HIGH PERFORMANCE INTER-CHIP SIGNALLING

Stefanos Sidiropoulos

Technical Report No. CSL-TR-98-760

April 1998

This research has been supported by ARPA contract DABT63-94-C-0054.

HIGH PERFORMANCE INTER-CHIP SIGNALLING

Stefanos Sidiropoulos

Technical Report: CSL-TR-98-760

April 1998

Computer Systems Laboratory
Departments of Electrical Engineering and Computer Science
Stanford University
Stanford, California 94305-4055

Abstract

The achievable off-chip bandwidth of digital IC's is a crucial and often limiting factor in the performance of digital systems. In intra-system interfaces where both latency and bandwidth are important, source-synchronous parallel channels have been adopted as the most effective solution. This work investigates receiver and clocking circuit design techniques for increasing the signalling rate and robustness of such channels.

One of the main problems arising in the reception of high speed signals is the adverse effects of high frequency noise. To alleviate these effects, a new class of receiver structures that utilize current integration is proposed. The integration of current on a capacitor based on the incoming signal polarity effectively averages the signal over its valid time period, therefore filtering out high frequency noise. An experimental transceiver prototype utilizing current integrating receivers was designed and fabricated in a 0.8 μm CMOS technology. The prototype achieves a signaling rate of 740 Mbps/pin operating from a 3.3-V supply with a bit error rate of less than 10^{-14}.

The second major challenge of inter-chip communication is the design of clock generation and synchronization circuits. Delay locked loops are an attractive alternative to VCO-based phase locked loops due to their simpler design, intrinsic stability, and absence of phase error accumulation. One of their main problems however is their limited phase capture range. A dual loop architecture that eliminates this problem is proposed. This architecture employs a core loop to generate finely spaced clock edges, which are then used by a peripheral loop to generate the output clock through phase interpolation. Due to its digital control, the dual loop can offer great flexibility in the implementation of phase acquisition algorithms. A dual DLL prototype was fabricated in a 0.8 μm CMOS technology. The prototype achieves 80KHz-400MHz operating range, 12-ps rms jitter and 0.4-ps/mV jitter supply sensitivity.

Key Words and Phrases: High Speed Signalling, Receivers, Delay Locked Loops

Acknowledgments

This work would not have been possible without the help and support of many people.

First, I would like to thank my advisor Prof. Mark Horowitz. There way too many things I could say to thank Mark. Instead of restating well known facts, I will just say that I really feel that I could not have had a better advisor. It has really been a privilege working with him these past years.

I would also like to thank Prof. Tom Lee for discussing many of my ideas, and for reading this thesis along with my associate advisor Prof. Kunle Olukotun. Thanks also are due to Prof. Bruce Lusignan who served in my orals examination committee, along with Prof. Bruce Wooley who also proofread many of my papers. I am also grateful to Dr. Norm Jouppi and Mark Johnson for being a source of technical information and criticism, and to my M.Sc. advisor Prof. Manolis Katevenis and Prof. Fouad Tobagi for being the original reason I came to Stanford.

I also want to acknowledge ARPA for their financial support, DEC-WRL for allowing me to do part of my research while working there over a summer, and the people at Rambus who helped me finish writing this dissertation in the middle of a tape-out.

It has been a real pleasure to work with the staff of both the Center for Integrated Systems and the Computer Systems Laboratory at Stanford. In particular I am grateful for the technical support of Charlie Orgish, and the administrative support of Darlene Hadding, Ann Guerra and Terry West.

My time at Stanford has been a memorable experience. This is mainly due to the interaction I had with other graduate students. In particular I would like to thank Ken Yang for our collaboration on the chip described in the third chapter of this thesis, Sha Rabii and Marc Loinaz for being the ideal office-mates providing me both with technical advise and friendship, and Birdy Amrutur, Ricardo Gonzalez, Hesham Al-Twaijry, and Gu Wei for being there to discuss and challenge my ideas. I am also grateful to John Maneatis, Tom Chanak, Don Ramsey, Clem Portmann, and Drew Wingard for helping me in my early years of research. John, Tom and Don were particularly instrumental in convincing me to

join Mark Horowitz’s group – a decision I will never regret. Drew left me with the mixed blessing of supporting the VLSI CAD environment at Stanford – a decision I will never regret as well.

It is certainly impossible to find appropriate words to thank my wife Lena Tsakmaki. I would not have completed this effort without her, and so this thesis – for all it’s worth – is dedicated to her..

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Advances in IC fabrication technology coupled with aggressive circuit design have led to an exponential growth of the speