the number of pins and wires provide attractive alternatives. One such scheme is single-ended signalling, where the signal alone is transmitted and compared to a shared reference at the receiver. This eliminates all the pins and wires associated with transmitting the complement signals. An alternative option is to allow data transfer in both directions on the same wire. Half-duplex systems allow data flow in either one direction at any time, reducing the number pins and wires needed for bidirectional data communication. However, this scheme is beneficial only in applications that do not need to utilize the full channel bandwidth in both directions. It can at most deliver the same effective total bandwidth as a unidirectional line. Often times, the effective bandwidth is reduced due to the overhead in the handshaking. For applications that require continuous data flow in both directions simultaneously, superimposing the data streams in both directions on the same wire, or simultaneous bidirectional signalling, is one way to cut the pin count and wire count while keeping the same bandwidth capability. Single-ended signalling and simultaneous bidirectional signalling, however, both create larger voltage noise that may ultimately limit the achievable performance of a link.

1.2 Organization

In the rest of this thesis we study how the traditional parallel link architecture can be modified to increase the bandwidth per pin while keeping the cost per link modest.

Chapter 2 describes a conventional unidirectional and differential source-synchronous point-to-point parallel link architecture, and examines the three fundamental challenges in increasing the overall system performance of this design: overcoming the voltage noise, recovering timing at the receiver, and keeping the cost per I/O low.

To better understand the design trade-offs, we built a parallel link transceiver test chip [27], [28] which is the topic of Chapter 3. Specifically, the chip architecture and circuits are discussed, and the measurement results are presented. In addition to the core functions, many testing and measurement capabilities were implemented to aid the voltage and timing noise studies in Chapters 4 and 5. These blocks are also described in detail.

Chapter 4 examines timing noise sources in high-speed parallel links: Section4.1 examines the static inter-signal timing skew problem, and explores the design trade-offs in the implemented per pin skew compensation architecture as well as in alternative approaches; Section4.2 studies the dynamic phase noise of the interface signals and compares the three different receiver timing clock generation strategies supported in the test chip.

Chapter 5 investigates voltage noise sources in low-cost signalling systems. It characterizes t he voltage noise sources present in the implemented single-ended and simultaneous bidirectional links , and compares the results obtained analytically from noise models and experimentally from measurements of the test chip.

CHAPTER 2

BACKGROUND

The simplest way to communicate among modules is to provide a shared set of wires that they can all read or drive. This is precisely what a broadcast architecture such as a multi-drop bus does. However, as the system performance requirements increase, the conventional bus architecture fails to deliver the desired performance. Consequently, high-performance digital systems are abandoning the bus for point-to-point links.

Section2.1 presents an architecture of modern source-synchronous point-to-point parallel links. Section2.2 through Section2.5 discuss the challenges in overcoming voltage noise, recovering timing at the receiver, and minimizing overall system cost, and previous solutions to these challenges. Finally, in Section2.6, we identify a set of questions unanswered or raised by these designs, and this set of questions directs the work in the rest of the thesis.

2.1 Conventional Parallel Links

While variations exist in different parallel interface systems, the constraints for designing the links are similar. Figure2.1 shows a conventional interface architecture that forms the framework of modern source-synchronous point-to-point parallel link designs. This interface is similar to the one in Figure1.3 , except that here, signalling is differential and signals transition on both edges of the transmitter clock (TxClk). To save power, voltage-mode drivers with their resistance matched to the characteristic impedance of the interconnects are used to series-terminate the transmission lines at the source, and the transmitted signal swing is much lower than the on-chip power supply value (usually on the order of hundreds of millivolt).

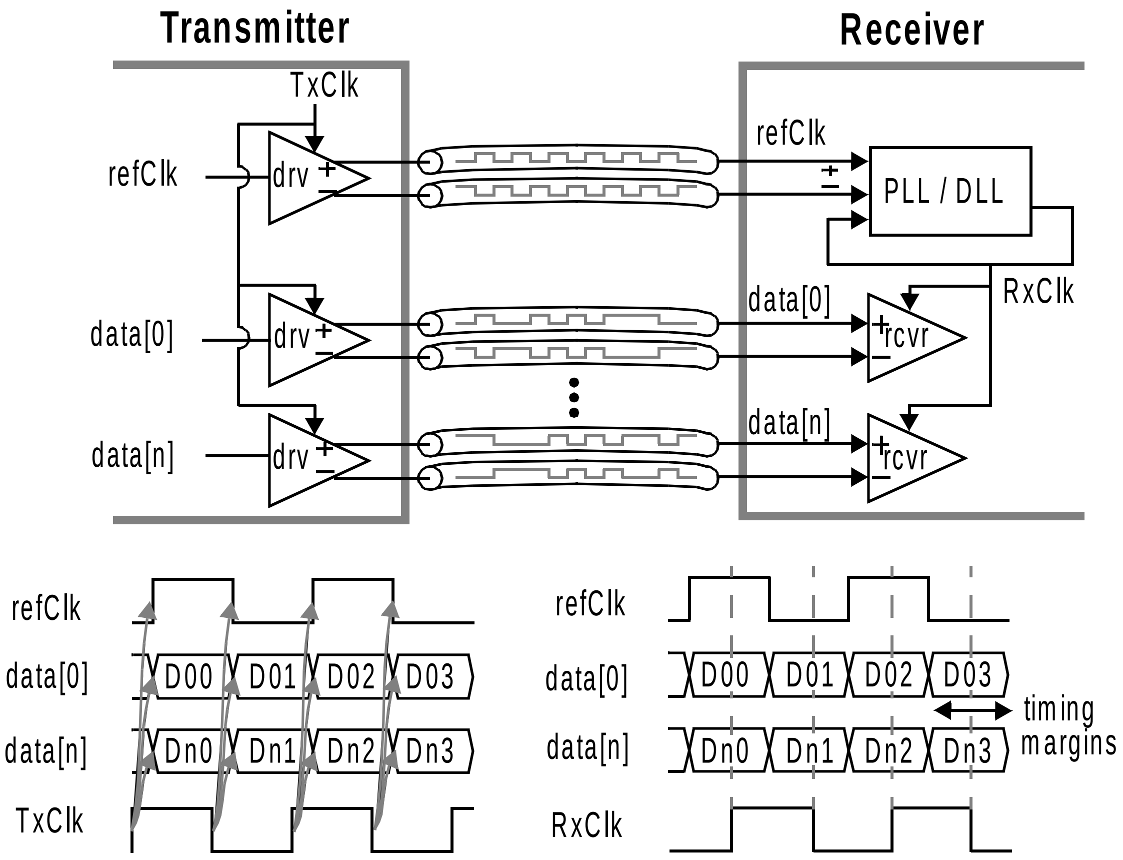


Figure 2.1: A conventional source-synchronous unidirectional and differential point-to-point parallel link.

A good design presents reasonable trade-offs among the performance metrics -- bit rate, latency and bit error rate (or robustness to be more general), and the cost metrics -- power, area, number of pins, wires and other electrical components, the required quality and hence cost of these parts, and design time and complexit .

The most common metric that designers use to report the performance of a link design, whether parallel or serial, is the highest data rate attained. Ambiguity may exist, such as whether this is the data rate per link or per pin[1]. A better measure is the minimum achievable bit time, which is the reciprocal of the maximum achievable bit rate. The quantity can be expressed in a time unit, such as in picoseconds. However, since the

[1] For instance, electrical serial interconnects are mostly differential and the data rate is often reported in bit rate per differential link, while parallel link performance can be described by the total data rate across all links or the bit rate per link.

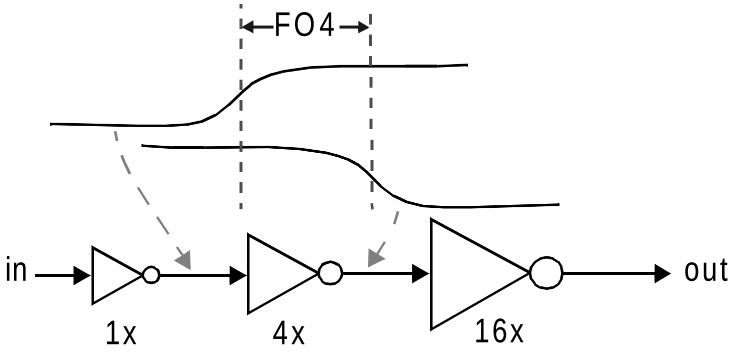


Figure 2.2: Fanout-of-four delay (FO4).

minimum bit time varies with the CMOS technology used, with the supply voltage, and with the temperature, normalizing the minimum bit time to create a technology-independent metric allows comparison or estimation of link performance in different technologies and extrapolation to future technologies in designs that scale with technology -- often true in digital CMOS circuits [29], [30].

The fanout-of-four delay, defined to be the delay of an inverter driving a capacitive loading (fanout) equal to four times its input capacitance as illustrated in Figure2.2 and denoted by FO4 in this dissertation, is a natural choice for the standard metric. It represents the delay of an inverter near its ideal fanout point, and has been previously shown to track the delays of other gates. Measuring the bit time in units of FO4 yields a technology-independent value[2].

Latency is another important metric in parallel link designs, even though it is of secondary importance in most serial links where latency is dominated by channel delay. Latency is often measured in terms of number of bits or number of transmitter or receiver clock cycles.

Bit errors reduce the effective bandwidth usage of the links, and m ost digital transmission systems have some mechanisms to handle or correct them. Therefore, reliability or robustness of a link at the desired operating speed is also an important consideration. This is measured by the bit error rate (BER). Different applications have

[2] Interestingly, FO4 in sub-micron CMOS processes is roughly 500ps/µm of effective channel length [30].

different BER requirements, and excess bit errors force a link to lower its operating bit rate. Bit errors are caused by voltage (amplitude) and timing (phase) noise sources as pictured earlier in Figure1.2. Voltage and timing noise issues are discussed in the next two sections.

2.2 Voltage Noise

Voltage noise directly reduces s ignal voltage margins . It also reduces signal timing margins by shifting signal transition edges. The major voltage noise sources present in the parallel link illustrated in Figure 2.1 are channel attenuation and inter-symbol interference, fabrication offsets, reflections, inter-signal cross-talk, and power supply noise.

2.2.1 Channel Attenuation and Inter-Symbol Interference

The channel filters the transmi t signal and causes frequency-dependent channel attenuation and signal distortion, leading to reduced receive signal amplitude and inter-symbol interference. Channel attenuation and inter-symbol interference are present in all links, but their magnitudes depend on the characteristics of the channel and the signal frequencies relative to the channel bandwidth.

The resistance of the channel attenuates the traversing signal, and the conduction in the surrounding dielectric causes further loss. Furthermore, high-frequency current flows closer to the conductor surface resulting in higher series resistance (i.e. skin effect); and the dielectric conduction also increases with signal frequency. Therefore, the channel acts like a frequency-dependent band-limited filter that disperse s the traversing signal. Moreover, the channel has some group delay (i.e. delay dependent on signal frequency), and hence the different frequency components reach the receiver with different delays, adding to the signal dispersion. As a result, the channel reduces signal amplitude, and adds residual error from previous bit leading to ISI.

Other ISI sources are reflections of previous bits due to termination mismatches or impedance discontinuities in the channel, which we discuss later in Section2.2.3 ; and incomplete settling of the transmit signal within one bit time, which needs to be resolved at the circuit level by speeding up the transmit datapath or damping any ringing at the transmit output.

To combat channel attenuation and ISI, equalization has been widely used in communication systems [16], [17]. The basic idea is to intentionally insert filters in the signal path to provide the inverse filtering effect of the channel. However, with current technologies, the complex equalization schemes used in communications systems cannot operate at the GHz frequency range and hence cannot be used in multi-Gigabit link design, where only simple equalization schemes can be applied. Equalizers can be implemented only at the transmitter or the receiver, or at both. The easiest and hence most common equalization technique used in Gigabit links is transmitter pre-emphasis [19], [20], [21], [31], where a short FIR (finite impulse response) filter pre-distorts the transmit signal to boost the signal energy of the high -frequency components. This scheme, however increases power consumption; and while it amplifies the high -frequency signals, it enhances the high-frequency noise as well. Alternatively, the same mechanism can be implemented at the receiver to increase high frequency gain [32], but the high-frequency noise enhancement problem still exists.

Realizing that equalization does not make efficient use of the channel bandwidth, designers have also explored multi-level signalling, where the transmitter sends more than one bit at a time. The simplest multi-level transmission scheme, called pulse amplitude modulation (PAM), is to encode N data bits into a symbol consisting of 2^N voltage levels. 4-PAM signalling has been demonstrated to increase the achievable data rate over band-limited channels [22], [33].

2.2.2 Offsets

Even in a carefully matched layout, transistor mismatches in the transmitter and receiver circuitry can induce fixed voltage offsets [34] whose magnitudes are independent of transmission signal swing but rather are determined by the transistor sizes and process parameters. The induced offsets increase for smaller transistors. Transmitter mismatches cause the actual output signal swing to deviate from the nominal swing, and receiver mismatches increase the minimum transmit signal swing required for accurate signal detection. Offset-cancellation techniques commonly used in op-amp designs, using either analog or digital control, can be applied to reduce the effects of circuit mismatches [35]. Offset-cancelled systems have successfully reduced the magnitude of the receiver offsets to below 10mV [36] which can be easily overpowered by the signal swings used in practical links. While offsets need to be considered especially in very low-power (low-swing) systems, they are not dominant error terms.

2.2.3 Reflections

Reflections can impact signal margins in two different ways. Firstly, reflections at mismatched terminations and impedance discontinuities come back as noise signals and add to future signal bits in the line. Therefore, reflection noise is another type of inter-symbol interference, and reduces receiver signal margins. The reflection of a signal is given by

$$V_{reflected} = \Gamma \cdot V_{incident} \quad , \tag{2-1}$$

where Γ, the reflection coefficient, is related to Z_L, the load impedance at the reflection point, and Z_0, the characteristic impedance of the line, by

$$\Gamma = \frac{Z_L - Z_0}{Z_L + Z_0} \quad . \tag{2-2}$$

In unidirectional links, the only reflections of the transmit signal that reach the receiver are even reflections (i.e. signal reflected twice, four times, etc.). These reflection points can be the two ends of the link, or impedance discontinuities in the channel. The magnitude of the second reflection depends on the product of the termination coefficients at the two reflection points, and is often quite small in doubly-terminated unidirectional lines; and the magnitudes of the higher even reflections are negligible.

Secondly, signals can be attenuated at impedance discontinuities because the reflections carry energy but never reach the receiver end. The receive signal energy is smaller than the transmit signal energy, and hence the receive signal amplitude is smaller than the transmitter output swing. This also results in a reduction in receiver voltage margin. This effect, however, is generally much smaller than the impact of the reflection noise itself.

Fabrication process variations and non-linearities in driver transistor resistances cause termination resistances to deviate from their nominal values, and bond wires, packages, connectors and other board components all introduce impedance discontinuities. Chip level variations are often much larger board level variations, and they result from environmental variations such as changes in supply voltage and temperature, as well as from variations in processing steps such as variations in dopant concentrations and transistor feature sizes.

Since a reflection is proportional to the swing of the inducing signal, increasing the swing does not increase the signal-to-noise ratio (SNR). Reflections need to be reduced or minimized by other methods. Automatic impedance control has been the most popular technique to reduce reflection noise by dynamically adjusting the termination resistor to match the interconnect characteristic impedance. This is usually done by comparing the voltage across a dummy driver in a potential divider formed by the driver [3] and the transmission line to half of the signal swing, or alternatively by monitoring the reflected waveform of an incident step voltage. Implementations can be completely analog [37] where an on-chip sense-amplifier comparator dynamically adjusts the gate overdrive voltages of the transistors in the dummy driver, or completely digital where the driver transistors are broken down into differently (often binary) weighted legs and digital controls select the right combination of legs to be turned on [38], [14], [39], [40]. Automatic impedance control usually takes a few extra I/O pins. The impedance information collected from the dummy driver is used to adjust all drivers, therefore small impedance variations can still exist because of process matching issues.

Reflection noise becomes more of a problem as signal frequencies increase. When the transition times become comparable to the signal propagation delays through the bond wires, package traces, and connectors along the signal path, these components can no longer be treated simply as lumped elements, but rather transmission line stubs. They

[3] In these cases, the drivers serve as source (transmitter) terminations.

create more impedance discontinuities which induce more reflections. These frequency-dependent effects make controlling the slew rate of transmitted signals important since faster transition edges contain higher frequency signals [41], [42], [40], [43]. By limiting the high-frequency components in the signals, high-frequency reflections are reduced. As we will see later in this section, slew rate control has other benefits: it also reduces the power supply noise caused by switching large current in the big output drivers [42], and cross-talk between neighboring signals.

2.2.4 Inter-Signal Cross-Talk

In parallel data channels , flux coupling to and from nearby signals due to mutual capacitance and mutual inductance leads to cross-talk. The size of this cross-talk depends largely on the signal layout geometry. For instance, in a twisted pair where the two wires are twisted precisely, the electrical and magnetic flux from each wire cancel that from the other wire (almost perfectly) so that the wire pair as a system does not radiate electromagnetic energy. The amount of radiation can be further reduced by shielding the signal pair. On the other hand, inter-signal cross-talk is the worst when unshielded wires are bundled together, unless if the cross-talk from all other signals is a common-mode noise on the pair of signals in consideration (this is almost impossible to accomplish). Nevertheless, differential links usually have low inter-signal cross-talk as efforts are taken to ensure equal couplings from other signals to each signal pair.

In unidirectional links, receiver voltage margins are affected only by the far-end cross-talk (FEXT), which is usually smaller than the near-end cross-talk (NEXT). The near-end cross-talk in these links is usually terminated at the source and hence does not affect the receiver signal margins. For two transmission lines in a homogenous medium, the forward travelling disturbances from the exciting signal line to its neighbor caused by the mutual capacitance and mutual inductance exactly cancel each other, and these two components partially cancel even if the transmission lines are in a non-homogenous medium. Therefore, the far-end cross-talk between transmission lines is often negligible. Much of the far-end cross-talk is from cross-talk at the packaging level, i.e. bond wire, package, connecto , and so on. The high-frequency components of the transmitter signals are usually attenuated by the time they arrive at the receiver, making the far-end cross-talk at this level also smaller than near-end cross-talk.

Inter-signal cross-talk may also be induced via a shared signal return or the power supply. The different cross-talk noise sources are studied in great detail in Chapter 5. Regardless of the cause, any form of inter-signal cross-talk is proportional to swing of the inducing signal, as in the case of reflection noise. Likewise, the SNR loss due to cross-talk cannot be compensated by increasing the transmit signal swing, and cross-talk needs to be reduced or compensated by other circuit or system techniques [44] , [33]. For the same configuration, and hence the same mutual capacitance and mutual inductance, capacitive coupling increases with a higher rate of change in voltage in the exciting signal, while inductive coupling grows larger when its rate of change in current is faster. Therefore, cross-talk also becomes worse for higher frequency signals, and techniques like slew rate control of driver output signals can also help reduce cross-talk.

2.2.5 Power Supply Noise

Power supply noise is induced by switching large currents in short durations across the parasitic inductance in power distribution network, and is therefore also called dI/dt noise. It is a problem in almost every digital system. The problem is becoming more serious as modern chips integrate more gates that switch at higher frequencies. Power supply noise can be induced by the switching activities of on-chip logic gates, and the magnitude of this component is independent of the I/O driver output signal swing. Power supply noise is also caused by the large output drivers switching large currents, and the magnitude of this disturbance is proportional to the output signal swing. Wide parallel interfaces integrate a large number of I/Os and can therefore suffer serious supply noise, both fixed and proportional.

Power supply noise is, in general, not a dominant voltage noise in differential links. Sending complementary signals allows the total current drawn from (and discharged to) each power supply to be constant, eliminating large current spikes across the power pin inductors or power distribution line inductance. Moreover, since the two halves are nicely balanced, to the first order, any power supply noise coupled to the signal pair at both the transmitter and the receiver is common-mode.

Although power supply noise affects different systems by different degrees, its omnipresence in digital systems has stimulated enormous research efforts in techniques to reduce dI/dt noise. Such techniques include minimizing the inductance of power distribution networks, employing constant-current drivers or more generally keeping the total current drawn from each supply constant, increasing bypass capacitance both on the chip and on the board, using separate power supplies for noise-sensitive circuits, generating on-chip supplies using voltage regulators, slowing down signal transitions using slew rate control [42], and using coding schemes that reduce switching frequency of signals [45].

2.3 Timing Noise

Unlike conventional synchronous systems, source-synchronous parallel interfaces eliminate the restrictions on the clock cycle time imposed by signal propagation delay or inter-module skew. However, the receiver now needs to recover timing information from the received signals.

The first step in receiver timing recovery is phase recovery -- figuring out where signal transition edges are and then sampling at the point that gives maximum timing margins in between transitions. This is relatively easy in the conventional parallel link design since timing information is carried in the source-synchronous reference clock line (refClk) sent along with the data signals (data[n:0]), assuming that all data signals and refClk reach the receiver at the same time[4].

The presence of timing errors, however, shifts the transition edges of the received data signals relative to the transition edges of refClk and narrows timing margins. In parallel links, the phase error of concern is the inter-signal phase error or, more precisely, the deviation in phase of each data signal relative to RxClk, or to refClk if RxClk tracks

[4] Or the arrival time of refClk is centered in the distribution of the arrival times of the signal pins, and the maximum difference in the arrival times is much less than the bit time.

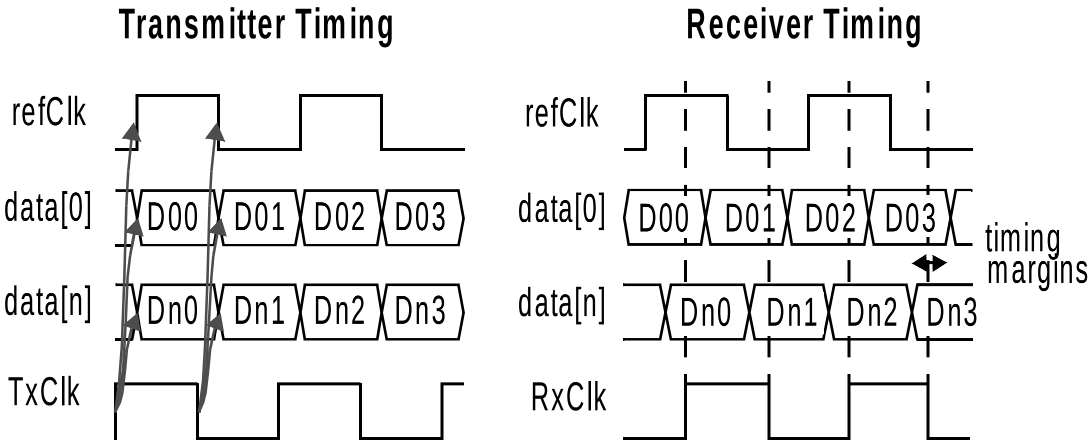


Figure 2.3: Inter-signal skew reduces timing margins at receiver. As inter-signal skew increases, overall timing margin of the link decreases.

refClk perfectly[5]. This phase error can be decomposed into a DC phase offset (skew) and the dynamic phase noise (jitter)[6].

Inter-signal timing skew is caused by differences in signal propagation delays from the transmitter to the receiver. This skew problem is illustrated in Figure 2.3. Signals arrive at the receiver at different times. So even if the global receiver clock (RxClk) is properly aligned with respect to the received refClk, timing margins are still greatly reduced. The achievable bandwidth across the parallel interface is then limited by the timing mismatches. Any static phase offset in clock recovery shifts the sampling point away from the optimal center and further narrows the receiver timing margins.

Mismatches in the transmitter and receiver circuitry, in the transmitter and receiver clock distributions, and in the interconnect wires (cables, printed-circuit board traces, package traces, and connectors) all create differences in signal delays and result in inter-signal timing skew [7]. While designers have different opinions on the magnitude of inter-

[5] 5. This is impossible in implementation since there is certain delay in the receiver timing recovery which limits the tracking bandwidth.

[6] 6. The problem of inter-signal timing errors, both skew and jitter, is unique in parallel links. In serial links, there is no inter-signal timing error, but the limited bandwidth of the receiver timing recovery may also create phase deviation of RxClk from the incoming data signal.

[7] 7. Synchronous voltage noise generated by a fixed data sequence can create skew too, but data streams are almost always, or assumed to be, random, therefore making the timing error caused by the voltage noise to be jitter.

signal skew resulting from circuit mismatches in careful designs, most agree that the skew coming from interconnect mismatches is becoming a problem. Delay measurements of commercial parts have shown skews as large as 50-60ps per meter of cable, per meter of printed-circuit board (PCB) trace, or per connector [46], [47]. Moreover, in many applications, exact matching of all signal traces is not possible for cost and marketing reasons. For instance, in wide or high-fanin parallel interfaces such as multiprocessor interconnections and network switches, the board traces and components are often laid out in a compact manner, resulting in variations in trace lengths between the traces near the center and those near the periphery. Sometimes efforts are taken to balance the traces by deliberately adding zigzag paths or turns to the inner shorter traces. The sharp corners, however, create large inductances at high frequencies leading to significant additional phase shift and attenuation in these signals, and hence such practice is avoided as data rate increases. The total mismatch resulted from all the above skew sources, as a percentage of the bit time, obviously gets worse as the bit time continues to scale down and as the parallel links run on longer wires.

Fortunately, skew is a static phase error and can be compensated. More and more interface designs have incorporated per pin deskewing functions [46], [48], [49], [44], [50], [9]. On start-up, a calibration mode is initiated, where each bit's skew relative to a timing reference is found using some digital control logic. The skew information is stored and is used to control the delay of an adjustable delay chain. Either the local transmitter clock [50] or the local receiver clock [46], [9] is shifted by this amount. The adjustable delay chain can be realized by activating a different number of stages [46], [50], by adjusting the delay per stage, or by using phase interpolation [9].

Jitter in the received signals also reduces timing margins of the links. The 'inter-signal jitter' of concern is the jitter of data transitions with respect to RxClk edges -- if RxClk and the data signals jitter by exactly the same amount *at any moment* and *correlated* edges are used, the timing margins are *theoretically* not affected no matter how big this jitter is.

Most jitter is induced by on-chip voltage noise in the signalling system itself[8]. All the major voltage noise sources in link designs are synchronous in nature, but random data streams make the effect of each, and hence the superimposed voltage noise, ‘random’. Each of these components contributes to an additive voltage noise that shifts the timing of the signals.

If both the transmitter and receiver clocks are derived from the same clock source, they are mesochronous [51], i.e. of exactly the same average frequency but may bear an indeterminate phase difference. This is often the case in smaller, localized systems where sharing the same crystal oscillator and routing the clock signal to all I/O ports are possible. The receiver timing recovery architecture is generally s impler because only the phase difference needs to be tracked. Fortunately, the short-distance parallel links that this dissertation focuses on fall mostly into this category. On the other hand, if the transmitter and receiver clocks are derived from two independent clock sources that have the same nominal frequency, i.e. plesiochronous [51], the receiver timing recovery needs to track both the instantaneous frequency and phase of the incoming signals.

Precise phase alignment circuits are needed to accurately extract timing information from the receive signals and position the receiver clock so that the effects of inter-signal timing skew and jitter are minimized and the receiver timing margins are maximized . Therefore, at the heart of the receiver timing recovery are the phase-locked loops (PLLs), whose importance and diverse applications have made the topic one of the most widely researched circuit topics in decades [52], [53], [54], [55], [56], [57], [58], [59], [60] , [61]. PLLs are described in the next section to better understand their jitter and offset issues.

2.4 Phase-Locked Loops

The basic idea behind PLLs is to ‘lock’ the output clock to a timing reference using negative feedback, or in other words, the output clock tracks variations in the timing reference at a fixed phase relationship. The bandwidth of the tracking and the exact phase relationship between the reference and the output depend on the properties of the components around the loop. PLLs can be classified into voltage-controlled-oscillator

[8] Jitter can also be caused by the clock sources but here the clock sources are assumed to be very good (stable).

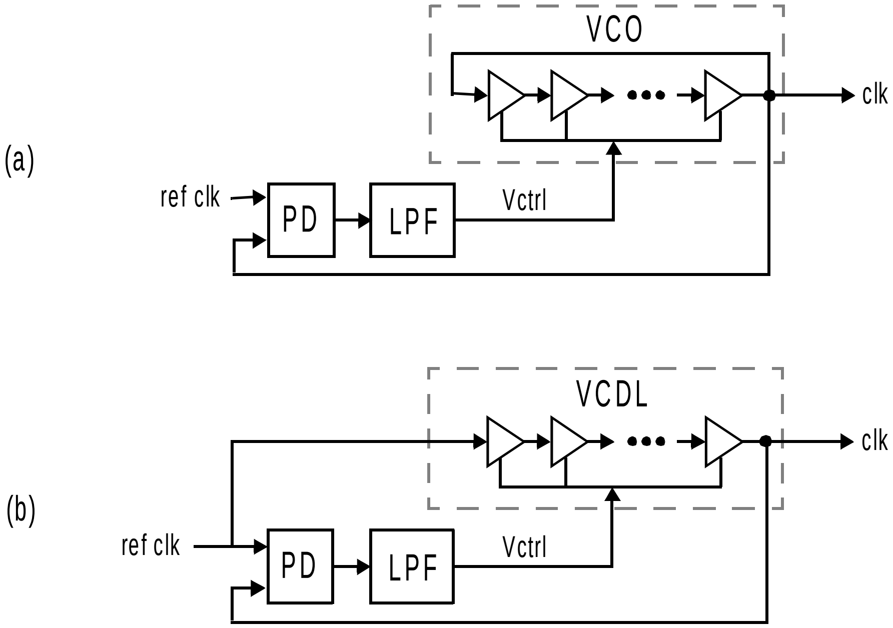


Figure 2.4: Phase-locked loops: (a) VCO based PLL, and (b) DLL.

(VCO) based, often referred to simply as phase-locked loops, or voltage-controlled-delay-line (VCDL) based, commonly referred to as delay-locked loops (DLLs). Figure 2.4 illustrates typical PLL topologies of both types: the VCO based PLL and the DLL.

A simple VCO based PLL, or simply PLL, consists of a phase detector (PD), a low-pass filter (LPF), and a VCO. The PD converts the phase difference between the input and output clocks, i.e. the phase error, into an output signal whose average DC voltage level is proportional to the phase error. This signal is ‘smoothed’ out by the LPF to generate the control voltage (*Vctrl*), which then drives the VCO and determines its oscillation frequency. The PD and VCO both provide gain to increase the loop gain and hence decrease the phase error. The filtering action in the LPF and the integration in the VCO each contributes a pole, so a stabilizing zero is needed in the LPF. Nevertheless, variations in process and environment shift the position of the zero and may cause problems in loop stabilit . A delay-locked loop, on the other hand, is intrinsically stable because the delay line is simply a gain stage (from the control voltage to the phase) and the integration pole associated with the VCO is eliminated. Therefore, a DLL has more relaxed stability vs. gain trade-offs.

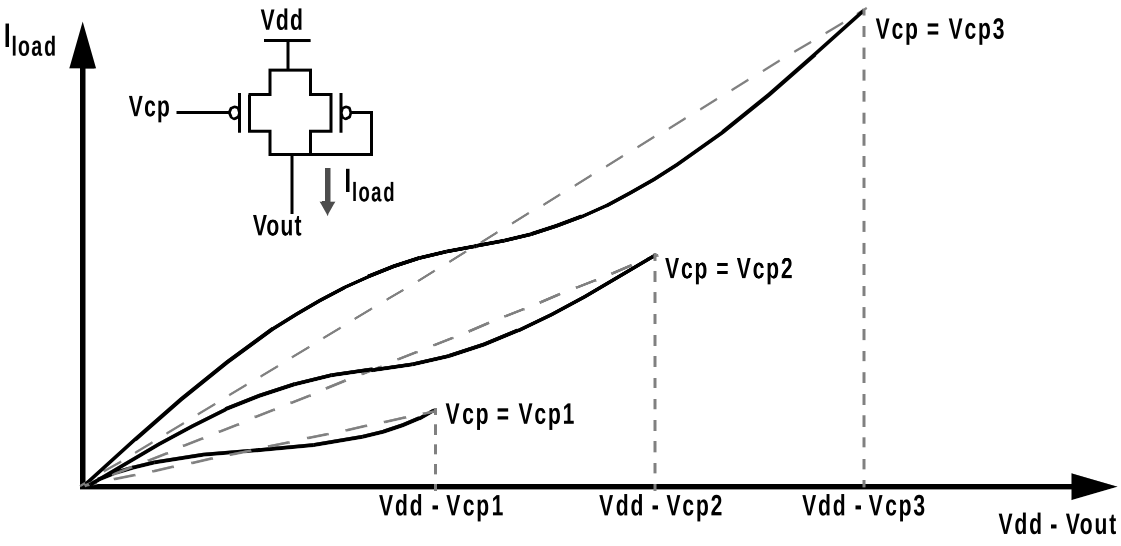


Figure 2.5: I-V characteristics of symmetric load as *Vcp* varies.

The output clock jitter is an important characterization parameter in PLLs and DLLs because the jitter eats directly into the timing margins of all signals clocked by any clock generated from this output. In large digital chips, power supply and substrate noise can be an important source of jitter. Therefore, one main focus in PLL and DLL research has been on techniques to reduce the jitter sensitivities of the clock to power supply and substrate noise.

The output clock jitter of a PLL or DLL is largely determined by the noise sensitivities of the buffer stages in the VCO or VCDL and of the subsequent clock buffer stages, and by the architecture of the PLL or DLL. In general, differential buffers with linear resistor loads provide high noise rejection because the output *RC* stays constant[9], giving a constant delay per buffer stage when the signal swing is fixed. Designers have used symmetric loads [62], [63], [61], illustrated in Figure2.5, to approximate voltage-controlled linear resistors. When the two transistors are equally sized, these loads exhibit perfectly symmetrical I-V characteristics, enabling differential buffers using them as loads to lower the supply and substrate noise sensitivities by an order of magnitude when compared to full-rail single-ended CMOS inverters.

[9] Part of the capacitive loading C is contributed by non-linear diffusion capacitance but the non-linear effect is negligible.

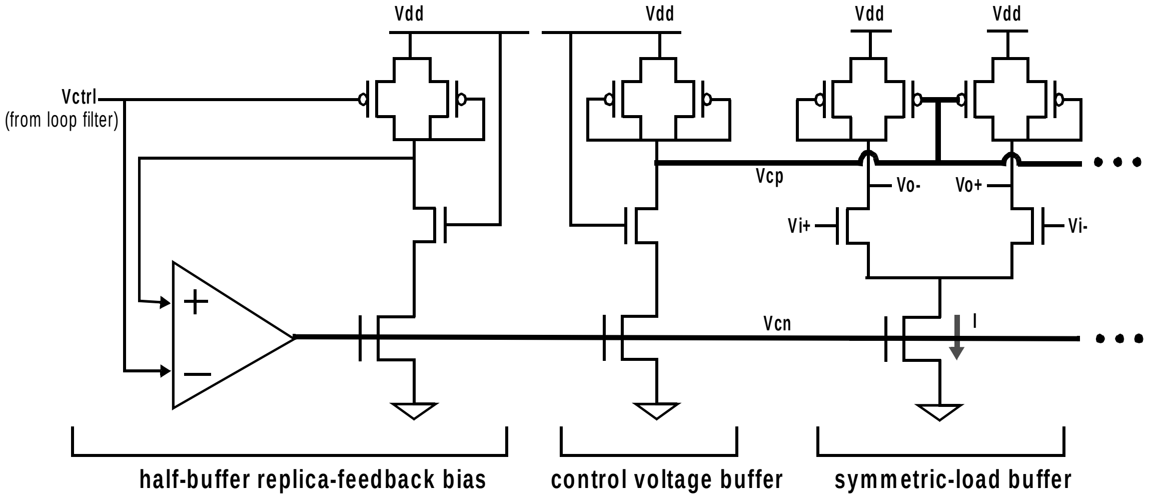


Figure 2.6: Self-biased replica-feedback circuit dynamically adjusts the bias voltages of the symmetric-load buffers to maintain constant bias current and signal swing.

To reduce supply and substrate noise sensitivities, the current of each buffer stage should be made to only depend on the control voltage, not the supply or substrate voltages. The problem with a simple current source is that movements in its tail voltage cause variations in its bias current due to its finite output impedance, leading to variations in signal swing. If the buffer’s effective resistance is also strongly current-dependent, both the signal swing and output *RC* time constant are affected. To maintain a constant voltage swing and a constant buffer delay, the bias current has to be kept unchanged. This can be accomplished by using a cascode current source to increase the output impedance. Alternatively, the same effect can be achieved with lower supply voltage requirement by dynamically adjusting the current source gate bias. One approach employs negative feedback with a half-buffer replica to generate control voltages for symmetric-load buffers used in the VCO or VCDL [62], [63], [61]. The essential components of the scheme are shown in Figure 2.6. The self-start, self-biased replica-feedback bias circuit generates the current source bias (*Vcn*) from the loop filter output control voltage (*Vctrl*) using a half-buffer replica. To avoid noise coupling onto *Vctrl* from the lower supply (Gnd) and the substrate, the buffers are not biased directly by *Vctrl*, but rather by *Vcp* which is generated by another half-buffer. These two control voltages *Vcp* and *Vcn* also bias circuit elements outside the VCO or VCDL such that the total delays of all clock paths scale with the PLL or DLL clock frequency. This replica feedback bias approach provides high isolation from either supply and minimizes the supply voltage requirement. The symmetric-load buffer exhibits low supply and substrate noise sensitivities. As a result, the above design and its variants have been used in numerous designs in recent years.

The problem of jitter propagation from the timing reference to the DLL output clock can be acute in the source-synchronous parallel link introduced earlier in Figure2.1 if a DLL is used in the receiver timing recovery. The received refClk can be noisy and jitter . If it is fed directly to the delay line input, the jitter is passed to the output clock, making the output worse than the input. To alleviate the problem, the delay line takes a separate clean clock source as input. While this technique normally suppresses the jitter propagation problem, it amplifies the limited phase locking range problem in a single-loop DLL -- the incoming refClk can be at arbitrary phase relationship with this clean reference clock input to the delay line. Worse still, the DLL would not lock if these two clocks are plesiochronous. A PLL would be needed in this situation.

One DLL design, capable of locking to an arbitrary-phase and plesiochronous clock, utilizes a dual-loop architecture [64], [65] as illustrated in Figure2.7 : a conventional first-order core DLL with its six-stage delay line locked at 180^o to give six equally spaced clock phases at 30^o spacings; and a peripheral digital loop that picks the appropriate pair of clock phases, selectively inverts them, and then interpolates between them. The interpolator output, and hence the DLL output (clk), can assume any of the quantized phase steps allowed by the phase interpolator, which in this case is any of the 16 steps within each 30^o interval. In this way, the DLL output can span across the full 360^o phase range and be rotated. This output clock then drives the main loop phase detector, wh ose output in turn drives the peripheral loop finite state machine (FSM). The FSM controls the phase selection, selective inversion, and interpolator phase mixing weight, and closes the peripheral loop, allowing the output clock to lock to the reference clock (refClk). The “bang-bang” nature of this control loop results in dithering around the zero phase error point when the loop is in lock, with the dither jitter determined by the interpolator phase step and the delay through the peripheral loop.

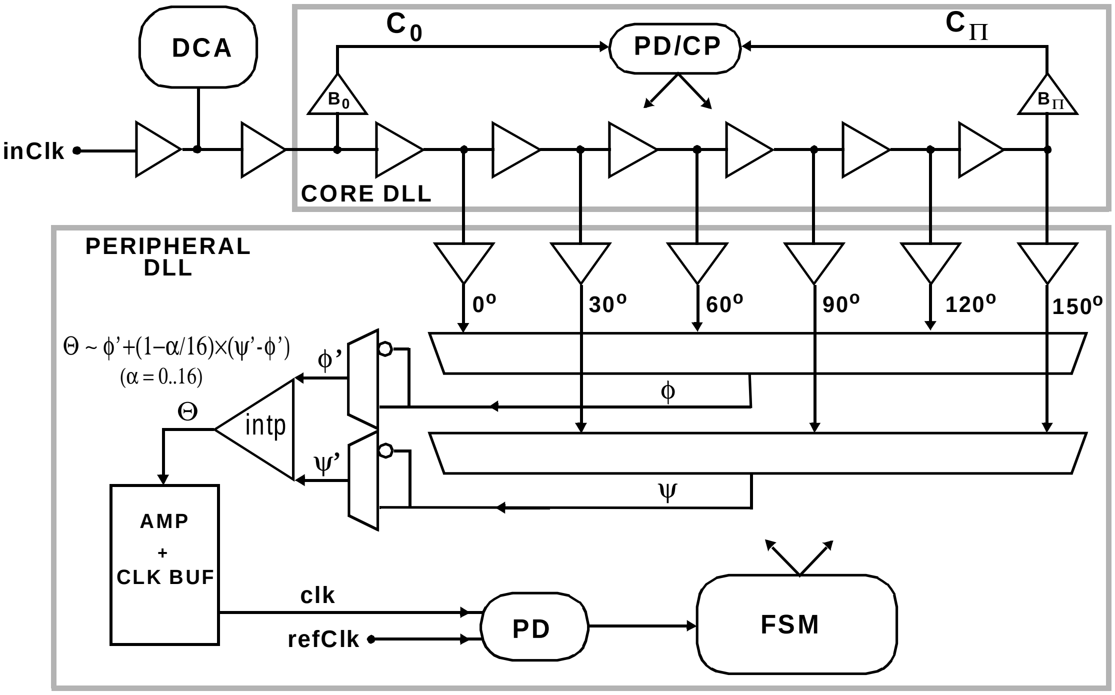


Figure 2.7: Dual-loop DLL.

The architecture in Figure 2.7 offers many benefits which make it useful for high-speed interface timing recovery. It solves the problem of limited phase capture range seen in conventional DLLs, and lifts any phase relationship constraint between the incoming clock and data signals and the on-chip clocks at the receiver. Even better, the architecture can handle plesiochronous timing between the transmitter and receiver : the DLL output phase is constantly being updated to track the phase of the incoming refClk, and can be rotated at the 360^o phase boundaries. Compared to conventional PLLs, this architecture offers a lower jitter output clock due to the elimination of the phase error accumulating VCO. The digital phase capture algorithm also provides greater stability compared to tuning a VCO in an analog loop as done in PLLs. For parallel link applications, the same input receiver design used for data reception can also be used as the phase detector in the peripheral loop. Hence, the on-chip signal paths at the receiver for refClk and all the data signals are balanced and matched. Finally, the 90^o phase shift required in RxClk (relative to refClk) to maximize receiver timing margins is easily done by a change in phase controls -- the phase moves by three 30^o clock spacings.

2.5 Low Cost I/O

High performance alone does not make a parallel link design superior. Since the same hardware is duplicated for every I/O signal, keeping the cost per I/O signal low is another important challenge in parallel link design. This low cost requirement can be easily seen from the system environment constraints discussed earlie , and its effect is two-fold: not only does it present a challenge itself to the designers, it can also make the other two challenges mentioned earlier, namely, overcoming voltage noise and recovering timing at the receiver, more of a problem, as we will see subsequently.

One way to reduce the cost of a system is to use lower cost electrical components. However, these parts are usually poorer in quality and in matching. One important adverse effect is that the badly matched wires create larger differences in the delays of the reference clock line and the data lines, and hence increase the inter-signal timing skew at the receiver.

As noted previously , scaling in fabrication technologies decreases the cost of transistors faster than the cost of I/O pins, making pin saving an important cost paramete . Thus, signalling setups that reduce the number of pins and wires are attractive alternatives to the traditional unidirectional differential system. Two such schemes are single-ended signalling and simultaneous bidirectional signalling.

Single-ended signalling has gained popularity [5], [66], [67], [68], [42] because it reduces the number of pins and wires of a system while delivering the same total bandwidth, but operates with reduced voltage margins because of the presence of larger noise sources. A unidirectional, single-ended parallel link interface is shown in Figure2.8 . At the receiver, all incoming signals are compared against a shared reference voltage (Vref) placed at the middle of the signal swing. This reference voltage can be generated in a number of different ways: on-chip at the receiver, fed as an adjustable input from the board, or on-chip at the transmitter and then connected to the receiver[10]. Reference voltage generation is a widely researched topic for single-ended links since the reference

[10] This type of interface can also be called pseudo-differential.

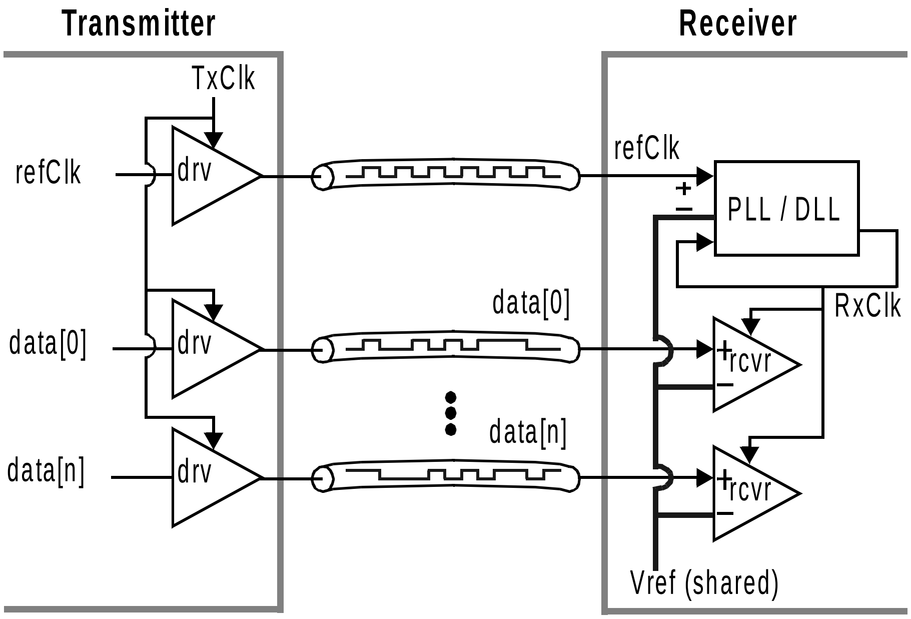


Figure 2.8: A unidirectional, single-ended parallel link interface. Received signals are compared against a shared reference voltage (Vref) placed at the middle of signal swing.

voltage is the source of many additional noise sources introduced in this signalling scheme, as we will see later in Chapter 5.

Since each input signal is now compared to the reference voltage, any noise on this reference affects signal margins. The DC component of this noise is usually called reference offset, and is caused by mismatch between the reference value and the signal swing. Most single-ended parallel link designs employ automatic control to adjust either the signal swing based on a preset Vref [66], [67] or to adjust the Vref value based on a fixed swing. Implementing swing control also helps to reduce power consumption as a side benefit [69].

The major source of AC noise is from the coupling of on-chip Vdd and Gnd onto the signal wires. Vref is more heavily coupled to the power supply at high frequencies than each data signal, reducing the common-mode rejection of power supply noise in the system. As a consequence, the effect of power supply noise also becomes much more prominent in a single-ended system than in a differential system. Even worse, the magnitude of the power supply noise may also increase in a single-ended system, because the power supply now acts as a shared current return path for the single-ended I/O signals.

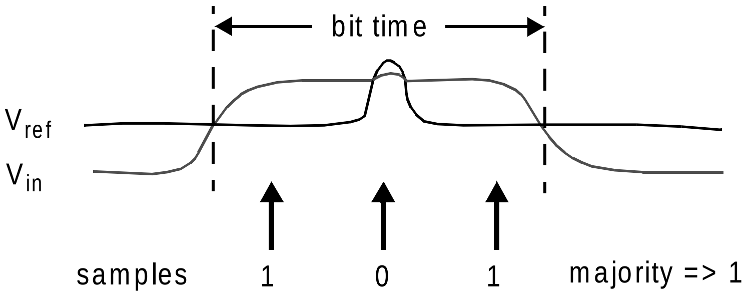


Figure 2.9: Oversampling as a means to overcome high-frequency noise.

The return current for one signal flowing in other signal paths and the direct capacitive and inductive coupling between wires also lead to increased inter-signal cross-talk. As discussed earlier, unidirectional link designers need to worry about far-end cross-talk only, which is often smaller than near-end cross-talk. One way to reduce cross-talk is to isolate the single-ended signals with power supplies. However, this reduces the cost-saving benefit of using single-ended links. In the extreme case, where a signal pin is always accompanied by a supply pin, the cost in terms of the number of wires and pins approaches differential links.

Coping with high-frequency noise (power supply noise, cross-talk, reflections, etc.) is not a unique challenge in single-ended links; even nicely balanced differential links suffer from the problem. Single-ended signalling accentuates the problem and necessitates architectural and circuit level solutions to combat these high-frequency noise sources. The most widely adopted approach is to use noise-insensitive receivers. One approach is to sample the received signal multiple times during one bit time -- a technique called oversampling -- and take a majority vote of all the samples [70]. Hence the oversampling factor should be an odd number of at least three. The idea is illustrated in Figure2.9 . Theoretically, oversampling can be done by generating a sampling clock at a frequency that is a multiple of the data signalling frequenc . Practically, this is rarely the case since the I/O signal is often at much higher frequency than the on-chip clocks. Generating an on-chip clock at a frequency multiple of the high I/O signalling frequency and designing a receiver that can run that fast are very difficult , if not impossible, tasks. Therefore, oversampling is often accomplished by generating multiple clock phases at equal phase

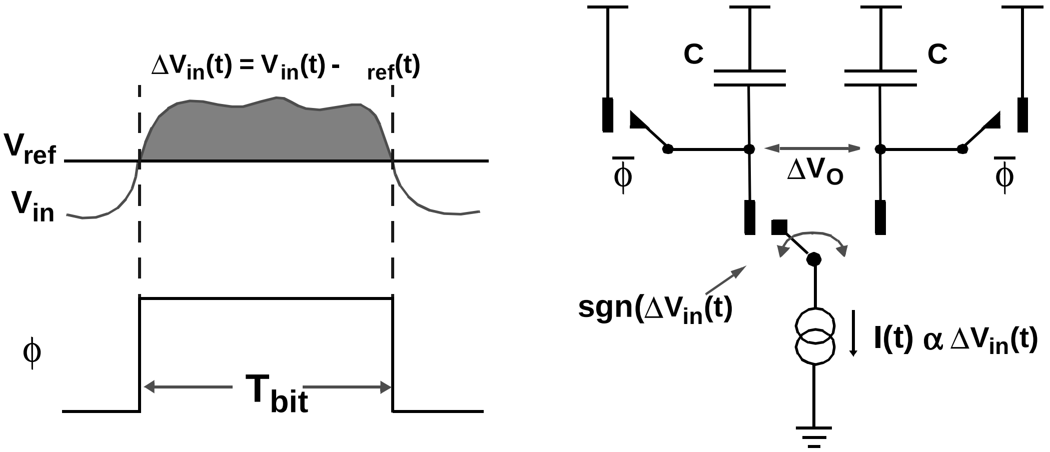


Figure 2.10: Current-integrating implementation of an ideal matched filter receiver. The instantaneous integration current is proportional, both in polarity and in magnitude, to the instantaneous difference in input signals.

spacings across the clock period and duplicating receiver samplers by the oversampling factor. This approach increases both the power and the area of each I/O cell, and thus it is unsuitable for parallel link designs.

Oversampling, with or without majority vote detection, has been widely used in serial link designs [21], [71] where the technique, more importantly, aids phase recovery. (For such a purpose, the oversampling factor can be less than three, i.e. two, depending on the phase detection algorithm used.) Parallel links, however, need more cost-effective schemes.

A more cost-effective solution is to implement the analog equivalent of majority vote. The optimal receiver is an ideal matched filter, which in this context means a receiver matched to the input signal pulse[11] [16], i.e. for an input pulse p, the ideal matched filter implements $p^*/\|p\|$, where p^* is the conjugate of ,$p$and is$\|p\|$ its m agnitude. Therefore, for a square wave input, the impulse response of the ideal matched filter is a unit-swing square wave. A current integrating implementation would integrate a current proportional to the actual input differential voltage (Vin - Vref) over the entire bit time, as illustrated in Figure2.10 . A pair of capacitors at the output nodes performs the current

[11] The channel impulse response is ignored here. In general, p is the signal pulse convoluted with the channel impulse response.

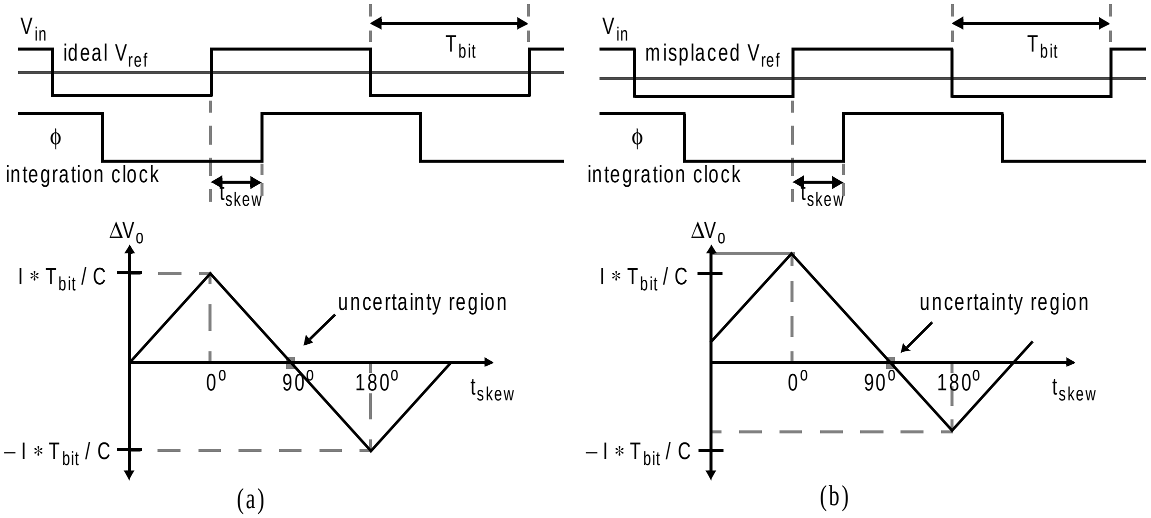


Figure 2.11: Phase characteristics of the current-integrating receiver shown in Fig ure2.10: (a) when Vref is centered at the middle of signal swing, and (b) when Vref is misplaced.

integration. The capacitors are reset prior to the integration bit period to eliminate intersymbol interference. The instantaneous integration current is proportional, both in polarity and in magnitude, to the instantaneous difference in input signals -- accomplished by a current proportional to the differential input magnitude and switched according to the differential input polarity. The resulting output voltage is therefore

$$\Delta V_O(T_{bit}) = \frac{G_m}{C} \cdot \int_0^{T_{bit}} [V_{in}(t) - V_{ref}(t)]dt \quad , \tag{2-3}$$

where G_m is the receiver transconductance transforming the input voltage to an integrating current, C is the value of the integrating capacitor, and T_{bit} is the bit time. The ideal phase characteristic curve of this receiver is shown in Figure2.11 (a). The problem with integrating the magnitude of the differential input signal is that any voltage noise on it changes the integrated voltage output: in particular, any reference offset shifts the phase characteristic curve away from its ideal position , as illustrated in Figure2.11(b), a nd consequently reduces the timing and voltage margins of this receiver.

To remove the signal margin dependency on reference offset, current-integrating receivers which integrate current based solely on the polarity of the input differential signal [72], [73], [74] have been used to filter the high-frequency reference noise in

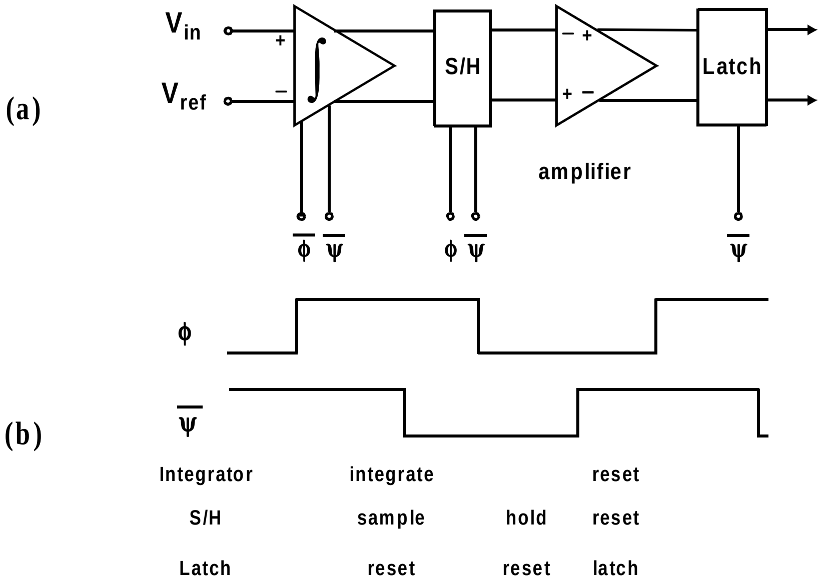


Figure 2.12: Current-integrating receiver block diagram and timing.

single-ended links. Using the same implementation as illustrated in Figure 2.10 -- but a *constant* integration current steered by the current switch based on the polarity of the differential input -- the output voltage is then given by

$$\Delta V_O(T_{bit}) = \frac{I}{C}\cdot\int_0^{T_{bit}} \mathrm{sgn}[V_{in}(t) - V_{ref}(t)]\,dt. \tag{2-4}$$

Theoretically, this type of receiver exhibits the ideal phase characteristic as shown in Figure 2.11(a) even in the presence of reference offset or any voltage noise source.

In a real implementation, the phase characteristic curve may shift away from the ideal curve due to circuit offsets [72]. Nevertheless, measurement results from current-integrating receivers implemented have shown that this type of receivers increase signal margins and robustness of high-speed links. In one careful design [73], [74], the simulated phase characteristics are very close to ideality across process corners, reflected in the measurement results as a minor loss in the receiver timing margin. The receiver architecture and timing are shown in Figure2.12 , The integrator stage, shown in Figure 2.13, consists of a differential pair (M_1 and M_2) used to low-pass filter the differential signal (Vin - Vref) by integrating currents using two capacitors at the output

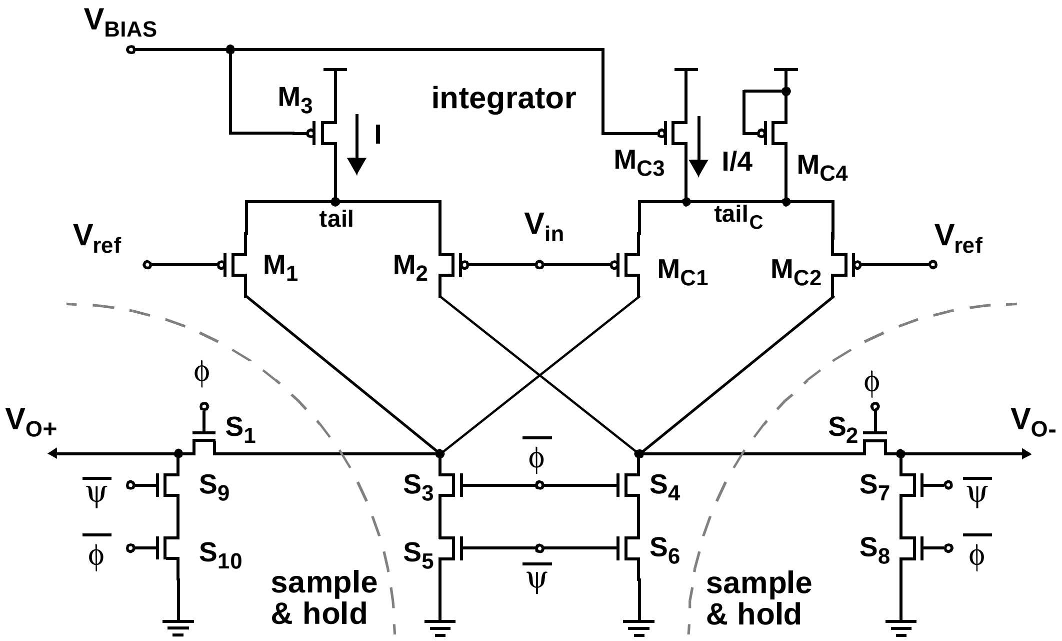


Figure 2.13: Integrator and sample-and-hold stages in the current-integrating receiver in [74].

nodes of the integrator stage for the entire bit time. The accumulated charges on these two output nodes pass through a sample-and-hold stage, also shown in Figure2.13 , and their difference is subsequently amplified and latched. In order to integrate the polarity of the differential input, the tail current should be completely steered to one side. (The integrator does not integrate the polarity when the differential input voltage is small.) T he two integrating capacitors are realized with parasitic drain capacitances of the transistors connected to the output nodes. The auxiliary differential pair (M_{C1} and M_{C2}), consisting of identically sized transistors operating at only one-fourth the tail current , is used to compensate for charge injection [74].

Further pin saving is achieved in a simultaneous bidirectional parallel interface [75], [76], [14], [77], [39], [40]. A single-ended, simultaneous bidirectional parallel interface using voltage-mode drivers is illustrated in Figure2.14 . The transceivers on both ends of the parallel link are identical. Signals travelling in both directions are superimposed on the same wire, giving a tri-level resultant waveform[12]. To recover the incoming signal, the receiver in each transceiver must subtract its own transmitted waveform. This is usually

[12] Assuming that the output signal swings of the two transmitters are the same.

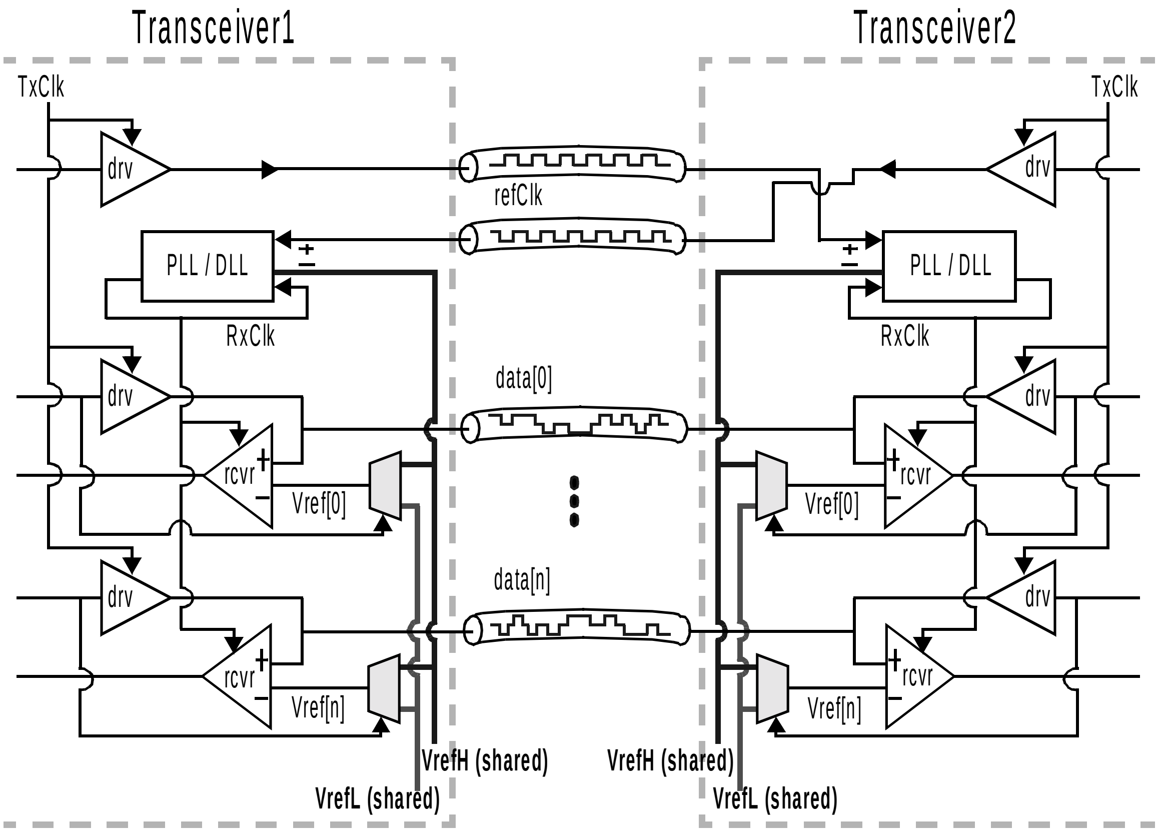
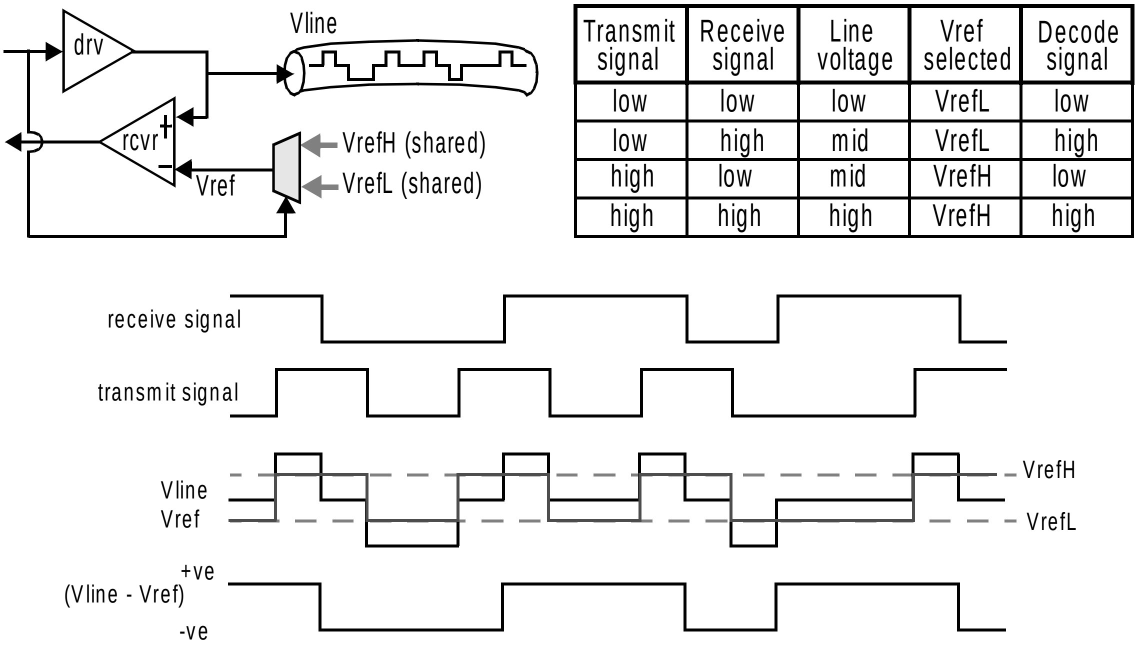


Figure 2.14: A single-ended, simultaneous bidirectional parallel link interface. Signals travelling in both directions are superimposed on the same wire. Each receiver subtracts out its own transmitted waveform to recover the incoming signal.

done by multiplexing two shared reference voltages (VrefH and VrefL) to generate the local reference voltage (Vref[n]), which switches to track the transmit signal. VrefH and VrefL are generated by similar schemes as used in unidirectional single-ended links.

The decoding scheme is explained further in Figure2.15 . Since the line voltage is the sum of the transmit signal and receive signal, if an analog copy of the transmit signal is subtracted from the line voltage, the receive signal is recovered. This copy is generated locally by multiplexing VrefH and VrefL. In transmitting a ‘low’, VrefL is used, and in transmitting a ‘high’, VrefH is used. The receiver accomplishes the subtraction by taking the differential input of (Vline - Vref).

However, noise issues grow even worse for this design due to the extra noise sources induced by the coupling of the transmit signal to the receive signals on both the same wire



Transmit signal	Receive signal	Line voltage	Vref selected	Decode signal
low	low	low	VrefL	low
low	high	mid	VrefL	high
high	low	mid	VrefH	low
high	high	high	VrefH	high

Figure 2.15: Decoding scheme used for simultaneous bidirectional signalling.

and the adjacent wires. Any amplitude mismatch between the transmit data swing and Vref swing results in an inexact cancellation in the receiver decoding , while any delay mismatch between the timing of the two causes a glitch at the receiver input. This induced glitch may lead to reception errors if a sampling receiver happens to sample at the glitch point. Hence, reference noise problem, or noise on the reference line, is even worse for simultaneous bidirectional links.

Designers have attempted to cope with the glitch problem. One reception and decoding scheme uses two sets of receivers in parallel and decoding happens after the analog front-end [70]. In this way, the errors introduced by amplitude mismatch and delay mismatch are eliminated, but this scheme duplicates the hardware per pin and works only when sampling receivers are used. Another scheme employs a low-bandwidth sampling receiver front-end so that the high-frequency glitch is filtered out without affecting the receiver output [77].

Even worse from a noise perspective is that reflections now directly reduce voltage margins. A single reflection of the transmit signal due to impedance discontinuities and termination mismatches, often called echo, will appear as noise to the incoming signal.

The coupling of the transmit signal to the receive signals on adjacent wires is caused by direct capacitive and inductive cross-talk. In simultaneous bidirectional links, both near-end cross-talk and far-end cross-talk reduce voltage margins. As explained earlier, near-end cross-talk is often larger than far-end cross-talk, and hence inter-signal cross-talk also becomes more important an issue in simultaneous bidirectional links.

In long-distance full-duplex communication lines where echo and near-end cross-talk are both dominant voltage noise sources, such as in Gigabit Ethernet [44] running on Category-5 (Cat-5) standard cables, echo and NEXT can be partially cancelled in the analog domain using programmable filters whose transfer functions a re programmed during the training sequence before actual data transmission. The residual noise sources can be cancelled in the digital domain using complex digital filters that are often incorporated with the channel equalizers. However, such highly advanced signal processing techniques are complex and require high hardware overhead, and can operate at data rates up to only a few hundred Mega symbols per second, making them unsuitable candidates for the Gigabit parallel I/O interfaces.

2.6 Summary

We can see from the previous sections that every design choice involves *trade-offs* among various performance and cost parameters. There has not been one single *best* design and different designers have different preferences and make different design decisions. It is these trade-offs that make parallel link an interesting research area.

As bit time continues to scale down, the loss in timing margin in parallel links due to inter-signal timing skew is becoming a larger percentage of the bit time and may present a performance bottleneck. Therefore, some type of skew compensation scheme will soon be needed in any high-performance parallel interface. While a few skew compensation schemes have been demonstrated previously, it is not clear what the design trade-offs in these schemes are. In particular, the skew compensation circuitry adds hardware to every I/O signal, and cost is a primary design parameter in parallel links. In this thesis, we investigate the design trade-offs between the skew compensation range of an architecture and the amount of hardware overhead needed and the added design complexity, by comparing architectures for different design environments scattered across the entire design space. To study the upper bounds on the cost overhead, we implemented a per pin skew compensation architecture that uses phase interpolation to enable full-range compensation.

Given the balanced nature of the refClk and data signals at the transmitter, the jitter in the received data signals may be correlated with the jitter in the received refClk; therefore trying to track the jitter in refClk in the receiver timing recovery may be beneficial. In this thesis, we study the dynamic phase noise characteristics of high-speed interface signals and the correlation in jitter in different signalling pins, and evaluate the benefits and problems of the traditional way of dynamically tracking the phase noise of a source-synchronous reference clock in the receiver timing recovery. Three different receiver timing recovery clock generation strategies are supported in the test chip, and the results are compared.

Fabrication technology scaling makes I/O pin count a more notable cost parameter than transistor count or gate count. Accordingly, pin-saving signalling setups such as single-ended signalling and simultaneous bidirectional signalling provide attractive low-cost alternatives to the traditional differential, unidirectional links. Unfortunately, they create a much noisier signalling environment. Exactly how much noise margin is being given up? In this thesis, we characterize the voltage noise sources in single-ended signalling and simultaneous bidirectional signalling , and extend the use of current integration to simultaneous bidirectional receivers.

Chapter 3 describes the test chip that provides the tools to experimentally study the noise sources, while Chapter 4 and Chapter 5 take on the timing noise and voltage noise aspects respectivel .

Chapter 3

Parallel Link Transceiver Test Chip

This chapter describes a transceiver test chip that was designed to help answer the questions from Chapter 2 and to explore some of the design trade-offs in parallel link design. Section3.1 gives an overview of the chip -- the components on the chip, the features supported, and the reasons behind the design choices. Section3.2 presents design details: the overall architecture is first described, followed by the design of the I/O front-end, and the transmitter and receiver building blocks. Section3.3 explains the test setup and environment, presents the system performance measurement and the experimental characterization results of different circuit blocks, and discusses the limitations of the setup and the accuracy of the measurement results. In addition to the core functions, the chip implements many testing and measurement capabilities to aid the voltage and timing noise studies in Chapters 4 and 5: the transceiver architecture supports per pin timing adjustment which allows measurements of timing margins of the links, while the adjustable reference voltage generation allows measurements of voltage margins. Therefore, we close this chapter, in Section3.4, by describing a systematic way with which the internal signal (voltage and timing) margins are measured.

3.1 Chip Overview

The parallel link transceiver test chip was fabricated in a 0.35μm (0.4μm drawn) CMOS process[1]. Figure 3.1 shows an interface between two chips. Each test chip has 8 single-ended data lines (data[7:0]) that are capable of simultaneous bidirectional data transmission. Each pin contains high-speed voltage samplers to display on-chip signals

[1] HP CMOS10 available through MOSIS.

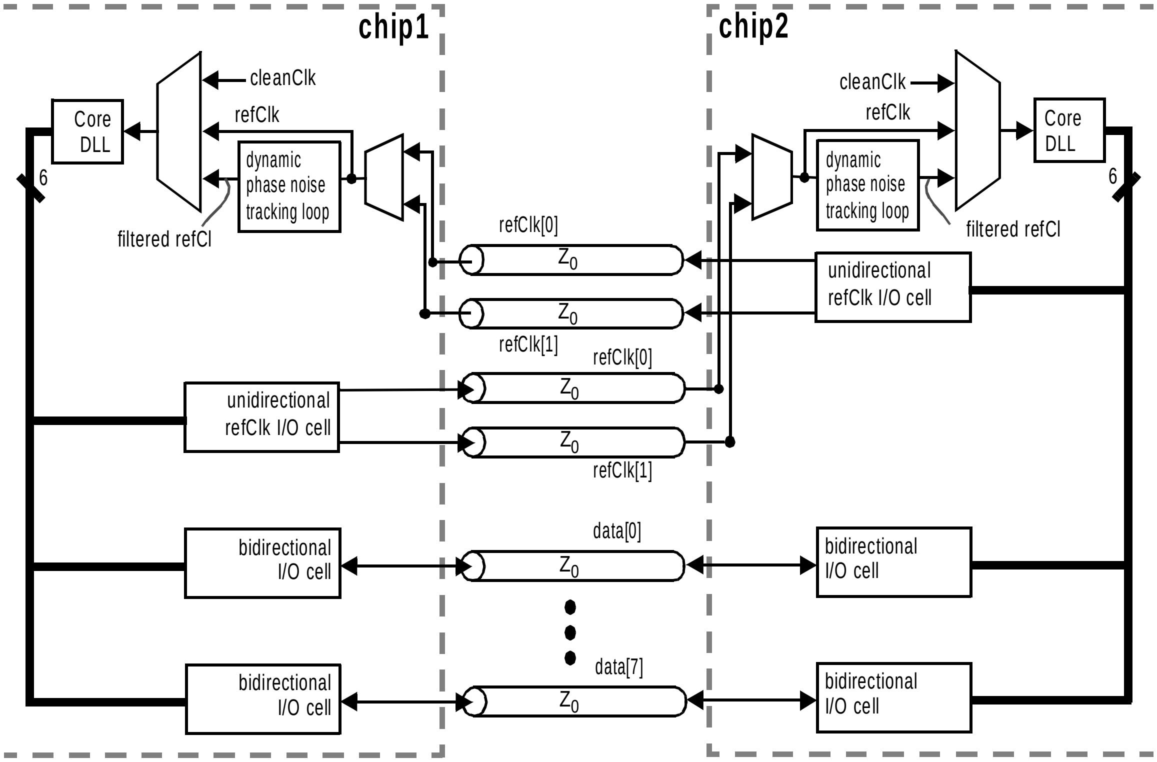


Figure 3.1: Chip Interface. Each chip has 8 single-ended, simultaneous bidirectional data and 2 sets of optional unidirectional refClk lines, and can operate in 3 different modes.

and to measure internal voltage margins of the links and inter-signal cross-talk, and per pin timing adjustment to compensate for inter-signal skew and to measure timing margins.

The chip also has two sets of optional unidirectional reference clock (refClk) lines, located in different parts of the chip as shown in Figure 3.2 . The test chip has three operational modes which differ in the receiver clock generation: in the default mode, a refClk signal is unnecessary and a ‘clean’ system clock is used for receiver clock generation; in the second mode, the receiver timing dynamically tracks the phase noise of one of the two source-synchronous refClk signals (refClk[1:0]); and finally in the third mode, the receiver timing dynamically tracks the phase noise of a filtered version of one of these two refClk signals using an additional dynamic phase noise tracking loop, which is effectively another dual-loop DLL that filters out any high-frequency phase noise in the selected refClk signal beyond the DLL’s update bandwidth. The motivation for using two refClk lines in this particular pad arrangement is to experiment whether refClk[0] carries more switching noise on the supply due to the switching activities of its neighboring

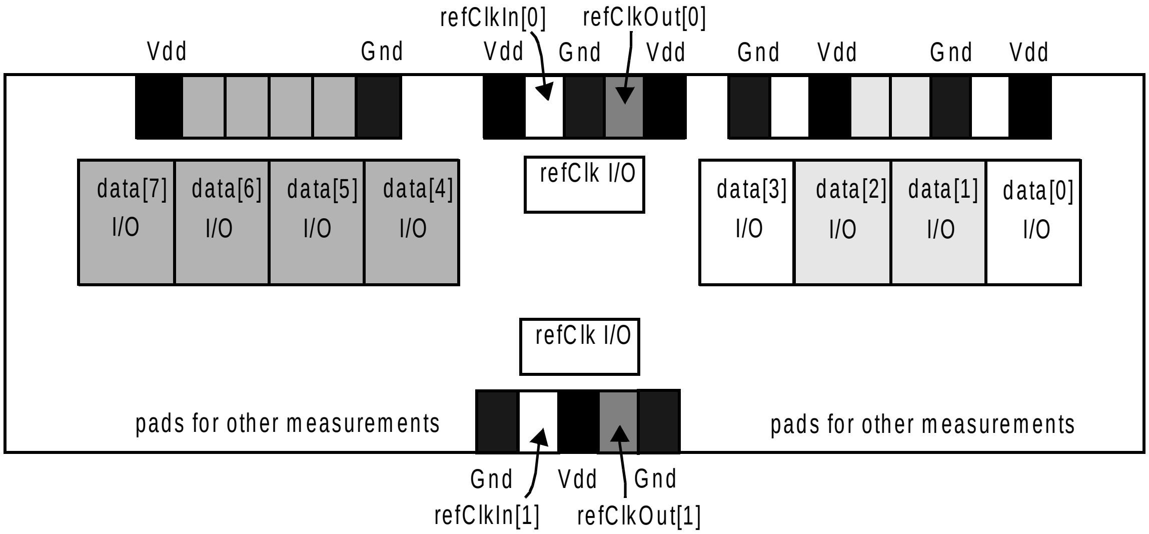


Figure 3.2: I/O pad placement. Data pads are laid out with different signal return configurations to study cross-talk and refClk pads are located in different parts of the chip to evaluate dynamic phase noise tracking.

signals pins, and to study whether refClk[0] shows more phase noise correlation with the data signals, when compared to refClk[1]. The three different receiver clock generation strategies also allow an evaluation of the dynamic phase noise characteristics of the interface signals and the advantages or disadvantages of jitter tracking in receiver timing recovery.

The I/O pads are laid out with different signal return configurations, also shown in Figure 3.2, to study cross-talk in parallel links: data[0] and data[3] are isolated signals situated in the middle of a pair of supply and ground pads; data[1] and data[2] share the same supply and ground pads; and finally data[4], data[5], data[6], and data[7] form a signal cluster of 4. This pad arrangement permits a comparison of the inter-signal cross-talk at the package and bond wire levels of the three different signal return configurations to help study the potential pin saving in single-ended signalling. The die occupies a total area of 1.7 x 3.8mm^2, and a die photo is shown in Figure3.3.

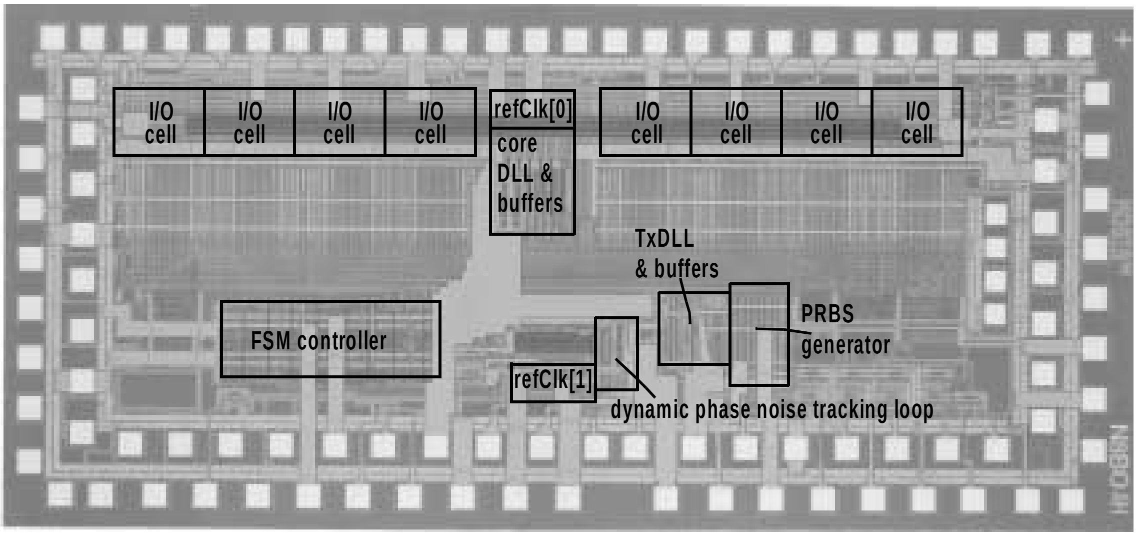


Figure 3.3: Die micrograph.

3.2 Design

The transceiver architecture supports per pin timing adjustment by adding a variable delay to the global receiver clock in each I/O cell. Figure3.4 shows the receiver portion, which uses current-integrating receivers [74] for data reception for the benefits described earlier in Section 2.5. In the calibration phase, a clock sequence is sent along each data line, and each variable delay element is adjusted so that the local receiver clock (RxClk[0],..., RxClk[n]) is centered around the transition edges of the calibration clock sequence at the end of the calibration phase. Then a 90^o phase shift is added to each delay element such that during the subsequent data transmission phase the local RxClk is aligned in phase with the incoming data stream. Since the actual receiver is used for timing calibration, this architecture calibrates and compensates for all static inter-signal timing errors at the receiver.

3.2.1 Transceiver Implementation

Figure 3.5 shows the actual transceiver implementation, which uses phase interpolation to realize the variable delay element. Using interpolation allows a 360^o unlimited phase adjustment range and hence there is no restriction on the timing of the incoming refClk and data signals relative to the on-chip clocks at the receiver[2].

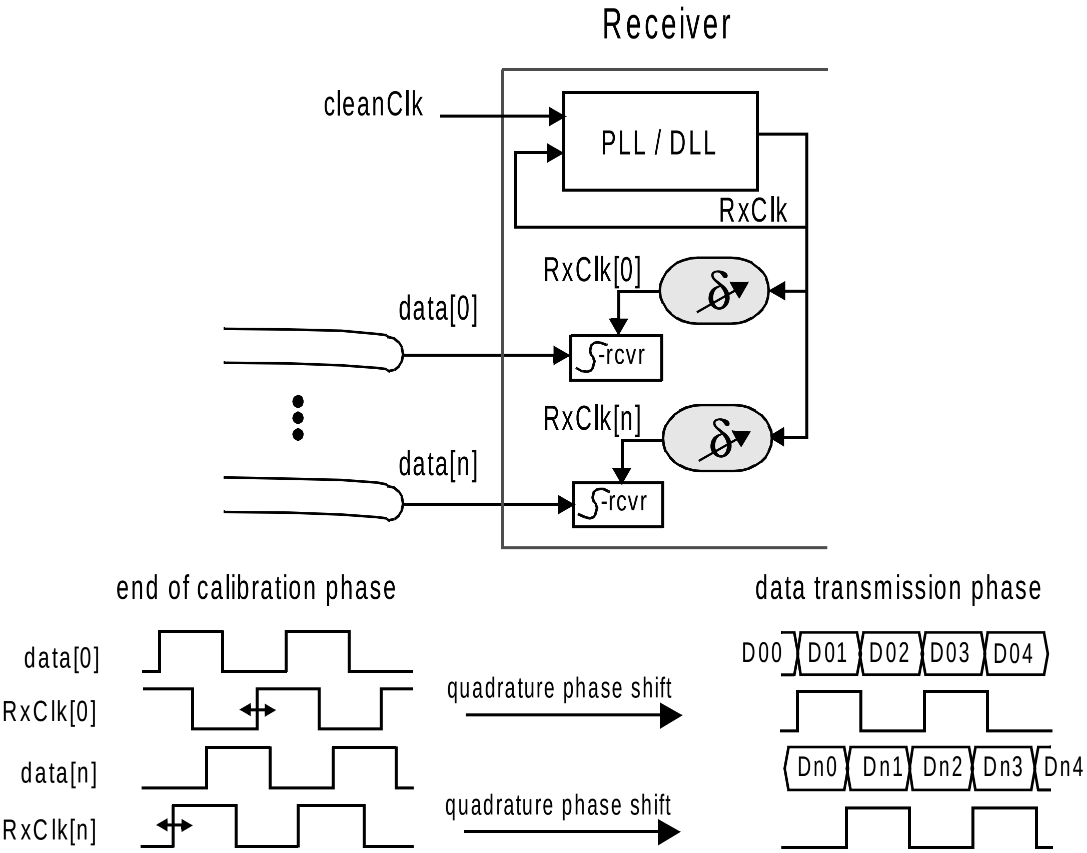


Figure 3.4: Receiver section of transceiver architecture. A variable delay is added locally to the global receiver clock to support per pin skew compensation.

The core data loop consists of a shared core delay-locked loop (DLL), a shared finite-state-machine (FSM) controller, and the 8 bidirectional I/O cells. This core data loop is based on the design of the dual-loop DLL introduced earlier in Section 2.4 [64], [65]: the core analog loop is shared, but the outer digital loop components, like the phase muxes and phase interpolator, are replicated within each I/O cell. The digital control logic is shared not only among the I/O cells, but also with the dynamic phase noise tracking loop that will be discussed in detail in Chapter 4. A transmitter delay-locked loop (TxDLL), not shown in the figure, generates a transmitter clock (TxClk), and a finite-state-machine clock (FSMClk) at a divided-by-4 frequency. The data source to each I/O transmitter can either be a pseudo-random bit sequence (PRBS) or an externally loaded data pattern.

[2] The interface functions correctly when the maximum inter-signal timing skew between any pair of the refClk and data signals is within one cycle time (i.e. two bit times).

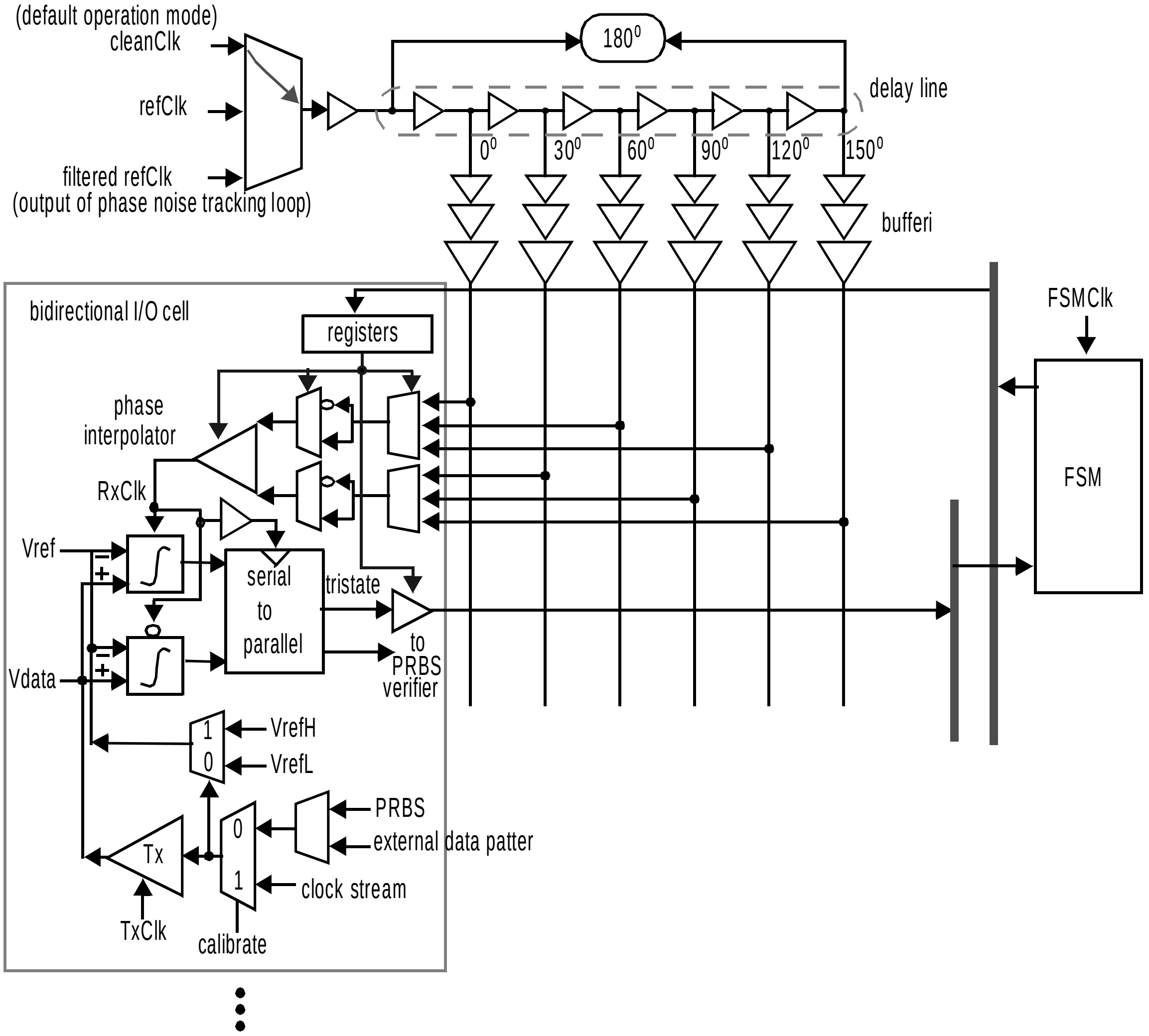


Figure 3.5: Transceiver implementation (core data loop). Core DLL generates 6 differential clocks at 30^{o} spacings that are phase-interpolated to generate a local receiver clock (RxClk) of unlimited phase range in each I/O.

The core DLL generates 6 differential clocks at 30^{o} phase spacings that are distributed to all the I/Os using the low-swing differential symmetric-load buffers described earlier in Section2.3 [62], [63], [61]. In the default operation mode, a 'clean' system clock (cleanClk) is used for clock generation [3]. As mentioned earlier, on start-up, the chip undergoes a calibration phase during which the transmitter sends a clock stream along each data line. The data pins are calibrated sequentially using the shared FSM. Inside each

[3] For simplicity, the same 'clean' system clock is used in both the transmitter chip and the receiver chip to avoid the overhead needed to handle any frequency difference, i.e. plesiochronous timing, between the transmitter and receiver clock sources.

I/O, the two current-integrating receivers serve as phase detectors that compare the phase of the incoming clock stream to the phase of the local RxClk. In calibrating a data pin, the FSM takes a majority vote of all 8 early/late samples collected from its current-integrating receivers in each cycle and decides which direction to adjust the phase controls. When its RxClk is centered around the transition of the incoming clock stream, as shown in Figure 3.4, the FSM quadrature-shifts the phase controls and stores them inside the registers in the I/O cell. This required quadrature phase shift is easily performed by a change in phase controls -- the phase moves by three 30^{o} clock spacings. Then the FSM advances to calibrate the next pin. After all pins are calibrated, the FSM turns off. Data transmission begins, and the stored phase controls inside each I/O keep its RxClk aligned in phase with the incoming data stream. A refClk signal is not needed in this operation mode.

3.2.2 I/O Front-End

Figure 3.6 is a schematic of the I/O front-end. The transmitter employs 2:1 multiplexing to transmit data on both clock phases. The open-drain output driver is broken down into 4 segments at a ratio of 1:2:4:4 to give 11 levels for swing control. The swing control logic is embedded inside the transmitter datapath. The reference-select mux is broken down into 4 similarly weighted segments to adjust the delay of Vref to match the delay of the transmit signal. The two shared reference voltages (VrefH and VrefL) are externally adjusted to measure internal voltage margins. The signal wire is terminated on each side with a PMOS resistor, whose gate voltage is adjusted externally for impedance control. On-chip voltage samplers are placed at both the data and Vref nodes to probe the internal signals. Finally, two current-integrating receivers [74] are used to integrate the input over the entire bit time, filtering out the high-frequency noise and the potential glitch caused by mismatched Vref and transmit data delays.

The maximum signal swing is bounded by the drain voltage that makes the open-drain driver NMOS transistors go out of saturation. In older technologies, this is normally set by the threshold voltage of the transistors[4]. The allowable transmit swing can only be half of

[4] Designers have tried to cope with this limit by using a separate (lower) supply to power the pre-drivers.

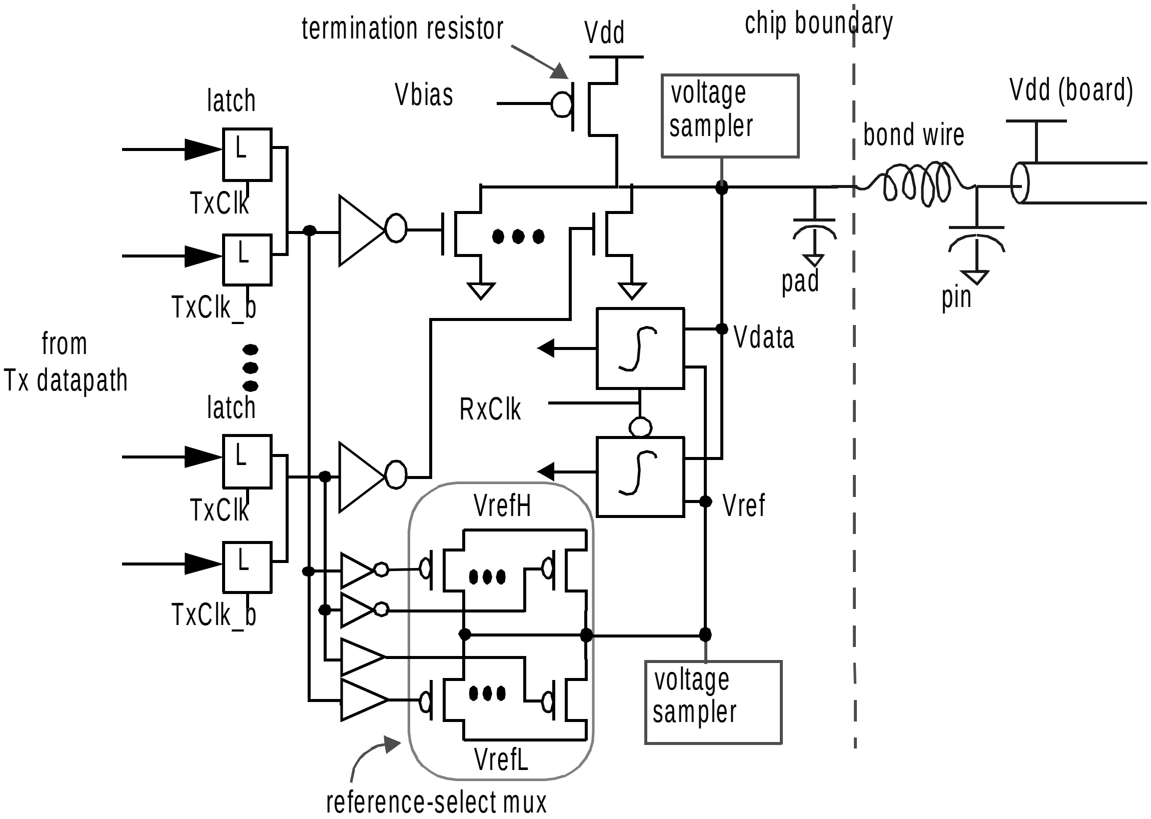


Figure 3.6: I/O front-end, using segmented open-drain output driver and current-integrating receivers.

this voltage limit because of the simultaneous bidirectional operation. However, in most sub-micron processes, minimum channel length devices are usually velocity saturated. This is the case with the NMOS transistors used for the output drivers in the test chip, and hence helps to increase the allowable total swing while keeping the output resistances of the devices high: simulations show that the output resistance of the widest device is well above 1kΩ when its drain voltage falls 1V below its gate voltage.

The output driver is effectively a 4-bit hybrid-code, current-summing DAC (digital-to-analog converter). The linearity in its output current is determined mostly by mismatches in the driver legs, while the output voltage linearity depends on three factors: linearity of the termination resistor, mismatches in the driver transistors, and maintaining high output resistances in these transistors. Unlike in DAC designs, the output linearity is not an important design issue here. The purpose of having adjustable swing is simply to facilitate the voltage noise measurements.

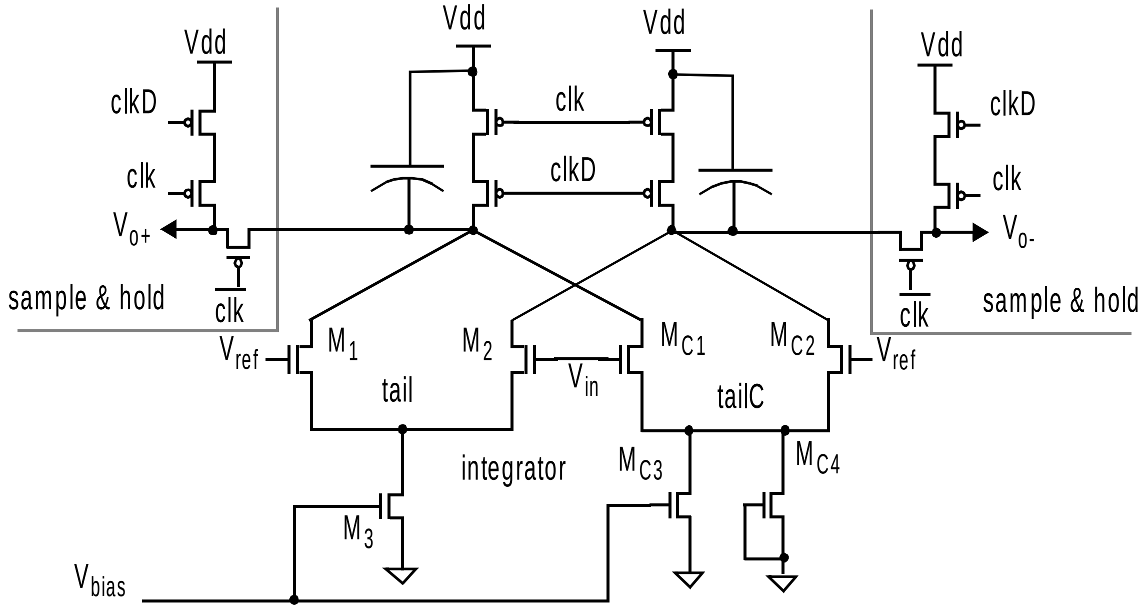


Figure 3.7: First two stages of current-integrating receiver used in test chip.

As signal frequency increases, the effectiveness of an off-chip termination resistance decreases due to the presence of the stub contributed by packaging elements. On-chip termination has been shown to be a better approach. Since well-controlled resistor options were unavailable in the process used, the termination is realized with a PMOS transistor with adjustable gate voltage, designed to give 50 Ω at the middle of the maximum bidirectional signal swing (or approximately the bottom of the maximum unidirectional swing) when its gate bias is at Gnd. As signal swing increases, the rising termination resistance is counteracted by the decreasing output current as the drain voltage of the NMOS drivers drops, leading to fairly linear output signal swings as found in simulations. For the same reason, the transmit output swing is also fairly linear in bidirectional signalling, meaning that the bidirectional signal swing is roughly equal to the sum of the two superimposed unidirectional swings.

The input receiver is a current-integrating receiver based on the design of [74]. Figure 3.7 is a schematic of the first two stages. In this implementation, an NMOS differential pair is used for the integrator because the input signals are referenced to Vdd.

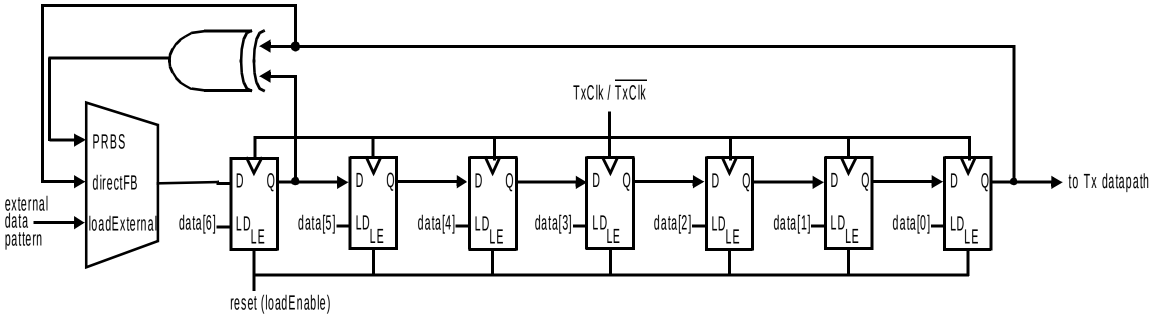


Figure 3.8: Linear feedback shift register (LFSR) chain used for PRBS data generation.

3.2.3 Transmitter

3.2.3.1 Data Generation and Datapath

To facilitate the proposed noise measurements, the transmitter data source must be programmable; to test the functionality, the links should operate correctly for any random data pattern. Therefore, the transmitter can transmit either a pseudo-random bit sequence (PRBS) or an externally loaded data pattern. The external data pattern can be recycled so that the noise sources induced are also periodic signals and can be easily measured using the voltage samplers or oscilloscopes. Besides, concatenating the serial-load paths of all the PRBS generator chains forms part of the scan chain for debugging purposes . Figure 3.8 shows the 7-bit maximum-length linear feedback shift register (LFSR) chain used: an M-sequence with primitive function

$$f(x) = x^7 + x^6 + 1 \tag{3-1}$$

and a repeating period of 127 (= 2^7-1) bits. The outputs of two such chains are multiplexed at the pre-driver as shown in Figure3.9, which illustrates the entire transmitter datapath. This structure also allows an external 14-bit data pattern to be transmitted repeatedly. Swing control logic is embedded inside the latches so as to reduce the number of stages after the 2:1 multiplexing, hence reducing the jitter of the output signal. The transmitter datapath is designed for a clock cycle time equal to 8*FO4 (FO4 is equal to 193ps -- under nominal conditions using typical transistor models -- in this process, giving a maximum clock rate of 650MHz or a data rate of 1.3Gbps).

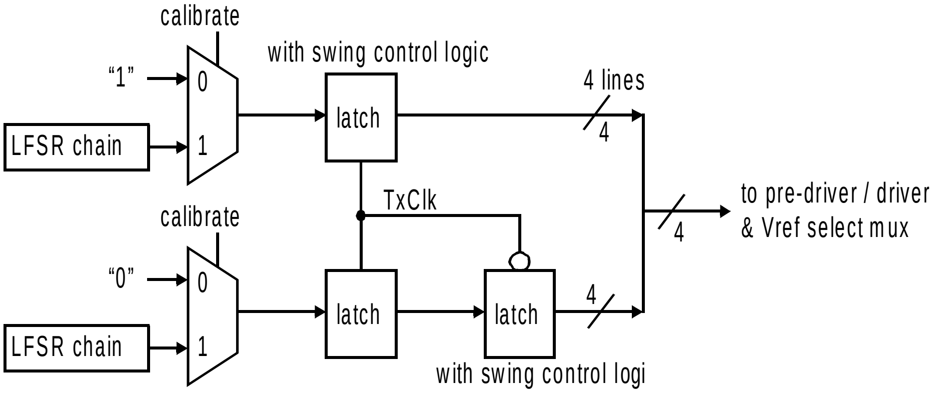


Figure 3.9: Transmitter datapath. Swing control logic is embedded inside the latches.

The multiplexed output of two maximum-length PRBS sequences is not necessarily a maximum-length PRBS sequence. As there is significant inter-signal cross-talk among signal pins in a parallel link, it is also important that the data across the entire interface is of maximum randomness to test the robustness of the interface. These two concerns are addressed using 8 parallel LFSRs that are appropriately time-shifted from one another and multiplexed in combinations as suggested by [78]. Load-able flip-flops are used to set the correct initial values in each chain upon reset. Therefore, the multiplexed transmitter output in each data pin transmits a maximum-length sequence, and a sequence formed by traversing all data pins in consecutive bit times is also a maximum-length sequence.

3.2.3.2 Clock Generation

A transmitter delay-locked loop (TxDLL), shown in Figure3.10, phase-locks the output of the pre-drivers by locking the output phase of a dummy pre-driver. The design is a single-loop analog DLL using symmetric-load buffers in the delay line and replica feedback bias. To accommodate the large number of inverter stages in the dummy inverter chain and to increase the adjustable delay range of the delay line (by increasing the number of buffer stages), the entire feedback delay does not fit into a half clock cycle. Therefore, the feedback path (highlighted in the figure) h as odd inversions, effectively locking all the stages in this feedback path to 360^o. Both the transmitter clock (TxClk) and the finite-state-machine clock (FSMClk), at a divided-by-4 frequency, are generated from TxDLL.

3.2.3.2 Clock Generation

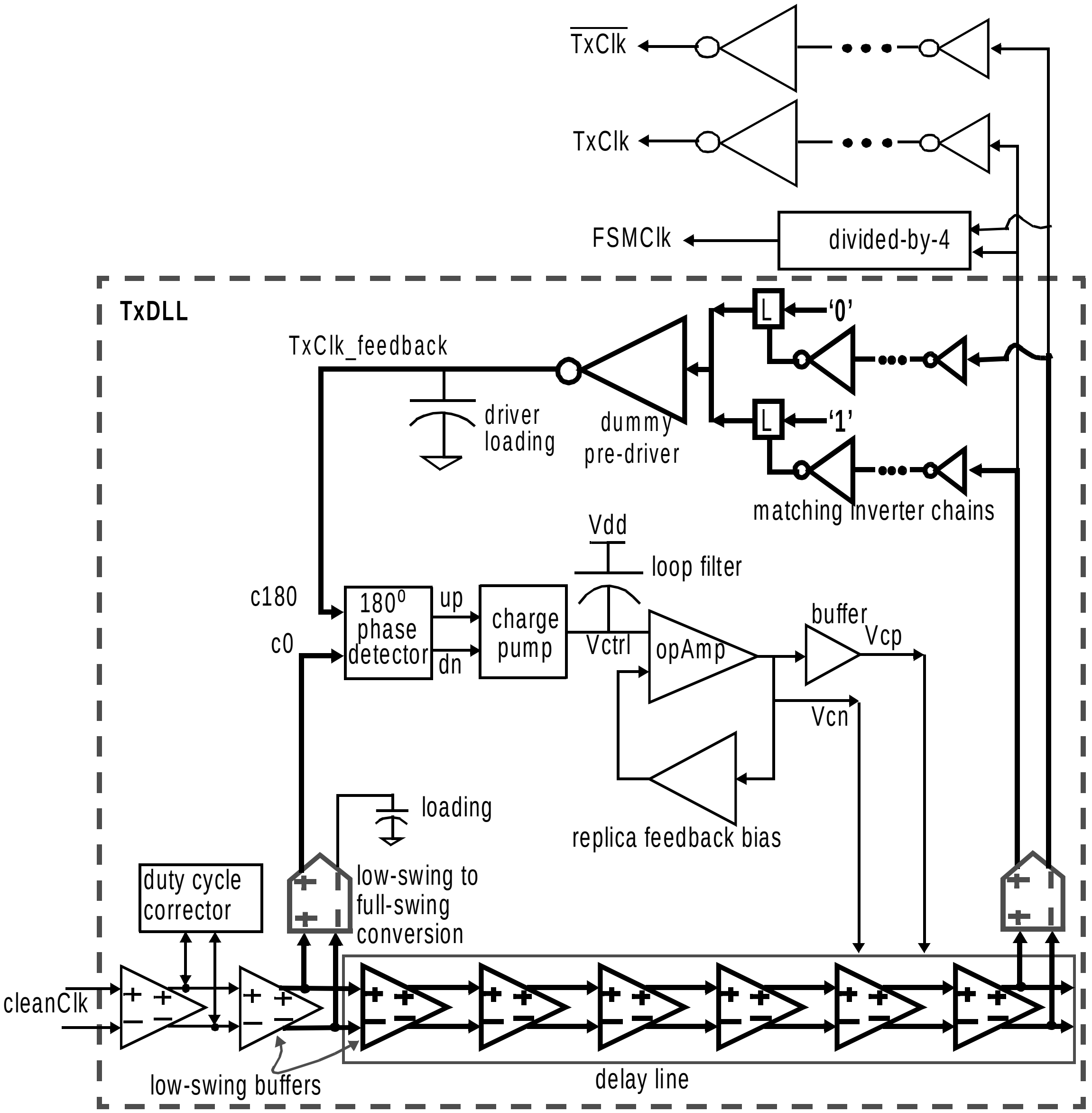


Figure 3.10: Transmitter clock generation. The enclosed area is the transmitter delay-locked loop (TxDLL). Both TxClk and FSMClk are generated from TxDLL.

TxDLL operates from its own isolated power supply (Vdd_TxDLL). The adjustable delay range of the delay line limits the DLL's locking range. Nevertheless, the simulated locking range at nominal conditions is from below 250MHz to above 750MHz and bounds our intended operating frequency by large margins in both directions. (However, just like the PRBS generator, the subsequent CMOS clock buffers are designed for a clock period equal to 8*FO4, corresponding to about 650MHz). The DLL has to be properly reset in the event of exhausting the delay range on either end before finding lock.

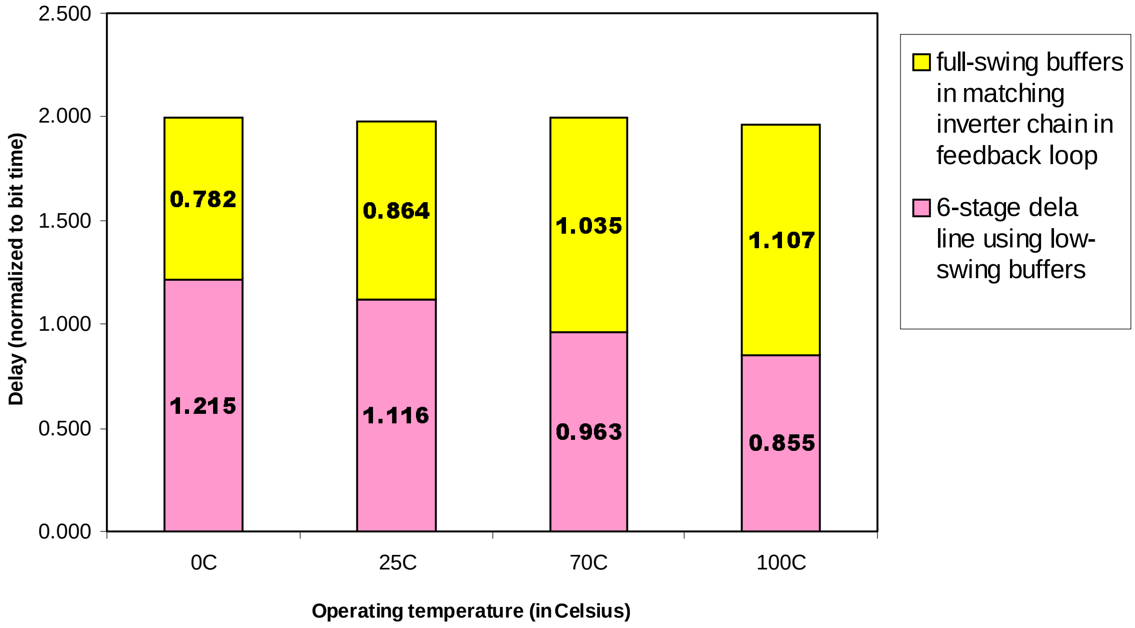


Figure 3.11: Total delay in feedback loop in TxDLL, simulated at different operating temperatures using measured process parameters. The feedback loop locks to one clock cycle.

The performance of TxDLL directly affects the performance of the links: any phase error in TxClk translates into phase errors in the transmitted refClk and data signals. The phase output of TxDLL is arbitrary, hence, so is the phase of any transmitted signal. This is not a problem as the receiver timing recovery allows full-range clock recovery. However, as the links are calibrated only at chip start-up in the default operation, any deviation in phase in the transmitter outputs after the c alibration reduces the receiver timing margins. Therefore, both high-frequency jitter and low-frequency phase drift in TxClk may impact performance and a periodic calibration may be needed if the low-frequency phase drift is significant.

Figure 3.11 plots the simulated total delay in the feedback loop in TxDLL, at different operating temperatures using nominal values of measured process parameters from the wafers. The feedback loop locks to one clock cycle (i.e. two bit times)[5]. As temperature increases, the CMOS full-swing buffers in the matching inverter chain are slower, forcing

[5] There is about 30ps difference between the shortest and longest total delays, and the behavior is not monotonic with temperature change. Possible reasons are simulation artifacts as well as differences in the phase error in TxDLL when in lock at different temperatures.

the delay of the low-swing buffers in the delay line to speed up such that the total delay locks to one cycle.

Duty cycle distortions in TxClk also translate into duty cycle distortions in the transmitted refClk and data signals, and reduce their timing margins at the receiver. Therefore, in all our measurements, the input cleanClk is set to 50% duty cycle, and any distortion is corrected or reduced by a duty cycle adjuster before the delay line , as illustrated in Figure3.10. Furthermore, as the low-swing to full-swing conversion and the subsequent CMOS inverters (clock buffers) may introduce duty-cycle distortions in the single-ended TxClk output, a well-matched complement $\overline{\text{TxClk}}$ is generated in parallel and distributed to the output drivers to avoid duty cycle distortions propagating to the transmit output signals[6].

On the other hand, any phase error in FSMClk is unimportant in this design because its long cycle time is much larger than the minimum cycle time needed for the FSM to function correctly.

3.2.4 Receiver

In addition to the timing errors in the received signals, the receiver timing margins of the links are also greatly affected by the accuracy of the skew calibration, the accuracy of the quadrature phase shift performed subsequent to the skew calibration, and the jitter in the local receiver clocks (RxClk[7:0]). These factors depend largely on the performance of the DLL used to generate RxClk[7:0] and the phase control logic used to calibrate the timing skew.

3.2.4.1 Clock Generation

The receiver architecture is based on the design of the dual-loop DLL described in Section2.4 [65]. The shared core DLL generates 6 differential clocks at 30 ^{o} phase spacings that are distributed to all the I/Os . Each I/O cell has it own set of phase muxes and phase interpolator, whose settings are determined in the skew calibration when the chip starts up to position the local RxClk to maximize the link timing margin.

[6] For simplicity, however, only TxClk is shown in most figures.

The dual-loop architecture allows a much wider locking range than the single-loop TxDLL: the simulated locking range at nominal conditions is from below 50MHz to about 1GHz. However, the CMOS clock buffers for the local RxClk are designed for a clock period equal to 8*FO4, corresponding to about 650MHz, to match in speed with the transmitter. The design issues of the DLL are discussed in detail and its design trade-offs thoroughly explored in [65] and [79]. In our design, special attention was paid to factors contributing to timing errors.

The DLL (the core loop and all the clock buffers) runs on another isolated power supply (Vdd_dataloop) to prevent the power supply noise generated by the I/O output drivers and by the digital blocks from affecting the noise-sensitive DLL circuits.

The 6 differential clocks are distributed to all the I/Os using low-swing differential symmetric-load buffers. These clock lines are extremely long, running on top of the I/O cells across a total distance of approximately 2.3mm in the top-layer metal (Metal-4), and shielded completely with a Metal-3 power supply ‘plane’ (a Metal-3 wire wide enough to terminate all fringing fields from the clock lines) underneath except at points of connections to the phase mux inputs in each I/O. The core DLL and the clock buffers are placed in the middle, as can be seen from the die micrograph in Figure3.3, so that the clock lines are driven from their midpoints to reduce the RC delays to the furthest away cells. For the DLL to operate at the required high speed with low jitter outputs, all low-swing stages are designed to have a fanout equal to roughly 2.67. Therefore, three clock buffer stages are needed to drive the 8 sets of phase muxes and the large wire loads. These buffer stages prevent any difference in loadings between the selected clock lines and the unselected clock lines from inducing delay mismatches among different buffer stages in the delay line. They also prevent the toggling phase mux and interpolator controls from coupling back to the delay line during calibration.

Using low-swing differential clock buffers results in very low jitter clocks, but dissipates substantially more power than using CMOS inverters. The area of each low-swing buffer is also at least 3 times larger than a CMOS inverter of comparable strength. As a result, the area overhead of the clock buffers can be substantially reduced if CMOS inverters are used instead. M oreover, using CMOS inverters allows higher fanout per buffer stage and hence potentially reduces the number of buffer stages and further saves area. A higher buffer fanout also makes the architecture more scalable in terms of expanding to wider parallel interfaces. In Chapter 4, we will evaluate the cost overhead in implementing this optimal skew compensation scheme with full-range compensation and minimal-jitter clocks.

As the local RxClk is quadrature-shifted in phase after skew calibration, the precision of the quadrature phase shift directly affects the receiver timing margin. Therefore, having even 30^o clock spacings is very important. The buffer stages in the delay line need to be well matched, and the phase error of the PD must be minimized.

An imperfect clock duty cycle in cleanClk causes the clock spacings in region #6 and region #12 to increase or decrease at the expense of each othe . Therefore, similar to TxDLL, a cleanClk with 50% duty cycle is used as the DLL input in the default operation mode, and any duty cycle distortion is further corrected by a duty cycle adjuster before the delay line. This duty cycle adjuster is especially important in the second operation mode where the receiver timing recovery tracks dynamic phase variations in refClk: a differential buffer[7] takes the single-ended incoming refClk and VrefH as inputs, and the output passes through two additional buffer stages before going into the delay line. Any duty cycle in this refClk signal propagates through all subsequent stages. Moreover, if VrefH is not exactly at the middle of the refClk swing, the duty cycle of the subsequent stages would be even worse.

Even with perfectly matched buffer stages and 50% duty cycle in the delay line outputs, a phase interpolator can introduce non-linearity in the interpolation steps. The linearity depends on the interpolator design itself, and on the relationship among the transition times (time constants) of the two input clocks being interpolated, their phase separation, and the transition time of the interpolator output. In general, good linearity is achieved when the transition times of the two input clocks [8] are comparable to the

[7] This is a simplified picture: the input stage is actually a differential 2:1 mux choosing between refClk[0] and refClk[1].

[8] The transitions times of these two inputs clocks are almost always the same, at least nominally.

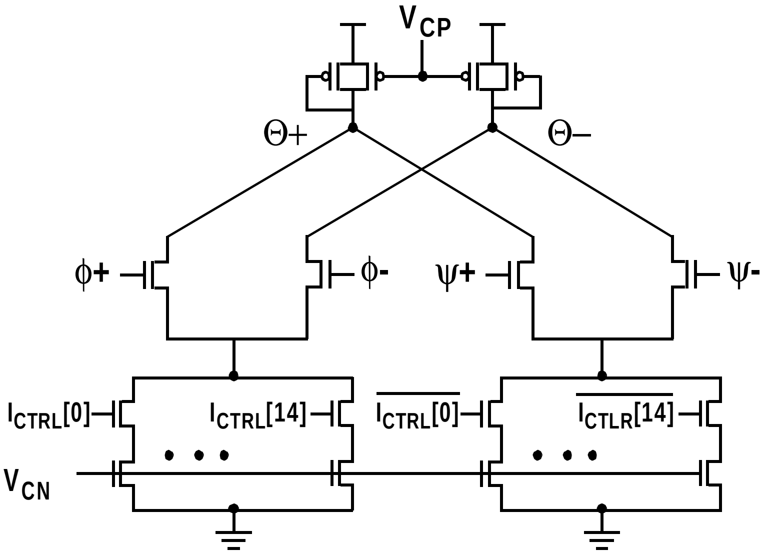


Figure 3.12: Phase interpolator used in the test chip.

transition time of the interpolator output, *and* when their phase separation is less than 2/3 of their transition times [79]. Figure 3.12 illustrates the phase interpolator design used in the test chip. All these interpolators, as well as all the phase muxes and clock buffers in the entire DLL, share the control voltages generated in the analog loop for the delay line buffers. Therefore, the delays of all low-swing stages scale with the link operating frequency. This delay scaling results in a number of benefits. For instance, the transition times of the phase interpolator inputs and output scale proportionally, allowing the design to be optimized for phase linearity across different link speeds. Another advantage is the scaling of the dynamic phase noise tracking bandwidth with the data rate, which we will study in detail in Section4.2 .

The interpolator uses 15-bit thermometer code current control. Using thermometer code increases the number of registers needed inside each I/O cell and the number of control wires routed across the chip from the FSM, but guarantees monotonicity in the phase output and avoids glitches that may otherwise result from delay mismatches in the interpolator controls had binary codes been used instead. With perfect device matching, Figure 3.13 shows the simulated phase steps when the on-chip clocks are running at 600MHz. There are altogether 16 phase steps in each 30^o clock region. The largest phase steps (marked type C in the figure) occur not with a change in interpolator weights, but

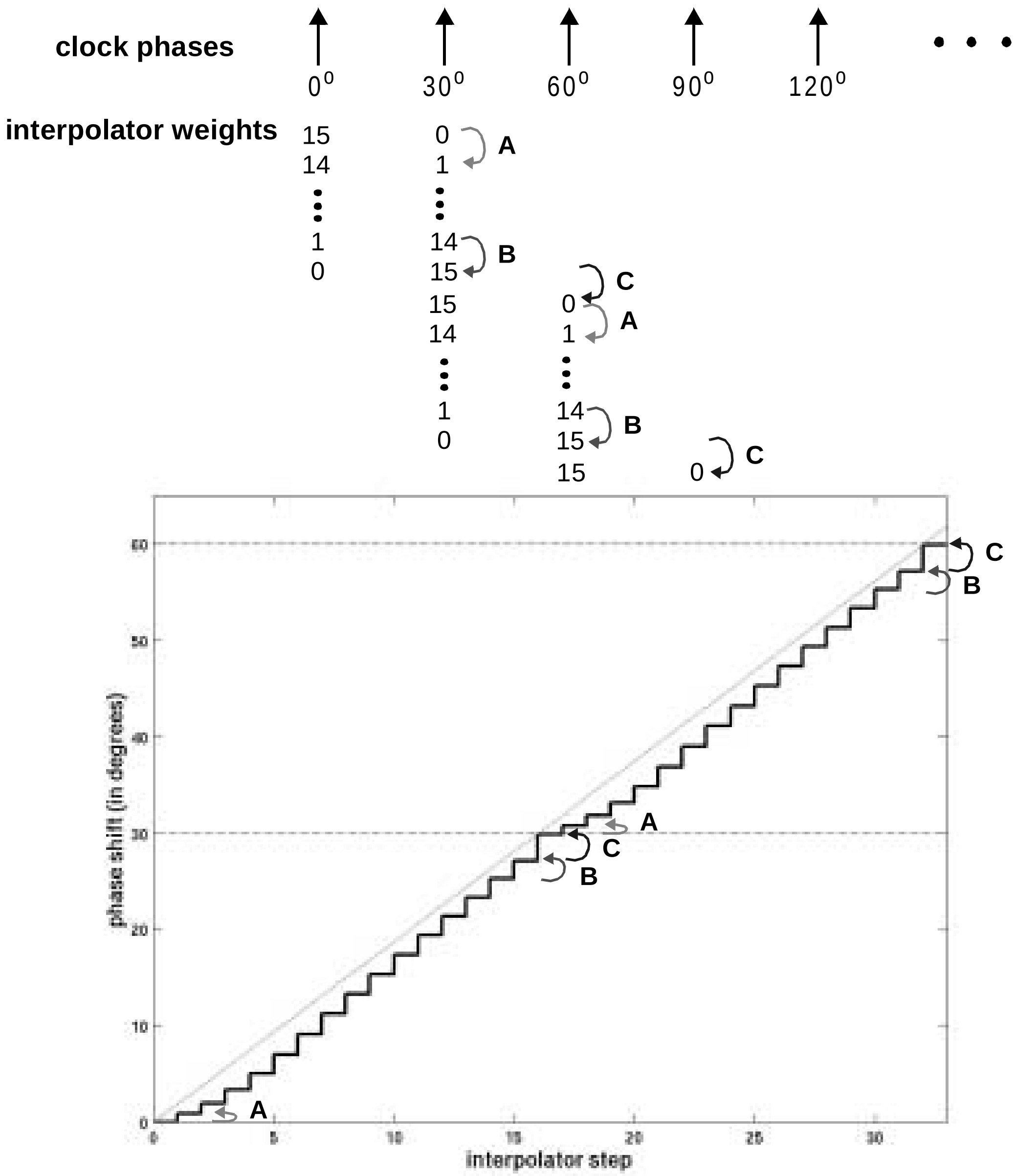


Figure 3.13: Simulated phase steps for two consecutive 30° clock regions. The horizontal broken lines represent the ideal phase boundaries, while the angled line represents the ideal phase shift if all the steps are even.

rather with a change in one of the input clocks at the boundaries[9]. Had all the steps been even, the nominal step size would be about 8.7ps. The simulation results show that the smallest steps (marked type A in the figure) are 4ps in size, while the largest ones (type C) are 12.7ps, which set the limit for the maximum phase error in the timing skew calibration.

[9] Had the phase boundaries been seamless, there would only be 15 phase steps inside each 30° clock region.

The interpolator output goes through a low-swing to full-swing converter to generate RxClk. A well-matched $\overline{\text{RxClk}}$ is also generated in parallel. Since the two current-integrating receivers operate on both RxClk and $\overline{\text{RxClk}}$ (using opposite phases), the effects of duty cycle distortions are greatly reduced.

Finally, any low-frequency phase drift in the local RxClk can also reduce a link's timing margin. As most of the circuit stages in the entire DLL share the control voltages generated in the analog loop for the delay line buffers, their delays also scale with the delay of the delay line, and hence are kept fixed as temperature changes -- the only exception is in the differential-to-single-ended converter and its subsequent buffers, which have positive temperature coefficients. In simulations, the longest possible total delay from cleanClk to the RxClk (i.e. when the RxClk is at the last phase step of clock region #12) drifts (increases) by about 8.4% of T_{bit} (69.8ps) when the temperature changes from 0^{o}C to 100^{o}C.

3.2.4.2 Datapath and Data Verification

On-chip data processing often runs at a lower speed than the off-chip I/O. Therefore, the receiver usually implements serial-to-parallel conversion and byte alignment in addition to error detection (and correction). In this per-pin skew-compensated system, the local receiver clocks RxClk[7:0] are all skewed relative to one another. Since the compensation is full-range, the skew between two local RxClk outputs can be as large as two bit times, making byte alignment a challenging task.

To enable testing of each individual data line and to facilitate various noise measurements on each individual pin in response to different combinations of excitations, the receiver datapath and data verification are done in a bit-by-bit manner, as illustrated in Figure 3.14. The PRBS verifier uses the same LFSR chain as the PRBS generator, and compares the value of the received data with the value of the feedback node using an exclusive-or (XOR) function. The PRBS verifiers are clocked by the locally skewed IOClk, which is generated from the locally skewed RxClk. In this way, the need for byte alignment is eliminated from the error detection function. The outputs from both verifiers are ORed to generate the bit error signal (bit_error), which is then muxed with the bit

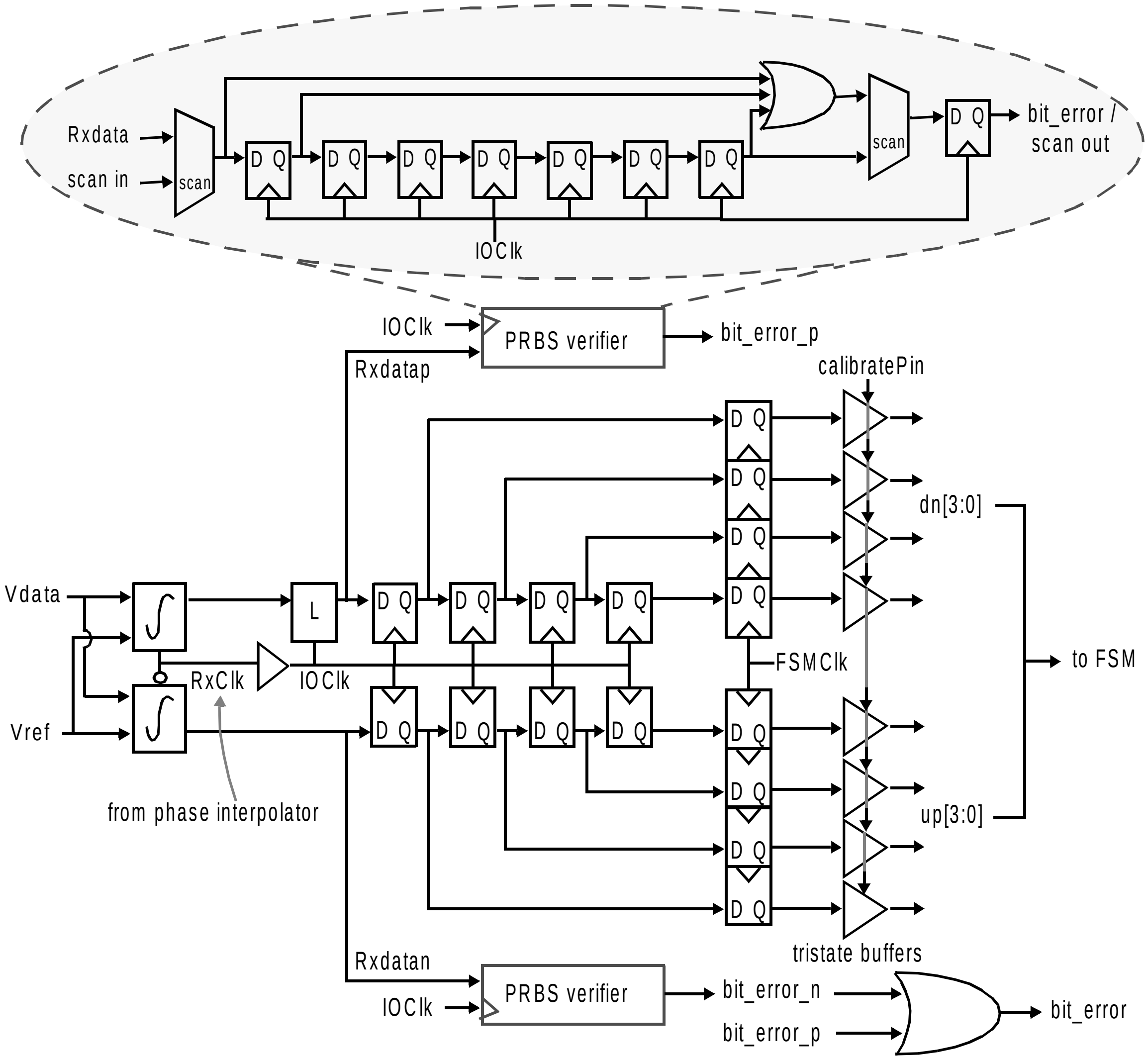


Figure 3.14: Receiver data path for each data line.

error signals from all other data pins and routed to an I/O pad. Consequently, the performance of each data channel can be measured separately. In addition, the bit error signals from all pins are ORed, and the output goes to a one-shot hold circuitwhose output also goes off-chip such that any bit error in any channel can be recorded.

3.2.4.3 Phase Control Logic Design

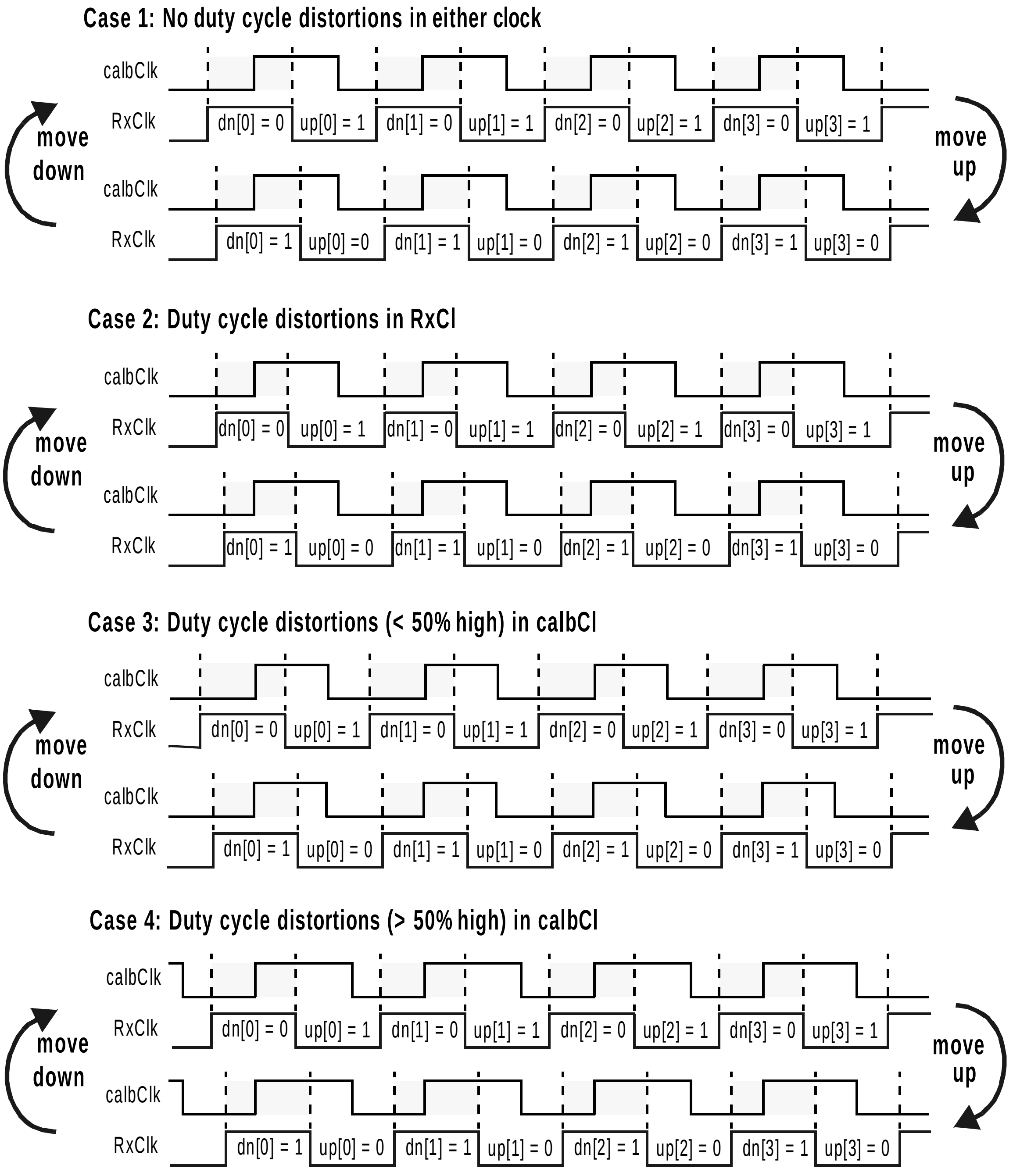
The design of the FSM is extremely important as it determines the accuracy of the static skew calibration and dynamic phase noise tracking. The FSM runs at a divided-by-4 frequency of the I/O clocks, and takes a majority vote of the 8 early/late phase bits (up[3:0] and dn[3:0]). The state machine goes through two modes during phase capture: a fast mode during which it updates the phase controls every (FSMClk) cycle to speed up calibration; and a slow mode, triggered when the FSM determines that the lock point is close, during which it updates the phase controls every 5 cycles to reduce the dither jitter to one interpolator step.

The calibration process proceeds as follows: in the default operation mode, no refClk is needed, and the FSM calibrates the data pins sequentially. As a measure to improve testability, the transmitter and the receiver have separate external calibration controls, calibrateOut and calibrateIn. On chip start-up, all storage elements in the chip are reset. To start the skew calibration process, calibrateOut is asserted so that the transmitter starts sending the calibration clock sequence in all channels. At the same time, calibrateIn is asserted to start the receiver and the FSM, which then asserts calibratePin[0] to enable the tristate buffers and registers in data[0]. As explained earlier, the current-integrating receivers serve as phase detectors that compare the phase of the incoming calibration clock to the phase of the local RxClk. The FSM takes a majority vote of the 8 phase bits collected in deciding which direction to move the phase controls. The latency of the feedback loop (from the transitions in up[3:0] and dn[3:0], to the logic delay in FSM, to the effect of the phase control update on the interpolator output, to the next up[3:0] and dn[3:0] transitions) is 5 (FSMClk) cycles. To speed up the calibration process, the FSM updates every cycle without waiting for its effect to be seen, in what we call the fast mode. When the first phase reversal is detected, the slow mode is triggered, and the FSM starts to update every 5 cycles. It discards the phase bits from 4 cycles, and then takes a majority vote of the 8 phase bits in the next cycle, and then waits for another 4 idle cycles. The phase bits for these 5 consecutive cycles are normally the same, thus no useful phase information is lost in discarding the phase information from the 4 prior cycles. To avoid false lock due to AC noise, the FSM declares a lock only after 6 consecutive phase reversals. In the worst case, RxClk traverses 180^o before finding lock. Since the phase acquisition is digital, the skew calibration time is directly proportional to the number of phase steps, or the total phase, traversed. The FSM *holds* the phase settings, while storing their 90^o-shifted values in the registers inside data[0]. It then advances to the next pin (data[1]), disabling calibratePin[0] and enabling calibratePin[1]. In the next skew calibration, the FSM thus starts with the phase controls of data[0]. This helps to reduce the calibration time of the subsequent pin given that the inter-signal skew between two consecutive pins is often much smaller than the 180^{o} worst case scenario. The process is repeated until the last data pin (data[7]) is calibrated. Then the FSM turns off, and data transmission proceeds.

Despite the measures taken to reduce duty cycle distortions, the incoming calibration clock (calbClk) and the local RxClk may still have imperfect duty cycles, and the effects can be detrimental if such distortions cause the phase locking algorithm to fail. Figure 3.15 illustrates the effects of duty cycle distortions in calbClk and RxClk on the resulting phase bits. All waveforms are shown with infinitely fast transitions, and the dithering phase step around the lock point is exaggerated to improve readability. Case 1 is the ideal scenario we have considered so far, where both calbClk and RxClk have perfect 50% duty cycle. RxClk dithers around the lock point, where it leads calbClk by 90^{o}. The majority vote decision is implemented using an 8-input adder, adding up up[3:0] and the complements of dn[3:0]. If the adder sum is 5 or above, RxClk moves up by one phase step; if the sum is 3 or below, RxClk moves down by one phase step.

If RxClk alone has duty cycle distortions, as shown in Case 2, RxClk would still be centered around the transitions of calbClk, and the resulting up[3:0] and dn[3:0] signals are similar to Case 1. It is worth noticing that a quadrature phase shift at the end of the calibration still maximizes and *centers* the timing margins of *both* current-integrating receivers as if there is no duty cycle error in RxClk. The figure illustrates the results when the duty cycle is less than 50% high, but it is easy to see that all the above observations also hold for cases where the duty cycle of RxClk is greater than 50%.

If the duty cycle distortions are in calbClk instead, the lock point moves around the ideal lock point by an amount that is determined by the amount of duty cycle distortion, as illustrated in Case 3 where the duty cycle of calbClk is less than 50% and in Case 4 where the duty cycle of calbClk is greater than 50%. In both Case 3 and Case 4, a quadrature phase shift from the lock point as determined by the FSM results in suboptimal timing


Case 1: No duty cycle distortions in either clock

Case 2: Duty cycle distortions in RxCl

Case 3: Duty cycle distortions (< 50% high) in calbCl

Case 4: Duty cycle distortions (> 50% high) in calbCl

Figure 3.15: Effects of duty cycle distortions in the incoming calibration clock calbClk) and local RxClk on phase bits. The dithering step is exaggerated in the figure.

margins -- one of the current-integrating receivers may now ha ve its timing margin reduced and shifted from the center, and the timing margin reduction and shift are both determined by the exact amount of duty cycle erro . In cases where RxClk and calbClk both contain duty cycle distortions, the situation can be even more complex.

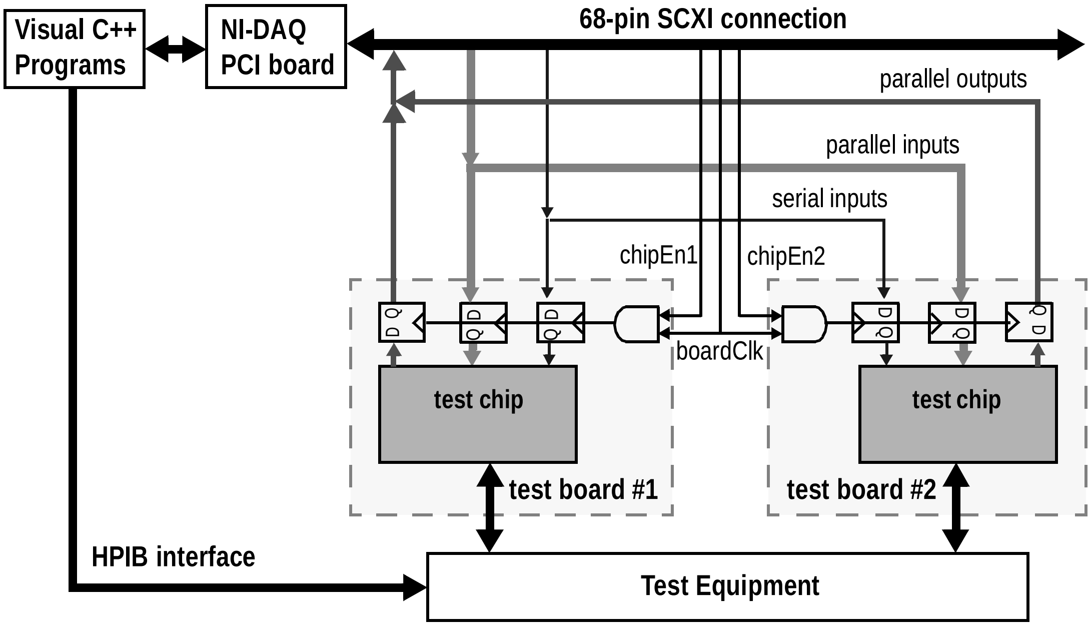


Figure 3.16: Test interface.

If the phases of the transmit signals drift significantly over time, the links can be recalibrated, perhaps periodically, by following the same sequence. This, however, reduces the effective bandwidth of the links. Alternatively, we attempt to track the transmitter output timing at the receiver by tracking the phase of one of the two refClk lines or of their low-pass filtered outputs. These two dynamic phase noise tracking modes share the same phase control logic as the skew calibration of the data links, and their operations are detailed in Chapter 4.

3.3 Measurements

We tested the data communication between two chips to measure link performance. Figure 3.16 shows the highly automated test interface used. All the digital control handshaking with the test chips is done using the NI-DAQ (National Instruments Data Acquisition) high-speed I/O interface [80], which allows all measurement tasks to be controlled using high-level programs (Visual C++ in this case), and simplifies measurement data collection and processing. The plug-in NI-DAQ PCI card supports a 32-bit parallel digital I/O interface with a wide range of configurable data transfer modes at speeds up to 76Mbytes/s (19Mbps per I/O line) through a multi-drop 68-pin SCXI connection. The various pieces of test equipment (pulse generators, scopes, power supplies etc.) are controlled manually, or in some cases by the HPIB (Hewlett-Packard Interface Bus) protocol for repetitive measurements.

The NI-DAQ board supplies the clock (boardClk) to the test boards. To allow the I/O lines to be shared between the two boards, boardClk is gated by complementary chip-enable signals (chipEn1 and chipEn2) on the boards. Input[10] controls are stored in registers. Start-up controls, such as swing controls, reference-voltage-select controls, and operational mode controls, are shifted into the test chips using a serial interface to reduce the number of on-chip I/O pads required since the design is pad-limited. These controls need to be reloaded on every new measurement run upon reset. This serial input also provides the scan function.

Interactive controls are implemented using parallel I/O. The interactive inputs include the 1-bit refClk_select to pick one of the two refClk signals, 3-bit pin_select to choose from the 8 data pins, 8-bit phase controls to set the RxClk in the selected pin to any of its 192 phase positions, calibration controls (calibrateIn and calibrateOut), as well as the input controls to the transmit datapath and receive data verification blocks. The interactive outputs include the 8-bit phase readings of the RxClk in the selected pin, bit error signals, and an output signal indicating the status of skew calibration.

As the transmit and receive signals are referenced to the chip supply (chipVdd) whereas the shields of the connectors in all the test equipment are shorted to the absolute Gnd (Earth), supply voltage translation is needed in our test setup. Figure3.17 illustrates the voltage levels in different components and the ir interconnections. The NI-DAQ interface is 5V TTL[11]. In order to provide the appropriate input logic levels for the chips, the lower supply of the PC (and hence of the NI-DAQ board) cannot be connected to Earth -- instead it has to float, and the logic “1” level is obtained by a 100 Ω-200Ω voltage

[10] ‘Input’ and ‘output’ as seen by the test chips. The opposite is the true from the NI-DAQ board’s perspective.

[11] Transistor-Transistor Logic, a widely used I/O standard in digital systems.

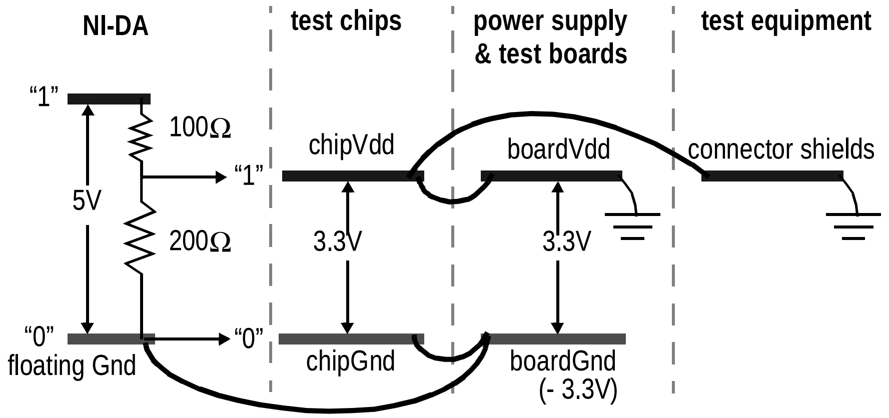


Figure 3.17: Voltage translation in the test setup.

divider. Larger resistors can be used to reduce power consumption, but we choose this resistance combination because it also provides proper t ermination for the 70Ω board traces used for these NI-DAQ interface signals. The logic "1" level produced by the test chips also falls in the logic "1" input range of the NI-DAQ (from 2V to 5V above the floating Gnd). The power supply voltage to the test chips is then set to -3.3V, and chipVdd is connected to the absolute Gnd -- the most stable supply. Hence, noise induced in the signalling system (where all I/O signals are referenced to chipVdd) is kept to a minimum. This arrangement also allows the interface signals in the parallel link to be connected directly to the test equipment; the measured voltage values are then negative.

3.3.1 Link Performance

Each data channel consists of bond wires, package wiring, PC board (GETEK) traces totalling more than 6 inches (3 inches on each board, drawn radially from the package to balance the traces), a coaxial cable ranging from 36" to 42", and two pairs of SMA connectors. Figure 3.18 illustrates one of the two test boards.

At 3.3V supply, the bidirectional links achieve a data rate of 2.4Gbps/pin (1.2Gbps in each direction) with no reception error observed for the entire testing period of more than 15 hours, representing a bit error rate (BER) less than 8 x 10^{-15}. At this data rate, the links require a minimum (quantized) signal swing of 193.5mV on each side in the pin with worst-case cross-talk (data[5])[12]. The chip dissipates less than 1W total power when all the links are running at 2.4Gbps/pin at their largest swings (about 430mV) and when all

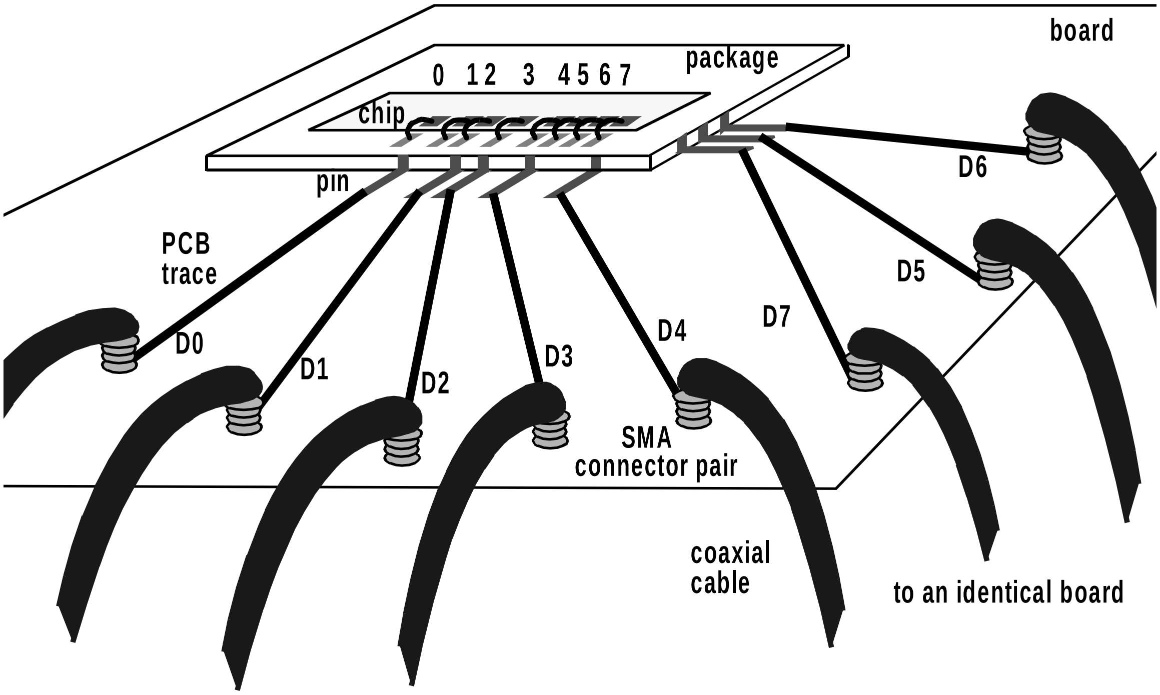


Figure 3.18: Test setup. Link performance is measured for data communication between two boards.

on-chip measurement blocks are active. Since the test board components draw a considerable amount of power, we obtain the total power figure by observing the difference in the total currents drawn from the power supplies when the chip is powered down compared to when the links are running under the above stated conditions.

The link speed is limited by the transmitter clock and data generation, which was designed for a clock period equal to 8*FO4. The transmitter fails to reliably transmit (or generate) the correct PRBS sequence above this speed as the chip heats up.

3.3.2 ransmitter Circuit Characterization

Despite the fact that the linearity of the output driver is not an important design issue, the measured unidirectional output voltage levels for 3 different pins, plotted in Figure3.19 , clearly show that the output (DC) levels are highly linear, with data[0] showing the worst differential nonlinear behavior with swing steps ranging from 32.4mV to 46mV. When we

[12] For unidirectional links running at 1.2Gbps/pin, the minimum (quantized) signal swing required in data[5] is 155.5mV

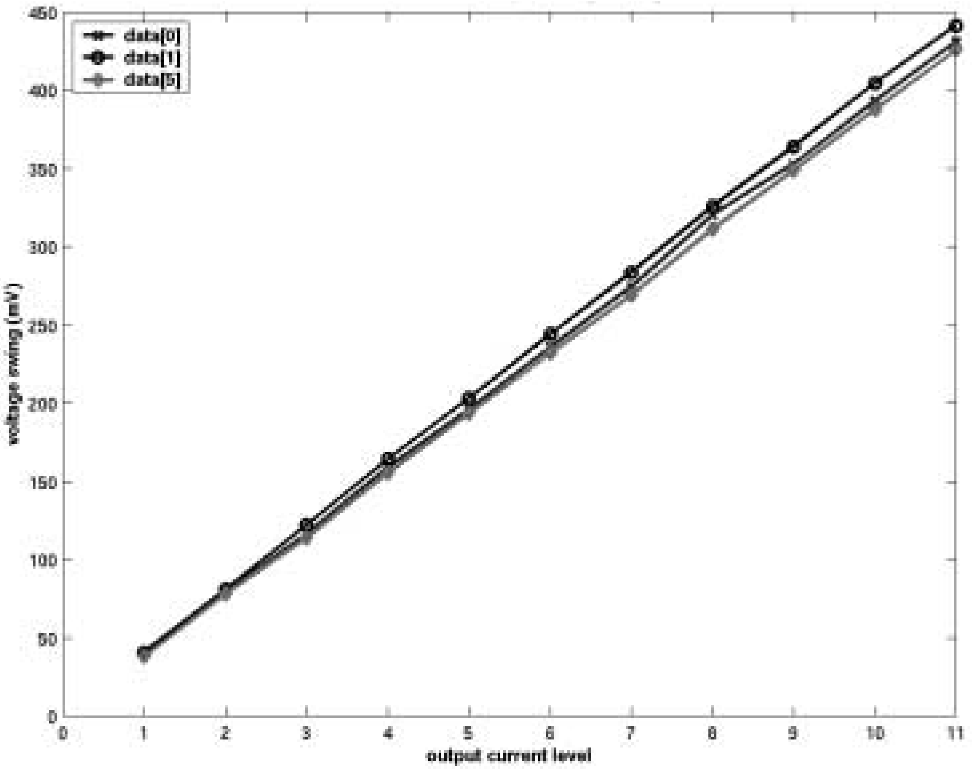


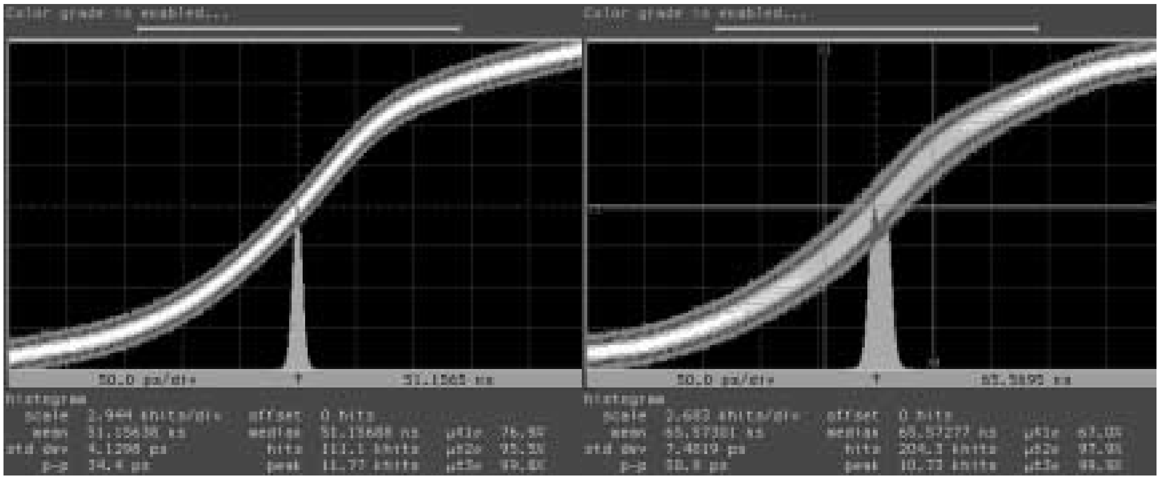
Figure 3.19: Measured unidirectional output voltage swings in data[0], data[1], and data[5].

superimpose unidirectional signals from both ends at zero phase, the resultant bidirectional swing is very close to the sum of the two unidirectional swings.

TDR (time domain reflectometry) measurements of the test chip show much higher termination resistance values than the design simulations because of the slower than expected process. The measured termination resistance varies from about 48 Ω to 57Ω at the two ends of the maximum unidirectional swing, and increases to 70 Ω at maximum bidirectional swing[13], even when we bias the gate of the PMOS at the lowest voltage we feel comfortable with (without gate stress and breakdown). The termination mismatch, however, does not introduce a significant voltage noise source, as we will see in Chapter 5.

TxClk is buffered and routed to an output pin. Unfortunately, since there are several full-swing buffers along the path, the buffered output is very jittery and an accurate jitter

[13] This measurement is the effective resistance of the parallel combination of the termination resistor and the output resistance of the output drive .



```
Color grade is enabled...
50.0 ps/div          ↑          51.1565 ns
histogram
   scale  2.944 khits/div     offset  0 hits
    mean  51.15638 ns         median  51.15688 ns    µ±1σ  76.9%
 std dev  4.1298 ps             hits  111.1 khits    µ±2σ  95.5%
     p-p  34.4 ps               peak  11.77 khits    µ±3σ  99.8%
```

(a)

```
Color grade is enabled...
50.0 ps/div          ↑          65.5895 ns
histogram
   scale  2.683 khits/div     offset  0 hits
    mean  65.57301 ns         median  65.57277 ns    µ±1σ  67.0%
 std dev  7.4819 ps             hits  204.3 khits    µ±2σ  97.9%
     p-p  58.8 ps               peak  10.73 khits    µ±3σ  99.9%
```

(b)

Figure 3.20: Jitter measurement of transmitted refClk[0]: (a) with no power supply noise, (b) with 200mV injected powe supply noise.

measurement of TxClk cannot be obtained directly. Instead, its jitter is measured indirectly by measuring the jitter on the transmitted refClk and data signals. Each transmitted output is clocked by TxClk, and then it passes through the pre-driver and output driver Therefore, the measured jitter of a transmitted signal is the sum of the jitter in TxClk and the jitter of the pre-driver and driver stages, providing an upper bound of the jitter in TxClk. Figure 3.20 shows the jitter measurements of transmitted refClk[0] when on-chip clocks are running at 600MHz and the chip operates in the default mode (using cleanClk as input to core DLL). The signal is quite clean, showing a nice Gaussian jitter histogram with a peak-to-peak jitter of 34.4ps (4.1% of T_{bit}) when no power supply noise is injected externally. This measurement suggests that the jitter of TxClk itself is even less. When a 200mV noise, generated by shorting Vdd_TxDLL and Gnd with a large transistor using a 1MHz square wave gate input, is injected on Vdd_TxDLL, the peak-to-peak jitter increases to 58.9ps (7.1% of T_{bit}). The injected noise frequency is below the bandwidth of the DLL, which is approximately 18MHz from simulations. The jitter histogram is bimodal with two peaks of about the same height separated by 8.3ps. This means that the DC power supply variation shifts the center of the Gaussian curve by 8.3ps, while the AC variation spreads out its base by 16.2ps. Hence the DLL has extremely low static jitter sensitivity of 0.0415ps/mV and dynamic jitter sensitivity of 0.081ps/m . The other transmitted signals (data[7:0]) have larger jitter due to noisier local supplies and inter-signal cross-talk, and the extra jitter subtracts directly from the receiver timing margins.

Any deviation in phase in the transmitter outputs after calibration reduces the receiver timing margins. Experimentally, the transmitted data[7:0] drift on average about 8.7% of T_{bit} (72.7ps) when we spray coolant on the package cover and leads continuously and then heat up the ambient with a hair dryer for 5 minutes, measured by shifts in the centers of the jitter histograms. In normal operating conditions, the time drift in each transmitter output signal due to heating up of the chip is less than 9.7ps, which is too small for the effect to be isolated from the high-frequency jitter.

Measures discussed earlier in the TxDLL design are taken to reduce duty cycle distortions and their effects on the transmit signals. Consequently, we see no significant duty cycle distortions in the transmit signals.

3.3.3 Receiver Circuit Characterization

Unlike in the transmitter where the jitter of TxClk can be measured indirectly by measuring the jitter of the transmitted signals, there is no good way to measure the jitter of the local receiver clocks. RxClk[0], the local RxClk generated to receive data[0], is buffered and routed to an output pad. This suffers from the same problem that the huge amount of jitter introduced by the full-swing buffers along the signal path m akes direct jitter measurement impossible. However, we can still obtain useful RxClk timing information from such a jittery clock: the mean value of the jitter histogram gives a good estimate of its center position.

We can, for instance, measure the low-frequency phase drift in this buffered RxClk[0]. Experimentally, the signal drifts by a much smaller amount than the simulated results: about 2.7% of T_{bit} (22.5ps) when we apply the ‘coolant and hair dryer’ test. Just like in the transmitted outputs, in normal operating conditions, the phase drift of this signal due to heating up of the chip is too small for the effect to be isolated from its high-frequency jitter.

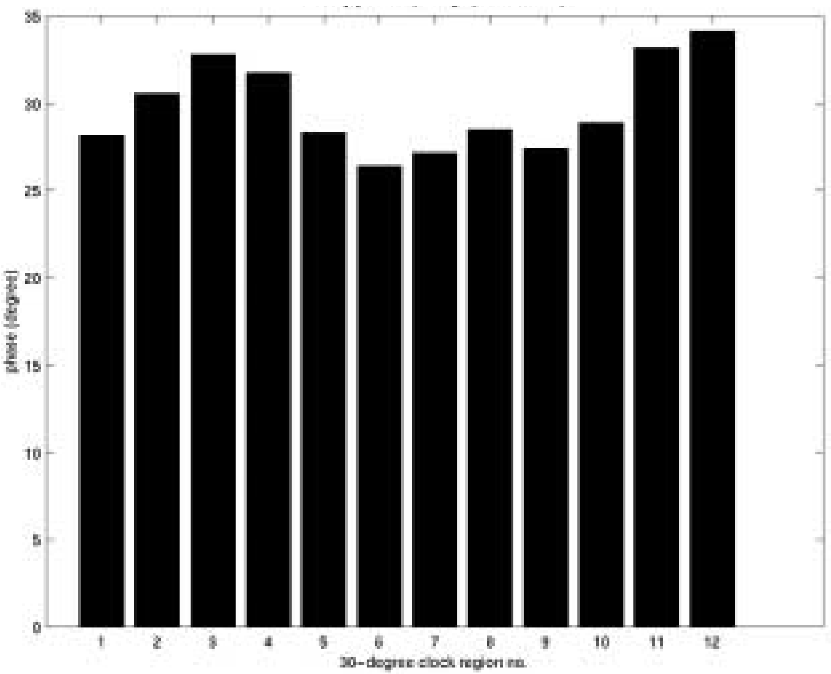
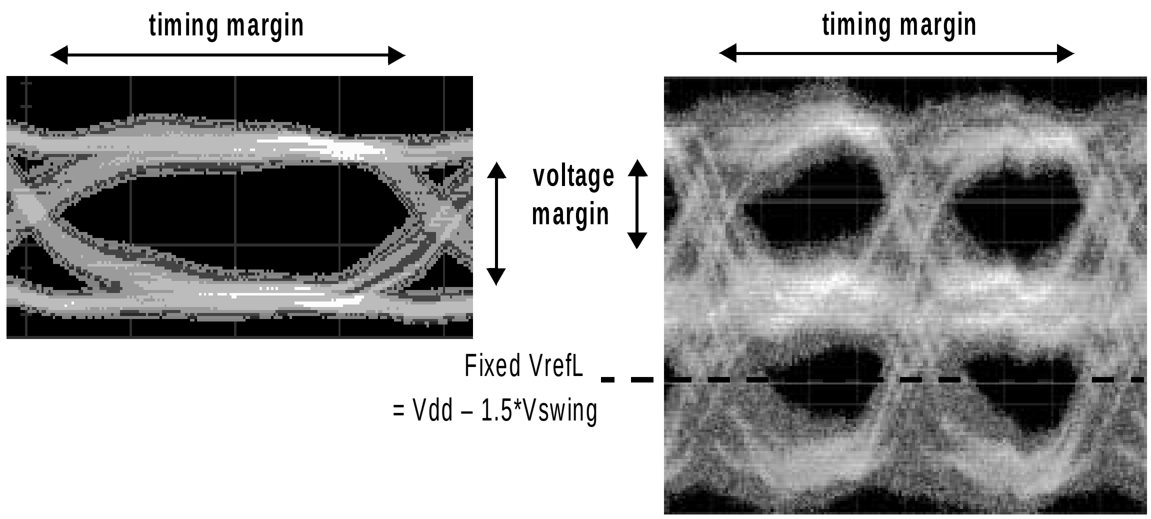


Figure 3.21: Measured clock spacings of RxClk[0], in degrees. The nominal spacing is 30°.

The accuracy of the quadrature phase shift performed after per pin timing calibration affects the timing margin directly. We measure the mean value of the clock (falling edge) jitter histogram when the interpolator controls are all steered to the latter of the two clock phases selected by the mux controls at 13 consecutive phase boundaries, then take the difference between each pair of consecutive values to obtain the clock spacings, as shown in Figure 3.21 for RxClk[0] running at 600MHz. These spacings are nominally at 30°, or 138.9ps. Measurement results span from 122.2ps to 158.1ps, or a range of 35.9ps, which represents a 25.8% deviation. A separate set of measurements taken using the median values, rather than the mean values, of the clock jitter histograms shows extremely close results that differ by less than 1ps in the worst case. The largest (region #12) and smallest (region #6) clock spacings are likely to be caused by duty cycle distortion in the clock input to the delay line t hat the duty cycle adjuster fails to correct. There are two main causes for the other clock spacing variations: mismatches in the delay line buffer stages; and offsets in these buffers and in the subsequent large clock buffers (that distribute the 30° clocks to the I/O cells) which result in additional duty cycle distortions in the 30°



*Transmit signal & receive signal in quadrature phase

Figure 3.22: Voltage and timing margins of links.

clocks being distributed. What really matters, however, is the accuracy of the quadrature phase shift, which can be found by the sum of 3 consecutive regions if the local RxClk happens to fall at these clock boundaries. This sum varies from 81.9^{o} to 96.2^{o}, meaning that in the worst case, 14.3^{o} (66.2ps or 7.9%) of the timing margin can be lost potentiall .

We also measure phase interpolator steps by taking the mean values of the jittery clock histograms and results for the first two 30^{o} regions show steps that span from a minimum of 3.9ps to a maximum of 14ps.

3.4 Measuring Signal Margins

The transceiver architecture supports per pin timing adjustment which allows measurements of timing margins of the links, while the adjustable reference voltage generation allows measurements of voltage margins. These built-in testing and measurement capabilities, combined with the NI-DAQ I/O interface and the HPIB interface, allow us to measure the internal voltage and timing margins of the links, illustrated in Figur e3.22, in a systematic way.

To measure voltage margins, the links are first calibrated by setting Vref at the middle of the nominal signal swing (V_{swing}). Keeping the phase controls inside all I/O cells fixed (and hence fixing the positions of RxClk[7:0]), Vref is moved up and down, and the first boundary points at which each link starts to fail are recorded, the difference of which is the voltage margin. This measurement has a 1mV resolution. To measure timing margins, we set Vref at the middle of the nominal signal swing and calibrate the links. Then, while keeping Vref fixed, we measure the timing margin of each link, by shifting the local RxClk at nominal timing steps equal to 8.7ps in both directions. The boundary points at which bit errors start to appear are recorded, and the interval between these two points is the timing margin.

The signal margins of bidirectional links are measured in similar steps. Because the transmitter output swing is fairly linear in bidirectional signalling as found earlier, a fixed VrefL equal to 1.5* V_{swing} below the supply is used. The voltage margin of each link is measured by varying VrefH while keeping RxClk fixed. The timing margin is measured by shifting RxClk while keeping VrefH fixed.

Each passing value in the signal margin measurements has a BER less than 10^{-11}. Unless otherwise specified, all measurements are taken with all of the circuit blocks turned on to simulate the power supply noise in a real mixed-signal system.

3.5 Summary

The chapter describes a n 8-bit single-ended, simultaneous bidirectional parallel link transceiver test chip implemented in a 0.35 µm CMOS process. The links achieve a bidirectional data rate of 2.4Gbps/pin with a BER less than 8 x 10^{-15}. The chip dissipates less than 1W total power from a 3.3V supply, and occupies a die area of 1.7 x 3.8mm^2.

The link performance is limited by clock and data generation at the transmitter. Experimentally, we find that the termination mismatch can be as large as 40% in bidirectional signalling and 14% in unidirectional signalling . TxClk carries less than 34.4ps of jitter, insignificant low-frequency phase drift, and negligible duty-cycle distortion. Such properties pass on to the transmit signals. On the receiver side, uneven 30^o clock spacings may introduce up to 66.2ps timing error in the quadrature phase shift performed on each local RxClk after the skew calibration. The phase interpolator steps also deviate from the 8.7ps nominal step size, taking on values ranging from 3.9ps to 14ps. In Chapter 4 and Chapter 5, we will study the implications and challenges presented by these, and many others that we will subsequently explore, timing and voltage noise sources.

CHAPTER 4

OVERCOMING TIMING ERRORS IN HIGH-SPEED PARALLEL LINKS

In Chapter 2, we identify the three fundamental challenges in high-speed parallel link designs. In this chapter, we specifically concentrate on receiver timing recovery issues. As explained earlier, phase recovery is relatively easy in parallel data channels that send a source-synchronous reference clock along with the data signals; the main challenge in the receiver timing recovery is then how to overcome timing errors that can potentially narrow the receiver timing margins and limit the data rate.

Section 4.1 studies the static timing error, inter-signal timing skew, and schemes to compensate for it. It examines, in particular, the benefits vs. cost overheads of different signal-to-signal skew compensation architectures. Section 4.2 concentrates on the dynamic timing error, inter-signal jitter, and studies the correlation of phase variations in different signalling pins.

4.1 Inter-Signal Timing Skew

To test the per pin skew compensation capability, two sets of experiments are carried out using the setup described earlier in Section3.3.1. In both tests, the unidirectional links run at 1.2Gbps at their maximum swings (about 430mV).

4.1.1 Skew Compensation Measurements

Calibration results are shown in Figure4.1 . The bars show receiver timing margins of different signal pins [1], their calibrated eye centers, and ideal (actual) eye centers. The

[1] Unfortunately, data[6] is mistakenly bonded to a non-I/O pin and its measurement results are ignored.

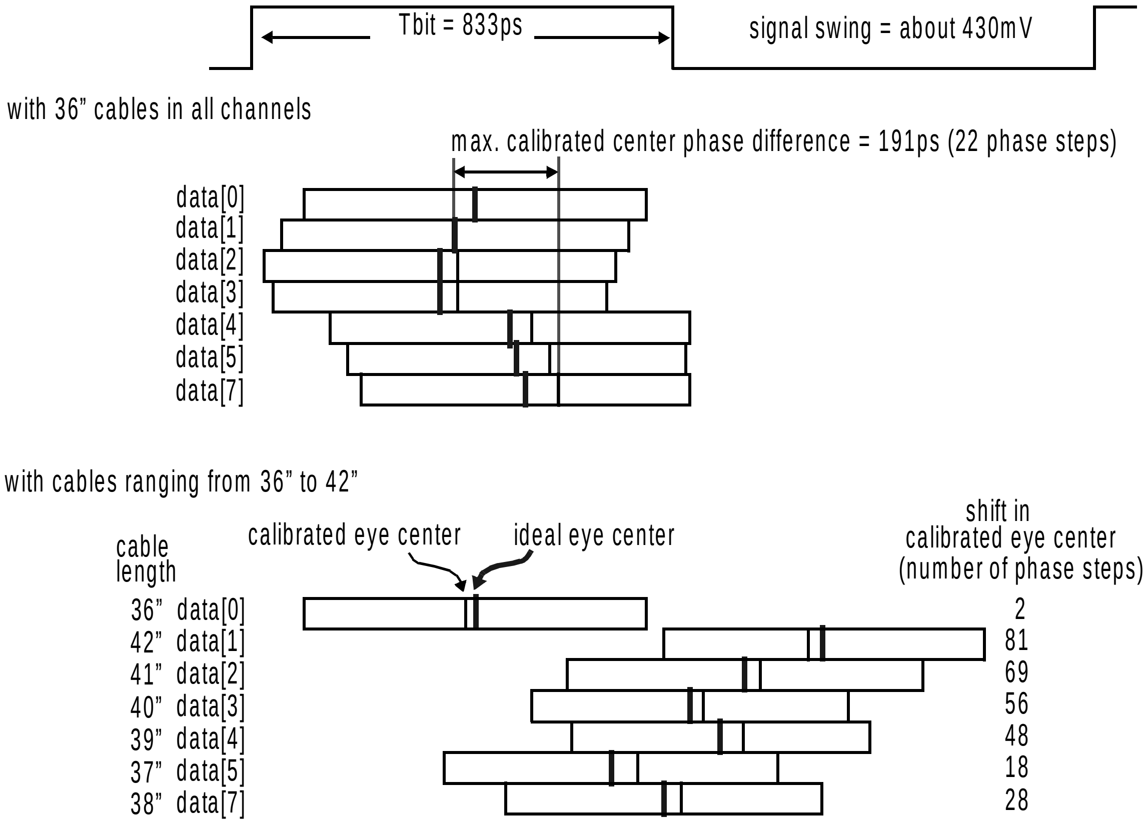

Tbit = 833ps
signal swing = about 430mV
with 36" cables in all channels
max. calibrated center phase difference = 191ps (22 phase steps)
data[0]
data[1]
data[2]
data[3]
data[4]
data[5]
data[7]
with cables ranging from 36" to 42"
cable length
calibrated eye center
ideal eye center
shift in calibrated eye center (number of phase steps)
36" data[0] 2
42" data[1] 81
41" data[2] 69
40" data[3] 56
39" data[4] 48
37" data[5] 18
38" data[7] 28

Figure 4.1: Receiver timing margins for skew compensation tests. Calibrated eye centers shift as skews increase.

positions of the centers and the widths of the eyes are scaled, in terms of the number of phase steps, by the bar charts.

Initially we use a 36-inch cable in each data channel and carefully match the delays of all paths. The links are calibrated, and the results show a maximum phase difference of 191ps (22 phase steps) in the calibrated eye centers between the fastest pin (data[1]) and the slowest pin (data[7]). The on-chip data waveforms, using the voltage samplers, indicate that approximately 100ps of this difference is due to inter-signal skew, about half of which can be attributed to differences in the signal traces in the packages used, as found by TDR measurements of the channels[2]. The calibration results show one possible problem with our calibration scheme. For the pins with significant coupling from their neighbors (data[4], data[5] and data[7]), the crossing point between the signal and the

[2] TDR measurements show that the signal propagation delays through the traces in each package are somewhere between 40 to 65ps.

corresponding Vref moves by about 90ps when the neighboring signals transition in sync with the signal compared to when the neighboring signals are idle. These two components account for the observed 191ps maximum phase difference in the calibrated eye centers. Then cables ranging from 36 to 42 inches in length are used to deliberately introduce more skew. Calibrated eye centers shift as skews increase, showing that the circuit is able to deal with larger skews without reducing timing margins.

It is worth mentioning that the 'time scale' we use in all the timing margin measurements is the *number of interpolator phase steps* , sometimes scaled back to absolute time by multiplying by the nominal phase step size (8.7ps). However, as we have seen from the RxClk measurements in Section3.3.3, the 30 ^{o} clock spacings are uneven, and the interpolator phase steps are non-linear. Therefore, phase steps differ in size, and the number of phase steps may not correspond to the exact phase shift. Nevertheless, statisticall , if the window of phase steps in question is large, the exact phase shift represented by this window becomes closer to the number of phase steps multiplied by the nominal step size. This is the case in our measurements, where the timing margins are 70 to 80 phase steps, and the effects of non-uniform phase steps are averaged, and we just need to be aware that any measured timing window can be off by a few phase steps.

4.1.2 Per Pin Skew Compensation Design Trade-offs

As we have seen in Chapter 3, the per pin timing adjustment architecture requires extra hardware which takes up area and power. A fair evaluation entails comparing and contrasting with the way we would have built the system had skew compensation not been implemented. The point of reference we would use for the comparison, which we refer subsequently as the reference design, is a parallel link system where the receiver timing recovery is done, similar to the conventional architecture shown earlier in Figure2.1, by phase-locking to a source-synchronous refClk signal using the same dual-loop DLL design implemented in the test chip. Two sets of phase muxes and interpolator are needed in such design: one to phase-lock to the incoming refClk and the other to give the 90 ^{o} shifted RxClk to sample the data signals.

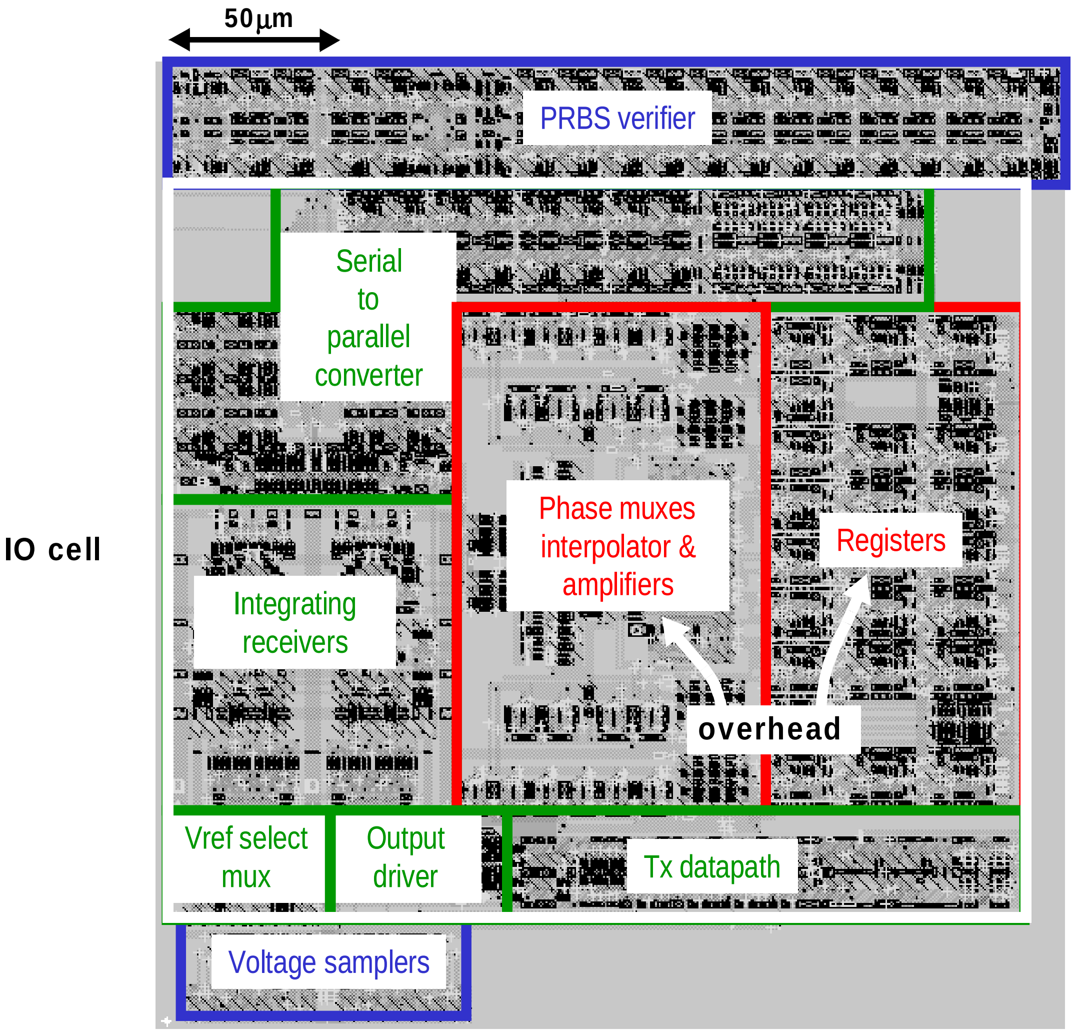


Figure 4.2: Layout of the bidirectional I/O cell.

Figure 4.2 shows the layout of each bidirectional I/O cell. The total area [3] is 192 x 275µm^2. The blocks added for skew compensation -- the phase muxes, the phase interpolator and the subsequent clock amplifier, and the associated registers for holding the mux and interpolator controls -- occupy slightly less area than the transmit and receive blocks, and hence approximately double the size of the I/O cell. As technologies continue to scale, even though the extra hardware always remains half of the total area, its a rea shrinks in proportion to the area of an I/O pad or pin.

[3] Excluding the voltage samplers and PRBS verifier since they are for measurement and testing purposes and would not exist in real applications.

The static currents in the phase muxes, phase interpolator and differential-to-single-ended converter draw an additional 13.7mW of power per I/O cell when the receiver runs at the 1.2Gbps maximum speed. As technologies continue to scale, the power supplies drop, and the circuit biasing currents required to achieve certain data rate decrease. As a result, the extra power per I/O also decreases.

In addition to overhead inside each I/O cell, the per pin skew compensation architecture also requires extra hardware that is shared among all pins. In the reference design, the global RxClk would most likely be distributed to all the I/Os, single-endedly, using CMOS inverters, and hence eliminating any static power dissipation in the clock buffers. In our design, the 6 differential clocks are distributed to all the I/Os using low-swing symmetric-load buffers to keep the clock jitter to a minimum. These buffers altogether dissipate 268mW of power at the highest link speed, and occupy an additional 220 x 220μm^2 of area.

Table 4-1 shows the simulated *static* power consumption in different parts of the test chip, totalling 853.4mW. The low-swing differential buffers that distribute the 30^o spaced

Table 4-1: Static power consumption in different parts of the chip, when each chip is running at 2.4Gbps bidirectional data rate at maximum signal swing.

Circuit block	Static power consumption
core DLL in data loop	**85.8mW**
6 sets of low-swing buffers distributing 30^o spaced clocks to all I/Os	**268m**
8 sets of phase muxes + interpolator + amplifier (data[7:0])	**109.6m**
8 output drivers (data[7:0], assuming each output is high 50% of the time)	**227m**
8 sets of current-integrating receiver pairs	**21.8mW**
TxDLL	**47.1mV**
dynamic phase tracking loop total	**94.1mW**
- core DLL - peripheral DLL - refClk[1:0] output drivers (transmit alternating zeros and ones) - 2 sets of current-integrating receiver pairs in refClk[1:0]	18.2mW 13.7mW 56.8mW 5.4mW

clocks across the chip to all I/Os account for 31.8% of the total static power, and the phase muxes, phase interpolators, and amplifiers inside the I/O cells account for another 12.8%. Even though jitter sensitivity to power supply noise is optimized, the potential timing margin improvement in this case is perhaps only about 50ps, which represents less than 7% improvement in the wide timing margins of the links even at the highest data rate.

Compared to the FSM used for the DLL in the reference design, the sequential skew calibration requires 3 extra flip-flops and a 3-bit counter in the FSM to keep track of which pin is currently being calibrated, and a small amount of extra logic to handle the quadrature phase shift performed after the skew calibration. All these extra circuits, however, amount to less than 10% of the total FSM area.

The skew calibration operation reduces the effective bandwidth of the links, but this is done only once initially at chip start-up, and perhaps periodically if the transmit signals drift considerably in phase over time. In these skew-calibrated systems, if tracking the phase of the incoming refClk does not improve the timing margins of the data signals, the refClk pin can be completely eliminated. Then altogether 8 sets of phase muxes and interpolator are required, compared to 2 sets (for refClk and the global RxClk) in the reference design.

The per pin skew compensation architecture we have implemented does not considerably increase the complexity of the design, either in the FSM or in the I/O cells, when compared to the reference design. Although the area overhead is significant inside each I/O cell, it is reasonable from the perspective of the whole chip. The major drawback with our implementation is the power overhead in the low-swing clock buffers, which, fortunately, is implementation-specific and not intrinsic to the architecture. As explained earlier, low-swing buffers are used to minimize the jitter of the clocks being distributed. The power overhead can be significantly lowered by replacing the clock buffers with CMOS inverters. While CMOS inverters have jitter sensitivities that are 2 or 3 times higher, as long as the buffer delay is kept short, the impact of the higher jitter is less than the power overhead. Therefore, replacing the power-hungry low-swing clock buffers with CMOS inverters is a good design trade-off.

This per pin timing adjustment architecture allows a full-range compensation for inter-signal timing skew up to two bit times. When the expected skew is small, the amount of hardware can be reduced by adopting alternative skew compensation architectures with smaller compensation ranges.

As described earlier in Chapter 2, most per pin deskewing circuits involve calibrating each data bit's skew relative to a timing reference during system initiation and storing the skew information in some adjustable delay. Skew calibration is often done by some digital control logic, while skew compensation is accomplished by skewing the local transmitter clock [50] or the local receiver clock [46], [9] based on the calibrated skew information so that each receiver data eye is recentered around the local receiver clock. The adjustable delay chain can be realized by activating a different number of stages [46], [50](e.g. using a delay tree of delay elements and tap ping off at different points), by adjusting the delay per stage in a fixed-stage delay line, or by using phase interpolation [9] . The jitter of the locally skewed receiver clock generated and the range of the skew that the system can handle are important design considerations. Often times, they also determine the overhead required to implement the deskewing function.

Considering the per pin skew compensation architecture we have implemented, the overhead can be greatly reduced if the phase muxes and interpolator in each I/O cell are replaced with a small variable delay line, as pictured earlier in Figure 3.4, at the expense of decreased skew compensation range. U sing a long delay chain extends the adjustable delay range, but at the same time increases the jitter of the local receiver clock generated. The delay range allowed is strongly dependent on process variations: the longest achievable delay at the fastest process corner sets the upper limit of the delay range, whereas the shortest achievable delay at the slowest process corner sets the lower limit. A simple fixed-stage tunable delay line with adjustable buffer loading, which can use similar phase control logic as implemented in the test chip, is shown in Figure4.3 . The delay steps are much coarser at the slowest process corner compared to at the fastest, reducing the resolution and accuracy of the skew compensation. Nevertheless, it allows substantial reduction in overhead not only inside each I/O cell but also in the FSM and global routing

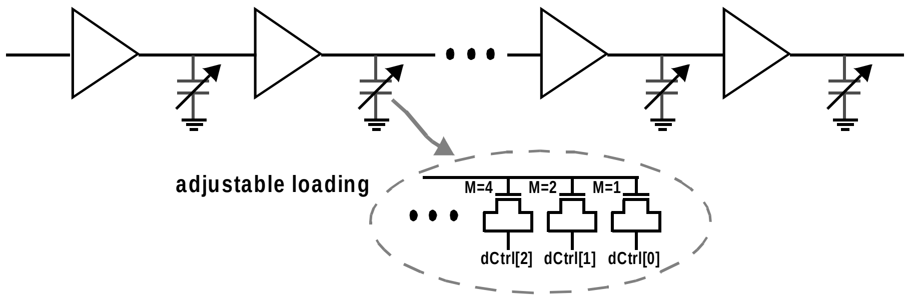


Figure 4.3: Delay control by adjusting buffer loading.

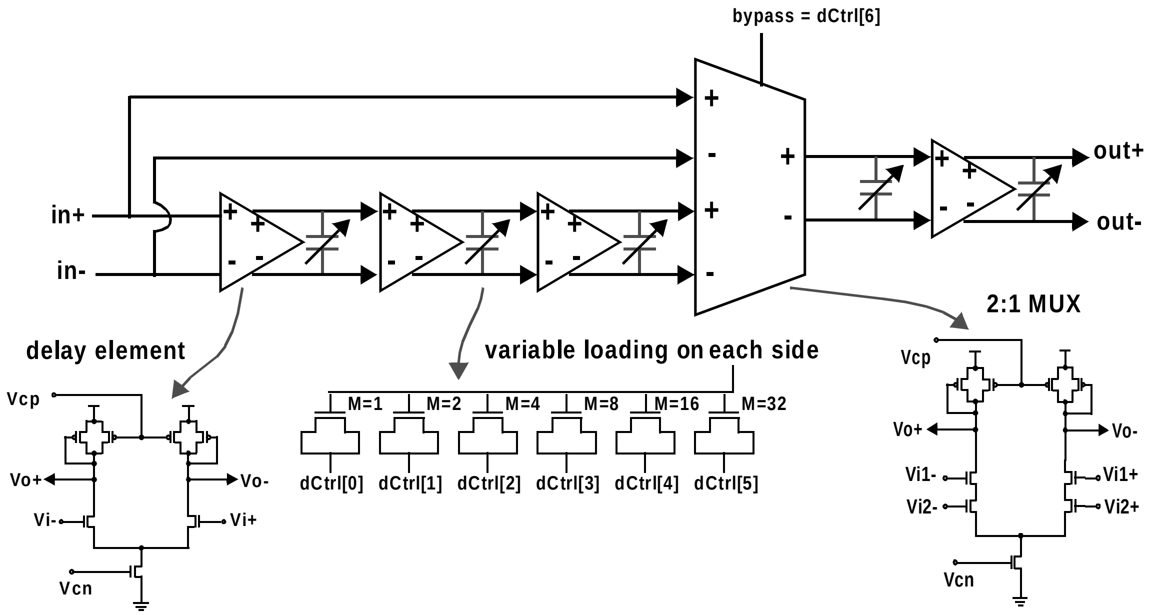


Figure 4.4: A 'bypass-able' delay line which extends the delay range.

as the phase control logic is much simplified and the number of phase controls is significantly reduced.

Regardless of the buffer stages and delay tuning method used, all designs have a limited delay range that depends on process variations. To extend the delay range, a mix of the approaches discussed earlier can be combined. A possible design -- a 'bypass-able' delay line as illustrated in Figure4.4 -- varies both the delay per stage and the number of delay stages by selecting between two delay paths that consist of different (but fixed) numbers of delay stages. In addition to the strongly process-dependent delay range and delay steps, this scheme may suffer an additional limitation that the phase transition may not be monotonic as the delay control switches from one path to the other. Hence, more complex phase control logic is needed to prevent false lock at the phase boundary, which requires a slight increase (but still substantial reduction when compared to the phase interpolation approach) in hardware overhead. On the other hand, by sharing the same buffer bias voltages as the delay elements in the delay-locked loop in the system, the delay range and phase steps also scale with the DLL operating frequency, therefore allowing the skew compensation range to scale with the data rate.

Skew compensation removes the performance bottleneck imposed by inter-signal timing skew in parallel links. The cost overhead in implementing it depends largely on the range and accuracy of compensation. Its increasing presence in interface designs clearly suggests its importance as the data rate in parallel links increases.

4.2 Dynamic Phase Noise

As explained earlier in Chapter 2, given the balanced nature of the refClk (especially refClk[0]) and data lines at the transmitter, the phase noise in each received data signal may be correlated with the phase noise in the received refClk; therefore, tracking the dynamic phase variations in refClk at the receiver timing (by moving the local RxClk of each data pin) may be beneficial. On the other hand, if there is no jitter correlation between the two signals, moving the local RxClk based on the dynamic phase variations in refClk in hope of tracking the phase variations in the data signal would actually create an even larger worst-case inter-signal jitter, which is equal to the sum of the jitter in refClk and the jitter in the data signal.

To test whether clock jitter tracking will help in this type of link, the clock for the core DLL has three possible sources as outlined earlier in Chapter 3. The delay in the core data loop clock generation, shown earlier in Figure3.5 and repeated in Figure4.5 with only the essential elements, limits the tracking bandwidth. Therefore if the phase noise is higher in frequency than this bandwidth, or the phase noise in the inputs is uncorrelated, trying to track the noise will decrease the overall quality of the link.

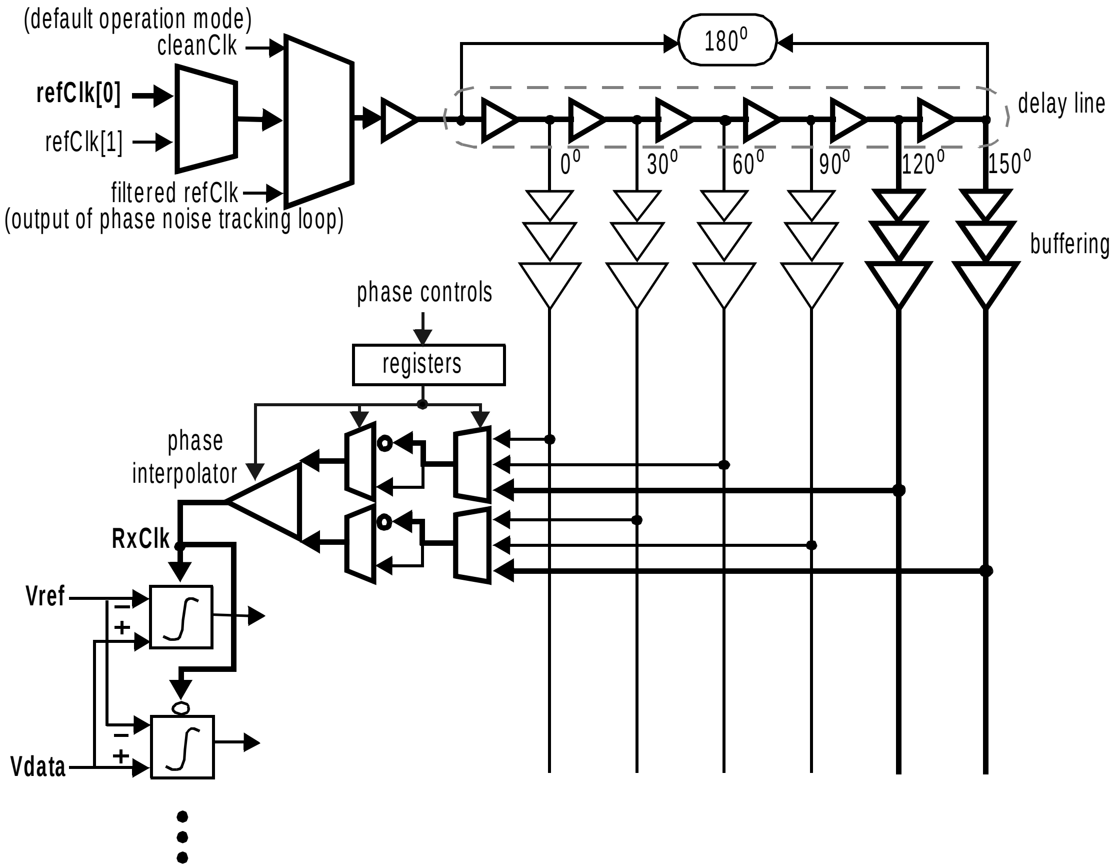


Figure 4.5: The delay in the core data loop clock generation, which depends on phase settings, limits the dynamic phase noise tracking bandwidth. One of the longest delay paths is highlighted.

The delay through most of the circuit stages in the clock generation loop scales with the bit time (T_{bit}) -- the only exception is in the differential-to-single-ended converter and its subsequent buffers. The delay also depends on the phase settings inside each I/O cell. The *maximum* total delay from one of the received refClk signals to the local RxClk[7:0] (T_d) is roughly[4] 3.3*T_{bit}+5*FO4. Theoretically, if tracking results in a phase shift of less than 90^o, the correction is in the right direction and hence is beneficial. Using this phase relationship, the maximum ‘ track-able’ noise frequency is equal to 1/(4* T_d). The maximum ‘track-able’ noise frequency as a function of unidirectional data rate (reciprocal

[4] The 2:1 and 3:1 clock muxes at the input and the phase muxes and phase interpolator inside each I/O cell all drive higher fanouts than the 6 buffers inside the delay line (which are locked to one bit time). Therefore, even though there are altogether 17 low-swing stages in the longest path, the delay scaling factor is 3.3. The portion of the total delay that does not scale with the bit time is due to the differential-to-single-ended converter and its subsequent buffers inside each I/O cell (5 stages in total) and scales with FO4.

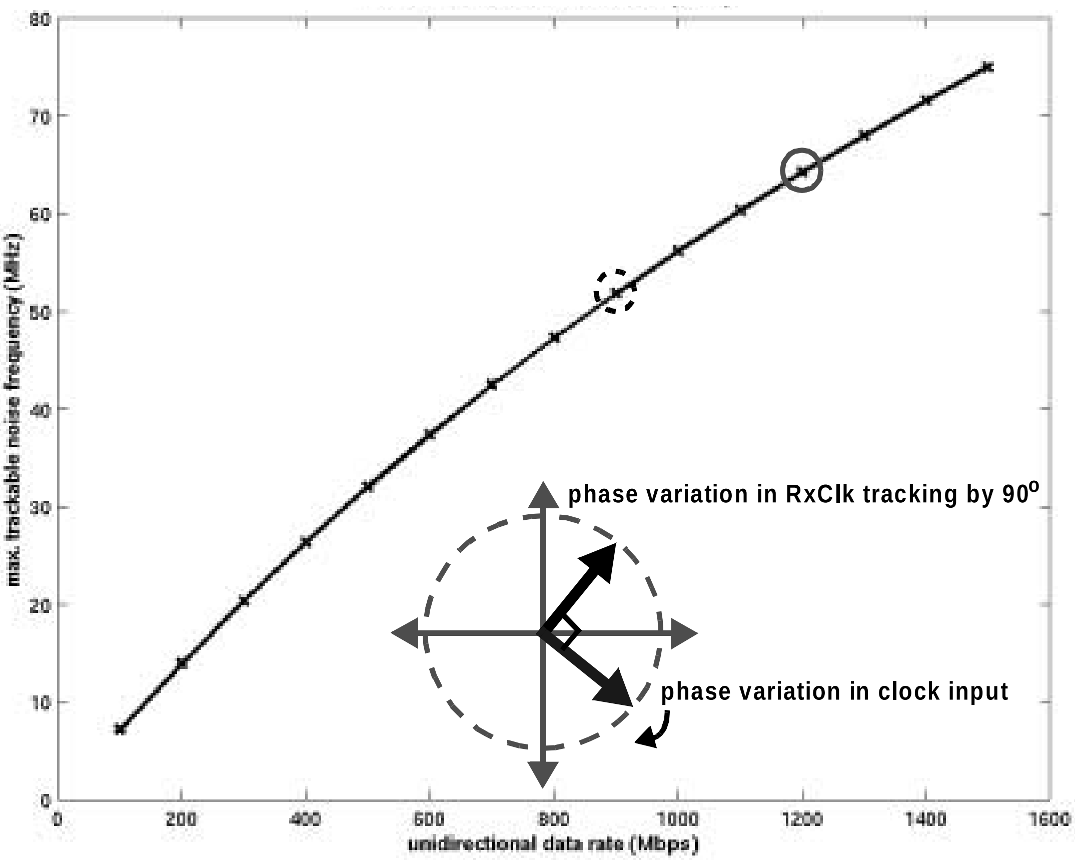


Figure 4.6: Maximum 'track-able' noise frequency as a function of unidirectional data rate. The ratio of maximum noise frequency to data rate (i.e. slope of the plot) decreases with increasing data rate.

of bit time) is plotted in Figure4.6, which also expresses the phase tracking signals in phasors. For example, at the 1.2Gbps unidirectional link speed quoted, the predicted noise tracking bandwidth is about 64MHz.

As explained earlier in Chapter 3, the default clock input to the core data loop is the cleanClk, a stable external reference that is driven into the chip. If the main phase noise is below the track-bandwidth allowed by the clock buffer delay, feeding in the received refClk directly should improve performance.

4.2.1 Low-Frequency Dynamic Phase Noise Tracking Loop

If the phase noise is mostly above the track-bandwidth, but there is also low-frequency phase noise, then using a filtered version of the received refClk would perform best. This option is also possible in the test chip by using a dynamic phase noise tracking loop: the

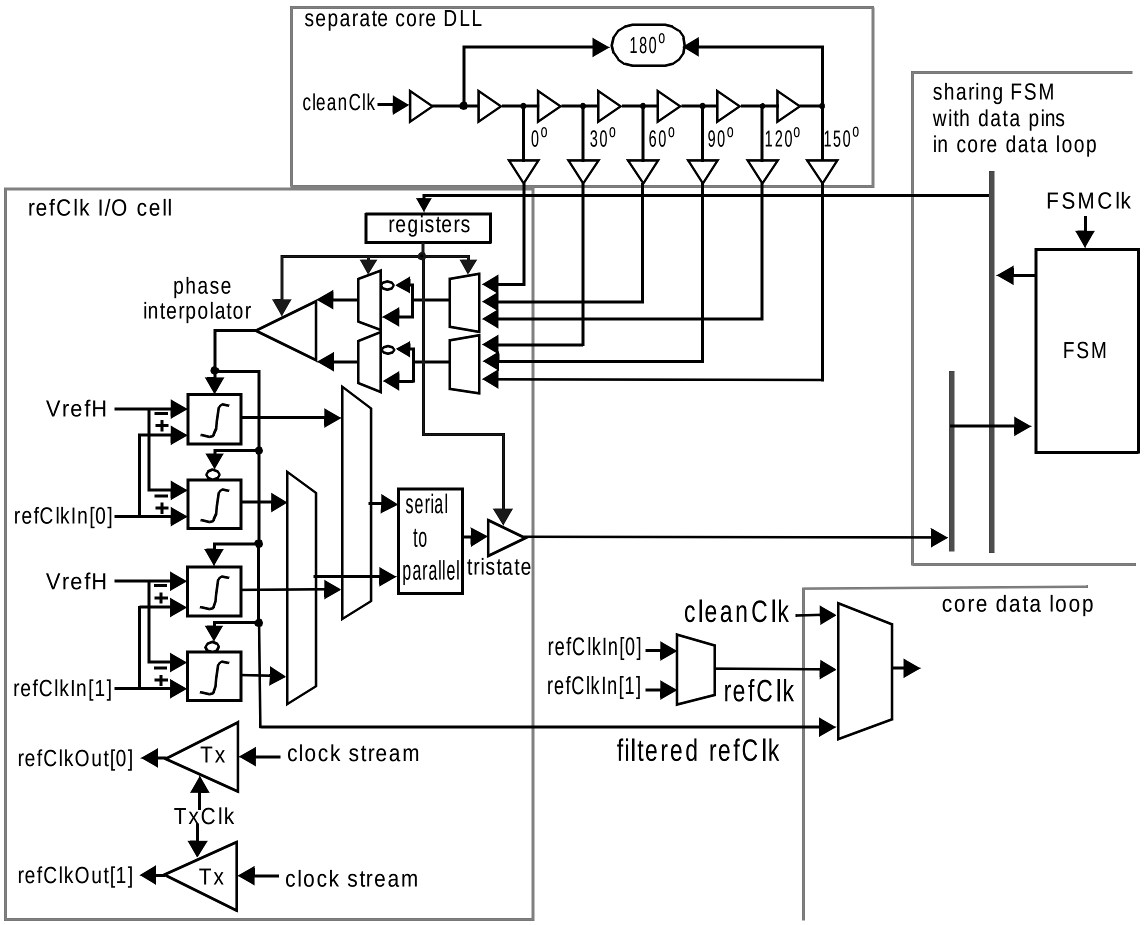


Figure 4.7: Dynamic phase noise tracking loop. The received refCl signal is filtered by another dual-loop DLL.

phase noise of the input refClk is filtered by using it to drive the feedback on another DLL. Thus the output of this DLL contains only the low-frequency phase noise of the received refClk.

The dynamic phase noise tracking loop, illustrated in Figure4.7, looks very similar to the core data loop. A delay-locked loop takes the clean system clock (cleanClk) as input and generates six differential clocks at 30^o phase spacings, which go to the refClk I/O cell where they are phase-muxed and interpolated. One of the two refClk signals is selected. On chip start-up, FSM first calibrates the refClk pins in the same manner it calibrates each data pin in the default mode, except that the phase controls are not quadrature-shifted before they are stored inside the registers. Then the FSM proceeds to the sequential calibration of all the data pins as in the default mode. After the calibration is complete, data transmission begins, and the FSM loads back the stored phase controls from the registers in the refClk I/O and monitors the phase changes in the selected refClk. In the phase tracking process, the FSM operates in the slow mode and updates the phase controls every 5 FSMClk cycles to allow for the latency in the digital feedback loop. On each FSM update, the filtered refClk (and therefore all the local RxClk[7:0]) moves one phase step, nominally equal to 8.7ps, in the same direction as the drift in refClk and helps to re-center each RxClk to its incoming data signal eye, whereas the high-frequency components of refClk are filtered by the bandwidth of this FSM update.

This mode allows plesiochronous operation. The update rate in the FSM in the slow mode is 1/(40*T_{bit}). The nominal phase step is T_{bit}/96. Therefore, the dynamic phase noise tracking loop can catch up with the variations in the selected refClk signal as long as it drifts by less than 260ppm. However, similar to the core data loop, mismatches in the 30^o clock spacings and non-linearity in the phase interpolator cause the interpolator output phase steps to deviate from the nominal value, which may slightly lower the phase drift allowed.

4.2.2 Phase Noise Measurements

To evaluate the dynamic phase noise characteristics of the interface signals, receiver timing margins of unidirectional links are measured using the five different clock inputs to the core data loop. The results are shown in Figure 4.8 for three pins with different signal return configurations, specifically data[0], data[1], and data[5], running at 900Mbps unidirectional data rate. Our earlier analytical model predicts that the maximum track-able noise frequency allowed by the core data loop is about 52MHz at this data rate, as shown in Figure 4.6. The FSM update rate is 22.5MHz.

The data clearly indicates that for this system the dominant phase noise is high-frequency noise, an expected result for a DLL-based system. Since there are no VCOs (voltage-controlled oscillators) to accumulate jitter near the loop bandwidth, most of the jitter is likely to be cycle-to-cycle jitter. As mentioned earlier, if the refClk signal carries both high-frequency and low-frequency noise, using the filtered refClk will give the best performance among all three options. Therefore, the fact that using one of the two filtered

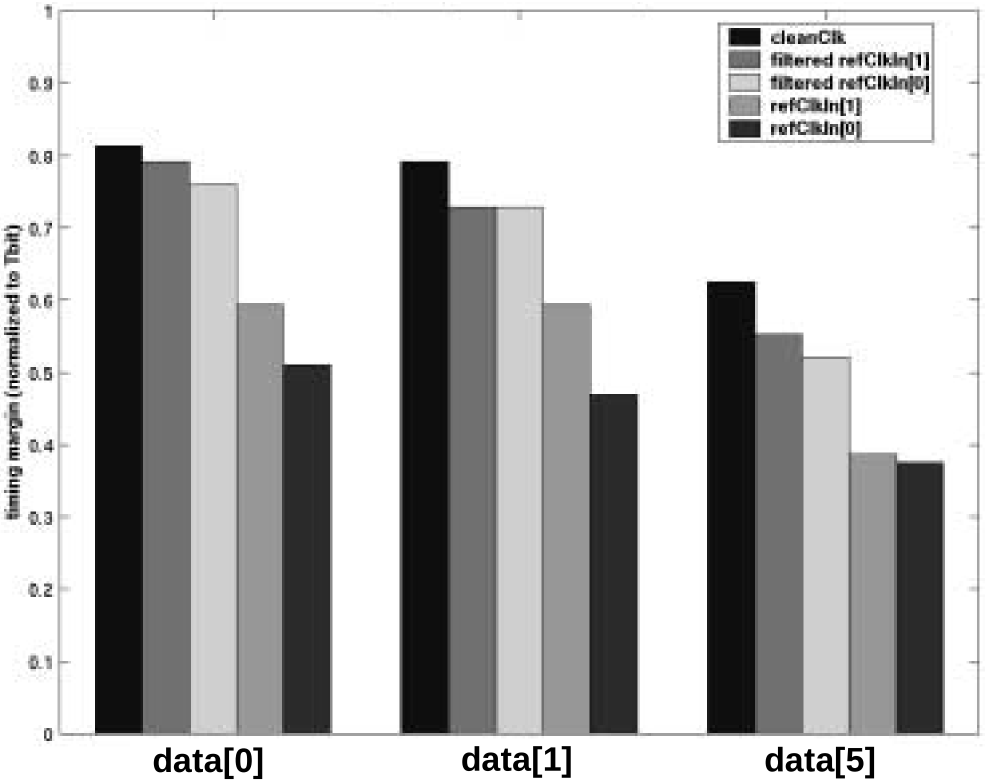


Figure 4.8: Receiver timing margins of unidirectional links using different clock inputs to the core DLL. Using cleanClk produces the largest timing margins, and using received refClkIn[0] gives the smallest margins, showing that the phase noise contains mainly high-frequency components.

refClk inputs is also worse than using cleanClk indicates that this system experiences very little low-frequency phase drift. The extra jitter that these filtered refClk signals pick up from the dynamic phase noise tracking loop circuitry reduces receiver timing margins. One interesting result is that timing margins degrade from data[0] to data[1] to data[5] in all operation modes. This degradation is caused by increased inter-signal cross-talk, which is described in the next chapter.

4.3 Summary

The main challenge in the receiver timing recovery in high-speed parallel links is in overcoming timing errors, and this chapter studies two major timing noise sources, namely, inter-signal timing skew and inter-signal jitter, which can create performance bottleneck and limit the achievable data rate.

Experimental results have demonstrated that per pin skew compensation helps to center and hence increase the receiver timing margins of parallel links. The cost overhead in implementing skew compensation depends largely on the range and accuracy of the compensation desired.

Experimental results have also shown that the dominant phase noise in the interface signals in a DLL-based system is high-frequency noise. Hence, using a stable clock source for receiver clock generation maximizes the receiver timing margins.

Chapter 5

Voltage Noise in Low-Cost Signalling Systems

Single-ended and simultaneous bidirectional links are both attractive alternatives to the traditional unidirectional and differential links because of their pin-saving potential, yet both are unattractive because they create more voltage noise sources that reduce receiver signal margins and may limit the attainable data rate or even cause data transmission errors. In this chapter, we study the extra noise sources introduced in these two low-cost signalling setups operating at high speed.

Sampling on-chip signals has been proved a useful technique for the testing and measurements of integrated circuits [81], [82]. In Section 5.1, the design and characterization of the on-chip voltage samplers are presented. Throughout the chip testing, we use these voltage samplers to probe high-speed on-chip signals, measure the internal signal margins of the links, and measure the individual voltage noise sources directly.

The voltage noise sources in single-ended and simultaneous bidirectional links are the topics for the rest of this chapter. In Section 5.2, the measured voltage margins are first presented and analyzed, and then fixed noise and proportional noise values of the links are extracted from these data points. Section 5.3 presents a complete noise model of the implemented signalling system that allows the magnitudes of the different noise sources present in the test chip to be predicted. The values of individual voltage noise components, measured directly using the voltage samplers, are presented, and compared against the values predicted by the proposed noise model. Discrepancies in the data are addressed, and the benefits of using current integration for simultaneous bidirectional links are also evaluated.

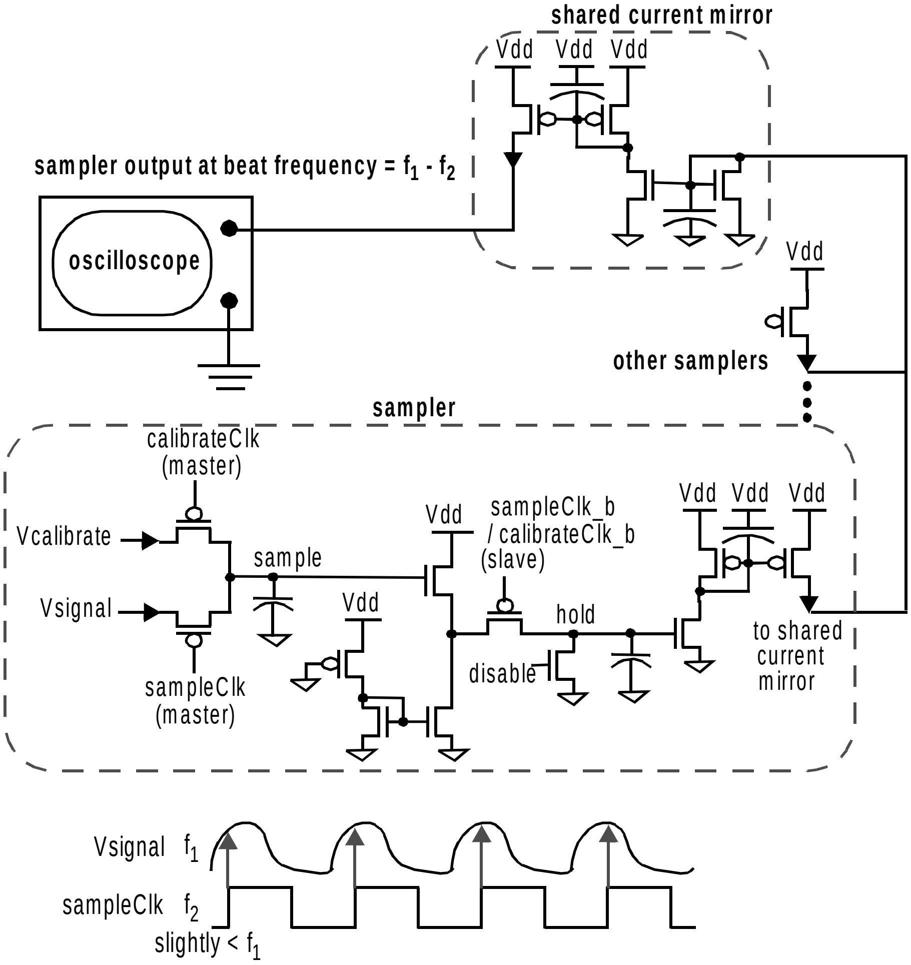


Figure 5.1: On-chip voltage sampler (buffered sample and hold).

5.1 On-Chip Voltage Sampler

Figure 5.1 is a schematic of the fast on-chip voltage sampler placed at every Vdata and Vref node in each I/O cell. The design is a conventional pass-transistor sample-and-hold, with a source follower stage between the master and the slave to prevent charge-sharing between the nodes marked ‘ hold’ and ‘sample’ which would otherwise impose a bandwidth limitation. PMOS transistors are used for the sampling and holding stages since the signals of interest are referenced to the on-chip supply (Vdd). An alternative sampling path is provided for calibrating the sampler. Each sampler can be enabled or disabled and hence different sampler outputs are multiplexed to reduce the total number of pins needed to implement this on-chip probing technique.

The operation of the sampler is also explained in Figure 5.1. If the on-chip periodic signal, at frequency f_1, is sampled with a sampling clock (sampleClk) at a slightly lower frequency f_2, the sampling point gradually moves along the signal period, giving a sampler output that is a replica of the on-chip signal at the beat frequency (f_1 - f_2).

The sampler bandwidth is determined by two factors. The first factor is the time spacing between the period of the input signal and the period of the sampling clock, which determines how fast a transition edge in the input signal the sampler can capture. In general, if $f_2 = x \cdot f_1$ where $0 < x < 1$, this time spacing is given by

$$\Delta T = T_2 - T_1 = \frac{1}{f_2} - \frac{1}{f_1} = \frac{1 - x}{x \cdot f_1} \quad , \tag{5-1}$$

and the bandwidth allowed by this time resolution is therefore

$$BW_{limit} = \frac{x \cdot f_1}{1 - x} \quad . \tag{5-2}$$

This bandwidth limitation can be easily removed by setting x close to unity.

The second factor is the sampling bandwidth allowed by the circuit. This is determined by the RC time constants at the nodes marked ‘sample’ and ‘hold’ in Figure 5.1. In our design, the RC pole at the sampling stage (i.e. at the sample node) sets the limit. Simulations during the design phase using process parameters supplied by the foundry showed a worst-case bandwidth of about 2.5GHz. However, the actual run turned out to be even slower than the SS corner of the process parameters provided, and the extracted capacitance values used in the design process were inaccurate. As a result, the bandwidth of the sampler in the silicon is only about 1.42GHz, which is much lower than expected. This means that the voltage sampler slows down, for instance, an on-chip signal with infinitely fast transition to an output waveform with 155ps rise or fall time. The signal transition times are approximately 300ps in our measurements. Hence, the effect of the bandwidth limitation is smaller: a 300ps transition on-chip is slowed down to 338ps in the sampler output waveform.

5.1.1 Sampler Calibration and Characterization

Voltage calibration of each individual sampler is necessary to ensure the accuracy of the mapped output waveforms. Each sampler is calibrated by passing a known DC voltage, measured externally using a multimeter, as the calibration input and recording the DC value of the sampler output. The DC input sweeps across the entire voltage range of interest at 5mV intervals.

Calibration results, using a 600MHz sampleClk, for the samplers on the two chips used in the link tests over the signal voltage range of interest are shown in Figure5.2 . The calibration curves show non-linear input-output voltage transfer functions where the gradients decrease by about a factor of two as the input signal gets closer to Vdd, and also suggest that the offsets in the samplers are random: in the first chip, the Vdata samplers have smaller mismatches than the Vref samplers, but the reverse is true in the other chip. The voltage calibration compensates for the non-linearity of the samplers, the random offsets which range up to about 100mV,and any voltage offset caused by clock coupling from sampleClk at the chip level and at the board level.

The samplers are re-calibrated multiple times to determine the accuracy of calibration results. The results from eight different calibration runs with the same set of samplers are shown in Figure 5.3. Changing the sampling clock frequency (from 500MHz to 650MHz) does not affect the calibration results, neither does changing any of the test controls: all the calibration curves are very close to one another when the measurements are taken one after another, without powering down the test setup. However, once the setup is powered down and then powered back up, the new calibration curve can move in any direction. For maximum accuracy in interpreting noise measurement waveforms, the samplers should be calibrated every time before they are used for voltage noise measurements, which is an extremely time-consuming process. Alternatively, we notice that the slopes of the calibration curves from different calibration runs are reasonably well-matched. The implication is that in interpreting voltage noise measurements, the *absolute value* of a measured voltage point (relative to chip Vdd) can be off by as much as 80mV due to the shift in calibration curve, but if we take two measured voltage points and map them using

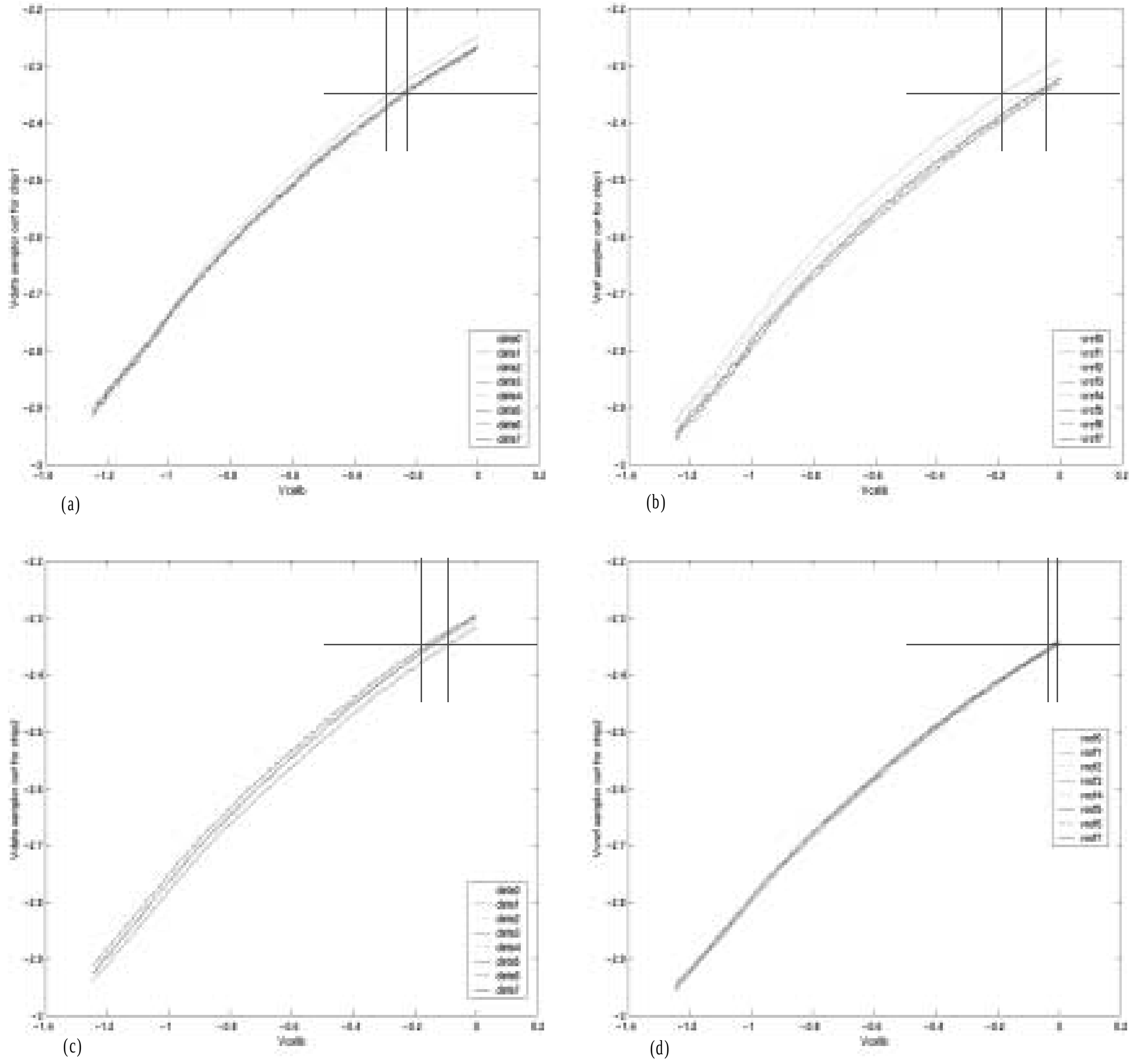


Figure 5.2: Voltage calibration results for samplers: (a) Vdata samplers in chip1, (b) Vref samplers in chip1, (c) Vdata samplers in chip2, and (d) Vref samplers in chip2. The vertical lines indicate the spread of the sampler offsets.

the same calibration curve, any error is cancelled out, and so the difference of the two mapped voltages, or the *amplitude* of a signal, is accurate. This serves our purpose as we are mostly interested in measuring the magnitude (amplitude) of each noise component.

A changing Vsignal value (source node voltage) causes the sampling PMOS pass gate to turn off at a slightly different point on the rising edge of sampleClk. Therefore, the sampler output exhibits a voltage-dependent time shift that also depends on the slew rate of sampleClk. The problem is illustrated in Figure5.4. This time shift is measured to be

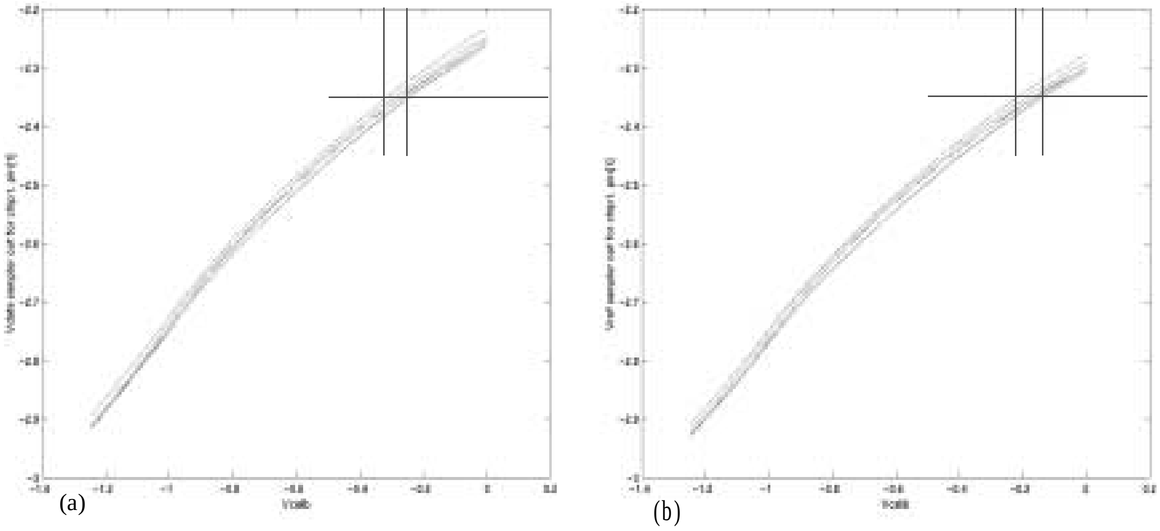


Figure 5.3: Results from multiple voltage re-calibrations of the same set of samplers: (a) at Vdata[1], and (b at Vref[1] in chip1. The vertical lines indicate the spread of the sampler offsets.

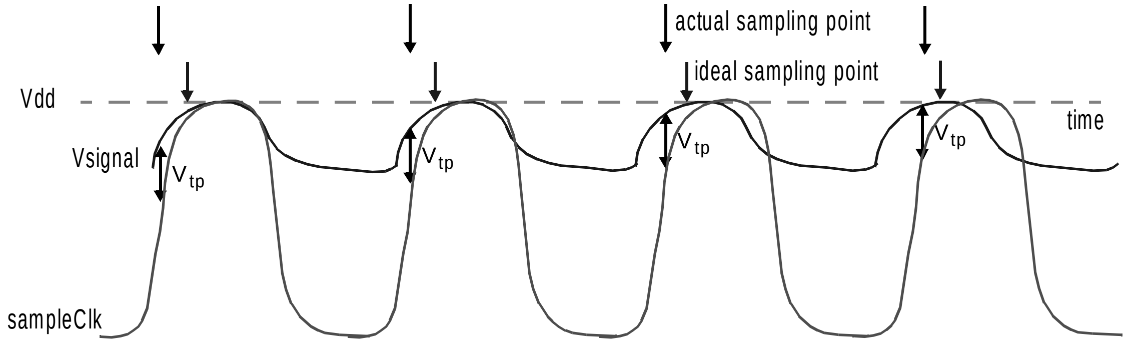


Figure 5.4: Sampler output exhibits voltage-dependent time shift.

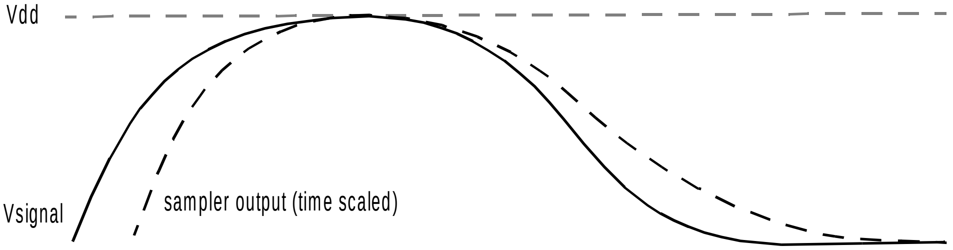


Figure 5.5: Sampler sharpens signal rising edge and slows down falling edge.

an almost linear function: 12ps for every 100mV that Vsignal is below the supply Figure 5.5 shows the resulting time shift. This time shift is mainly due to the rise time (slew rate) of sampleClk, as confirmed by simulations using measured process parameters from the wafers: the (body-effected) threshold voltage of the sampling PMOS

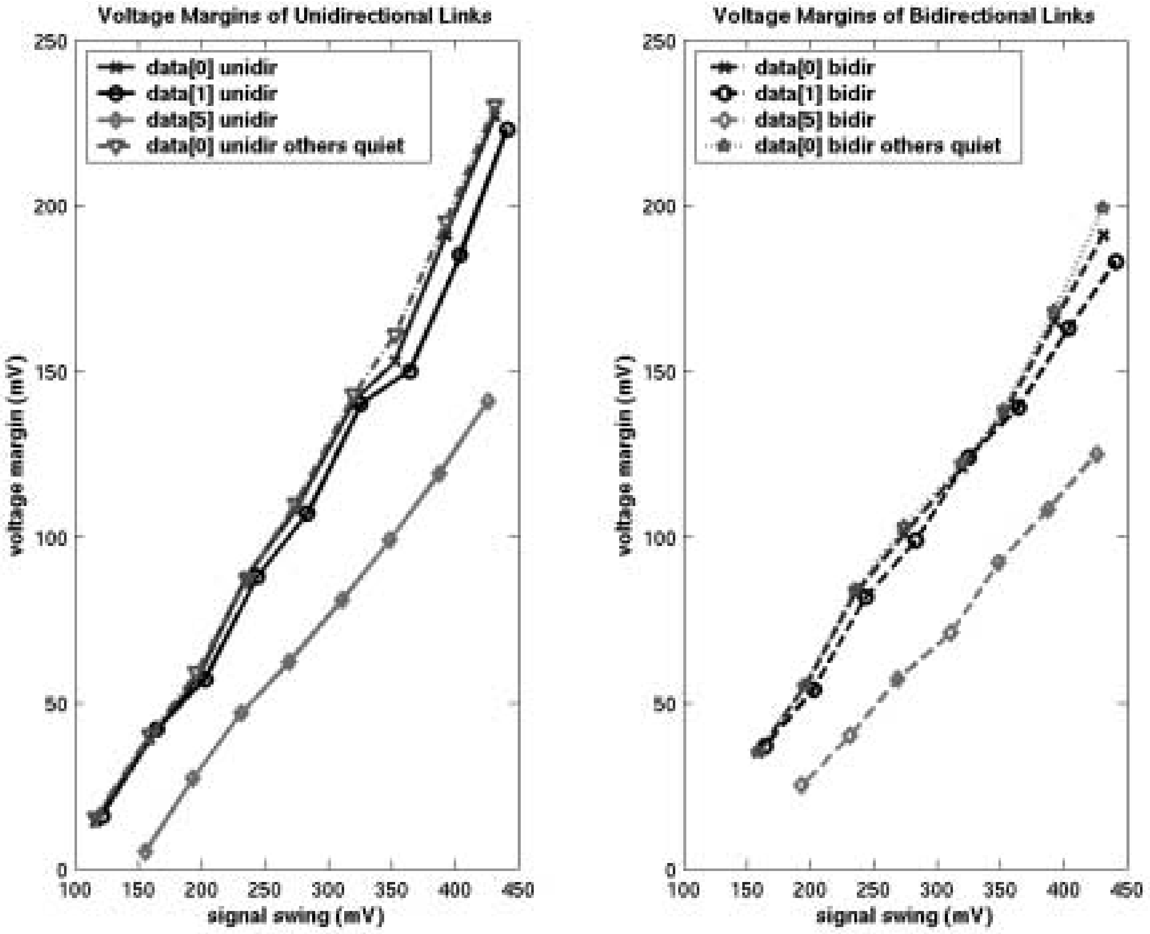


Figure 5.6: Measured voltage margins of unidirectional and bidirectional links as signal swings vary.

pass transistor (V_{tp}) is approximately 550mV, and simulations show that the rising clock edge of the internal sampleClk from 1.75V to 2.75V is 122ps.

5.2 Measurements

The voltage margins of different signal pins are measured, in both unidirectional and simultaneous-bidirectional operations, under different conditions and with different transmission signal swing. The data points give a set of straight lines. Figure 5.6 illustrates the measurement results for three signals with different signal return configurations, namely, data[0], data[1], and data[5], transmitting pseudo-random bit sequences (PRBS) at 1.2Gbps unidirectional data rate or 2.4Gbps bidirectional data rate.

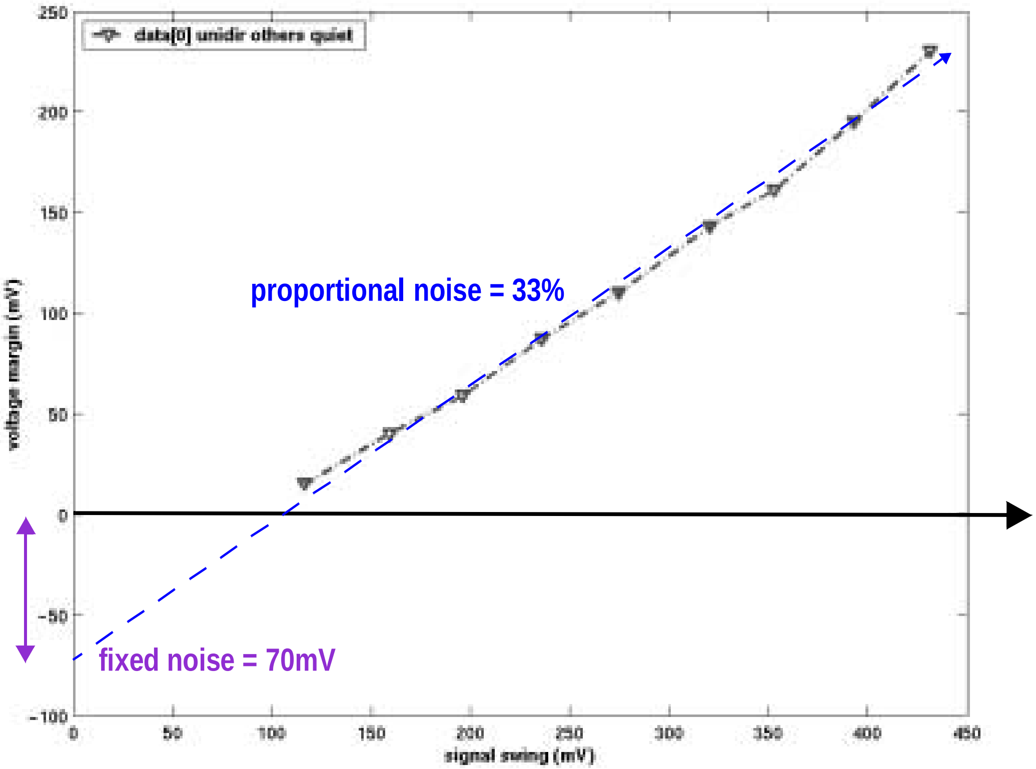


Figure 5.7: Extracting fixed and proportional noise values from voltage margin measurements.

We define voltage margin as the difference between the DC voltage swing and the total noise, and postulate that the voltage noise sources decompose into two groups: noise sources which are fixed in value and noise sources whose values change proportionally to the signal swing. The negative value of the y-intercept is the fixed noise, and the slope of the line corresponds to *(1 - proportional noise)* . Using a linear fit to analyze the data points for unidirectional data[0] when all the other data signals are idle, we see about 70mV of fixed noise and 33% proportional noise, as illustrated in Figure5.7 .

Then we measure the voltage margins when all the data signals transmit PRBS data[1]. A summary of the extracted voltage noise values for data[0], data[1], and data[5] is

[1] The two sets of unidirectional refClk lines are always active and cannot be turned off without turning *all* data signals off as well, i.e. they are active in all the measurements. In particular, they are active in the measurements where all data signals are active, and hence provide a signalling environment consistent with the noise models.

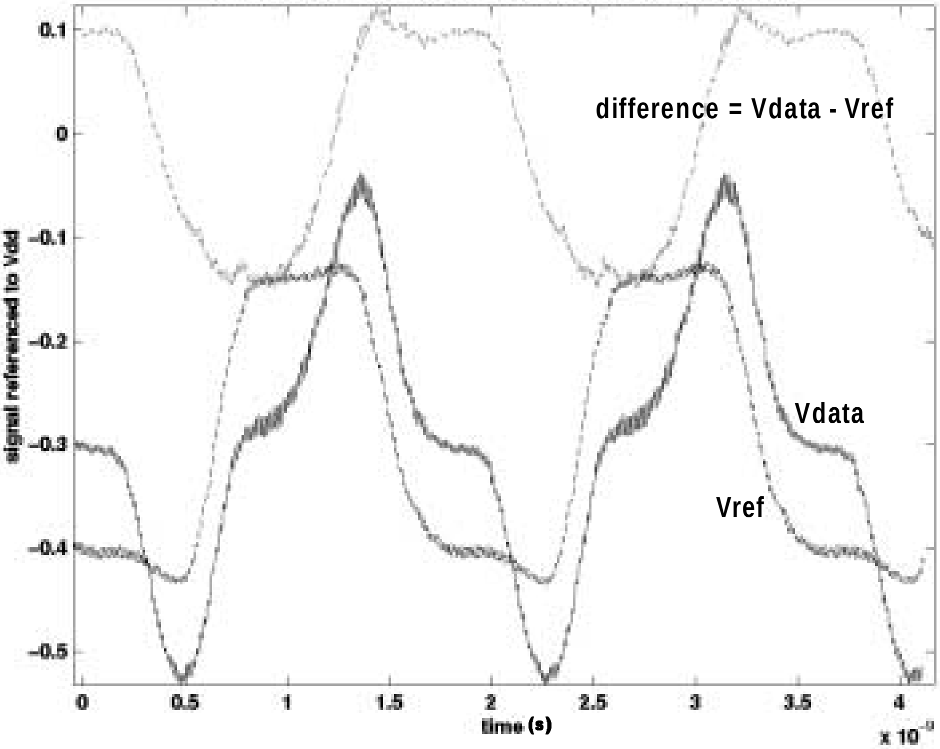


Figure 5.8: Bidirectional on-chip signals.Voltage margin falls to a minimum when the transmit and receive signals are in quadrature phase as shown.

shown in Table 5-1. The voltage margins in data[0] and data[1] are very similar to each other, while the voltage margins in data[5] are significantly worse.

Table 5-1: Fixed and proportional noise values in unidirectional and simultaneous bidirectional links extracted from voltage margin measurements.

		data[0], others quiet	data[0], all PRBS	data[1], all PRBS	data[5], all PRBS
unidirectional	**fixed noise**	**70mV**		**64mV**	**69mV**
	proportional noise	**33%**	**34%**	**37%**	**51%**
bidirectional	**fixed noise**	**57mV**	**53m**	**50mV**	**60mV**
	proportional noise	**42%**	**45%**	**47%**	**57%**

The voltage samplers are then used to capture on-chip waveforms and measure the voltage noise sources directly. For instance, Figure 5.8 shows 2.2Gbps 280mV-swing bidirectional on-chip signals (voltage-compensated). As we will see later in this chapter, the voltage margin of a bidirectional link may change as the phase relationship between the transmit and receive signals varies because of timing mismatch in Vdata and Vref. When the receive and transmit signals are set up to be in quadrature phase as shown, the voltage margin generally falls to a minimum.

In the next section, we identify the voltage noise sources present in the implemented signalling system, model and estimate their magnitudes analytically, and verify the accuracy of the noise model by measuring the noise sources directly using the voltage samplers.

5.3 Noise Sources in Implemented Signalling System

To evaluate the noise sources in our test chip, we apply the principles of superposition to progressively build up a complete noise model. Using simple hand calculations aided by simulations of the noise model, we estimate the values of the voltage noise components. We then measure each individual component from the test chip to check the accuracy of our assumptions and noise model.

The transceiver front-end of the single-ended, simultaneous bidirectional parallel link interface in the test chip is shown in Figure5.9. The multiple-segment structures of the output driver and the reference-select mux are collapsed to only one leg for each to simplify the figure. As mentioned earlier in Chapter 3, VrefH and VrefL are adjusted to measure voltage margins. As signal swing varies, their center values are also adjusted.

Figure 5.10 depicts the essential components in the signalling scheme that appear in the noise model . The figure also shows the current each component carries when the output driver turns on during a transmit operation, and when the chip receives. (Switching off the driver is essentially sending a negative current pulse through the circuit.) For our measurements, the signal transition times are approximately 300ps and the signal propagation delays through the package traces are somewhere between 40 to 6 5ps. Therefore each package trace is modelled by simply a lumped capacitor at its lead (C_{lead})

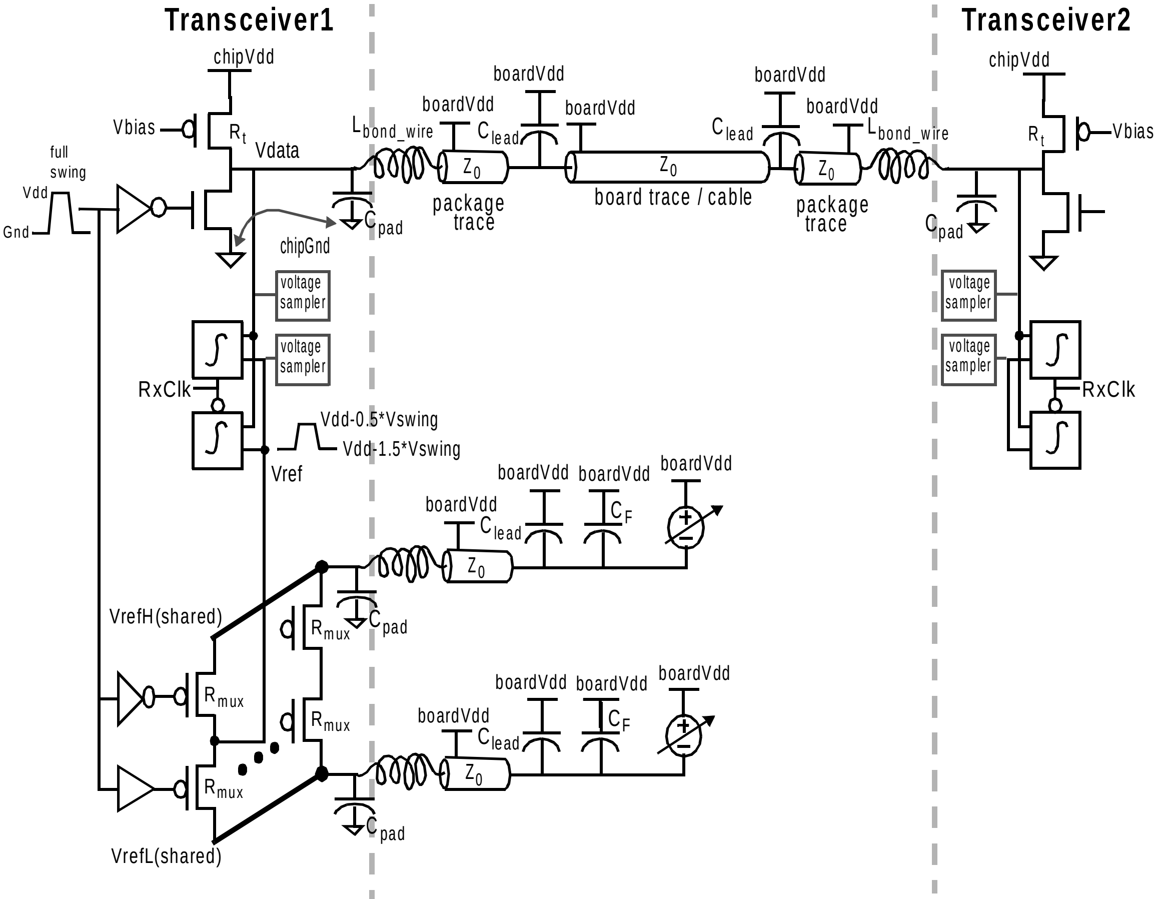
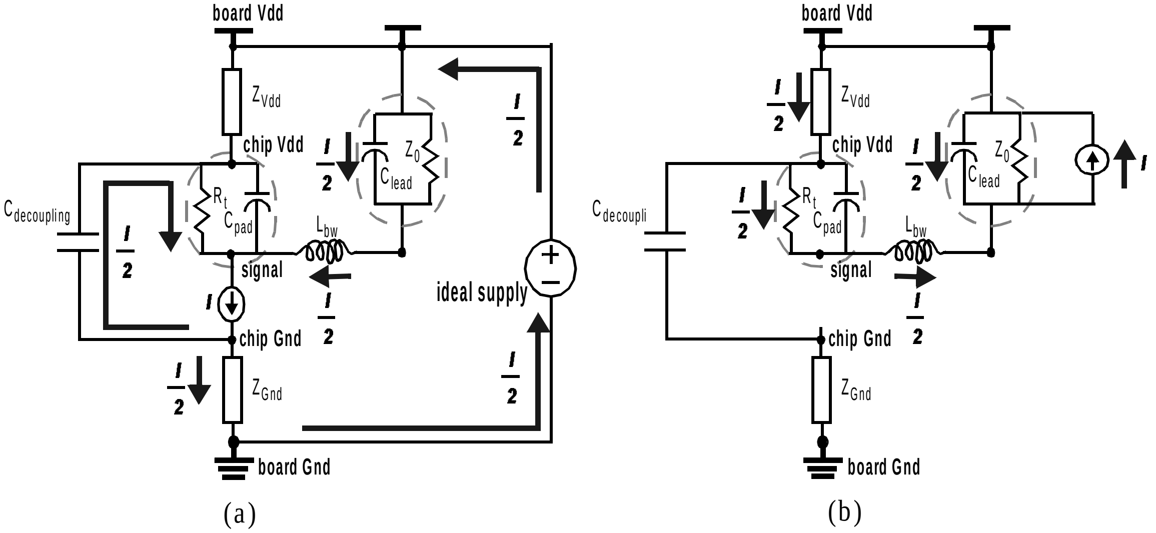


Figure 5.9: Implemented single-ended, simultaneous bidirectional parallel link interface. Both VrefH and VrefL are adjusted as signal swing varies.



(a)

(b)

Figure 5.10: Current flow in each I/O cell in the test chip when the chip (a) transmits, and (b) receives.

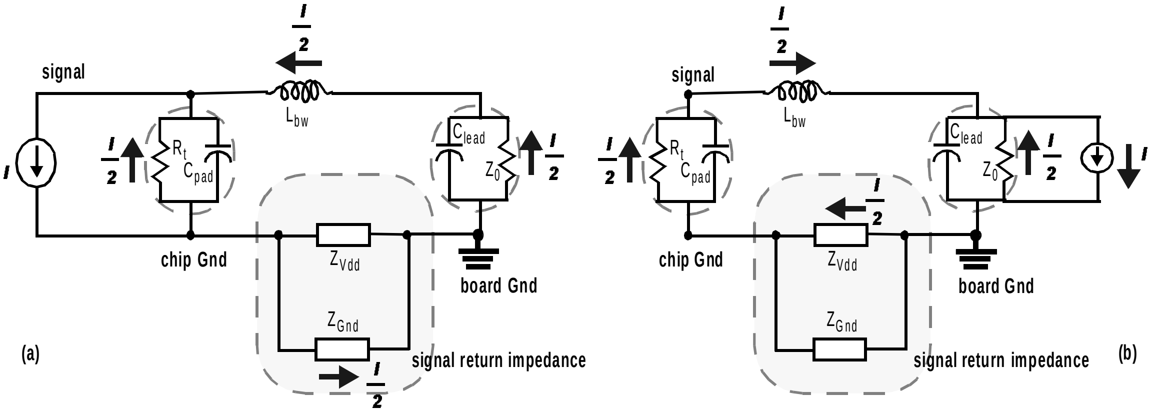


Figure 5.11: AC model of the signalling system in Figu re5.10.

where the package pin is soldered to the pad on the test board. If there is sufficient on-chip decoupling (for the frequency range of interest), chipVdd and chipGnd move together. Assuming infinite bypass between chipVdd and chipGnd, we can derive an AC model of the signalling system as shown in Figure5.11. All signals are ‘inverted’ so that they are now referenced to Gnd instead of Vdd. Measurements using on-chip samplers of chipVdd to chipGnd noise indicate that it is less than 20mV peak-to-peak in all cases (even when all signals are active), reassuring us that the above assumption is reasonable.

In the rest of this section, we will look at the noise sources in our signalling system. We first study, in Section5.3.1, channel attenuation and inter-symbol interference, which combine to form the largest proportional noise source in our signalling system. Then, in Section5.3.2, we look at noise coupling from on-chip clocks, which is found to be the largest fixed noise source. Next, we study the effects of on-chip power supply noise in our single-ended links in Section5.3.3, the different inter-signal cross-talk components in Section5.3.4, and reference offset in Section5.3.5. In Section5.3.6, the additional noise sources arising from the coupling between the transmit and receive signals on the same wire when the links operate in simultaneous bidirectional mode are considered. Finally, in Section5.3.7, we examine whether switching the Vref in one pin affects the voltage margins of the pin itself and of the other pins.

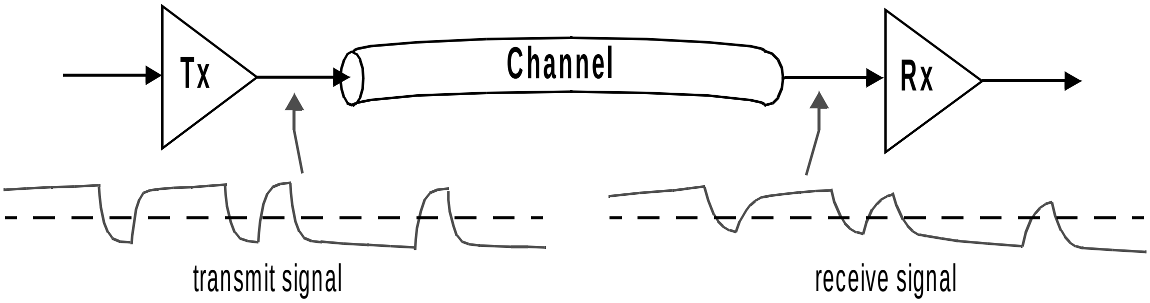


Figure 5.12: Channel attenuation and inter-symbol interference.

5.3.1 Channel Attenuation and Inter-Symbol Interference

As explained earlier in Chapter 2, the channel attenuates and disperse s the traversing signal pulse, leading to signal attenuation and inter-symbol interference. Channel attenuation and inter-symbol interference are present in all links. This problem is illustrated in Figure5.12. The worst-case receiver eye is bounded by an isolated '1' and an isolated '0'.

The resistance of the channel attenuates the traversing signal. The series resistance per unit l ength depends on the resistivity of the conductor and the conduction area. High-frequency current flows mostly near the surface of a conductor, and the current density falls off exponentially with its distance from the surface. This effect, the Skin Effect, leads to a smaller conduction area and hence higher series resistance for high er frequency signals. Dielectric conduction also causes channel loss , and increases with signal frequency. The above two mechanisms combine to make t he channel a frequency-dependent band-limited filter that reduces the signal amplitude at the receiver.

The band-limiting effect also broadens the traversing signal pulse. Moreover, the channel has some group delay (i.e. delay dependent on signal frequency), and hence the different frequency components reach the receiver with different delays, causing the received signal pulse to spread out in time. Both of these phenomena make the channel dispersive, with a long-tailed channel impulse response which causes ISI.

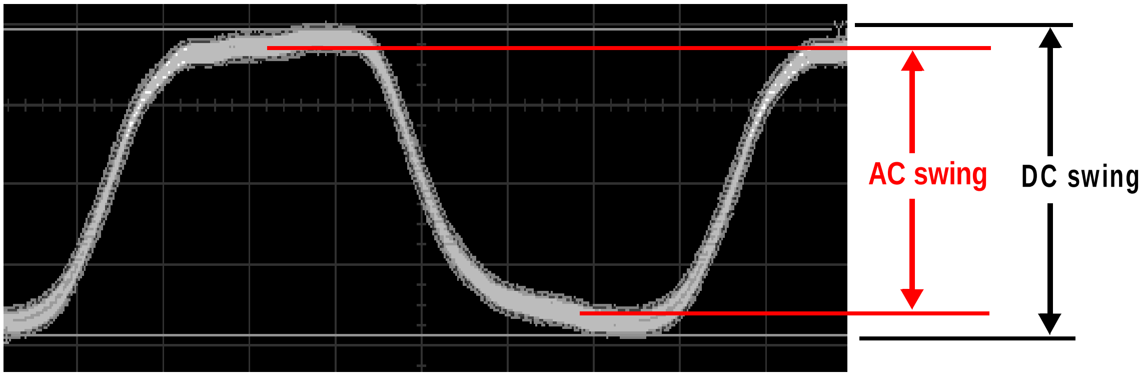


Figure 5.13: Difference between DC and AC signal swings.

Other possible ISI sources are reflections of previous bits due to termination mismatches or impedance discontinuities in the channel, and incomplete settling of the transmit signal within one bit time,.

The magnitudes of the channel attenuation and ISI depend on the quality of the channel, and the losses at the transmitter and at the receiver. In our system, they are found to be the largest voltage noise components. Over half of the proportional voltage loss in the voltage margin measurements can be attributed to the way we define the signal swing, which we have defined to be the difference in DC levels when the transmitter outputs a '1' and a '0' permanently. However, when a bit stream with alternating zeros and ones is transmitted, the signal swings to only 84%[2] of the DC swing at the midpoints of the bit time, as can be seen in Figure5.13. This problem is even more severe when PRBS data are used: the eye height bounded by an isolated '1' pulse and an isolated '0' pulse, as measured by the PRBS eye, is further reduced to only 80%[3] of the DC swing. These signal loss figures include the signal attenuation in the transmitter board trace, measured to be approximately 3% on a 600MHz sinuosoid using the network analyzer. Measurements show that another 3% is lost in the receiver board trace, and the loss in the cable is small enough to be ignored. Therefore, the total signal reduction is about 23% in each link, which is 70% of the proportional noise observed in data[0] when the other pins are idle.

[2] These measurements take into account the signal spreading caused by other voltage and timing noise sources, i.e. we take the midpoint of each thick line captured on the scope.

[3] See Footno te2.

5.3.2 Noise Coupling from On-Chip Clocks

Internal clock coupling is the major source of fixed noise in our measurements. We measure the contribution of each on-chip clock separately by considering its effect on the differential signal (Vdata[0]-Vref[0]) and comparing the difference when the clock is turned on and off. TxClk induces a 20mV peak-to-peak coupling noise which is correlated in phase with signal transitions; RxClk induces a 22.1mV peak-to-peak coupling noise whose phase relationship to signal transitions depends on the actual phase control settings; and cleanClk induces a 26.4mV peak-to-peak noise which can be uncorrelated to signal transitions. These large internal clock couplings are surprising but we have not been able to track down the exact causes.

The coupling from sampleClk, which can occur at any point within the bit time as it sweeps along, is measured by observing its effects on the voltage margins of data[0] at different signal swings when sampleClk is active and when sampleClk is shut off. The difference in the measured voltage margins is found to be less than 2mV at all swings.

The same fixed noise sources, of similar magnitudes, are measured in the other signal pins as well. Specifically, we observe similar clock couplings on data[1] and data[5], and hence expect similar voltage margin reductions due to these fixed noise sources.

5.3.3 On-Chip Power Supply Noise

The major noise path from on-chip Vdd and Gnd is differential coupling onto the signals Vref and Vdata at the receiver. Th ese signals are coupled to the supplies differently, making rejection of power supply noise imperfect. Specifically, Vref is more heavily coupled to the power supply at high frequencies than each data signal, as illustrated in Figure 5.14, and hence high-frequency power supply noise coupling is not common-mode. This differential noise coupling explains why the effect of power supply noise is much more prominent in a single-ended system than in a differential system.

We created a model, shown in Figure5.15 , to investigate the effect of receiver and transmitter power supply noise in the test chip. A 1V AC noise is injected at chipGnd to

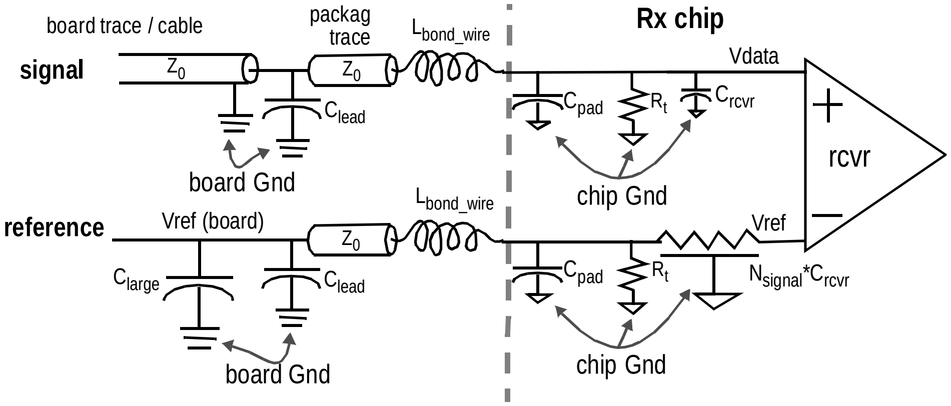


Figure 5.14: AC reference nois in a load-terminated parallel link. The data and reference lines are coupled to the power supply differently, making their supply noise rejection different.

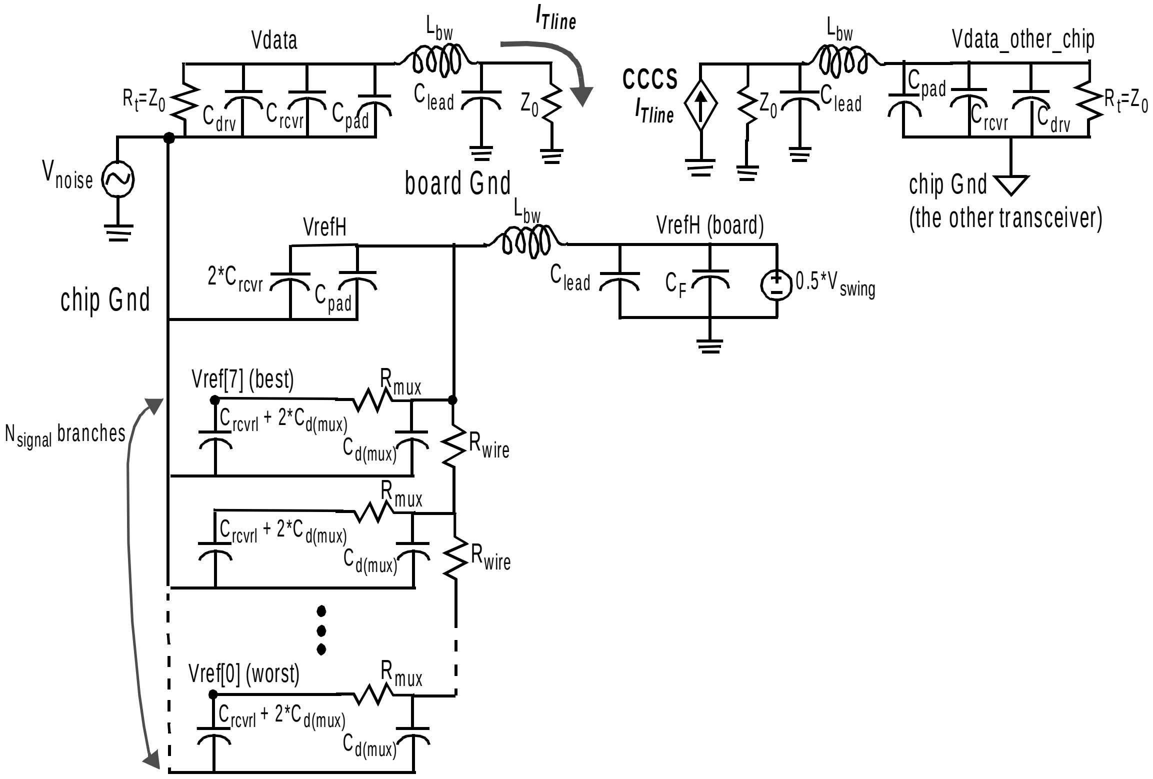


Figure 5.15: Model to analyze AC reference noise due to receiver and transmitter on-chip power supply noise in the test chip.

evaluate the extent it couples to the data and local Vref nodes in the chip, and to the data node in the other chip. (No noise is induced on the local Vref node of the other chip since the Vref voltages of the two chips are not connected.) For simplicity, all non-linear resistances (of transistors) and source and drain capacitances are linearized in the model.

VrefH is also tied to the current-integrating receivers in the two sets of unidirectional refClk lines and hence carries an extra capacitive loading of 2*C_{rcvr}[4], which is not present in VrefL. In unidirectional links, all the reference-select muxes are steered towards VrefH. This also represents the worst-case scenario in simultaneous bidirectional links because the difference in loadings on Vdata and VrefH is the largest. Because of layout constraints, the VrefH and VrefL lines are not distributed from the center of the chip. Instead, they are routed from one edge and hence the absolute worst-case local Vref is the one that is furthest away on the opposite edge (i.e. Vref[0]).

The signal pins have different signal return configurations and hence different self-inductance values, which are accurately calculated in Section5.3.4.2 using a 3-dimensional field solver. To generate a model usable for all signal pins here, L_{bw}=1.5nH is used. In addition, we assume there is no attenuation in the channel. The current arriving at the receiver is equal to the current pushed into the transmission line, modelled by a current-controlled current source (CCCS).

Given the above assumptions and simplifications, the model is simulated using Z_0=50Ω, L_{bw}=1.5nH, C_{lead}=5pF, C_{pad}=1pF, N_{signal}=8, C_{rcvr}=128fF, C_F=28nF[5], C_{drv}=160fF[6], R_{wire}=1.2Ω, R_{mux}=390Ω when the mux is on, and $C_{d(mux)}$=40fF[7].

The simulation results are summarized in Figure5.16. The difference in noise response between the furthest away local Vref (Vref[0]) and the nearest one (Vref[7]) is negligible. At low frequencies, about half of the power supply noise is coupled to Vdata but not to the local Vref, causing a 50% differential noise coupling with approximately 100° phase difference. From 600MHz to 1.8GHz (its third harmonic frequency), the magnitude of the differential noise increases from 56% to 108%. Therefore, the receiver power supply noise induces a huge differential noise.

[4] C_{rcvr} is the total gate capacitance for each pair of current-integrating receivers.

[5] C_F is the sum of a pair of surface mount capacitors (1nF and 27nF respectively) used to damp out high frequency noise transients from the power supplies at VrefH and VrefL on each transceiver board. The inductances of these capacitors are not modelled.

[6] C_{drv} is the sum of drain capacitances of the output driver and terminator resistor.

[7] $C_{d(mux)}$ is the drain capacitance of the PMOS switch in the mux.

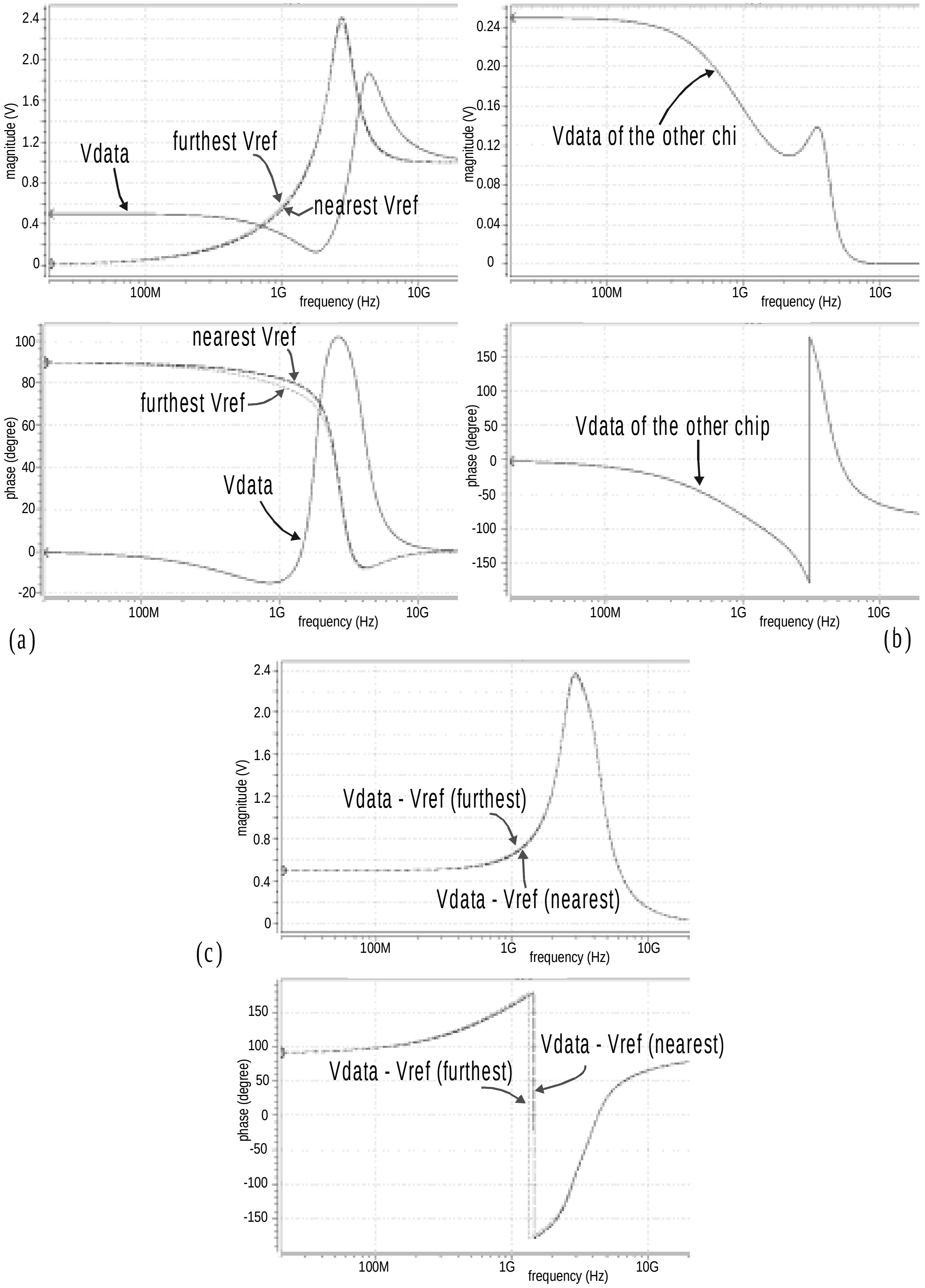


Figure 5.16: Magnitude and phase plots for the worst-case AC reference nois due to on-chip power supply noise: (a) noise response at receiver, (b) noise response at transmitter, and (c differential noise Vdata - Vref) at receiver.

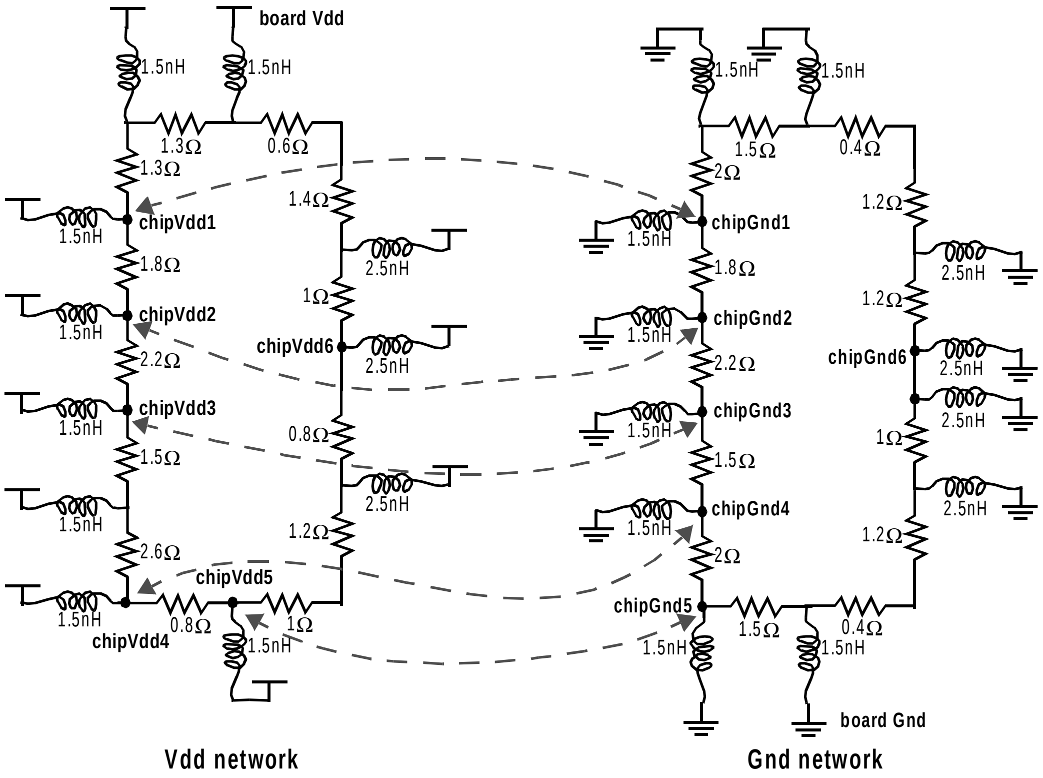


Figure 5.17: Model for the power supply networks in the test chip.

The supply noise also couples to Vdata at the other side of the link, by about 20% in magnitude at 600MHz and 11% at 1.8GHz with considerable phase shifts at both frequencies.

Knowing how on-chip supply noise couples to the data and local Vref nodes in both the transmitter and the receiver, the next step is to estimate the amount of supply noise present. One supply noise component is contributed by the switching activities of the other on-chip circuitr , and its magnitude is independent of the I/O signal swing. However, this term can be minimized by on-chip bypass capacitors, and only moves Vdd relative to Gnd. It does not move the common-mode.

On the other hand, the supply noise induced by the switching activities of the I/O signals tends to translate the chip Gnd, and is proportional to the signal swing. A simplified model of the power supply networks (Vdd and Gnd) is shown in Figure 5.17. The model ignores effects like self-inductance of the power distribution networks and

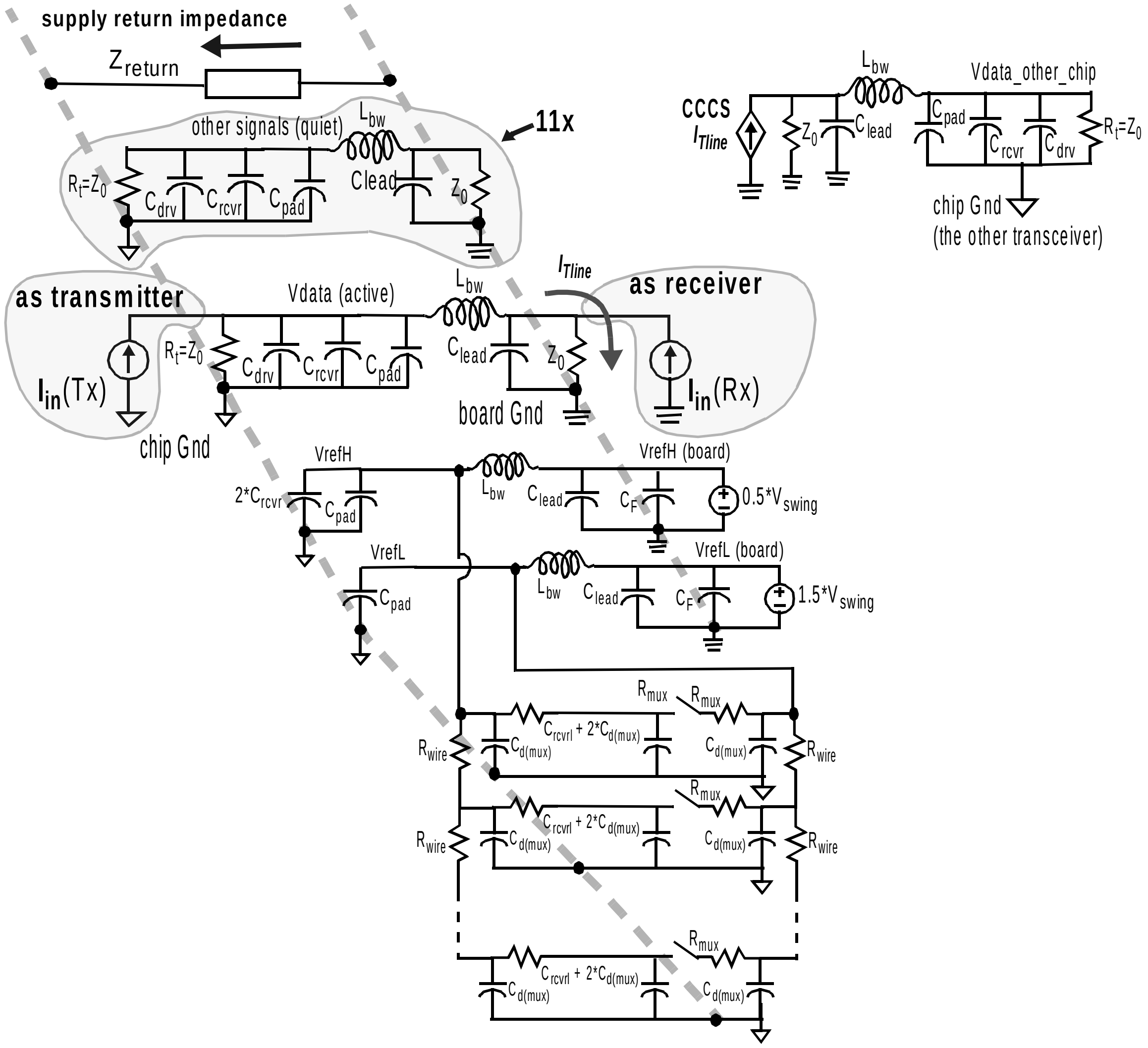


Figure 5.18: Model to analyze receiver and transmitter power supply noise induced by a switching signal. Details about local power supply nodes are omitted in this figure, but are taken into account in simulations.

mutual inductance between different bond wires. Since the supply pins are designed to occur in pairs (except for one), there are equal numbers of local Vdd nodes and local Gnd nodes.

We assume perfect decoupling between each pair of Vdd and Gnd pins, and hence they are shorted together in the AC model (i.e. chipVdd1 shorted to chipGnd1, chipVdd2 to chipGnd2, and so on). The circuit model in Figure5.18 is used to estimate the power supply noise a signal induces when the chip serves as a transmitter (I_{in}(Tx) is on) or when it acts as a receiver (I_{in}(Rx) is on). Therefore, the model contains the active signal branch itself, the other signal branches to account for their loadings, and the two Vref generation branches where all the muxes select VrefH. As a transmitter, I_{in}(Tx) toggles between 18mA and 0mA at 600MHz, at 833.3ps bit time with 300ps rise and fall times, and I_{in}(Rx) remains at 0; as a receiver, the same current excitation is applied in I_{in}(Rx) while I_{in}(Tx) is set to 0. To improve the accuracy of the simulation results, I_{in}(Tx) and I_{in}(Rx) are voltage-controlled current sources, controlled by the output voltages of actual output drivers that transmit the desired excitations (not shown in Figure5.18). This arrangement removes unreal sharp corners and hence high-frequency components in I_{in}(Tx) and I_{in}(Rx) which induce illusory ringings and high-frequency noise in the simulation results.

To match the signal paths, each of the unidirectional line s (i.e. refClkIn[1:0] and refClkOut[1:0]) is designed to be identical to a bidirectional line and hence the same model can be used. The Vref node, however, is different: a bidirectional line is hooked up to two muxes, and a unidirectional link connects to VrefH directly. This difference is also reflected in the model. As a result, from a loading perspective, the noise model has 12 signal lines altogether. In calculating the total power supply noise, however, there are 10 transmitters and 10 receivers.

Even though all chipGnd nodes are shown to be shorted together in Figure 5.18 for simplicity, we take into account the signal placement in simulations: data[0] returns to chipGnd1, data[2:1] to chipGnd2, data[3] together with refClkIn[0] and refClkOut[0] to chipGnd3, data[7:4] to chipGnd4, VrefH and VrefL to chipGnd5, and finally refClkIn[1] and refClkOut[1] to chipGnd6. Each reference-select mux is tied to the same chipGnd as the corresponding signal itself. This arrangement roughly follows the chip layout, and ignores the distance between each chipGnd/chipVdd pair and the exact positions of the decoupling capacitors.

To see whether the power supply nodes in Figure 5.17 are equipotential, a current pulse is transmitted at data[0] while all the other signals remain quiet. Figure 5.19 plots the simulated voltages at the labelled chipGnd nodes. The results clearly show that the supply nodes are not equipotential; the Gnd noise induced at each local supply node

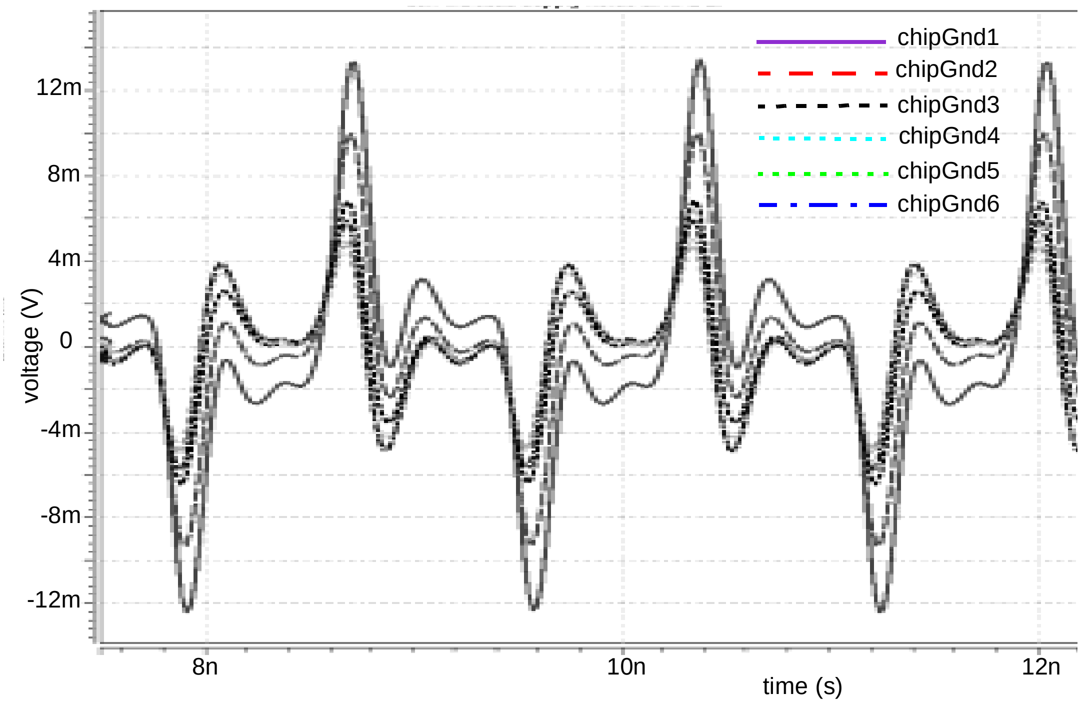


Figure 5.19: Power supply noise induced at different chipGnd nodes by transmitting a current pulse at data[0] while all the other signals are inactive. The magnitude of th Gnd noise induced at each node depends on its proximity to the excitation.

assumes a slightly different shape (and hence has different frequency compositions); its peak magnitude depends on the supply node's proximity to the excitation: the close r the node is to the excitation, the larger is the induced supply noise.

The noise model in Figure 5.18 allows the evaluation of many voltage noise sources. First of all, we activate the signals one at a time to find the self-induced chipGnd noise by a switching signal, either transmitting or receiving. Figure5.20 illustrates the noise induced at the *local* power supply node in each case. The plots indicate that, in general, the local supply noise induced is about the same. For data[0], the 450mV transmit signal induces a peak-to-peak noise of 25.6mV[8] (i.e. a 5.7% proportional noise) on chipGnd1 while the receive signal induces an 11.5mV (2.6%) local Gnd noise.

Combining this induced Gnd noise with the noise coupling mechanism described earlier shows that in unidirectional data[0], an active receive signal induces a *peak-to-*

[8] The induced noise does not exactly center around zero since the driver output rising and falling transitions are slightly unbalanced,

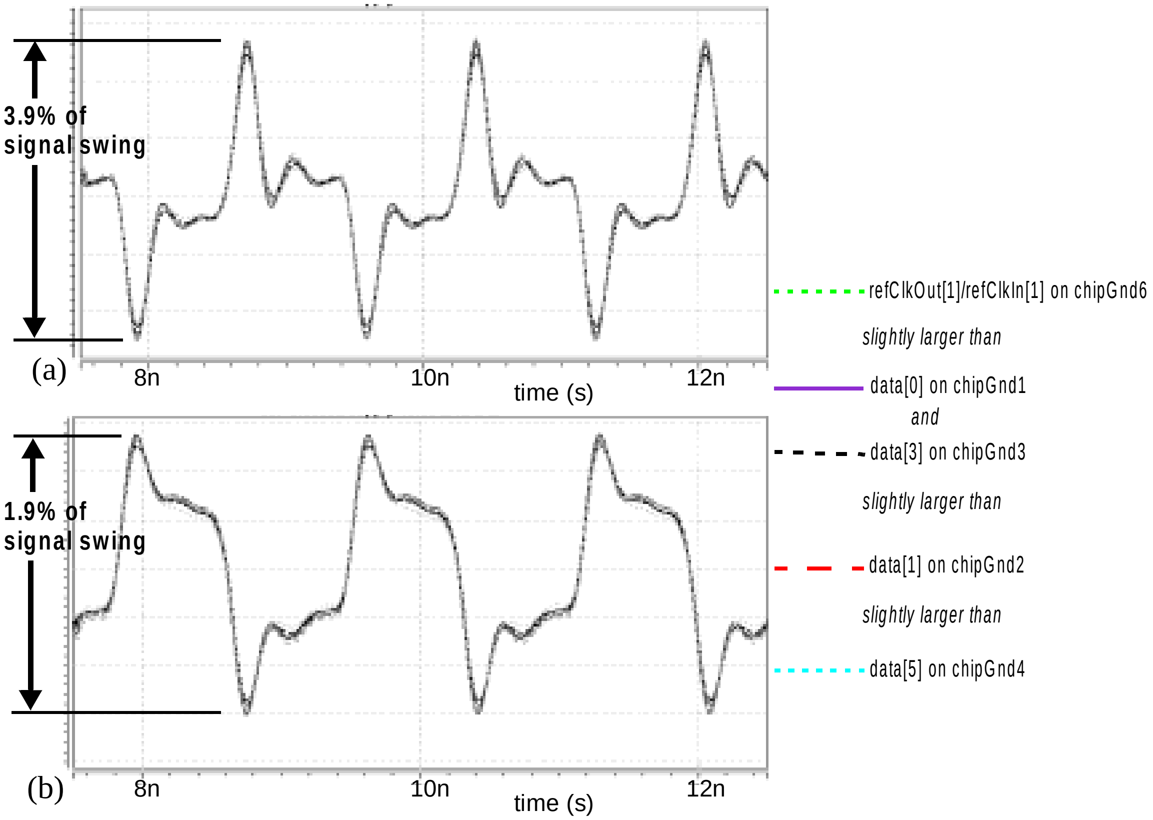


Figure 5.20: Noise induced by a switching signal at its local power supply node when all the other signals are inactive: (a) in transmitting, (b) in receiving. The results are about the same for all pins.

peak noise in (Vdata[0]-Vref[0]) that corresponds to a 1.9% voltage margin deduction; and in bidirectional data[0], the peak-to-peak differential noise induced by the transmit signal is 3.9%, while the coupling on Vdata from the other chip's power supply noise can add another 0.7%[9], making the total differential noise 6.5%.

Repeating the same analysis for data[1] and data[5], we conclude that the reference noise due to self-induced power supply noise is approximately the same in all data pins -- an expected observation since the magnitude of the local chipGnd noise is about the same as found in Figure 5.20.

The effect of the noise is significantly reduced in the case of ideal matched filter receiver, where the differential noise is integrated over one bit time. Results demonstrate

[9] The noise coupling from the power supply noise induced by an active transmit signal to the Vdata of the other chip is 0.7%. Therefore, by duality, the coupling on Vdata from the other chip's power supply noise can add another 0.7%.

that the receive signal induces just 0.4% peak-to-peak noise and the transmit signal induces an additional 0.5% peak-to-peak noise when integrated using a sliding window clock at the same frequency, while the effect of power supply noise in the other chip is negligible. Hence, the voltage margin reduction in unidirectional links is reduced to only 0.4%. Since superposition holds for integration in linear systems, we can sum the numbers to give a total of only 0.9% in bidirectional links. This huge reduction in proportional noise after integration can be attributed to the high-frequency composition of the induced differential noise on (Vdata-Vref).

We then check the accuracy of the above simulation results by measuring the noise sources directly from the test chip. To measure the self-induced power supply noise, all the other signals are kept quiet to get rid of all cross-talk components. At both the transmitter and the receiver, we first capture Vdata[0] and Vref[0] when all signals are idle, and capture the waveforms again when data[0] alone switches (unidirectional operation). In this way, we eliminate all the background noise sources that are not common-mode.

As data[0] signal swing varies, the movements in the sampler outputs suggest that the self-induced power supply noise is 3.7% at the transmitter and 2.8% at the receiver, and the peak-to-peak differential noise on (Vdata[0]-Vref[0]) is 2.1% at the transmitter and 1.6% at the receiver. It is encouraging to note that the measured power supply noise and differential reference noise figures at the *receiver* agree very well (within 15%) with the values predicted by the noise model. However, the measured noise figures at the *transmitter* are about 40% lower than the simulation results. Possible causes for such discrepancies are examined later in Section5.4. R epeating the same sequence of experiments with data[1] and data[5] yields similar measurement results, confirming the earlier simulation results that show little difference in the self-induced power supply noise across different pins.

5.3.4 Inter-Signal Cross-Talk

As discussed in Chapter 2, unidirectional links are often affected by far-end cross-talk only, but simultaneous bidirectional links are affected by both far-end cross-talk and the generally larger near-end cross-talk. The implemented interface suffers from both types of

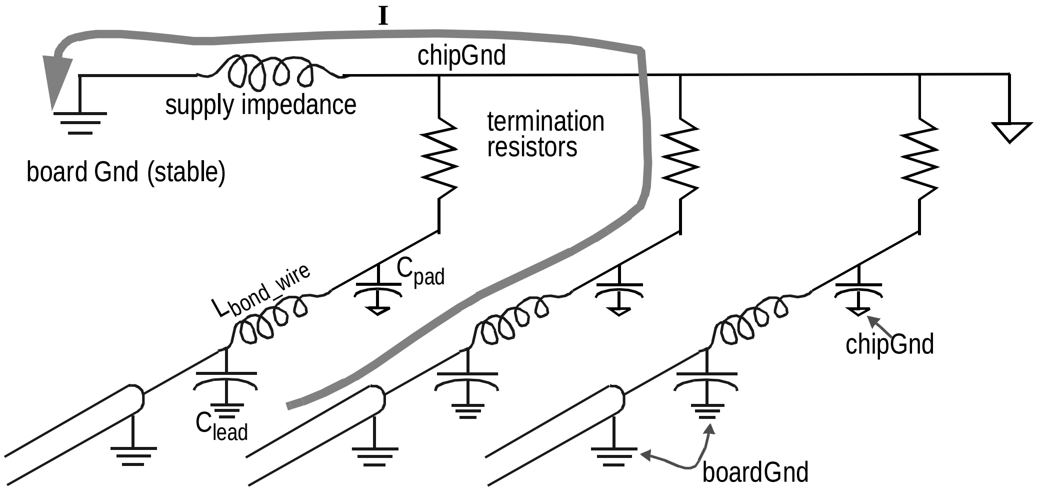


Figure 5.21: Power supply acts as a shared current return path for I/O signals in a single-ended parallel link.

inter-signal cross-talk: cross-talk via a shared power supply (or a shared signal return), and direct capacitive and inductive cross-talk between adjacent signals -- both are in general more significant at the transmitter than at the receiver. Hence, inter-signal cross-talk becomes a more important issue in simultaneous bidirectional links.

5.3.4.1 Cross-Talk Via Shared Signal Return

In Section5.3.3, we clearly see that power supply noise induces reference noise in single-ended links and hence reduces the receiver voltage margins. Compared to differential links, the magnitude of power supply noise, at both the transmitter and the receiver, is often larger in single-ended links. The power supply acts as a shared current return path for the I/O signals, leading to inter-signal cross-talk: the return current for a signal induces noise across the impedance of this shared return, therefore moving the on-chip supply, which couples back to the receiver inputs. Figure5.21 illustrates the problem in a load-terminated design. Similarly, on the transmitter side, the dI/dt noise induced on the transmitter supply when output drivers switch can affect other signals.

Strictly speaking, noise coupling through the power supply and noise coupling through a shared signal return can be two different noise sources, but since such shared signal returns are often times also power supplies, we do not make a clear distinction in the discussions here. Intuitively, the worst-case power supply noise induced in a system, and therefore the worst-case signal return cross-talk, increases with the number of signals sharing the same return and the impedance of the signal return itself.

We find the total cross-talk when one signal is quiet and all the others are switching in the same direction. Any cross-talk noise is synchronous with the signal that induces it. The presence of inter-signal timing skew changes the phase relationships between the quiet signal and the cross-talk components from other signals. Hence peak-to-peak noise values are used in our analysis. The peak-to-peak values of the total far-end cross-talk, due to shared signal returns and the other signals' return currents, on data[0], data[1], and data[5] are all about 11% of the signal swing. These represent the voltage margin reductions in unidirectional links using a sampling receiver. The corresponding total near-end cross-talk are all about 18%, meaning that the voltage margin reductions rise to 29% in bidirectional links! If an ideal matched filter receiver is used, however, the far-end cross-talk figures integrate to 1.3%; while the near-end cross-talk figures integrate to 3% using a sliding window clock at the same frequency. Therefore, the voltage margin reductions are greatly reduced to 1.3% in unidirectional data[0], data[1], and data[5], and 4.3% in the bidirectional links.

5.3.4.2 Capacitive and Inductive Cross-Talk between Bond Wires

The dominant inter-signal cross-talk source is from the direct capacitive and inductive coupling between signals, which can occur at any point in the signal transmission paths: between parallel signal traces inside packages or on the boards, or between bond wires and package leads.

When two long transmission lines run in parallel next to each other (as in the case of parallel package or board traces), their mutual inductance induces a positive backward travelling noise pulse and a negative forward travelling noise pulse, while their mutual capacitance induces positive noise pulses in both directions. Hence, the two backward travelling noise pulses superimpose while the two forward travelling noise pulses subtract, making the near-end cross-talk larger than the far-end cross-talk. The magnitude s of the cross-talk components depend on the surrounding medium. In a homogenous medium, i.e.

the transmission lines are surrounded by the same material (e.g. striplines), the inductive and capacitive couplings are equal in magnitude, and hence the forward travelling cross-talk components cancel each other exactly, making the far-end cross-talk equal to zero[10].

Bond wires and other package parasitics can be modelled using lumped elements -- capacitors (to supply) and (self) inductors along each signal path, and mutual capacitors and mutual inductors in between signal paths -- if the delays across the paths under consideration are short compared to the signal transition times. In general, the receive signal carries smaller high-frequency components after being attenuated along the signal path, making the capacitive and inductive cross-talk at the receiver smaller than that at the transmitter.

In the implemented interface, most of the inter-signal cross-talk happens at the bond wires and perhaps inside the package. The coaxial cables are well-shielded. The transmission lines formed by the board traces are far apart from each other above the board Vdd plane (AC Gnd) which is assumed to be stable. Adjacent bonding pads inside the package are routed to non-adjacent pins that are located next to supply pins[11], reducing the cross-talk between the adjacent signals . Since no detailed information about the internal design of the package is available to us, we do not have sufficient information to calculate the coupling coefficients inside the package. Nevertheless, based on TDR measurements of the channels, we have good reasons to believe that inter-signal cross-talk happens mostly between the bond wires and we model the system to be such.

Figure 5.22 illustrates the three-dimensional model used for field solver simulations. The bond wires are modelled using cy linders and their curvatures are ignored. The computed inductance and capacitance matrixes are

[10] Detail descriptions can be found in [83] or any books on transmission lines such as [84].

[11] This arrangement is not done deliberately. The high-speed package internal wirings are designed this way: the high-speed adjacent signal pads are routed to non-adjacent pins. This seems to be a common practice in high-speed packages with multiple tiers of pads. Our goal has been to study cross-talk from all packaging components, but here most of the observed cross-talk is from the bond wires.

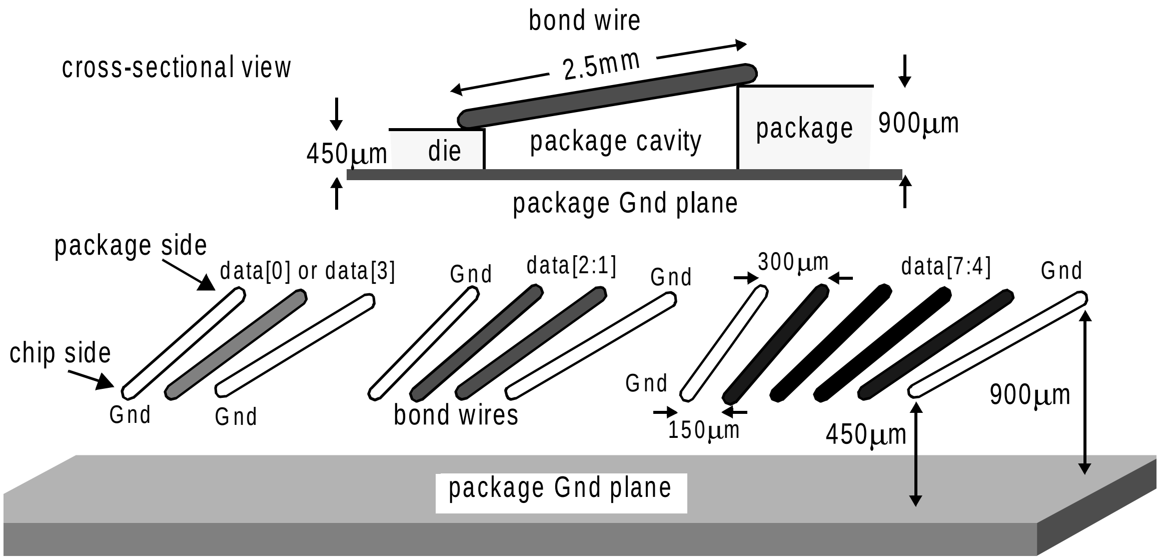


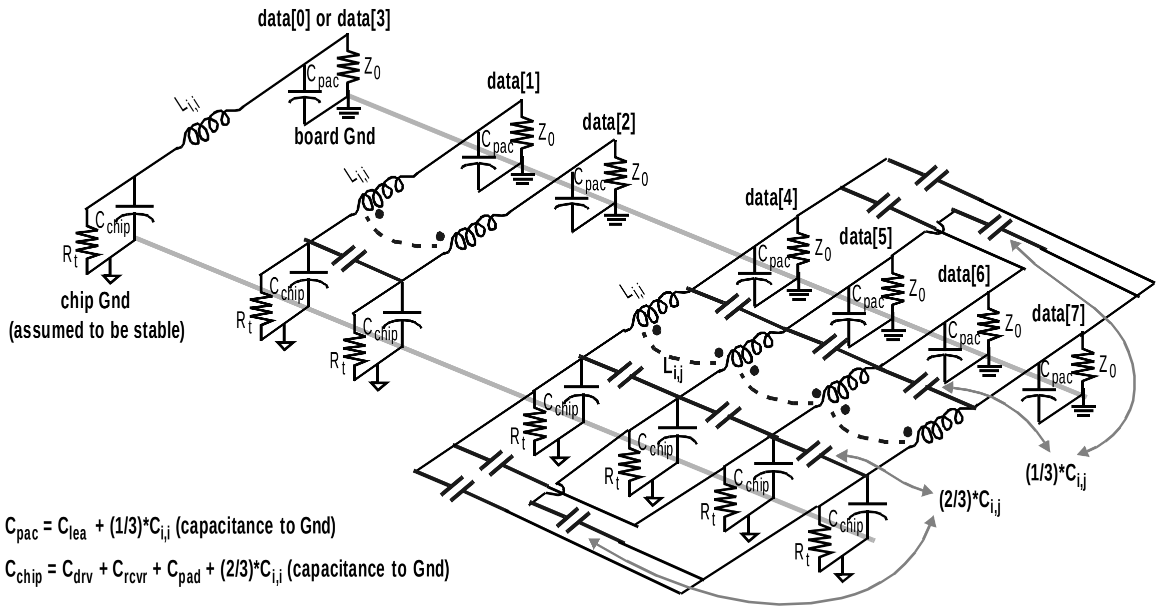
Figure 5.22: Three-dimensional field solver model of the signal and supply bond wires. All Vdd and Gnd bond wires are marked Gnd (AC Gnd).

$$\left[L_{i+1,j+1}\right] = \begin{bmatrix} 1.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.69 & 0.58 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.58 & 1.69 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.25 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.70 & 0.70 & 0.44 & 0.32 \\ 0 & 0 & 0 & 0 & 0.70 & 2.77 & 0.73 & 0.44 \\ 0 & 0 & 0 & 0 & 0.44 & 0.73 & 2.77 & 0.70 \\ 0 & 0 & 0 & 0 & 0.32 & 0.44 & 0.70 & 2.70 \end{bmatrix} \text{nH}, \qquad (5\text{-}3)$$

where i and j correspond to the signal numbers[12], and

$$\left[C_{i+1,j+1}\right] = \begin{bmatrix} 35.3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 35.3 & 8.2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 8.2 & 35.3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 35.3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 35.7 & 8.1 & 2.4 & 1.1 \\ 0 & 0 & 0 & 0 & 8.1 & 35.8 & 8.1 & 2.4 \\ 0 & 0 & 0 & 0 & 2.4 & 8.1 & 35.8 & 8.2 \\ 0 & 0 & 0 & 0 & 1.1 & 2.4 & 8.2 & 35.7 \end{bmatrix} \text{fF}, \qquad (5\text{-}4)$$

[12] The notation is confusing because the matrix row and column indices start with 1, but the data signal convention we have been using, which we also adopt here, starts with data[0].



$C_{pac} = C_{lea} + (1/3)*C_{i,i}$ (capacitance to Gnd)

$C_{chip} = C_{drv} + C_{rcvr} + C_{pad} + (2/3)*C_{i,i}$ (capacitance to Gnd)

Figure 5.23: Model to analyze inter-signal cross-talk due to bond wires.

where $C_{i,i}$ is the capacitance to Gnd. The computed self-inductance values are very close to the values obtained by area estimation of TDR results (if we assume most of the inductance in the signal path is contributed by the bond wire).

The noise model in Figure 5.23 is used to study bond wire level cross-talk. As shown in Figure 5.22, the distance between the bond wires and their distance from the Gnd plane on the package side are both double of the corresponding distances on the chip side. Since capacitance is inversely proportional to distance, all mutual capacitances and capacitances to Gnd are split into two in the model -- 2/3 on the chip side and 1/3 on the package side. To single out the effects of capacitive and inductive cross-talk, all on-chip Gnd nodes are assumed to be stable to eliminate the other noise sources such as power supply noise and cross-talk via shared current returns.

We simulate the cross-talk on a quiet signal when another signal switches, or when a combination of the other signals switch. Each active signal in question transmits the familiar current pulse (18mA, 600MHz with 300ps transition times , controlled by the output voltage of an actual output driver) to study near-end cross-talk, and receives the same current pulse to study far-end cross-talk. Using this model, there is no cross-talk to and from the isolated pins (i.e. data[0] and data[3]), and neither is there any coupling on the Vref nodes. Therefore, any induced noise on each data signal translates directly into a differential noise in the receiver inputs.

Results from a representative set of simulations show that the peak-to-peak value of the cross-talk from an immediate neighbor (i.e. data[2] to data[1], data[4] to data[5], and data[6] to data[5]) is about the same in each case, even though the waveform is slightly different in shape (and hence in frequency composition). The peak-to-peak value of the cross-talk from data[7] to data[5] is smaller, equal to only 2/3 of the above value. T he near-end cross-talk components are substantially larger than the ir far-end cross-talk counterparts, as we have predicted earlier. The worst-case cross-talk from data[2] to data[1] is 11% at the near-end and 5% at the far-end, while the worst-case cross-talk to data[5] when data[4], data[6], and data[7] transition in sync is 26% at the near-end and 14% at the far-end. These noise sources integrate to 2.6%, 1.1%, 8%, and 3.8% respectively using a sliding window clock at the same frequency. Therefore, the cross-talk due to bond wires can potentially decrease voltage margins by 5% in data[1] and 14% in data[5] in unidirectional signalling, and 16% in data[1] and 40% in data[5] in bidirectional signalling if sampling receivers are used. Ideal matched filter receivers can lessen the voltage margin losses to 1.1% and 3.8% in unidirectional data[1] and data[5], and 3.7% and 11.8% in bidirectional data[1] and data[5].

5.3.4.3 Measurements

For near-end cross-talk measurements, we cannot disable TxClk and the data itself to get rid of their coupling effects and isolate the inter-signal cross-talk to be measured directly. We first capture the internal waveforms of Vdata[0] and Vref[0] when data[0] alone transmits a clock stream and all the other data lines are quiet, and capture the waveforms again when all the other pins are toggling in sync. We repeat the same procedure for data[1] and data[5]. As the signal swings vary, the movements in the sampler outputs suggest that the near-end cross-talk from other signals via shared current returns is about 10.1% peak-to-peak in all three pins, while the near-end capacitive and inductive cross-talk at the bond wires adds an additional peak-to-peak differential noise of 5.3% in data[1] and 17.3% in data[5], but has little effect in data[0].

Far-end cross-talk can be measured directly by disabling all the on-chip clocks in the receiver chip. We follow a similar procedure in eliminating the background noise. As predicted, the far-end cross-talk components in each pin are smaller than their near-end counterparts, and the trends remain the same: data[0], data[1], and data[5] all experience approximately the same cross-talk (8.5% peak-to-peak) from other signals via shared current returns; the capacitive and inductive cross-talk between the bond wires adds an additional peak-to-peak differential noise of 4.7% in data[1] and 12.7% in data[5], but has little effect in data[0].

The peak of the cross-talk from the bond wires has a slight phase shift from the peak of the signal return cross-talk, whether at the transmitter (near-end) or receiver (far-end), making the superimposed total noise a smaller than the sum of their peak values. Moreover, near-end cross-talk is always in sync with signal transitions, while far-end cross-talk can be uncorrelated depending on the inter-signal timing skews at the receiver.

As mentioned earlier, the two sets of unidirectional refClk lines share the same controls as the data signals and cannot be turned on or off independently. In the noise models, their effects are lumped into the inter-signal cross-talk via shared signal return when all data signals are active. In all the measurements, they induce some ‘background noise’ even when data[0] alone is active while the other data signals are idle. Since these drivers share the same swing control as the signal output drivers, their coupling effect is also a proportional noise. Measurements using the samplers suggest that the toggling activities of these refClk lines induce an additional 2.1% peak-to-peak differential noise on (Vdata[0]-Vref[0]), due to far-end cross-talk through the power suppl y. Similar differential noise values are obtained on the other signal pins. This cross-talk component also appears in the near-end and should therefore be taken into account in bidirectional links.

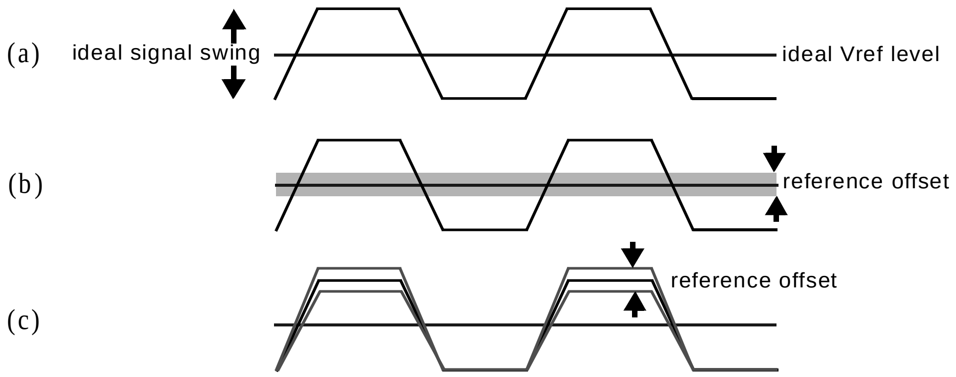


Figure 5.24: Reference offset is caused by a mismatch between the center of the signal swing and the Vref level. The figure illustrates 3 different cases: (a) ideal signal swing and Vref, (b) Vref deviates from nominal value, and (c) signal swing deviates from nominal value, assuming signal is referenced to the lower supply.

5.3.5 Reference Offset

The DC component of the reference noise, called reference offset, is illustrated in Figure 5.24. It is caused by mismatches between the reference value and the center of the signal swing. Process variations cause either the signal swing or the reference voltage level, or both, to deviate from their nominal values, and consequently Vref may not be at its optimal middle-of-swing value. Reference offset is a fixed voltage noise reducing the voltage margin of a link.

This noise source, however, does not affect our voltage margin measurement results. In the unidirectional link measurements, Vref is adjusted to find the voltage margin. In the bidirectional link measurements, VrefL is fixed while VrefH is varied, and the voltage margin of the upper eye is measured. Therefore, in either case, the receiver offset in either of the two current-integrating receivers in each pin does not affect the measured voltage margin. However, the *difference between these two receiver offsets* introduces a fixed noise.

An estimation using Pelgrom's equations [34] gives a +/-3σ value of +/-35mV for each receiver offset (which is modelled as a random variable), where σ is the standard deviation. The difference between two receiver offsets is then a random variable with a 3σ value of 49.5mV. With the ability to view the instantaneous digital outputs of the two current-integrating receivers in one of the eight channels, we measure this quantity by adjusting Vref and noting down its values when the digital outputs toggle. The difference between these two values corresponds to the difference in receiver offsets, which is measured to be bounded below 20mV for all channels.

5.3.6 Coupling between Transmit and Receive Signals on the Same Wire

Regardless of implementation details, all simultaneous bidirectional links suffer from one class of noise sources induced by the coupling between the transmit signal and the receive signal *on the same wire*. The transmit signal can couple to the receive signal on the same wire through an amplitude mismatch or a timing mismatch between the transmit signal and reference signal paths, through reflections of the transmit signal, or through the signal return impedance. On the other hand, the receive signal can couple to the transmit signal via the return impedance. These extra noise sources are sometimes termed reverse-channel cross-talk [85], and their exact magnitudes are strongly implementation-dependent.

For instance, the self-induced power supply noise that we investigate d earlier in Section5.3.3 is a form of coupling between the transmit and receive signals on the same wire via signal return impedance: the transmit signal induces power supply noise which couples to the receive signal, while the receive signal induces power supply noise which couples to the transmit signal, which then carries the noise with it to the other chip. In either case, the power supply noise is induced by the signal return current flowing through non-zero return impedance, and hence the transmit and receive signals couple to each other via the signal return impedance.

5.3.6.1 Amplitude Mismatch and Timing Mismatch

Clearly, mismatches between the two reference levels reduce signal margins, but a difference in the timing of the transmitter output and that of the reference also reduces signal margins or can even cause a glitch in the receiver differential input (Vdata-Vref). Figure 5.25 illustrates the amplitude mismatch and delay mismatch problems, which are both caused by fabrication process variations. The amplitude mismatch results in an inexact cancellation in the receiver decoding. A mismatch in the timing of Vdata and Vref, either in the delays of these two paths or in their transition times, can cause glitches

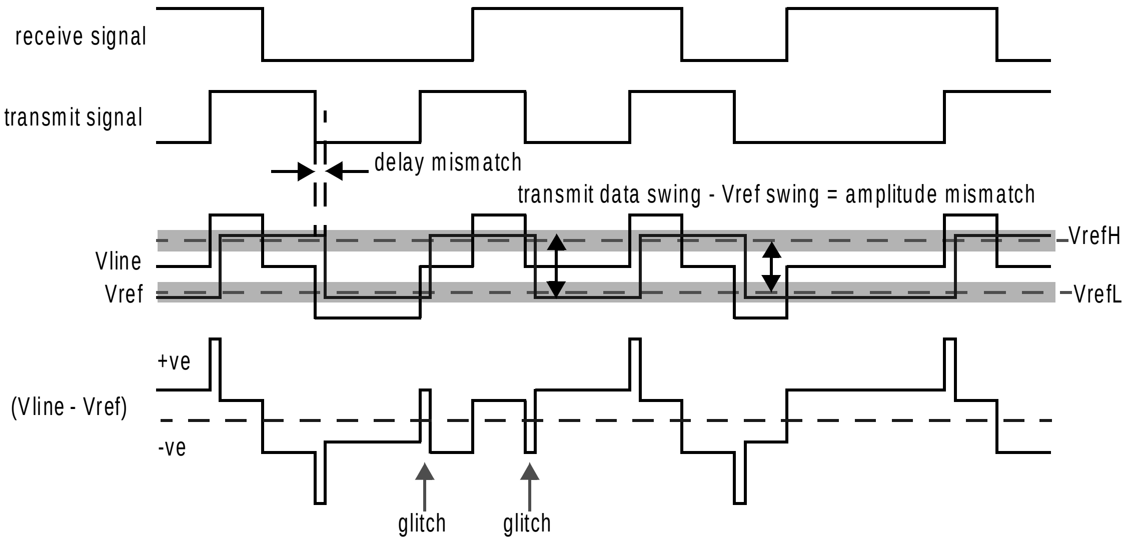


Figure 5.25: Mismatches in timing and in swing between the transmit signal and local Vref result in extra noise sources in simultaneous bidirectional links.

at the receiver differential inputs. (In a differential, simultaneous bidirectional link, the error comes from any mismatch in the two pairs of differential signals.) These glitches are most critical when the transmit and receive signals on the line are in *quadrature* phase. The receiver clock is positioned to be at the center of its data eye, which coincides with the glitches. Reception errors may result.

However, as long as VrefL is set at a level that does not cause any error (and it does not have to be at the middle of the lower eye), an amplitude mismatch between the swings of the local Vref and Vdata does not contribute to a fixed noise in our measured voltage margins as such a mismatch would normally do to simultaneous bidirectional links. On the other hand, an amplitude mismatch can increase the timing mismatch between the two signals.

In general, a timing mismatch between Vdata and Vref may convert to a proportional noise, a fixed noise, or a combination of the two depending on the specific link implementation. As mentioned earlier, the process run that our test chip was fabricated in turned out to be very slow. Fortunately, the multiple-segment structure in the Vref-select muxes allows tuning the timing of Vref to match Vdata. All legs in the muxes are activated to give the lowest R_{mux} value to produce a Vref timing well matched to Vdata, and the setting is fixed throughout all the measurements. If R_{mux} and the capacitance loading are both linear and constant, the RC time constant of the Vref transitions is fixed as signal swing varies. On the signal side, the RC time constant is determined, to the first order, by the parallel combination of the termination resistor and the transmission line characteristic impedance, and the capacitive loading presented to the output driver, mostly of the I/O pin and the pad but also of the output driver itself. Ignoring all the resistance and capacitive value changes as the source and drain voltages of the transistors vary, both RC time constants are unchanged as the signal and Vref swings change together. Therefore, we expect any timing mismatch present in the test chip to cause a proportional noise in the voltage margin measurements.

5.3.6.2 Reflections

As explained earlier in Chapter 2, reflections directly reduce voltage margins in simultaneous bidirectional links. A single reflection of the transmit signal due to impedance discontinuities in the signal path and termination mismatches will appear as noise to the incoming signal. Reflection noise is less of an issue for double-terminated unidirectional lines since only even reflections reach the receiver. Figure5.26 compares the reflection noise in double-terminated unidirectional and simultaneous bidirectional links. For simplicity, we assume that reflections happen only at the termination resistors and lumped capacitors contributed by the package leads, and that reflections higher than the second order are negligible. In reality, reflections happen at any impedance discontinuity. For unidirectional links, only the second reflections affect the receiver voltage margins. For simultaneous bidirectional links, all the first reflections of the transmit signal add to the receive signal in subsequent bit times and hence cause inter-symbol interference.

In the implemented signalling interface, since the propagation delay across any impedance discontinuity in the channel is short compared to the signal transition times, we believe that any reflection at the impedance discontinuities has small effect, and such high-frequency noise is further reduced by the use of current-integrating receivers.

5.3.6.2 Reflections

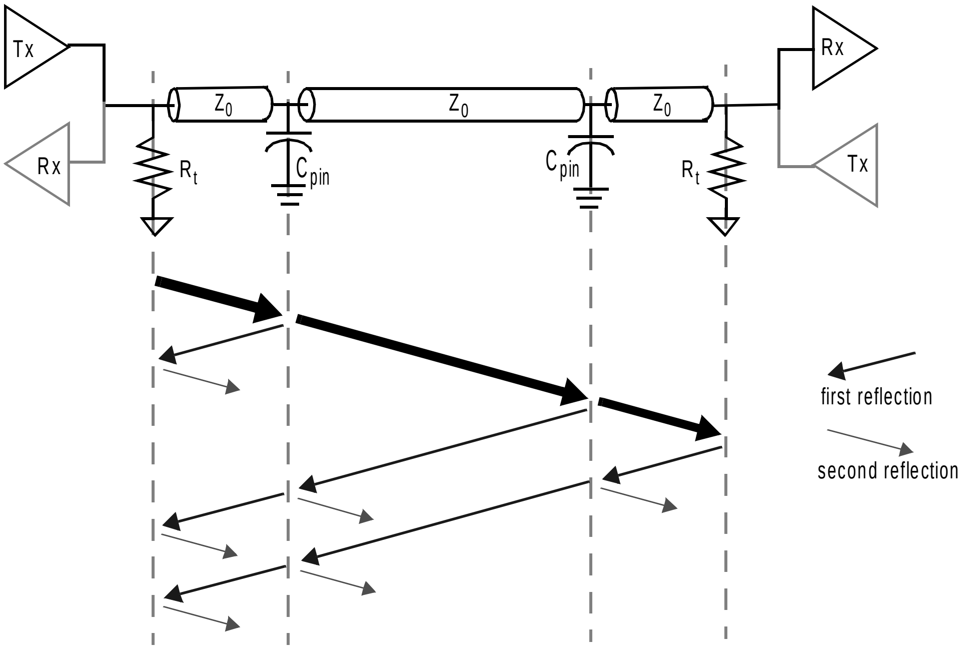


Figure 5.26: Reflection noise in double-terminated unidirectional and simultaneous bidirectional links.

As mentioned earlier in Chapter 3, even when we bias its gate at the lowest voltage we feel comfortable with (without excessive gate stress and breakdown), the measured impedance of the non-linear PMOS termination is still skewed towards the high side because of the slower than expected process: its resistance increases from 48 Ω at V_{ds}=0, to 57Ω at V_{ds}=450mV (unidirectional swing), to 70Ω at V_{ds}=900mV (twice of the unidirectional swing), to 78Ω at V_{ds}=1.1V. Using bounce diagrams that account for up to the second reflections due to termination mismatches on the two ends of the links and the instantaneous V_{ds} and resistance in all possible transmit signal and receive signal combinations, we expect the bidirectional eye diagram to assume voltage levels as shown in Figure 5.27. The non-linear resistance only ‘spreads’ out the lower eye and the middle and bottom voltage levels because the termination mismatch becomes larger at these voltages, but it does not significantly affect the upper eye opening. Based on this eye diagram, the termination mismatch can theoretically reduce the measured voltage margin by 6.4% since we measure the upper eye height.

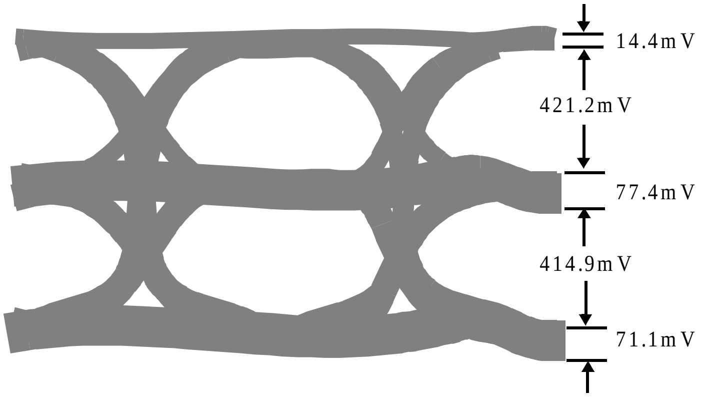


Figure 5.27: Bidirectional eye diagram predicted using bounce diagrams.

5.3.6.3 Measurements

Timing Mismatch

The switchings of the reference and of the transmitter output are well matched, and the induced glitch is small. The effect of this glitch is further reduced by the current-integrating receivers. In fact, varying the phase relationship between the transmit and receive signals across the bit time changes the voltage margins of the bidirectional links by only 20mV, which represents a proportional voltage noise of about 7%, and has no appreciable effect on timing margins. This observation is the combined effect of mismatched timing and any reflection of the transmit signal.

Reflections

To isolate the effects of reflections, we change the length of the cables used and add additional connectors in the signal paths to vary the position in a bit time at which any reflection hits the receive . However, no significant changes in voltage margins are observed in either unidirectional or bidirectional links. This confirms the projection that reflection noise is insignificant in unidirectional links, and that the first reflections resulted from the other chip’s termination mismatch and impedance discontinuities at the *far-end* are not significant in bidirectional links.

However, there is no easy way to determine the effect of the first reflections of the transmit signal by the impedance discontinuities at the *near-end* (i.e. due to the packaging components for the transmitter chip). The reason is that any observed change in voltage margin could very well be the effect of a timing mismatch in Vdata and Vref. We believe most of the 7% proportional noise seen above is due to a timing mismatch. The round-trip delay from the transmitter output to the discontinuities is actually shorter than the signal transition times, making the effect of reflections small. Th e effect is further reduced by current integration at the receivers.

5.3.7 Noise Induced by Switching of Reference Voltage

Switching activities in the Vref-select muxes move VrefH and VrefL, which couple even more noise onto the local Vref nodes. Therefore, this Vref switching noise has a self-induced component and an inter-signal cross-talk component similar to what we have seen earlier for the power supply noise induced by switching activities of Vdata; the difference is that the peak currents being switched by the Vref-select muxes are about 20 times smaller than the currents being switched by the output drivers. A model similar to the one in Figure 5.18, but replaces the resistors that model the muxes with the actual PMOS transistors, is used so that these transistors' dynamic resistances and capacitances , and turn-on and turn-off behavior are modelled more accurately[13].

5.3.7.1 Self-Induced Vref Switching Noise

The noise induced on the local Vref nodes is mainly caused by the noise induced on VrefH or VrefL, which is found to be almost the same magnitude in simulations regardless of the position of the active Vref. This is analogous to the self-induced power supply noise on the local supply node, w hich is similar in magnitude regardless of the position of the active Vdata.

At 450mV swing, the toggling activity of Vref[0], at the same frequency (600MHz) and slew rate (300ps transition times) as the signal , generates a 1.3% peak-to-peak differential noise on (Vdata[0]-Vref[0]) which integrates to 0.3% over the entire bit time using a receiver clock with sliding integration window. The simulations also show a fixed noise of 6.2mV (0.4mV after integration) due to unbalanced noise couplings from the two

[13] In order to turn on the PMOS switches, the DC biases of VrefH and VrefL have to be set correctl , and so the model is no longer a pure AC model . Also, $C_{d(mux)}$ should be set to 0. A lternatively, we can still use resistors -- controlled resistors whose values ramp from 390Ω to infinity while turning off and vice vera while turning on, but that approach generates less accurate results.

digital control signals that drive the gates of the muxes. Such large noise figures can be attributed to high-frequency ringings on VrefH and VrefL; in reality, ringings would be damped by wire resistances in the VrefH and VrefL lines, which are absent in the noise model. The observed peak-to-peak noise values decrease by more than 30% when a 5Ω series resistor is inserted in each VrefH/VrefL line. Some non-linearity is introduced by the non-linear mux resistance and source and drain capacitances, but the effects cannot be isolated. When the switching happens in Vref[1] or Vref[5] instead, similar noise figures are obtained.

5.3.7.2 Inter-signal Cross-Talk due to Local Vref Switching in Other Pins

The worst-case cross-talk on a quiet local Vref node occurs when the local Vref nodes in all the other pins transition high or low simultaneously. This cross-talk is almost the same magnitude whether the quiet line is Vref[0], Vref[1], or Vref[5], because the noise induced on VrefH and VrefL is about the same in each case. The simulations suggest thatthe cross-talk amounts to about 10% of proportional noise and 46mV of fixed noise, which integrate to 1.8% and 4mV. The large fixed noise is due to couplings from the gate controls of the muxes which induce oscillations on VrefH and VrefL. Both of the noise values reduce by more than 30% when each VrefH/VrefL line is damped using a 5Ω series resistor.

5.3.7.3 Measurements

Since the test chip allows activating or deactivating the output drivers and Vref-select muxes independently, we are able to study the effects associated with the switching activities on the reference lines. To measure the self-induced Vref switching noise, we capture the waveforms of Vdata[0] when it is idle, with and without Vref[0] switching, while all the other signals and their corresponding Vref lines are also idle. The changes in the captured waveforms cannot be distinguished from the fuzziness in the displays. To measure the cross-talk from other Vref lines switching, we capture the waveforms of Vdata[0] and Vref[0] when all signal and Vref lines are idle, and then turn on Vref[7:1]. Again the changes in the captured waveforms cannot be distinguished from the fuzziness in the displays. These observations may be accounted by the fact that the self-induced and cross-talk components of Vref switching noise consist m ostly of high-frequency components which are filtered out by the bandwidth-limited voltage samplers.

Unfortunately, the effects of Vref switching cannot be isolated and quantified by voltage margin measurements of bidirectional links either. In order to measure voltage margins, Vdata has to switch as well, and so any noise measured can very well be the effects of switching Vdata.

5.3.8 Gaussian Noise

One last interesting point -- the Gaussian noise fall-off is very sharp. In the voltage margin measurements, a small change (1 or 2mV) in reference voltage at the edges of the eye causes burst errors instantaneously. This observation confirms our fundamental assumption that the Gaussian noise sources are extremely small in magnitude in the type of electrical links we focus on.

5.4 Summary

A summary of all the voltage noise sources we have modelled analytically and measured experimentally is given in Table5-2: the peak-to-peak values obtained from the noise model analysis should be compared against the measured peak-to-peak values in the individual voltage noise component measurements, while the noise figures after integration from the noise model analysis should be compared against the voltage margin measurements presented earlier in Table5-1. While the current-integrating receivers we have implemented are not ideal matched filter receivers, they should yield results that are closer to those generated using ideal matched filter receivers than using sampling receivers. E ven though we are only able to extract the total fixed noise and the total proportional noise, and not the individual components, from the voltage margin measurement results, the differences in the fixed and proportional noise figures of different signal pins operating under different conditions allow us to break down the noise numbers as shown in Table 5-2.

Channel attenuation and inter-symbol interference combine to form the largest proportional noise source in the system. These noise sources are not characteristics of single-ended or simultaneous bidirectional signalling; they are also present in differential, unidirectional systems. The largest fixed noise comes from on-chip clock coupling,

Table 5-2: Identified voltage noise sources in implemented system, from noise model and measurements.

	Noise Measurements				Noise Model								Voltage Margin Measurements			
signal & link operation conditions[a]	D[0] others quiet	D[0] all PRBS	D[1] all PRBS	D[5] all PRBS	D[0] others quiet	D[0] all PRBS	D[1] all PRBS	D[5] all PRBS	D[0] others quiet	D[0] all PRBS	D[1] all PRBS	D[5] all PRBS	D[0] others quiet	D[0] all PRBS	D[1] all PRBS	D[5] all PRBS
	peak-to-peak				peak-to-peak (sampling Rx)				integrated over T_{bit} (ideal matched filter Rx)				voltage margin reduction measured using current-integrating Rx			
coupling from on-chip clocks TxClk RxClk cleanClk sampleClk	20mV 22.1mV 26.4mV < 2mV	similar to values found for data[0]			based on measurements								70mV		64mV	69mV
coupling from Rx chip power supply noise due to switching activities of non-I/O circuitry	cannot be isolated in the noise measurements				the effect is a fixed noise											
difference between Rx offsets	< 20mV				statistically 3σ is 49.5mV											
channel attenuation & ISI	23%				based on measurements (23%)								33%			
reference noise by self-induced power supply noise induced by receive signal	1.6%				1.9%				0.4%							
FEXT from active refClk lines	2.1%				effect lumped into far-end cross-talk via shared current return											
second reflections	cannot be isolated				negligible											
FEXT via shared current returns at Rx	0	8.5%			0	11%			0	1.3%			0	1%		
FEXT from signals in the same cluster between bond wires	0	0	4.7%	12.7%	0	0	5%	14%	0	0	1.1%	3.8%	0	0	3%	17%
SUBTOTAL (total noise in unidirectional links)																
fixed	< 91mV				< 49.5mV				< 49.5mV				70mV		64mV	69mV
proportional	27%	35%	40%	48%	25%	36%	41%	50%	23%	25%	26%	29%	33%	34%	37%	51%

Table 5-2: Identified voltage noise sources in implemented system, from noise model and measurements.

	Noise Measurements				**Noise Model**								**Voltage Margin Measurements**			
signal & link operation conditions[a]	D[0] others quiet	D[0] all PRBS	D[1] all PRBS	D[5] all PRBS	D[0] others quiet	D[0] all PRBS	D[1] all PRBS	D[5] all PRBS	D[0] others quiet	D[0] all PRBS	D[1] all PRBS	D[5] all PRBS	D[0] others quiet	D[0] all PRBS	D[1] all PRBS	D[5] all PRBS
	peak-to-peak				peak-to-peak (sampling Rx)				integrated over T_{bit} (ideal matched filter Rx)				voltage margin reduction measured using current-integrating Rx			
unknown fixed noise	of unknown sources but most likely due to small errors in modeling and measurements, and in curve-fitting measurement data												-13mV		-10mV	-5mV
fixed noise induced by self-induced Vref switching	cannot be measured accurately				6.2mV				0.4mV				0			
fixed noise induced by cross-talk due to others' Vref switching	0	cannot be measured accurately			0	46mV			0	4mV			0	-4mV		
reference noise by self-induced power supply noise induced by transmit signal	2.1%				3.9%				0.5%				9%	8%	5%	-9%
single-ended coupling from Tx power supply to Vdata	small, cannot be isolated				0.7%				negligible							
NEXT from active refClk lines	2.1%				effect lumped into near-end cross-talk via shared current return											
Vdata and Vref timing mismatch	7%				proportional noise of unknown values											
reflections	cannot be isolated				6.4% due to first and second reflections caused by termination mismatches				too complex to model							
prop. noise induced by self-induced Vref switching	cannot be measured accurately				1.3%				0.3%							
prop. noise induced by cross-talk due to others' Vref switching	0	cannot be measured accurately			0	10%			0	1.8%			0	3%		
NEXT via shared current returns at Tx	0	10.1%			0	18%			0	3%			0			
NEXT from signals in the same cluster between bond wires	0	0	5.3%	17.3%	0	0	11%	26%	0	0	2.6%	8%	0	0	2%	12%
TOTAL (total noise in bidirectional links)																
fixed	**< 91mV**				**< 56mV**	**< 102mV**			**< 50mV**	**< 54mV**			**57mV**	**53mV**	**50mV**	**60mV**
proportional	**38%**	**57%**	**67%**	**87%**	**37%**	**76%**	**92%**	**116%**	**24%**	**30%**	**34%**	**42%**	**42%**	**45%**	**47%**	**57%**

[a] Because of table margin issues, data[0], data[1], and data[5] are abbreviated to be D[0], D[1], and D[5] here.

possibly through the power supplies, even though the exact cause cannot be confirmed. Differential signalling, where the signal and its complement are coupled equally to the supplies, may alleviate the problem if the clock couplings are indeed through the power supplies.

Comparing the peak-to-peak values obtained from the noise model against their counterparts from the noise component measurements, we notice that these two sets of noise figures agree quite well. The measured values are smaller in general, and the discrepancies are larger for noise sources on the transmitter side compared to those on the receiver side: the receiver side noise figures match within 20% and the differences can be accounted for by measurement errors; t he measured values for transmitter side noise sources are only about 60% of the simulated values , suggesting that our simple noise model needs refinement.

In a single-ended parallel link, the power supply induces differential noise coupling onto the data signal and reference voltage in each pin. It also acts as a shared current return path for different I/O signals, which creates inter-signal cross-talk via the shared signal return (power supply), and increases the on-chip power supply noise which in turn increases the differential noise coupling onto the data signal and the reference. These effects are intensified when the single-ended link is simultaneous bidirectional, because the transmitter end noise sources have to be considered in addition to the receiver end noise sources. In simultaneous bidirectional signalling, the receiver voltage margin is also reduced by timing and amplitude mismatches between the signal and the reference, and by first reflections of the transmit signal. However, these effects are found to be relatively small, from both simulations and measurements, when compared to the direct inductive and capacitive couplings from bond wires in the same signal cluster when the signals on both of the adjacent bond wires transition in sync. Cross-talk between bond wires, especially at the far-end, can also be a problem in differential, unidirectional links.

It is interesting to note that even though the corresponding entries for far-end cross-talk via shared current returns match very well in the noise model and measurement results, so are those for near-end cross-talk via shared current returns, the results are quite counter-intuitive: intuitively, we had expected a larger cross-talk via shared current returns in data[5] compared to in data[1], which should in turn be larger than in data[0]. While this relation is indeed true in our noise model simulations, the differences are very small (less than 1%) as found earlier in Section5.3.4.1. The reason is that the signal clusters are close to one another and the return current for each I/O signal does not flow entirely through the closest set of Vdd/Gnd pins (i.e. the local supply). Moreover, since the test chip is small, the non-I/O Vdd/Gnd pins also help to divert some return current from each I/O signal. Therefore, the cross-talk is about the same in each pin -- even in the isolated pins. The cross-talk would be much worse in the data[7:4] cluster had the power supplies of the different signal clusters been disconnected from one anothe .

Therefore, single-ended signalling and simultaneous bidirectional signalling are viable pin-saving alternatives to the traditional differential and unidirectional signalling. The additional on-chip noise sources introduced are manageable by careful circuit design and are small compared to the off-chip noise sources, which need to be addressed by improvements in packaging technologies.

Chapter 6

Conclusion

In this dissertation, we examine the three fundamental challenges in increasing the overall system performance of high-speed parallel links: overcoming the voltage noise, recovering timing at the receiver, and keeping the cost per I/O low, and study how the conventional unidirectional, differential source-synchronous parallel link architecture can be modified to increase the bandwidth per pin while keeping the cost per link modest.

Voltage and timing error sources limit the performance of a link and affect its robustness. The voltage and timing noise sources unique in parallel links, such as inter-signal timing skew and inter-signal cross-talk, impose greater challenges as the performance increases. While many possible solutions exist to reduce the magnitudes of these noise sources or minimize their impacts on the system performance, almost every scheme requires adding extra hardware or increasing the per I/O cost,therefore conflicting with the low cost per I/O requirement which is essential for mass integration of parallel I/Os. Hence, every design choice involves *trade-offs* among various performance and cost parameters. As performance requirements increase, future parallel links will employ more traditional serial link techniques, often in their simplified forms, to extend the achievable data rate at the cost of added hardware and complexity.

As the bit time continues to shrink and the channel continues to grow longer, the loss in timing margins in parallel links due to inter-signal timing skew is becoming a larger percentage of the bit time and will eventually present a performance bottleneck. With carefully matched signal paths, we measure a maximum skew of 100ps . Using cheaper electrical components to reduce the cost per I/O would further increase the delay mismatches. Therefore, it is clear that some type of skew compensation scheme will soon be needed in any high-performance parallel interface. We propose a per pin skew compensation architecture that uses phase interpolation to enable full-range compensation, and have demonstrated that it recenters and improves receiver timing margins of the links, and is able to handle larger skews that may otherwise cause link failure.

The cost overhead in implementing skew compensation depends largely on the range and accuracy of the compensation desired. Setting an upper bound on the cost overhead, our full-range compensation architecture allows a compensation range up to two bit times with a minimal jitter local RxClk. The low-jitter differential clock buffers increase the static power consumption of the system by 55% but bring about less than 7% of timing margin improvement. Hence replacing them with full-rail CMOS inverters is a good design trade-off, and it also allows the architecture to scale to wider parallel links . Furthermore, the cost overhead can be significantly reduced when the required skew compensation range is smaller, and we propose the use of a local adjustable delay line in such cases and examine two different possible designs.

The phase noise in high-speed interface signals (especially in a DLL-based system) carries significant high-frequency components, and experimental results have shown that the receiver clock generation delay makes tracking the jitter of a source-synchronous refClk difficult. Using a 'clean' (stable) clock for receiver timing recovery clock generation is the best strategy for jitte . In general, good transmitter and receiver clock generation schemes, which lock delay paths that closely track those of the output clocks, result in little low-frequency phase drifts. In a system capable of skew compensation, if the phase drifts are significant or affect the performance, they can be compensated by periodic calibration, making the source-synchronous refClk signal unnecessar . Periodic calibration reduces the effective bandwidth of the links, and hence should not be done in systems where the low-frequency phase drifts are small compared to the receiver timing margins, as in the case of our implemented system.

Voltage noise sets the minimum signal swing, and hence power, required for robust link operation, and may cause bit errors. In high-speed parallel links, the major voltage noises are channel attenuation and inter-symbol interference, fabrication offsets, reflections, inter-signal cross-talk, and power supply noise. Their effects would intensify in the future: fabrication offsets increase as fabrication technology scales and transistor sizes decrease, while the other four noise sources become more problematic as signalling frequency increases. In Chapter 2, we have detailed the various approaches link designers have taken to combat these voltage noise sources. Our measurements show that the largest proportional noise comes from channel attenuation and inter-symbol interference, which lead to 23% loss in the link voltage margins even in our relatively short and high-quality communication channel.

The remaining noise sources clearly show that modern high-speed parallel link design is not just about designing circuits -- the design of the packaging and off-chip components, and even of the process, plays an increasingly important role as signalling frequency increases. First of all, most of our measured inter-signal skews are caused by mismatches in the package signal traces and cables; the delay mismatches in the transmitter and receiver circuitry are much smaller. Secondly, the accuracies of our analytic noise models are limited by the ability to accurately model the geometries and parasitics of the packaging entities. Thirdly, the direct inductive and capacitive couplings from bond wires in the same signal cluster when the signals on both of the adjacent bond wires transition in sync, both at the far-end and at the near-end, are found to be larger than the total of the other additional voltage noise sources introduced by single-ended signalling and simultaneous bidirectional signalling . Therefore, the ability to realize the pin-saving potential brought about by these two signalling schemes depends heavily on future research and advancements in packaging design and technologies.

References

[1] Point-contact transistor, invented by Walter Brattain and John Bardeen in 1947. One of the earliest commercially available point-contact transistors, built in germanium, is Raytheon CK703 in 1950.

[2] Junction (sandwich) transistor, invented by William Shockley in 1948. One of the earliest commercially available junction transistors, also built in germanium, is Raytheon CK722 in 1953.

[3] R. Heald et al., “A third-generation SPARC V9 64-b microprocessor,” *IEEE Journal of Solid-State Circuits*, vol. 35, no. 11, pp. 1526-1538, Nov. 2000.

[4] S. Rusu and G. Singer, “The first IA-64 microprocessor,” *IEEE Journal of Solid-State Circuits*, vol. 35, no. 11, pp. 1539-1544, Nov. 2000.

[5] E. Reese et al. “A phase-tolerant 3.8 GB/s data-communication router for a multiprocessor supercomputer backplane,” *1994 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, pp. 296-297, Feb. 1994.

[6] J. Kuskin et al., “The Stanford FLASH multiprocessor,” *Proceedings of the 21st International Symposium on Computer Architecture*, pp. 302-313, Apr. 1994.

[7] *Scalable Coherent Interface* (SCI), IEEE Standard 1596.

[8] M. Galles et al. “Spider: a high-speed network interconnect,” *IEEE Micro*, vol.17, no.1, pp. 34-39, Jan.-Feb. 1997.

[9] K. Gotoh et al., “A 2B parallel 1.25Gb/s interconnect I/O interface with self-configurable link and plesiochronous clocking,” *1999 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, pp. 180-181, Feb. 1999.

[10] A. Charlesworth et al., “Gigaplane-XB: Extending the Ultra Enterprise family,” *Hot Interconnects V Symposium Record*, pp. 97-112, Aug. 1997.

[11] A. Charlesworth, “Starfire: Extending the SMP envelope,” *IEEE Micro*, vol.18, no.1, pp. 39-49, Jan.-Feb. 1998.

[12] A. Mu. et al. “A 285 MHz 6-port plesiochronous router chip with non-blocking cross-bar switch,” *Proceedings of 1996 IEEE Symposium on VLSI Circuits*, pp. 136-137, June 1996.

[13] K.Y. Chang et al. “A 50 Gb/s 32×32 CMOS crossbar chip using asymmetric serial links,” *Proceedings of 1999 IEEE Symposium on VLSI Circuits*, pp. 19-22, June 1999.

[14] T. Takahashi et al., “A CMOS gate array with 600Mb/s simultaneous bidirectional I/O circuits,” *IEEE Journal of Solid-State Circuits*, vol. 30, no. 12, pp. 1544-1546, Dec. 1995.

[15] K. Lee et al., “A jitter-tolerant 4.5Gb/s CMOS interconnect for digital display,” *1998 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, pp. 310-311, Feb. 1998.

[16] J. Cioffi, *Digital Communication*, currently available from *http://www.stanford.edu/class/ee379a* and *http://www.stanford.edu/class/ee379b*.

[17] E.A. Lee and D.G. Messerschmit, *Digital Communication*, second edition, Kluwer Academic Pub., Aug. 1993.

[18] A.S. Tanenbaum, *Computer Networks*, third edition, Prentice Hall, 1996.

[19] A.X. Widmer et al., “Single-chip 4*500-MBd CMOS transceiver,” *IEEE Journal of Solid-State Circuits*, vol. 31, no. 12, pp. 2004-13, Dec. 1996.

[20] W.J. Dally and J. Poulton, “Transmitter equalization for 4-Gbps signaling,” *IEEE Micro*, Jan.-Feb. 1997, vol.17, no.1, p. 48-56.

[21] A. Fiedler, et al., “A 1.0625 Gbps transceiver with 2x-oversampling and transmit signal pre-emphasis,” *1997 IEEE International Solids-State Circuits Conference. Digest of Technical Papers* Feb. 1997, pp. 238-9.

[22] R. Farjad-Rad, et. al, “A 0.3-μm CMOS 8-Gbps 4-PAM serial link transceiver,” *IEEE Journal of Solid-State Circuits*, vol. 35, no. 5, pp. 757-64, May 2000.

[23] L.I. Andersson et al., “Silicon bipolar chipset for SONET/SDH 10-Gb/s fiber-optic communication links,” *IEEE Journal of Solid-State Circuits*, vol. 30, no. 3, pp. 210–8, Mar. 1995.

[24] Y.M. Greshishchev et al., “A fully integrated SiGe receiver IC for 10-Gb/s data rate,” *IEEE Journal of Solid-State Circuits*, vol. 35, no. 12, pp. 1949-57, Dec. 2000.

[25] M. Meghelli et al., “SiGe BiCMOS 3.3-V clock and data recovery circuits for 10-Gb/s serial transmission systems,” *IEEE Journal of Solid-State Circuits*, vol. 35, no. 12, pp. 1992-5, Dec. 2000.

[26] A. Momtaz et al., “Fully-integrated SONET OC48 transceiver in standard CMOS,” *2001 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, pp. 76-77, Feb. 2001.

[27] E. Yeung and M. Horowitz, “A 2.4Gb/s/pin simultaneous bidirectional parallel link with per pin skew compensation,” *2000 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, pp. 256-257, Feb. 2000.

[28] E. Yeung and M. Horowitz, “A 2.4Gb/s/pin simultaneous bidirectional parallel link with per pin skew compensation,” *IEEE Journal of Solid-State Circuits*, vol. 35, no. 11, pp. 1619-1628, Nov. 2000.

[29] M. Horowitz, C.K. Yang, S. Sidiropoulos, “High-speed electrical signalling: Overview and limitations,” *IEEE Micro*, vol.18, no.1, pp. 12-24, Jan.-Feb. 1998.

[30] C.K. Yang, *Design of High-Speed Serial Links in CMOS* , Ph.D. Dissertation, Stanford University, December 1998.

[31] J. Poulton et. al., “A Tracking Clock Recovery Receiver for 4Gb/s Signalling,” *Hot Interconnects V Symposium Record*, pp. 157-169, Aug. 1997.

[32] B.S. Song et al., “NRZ timing recovery technique for band-limited channels,” *1996 IEEE International Solid-State Circuits Conference Digest ofTechnical Papers,* pp. 194-195, Feb. 1996.

[33] J. Zerbe et. al., “A 2Gb/s/pin 4-PAM parallel bus interface with transmit crosstalk cancellation, equalization, and integrating receivers,” *2001 IEEE International Solid-State Circuits Conference Digest ofTechnical Papers,* pp. 66-67, Feb. 2001.

[34] M.J. Pelgrom, A. Duinmaijer, and A. Welbers, “Matching properties of MOS transistors,” *IEEE Journal of Solid-State Circuits*, Oct. 1989. vol.24, no.5, pp. 1433-1439.

[35] C.C. Enz and G.C. Temes, “Circuit techniques for reducing the effects of op-amp imperfections: autozeroing, correlated double sampling, and chopper stabilization,” *Proceedings of the IEEE*, vol. 84, no. 11, pp. 1584-1614, Nov. 1996.

[36] W. Ellersick et al., “GAD: A 12GS/s CMOS 4-bit A/D converter for an equalized multi-level link,” *Proceedings of 1999 IEEE Symposium on VLSI Circuits* , pp. 49-52, June 1999.

[37] T. Knight and A. Krymm, “A self-terminating low-voltage swing CMOS output driver,” *IEEE Journal of Solid-State Circuits*, vol. 23, no. 2, pp. 457-464, Apr. 1988.

[38] A. DeHon, T. Simon and T. Knight, “Automatic impedance control,” *1993 IEEE International Solid-State Circuits Conference Digest of Technical Papers* , pp 164-165, Feb. 1993.

[39] M. Haycock and R. Mooney, “A 2.5Gb/s bi-directional signaling technology,” *Hot Interconnects V Symposium Record*, pp. 149-156, Aug. 1997.

[40] T. Takahashi et al., “110GB/s simultaneous bi-directional transceiver logic synchronized with a system clock,” *1999 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, pp. 176-177, Feb. 1999.

[41] T. Sekiguchi et al, “Low-noise, high-speed data transmission using a ringing-cancelling output buffer,” *IEEE Journal of Solid-State Circuits*, vol. 30, no. 12, pp. 1569-74, Nov. 1995.

[42] B. Lau et al., “A 2.6-GByte/s multi-purpose chip-to-chip interface,”*IEEE Journal of Solid-State Circuits*, vol.33, no.11, pp. 617-626, Nov. 1998.

[43] G. Wei et al., “A Variable-Frequency Parallel I/O Interface with Adaptive Power Supply Regulation,” *IEEE Journal of Solid-State Circuits*, vol. 35, no. 11, pp. 1600-1610, Nov. 2000.

[44] *Gigabit Ethernet*, IEEE Standard 802.3z, 1998.

[45] K. Nakamura and M. Horowitz, “A 50% noise reduction interface using low-weight coding”, *Proceedings of 1994 IEEE Symposium on VLSI Circuits,* pp. 144-145, June 1996.

[46] D. Cecchi, M. Dina and C. Preuss, “A 1GB/s SCI link in 0.8-µm BiCMOS,” *1995 IEEE International Solid-State Circuits Conference Digest ofTechnical Papers* , pp. 326-327, Feb. 1995.

[47] SCIzzL Documents on IEEE Standards Project P1596.8 (Parallel Links for the Scalable Coherent Interface), 1996, available online from *ftp://ftp.SCIzzL.com/P1596.8/960926kbl.pdf*.

[48] S. Kim et al., “A 960Mbps/pin interface for skew-tolerant bus using low jitter PLL”, *Proceedings of 1996 IEEE Symposium on VLSI Circuits,* pp. 118-9, June 1996.

[49] HIPPI-6400 PH (High-Performance Parallel Interface - 6400 Mbit/s Physical Layer), 1998.

[50] T. Sato et al., “5GByte/s data transfer scheme with bit-to-bit skew control for synchronous DRAM”, *Proceedings of 1998 IEEE Symposium on VLSI Circuits*, pp. 64-65, June 1998.

[51] D.G. Messerschmitt, “Synchronization in Digital System Design,” *IEEE Journal on Selected Areas in Communications*, vol.8, no. 8, pp 1404-1419, Oct. 1990.

[52] F. Gardner, *PhaselockTechniques*, Second Edition, John Wiley, 1979.

[53] F. Gardner, “Charge-Pump Phase-Lock Loops,” *IEEE Transactions on Communications*, vol. COM-28, no. 11, pp. 1849-1858, Nov. 1980.

[54] R. Best, *Phase-Locked Loops: Theory, Design and Applications*, McGraw-Hill, 1984.

[55] M. Johnson and E. Hudson, "A Variable Delay Line PLL for CPU-Coprocessor Synchronization," *IEEE Journal of Solid-State Circuits*, vol. 23, no. 5, pp. 1218-1223, Oct. 1988.

[56] I.A. Young et al., "A PLL clock generator with 5 to 110 MHz lock range for microprocessors," *IEEE Journal of Solid-State Circuits* , vol. 27, no. 11, pp. 1599-1607, Nov. 1992.

[57] T.H. Lee and J. Bulzachelli, "A 155MHz clock recovery delay- and phase-locked loop," *IEEE Journal of Solid-State Circuits*, Dec. 1992, vol. 27, no. 12, pp. 1736, Dec. 1992.

[58] M. Horowitz et al., "PLL Design for a 500MB/s Interface," *1993 IEEE International Solid-State Circuits Conference Digest ofTechnical Papers*, pp. 160-161, Feb. 1993.

[59] T.H. Lee et al., "A 2.5 V CMOS delay-locked loop for 18 Mbit, 500 megabyte/s DRAM," *IEEE Journal of Solid-State Circuits*, vol. 29, no. 12, pp. 1491-1496, Dec. 1994.

[60] B. Razavi, Editor, *Monolithic Phase Locked Loops and Clock Recovery Circuits* , IEEE Press, 1996.

[61] J.G. Maneatis, "Low-jitter process-independent DLL and PLL based on self-biased techniques," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 11, pp. 1723-1732, Nov. 1996.

[62] J. Maneatis and M. Horowitz, "Precise delay generation using coupled oscillators," *IEEE Journal of Solid-State Circuits*, vol. 28, no. 12, pp. 1273-1282, Dec. 1993.

[63] J. Maneatis, *Precise Delay Generation Using Coupled Oscillators*, Ph.D. Dissertation, Stanford University, June 1994.

[64] S. Sidiropoulos and M. Horowitz, "A semi-digital DLL with unlimited phase shift capability and 0.08-400 MHz operating range," *1997 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, pp. 332-333, Feb. 1997.

[65] S. Sidiropoulos and M. Horowitz, "A semi-digital dual delay locked loop", *IEEE Journal of Solid-State Circuits*, vol. 32, no. 11, pp. 1683-1692, Nov. 1997.

[66] N. Kushiyama et al., “A 500-megabyte/s data-rate 4.5M DRAM,” *IEEE Journal of Solid-State Circuits*, vol.28, no.4, pp. 490-8, Apr. 1993.

[67] K. Donnelly et al., “A 660 MB/s interface megacell portable circuit in 0.3mm-0.7mm CMOS ASIC,” *1996 IEEE International Solid-State Circuits Conference Digest of Technical Papers,* pp. 290-291, Feb. 1996.

[68] S. Sidiropoulos, C.K. Yang and M. Horowitz, “A CMOS 500 Mbps/pin synchronous point to point link interface,” *Proceedings of 1994 IEEE Symposium on VLSI Circuits,* pp. 43-44, June 1996.

[69] Y. Morooka et al., “Source synchronization and timing vernier techniques for 1.2GB/s SLDRAM interface,” *1998 IEEE International Solid-State Circuits Conference Digest of Technical Papers,* pp. 160-1, Feb. 1998.

[70] K. Lee, S. Kim, G. Ahn and D.K. Jeong, “A CMOS serial link for fully duplexed data communication”, *IEEE Journal of Solid-State Circuits*, April 1995. vol.30, no.4, p. 353-64.

[71] C.K. Yang, R. Farjad-Rad, and M. Horowitz, “A 0.5-µm CMOS 4Gb/s transceiver with data recovery using oversampling,” *IEEE Journal of Solid-State Circuits*, vol.33, no.5 pp. 713-22, May 1998.

[72] S. Sidiropoulos and M. Horowitz, “Current integrating receivers for high speed system interconnects,” *Proceedings of 1995 IEEE Custom Integrated Circuits Conference,* pp. 107-110, May 1995.

[73] S. Sidiropoulos and M. Horowitz, “A 700 Mbps/pin CMOS signalling interface using current integrating receivers,” *Proceedings of 1996 IEEE Symposium on VLSI Circuits,* pp. 142-143, Jun. 1996.

[74] S. Sidiropoulos and M. Horowitz, “A 700-Mb/s/pin CMOS signaling interface using current integrating receivers,” *IEEE Journal of Solid-State Circuits*, vol. 32, no. 5, pp. 681-90, May 1997.

[75] K. Lam, L. Dennison and W. Dally, “Simultaneous bidirectional signalling for IC systems,” *Proceedings of 1990 IEEE International Conference on Computer Design*, pp. 430-433, Sept. 1990.

[76] L. Dennison, W. Lee and W. Dally, “High-performance bidirectional signaling in VLSI systems,” *Proceedings of Symposium on Integrated Systems, MIT Press* , pp. 300-319, March 1993.

[77] R. Mooney, C. Dike and S. Borkar, “A 900Mb/s bidirectional signaling scheme,” *IEEE Journal of Solid-State Circuits*, vol.30, no. 12, pp. 1538-1543, Dec. 1995.

[78] A. Lempel and W.L. Eastman, “High speed generation of maximal length sequences,” *IEEE Transaction on Computers* , vol. 32, no. 7, pp. 1115-1121, Feb. 1971.

[79] S. Sidiropoulos, *High-Performance Inter-Chip Signalling* , Ph.D. Dissertation, Stanford University, April 1998.

[80] National Instruments PCI-DIO-32HS. Product specifications available from *http://www.ni.com/catalog/pdf/1mhw402-403,414-415a.pdf.*

[81] P. Larsson and C. Svensson, “Measuring high-bandwidth signals in CMOS circuits,” *Electronic Letters*, vol. 29, no. 20, pp. 1761-2, September 1993.

[82] R. Ho et al., “Applications of on-chip samplers for test and measurement of integrated circuits,” *Proceedings of 1998 IEEE Symposium on VLSI Circuits*, pp. 138-9, Jun. 1998.

[83] H.B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*, Addison-Wesley, 1990.

[84] R. Matick, *Transmission lines for digital and communication networks*, IEEE Press, 1995.

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

[86] C.K. Yang and M. Horowitz. “A 0.8-μm CMOS 2.5Gb/s oversampling receiver and transmitter for serial links,” *IEEE Journal of Solid-State Circuits*, vol.31, no.12 pp. 2015-2023, Dec. 1996.

[87] H.O. Johansson, J. Yuan, C. Svensson, “A 4-GSamp/s line-receiver in 0.8 μm CMOS,” *1996 Symposium on VLSI Circuits. Digest of Technical Papers*, Jun. 1996, pp. 116-7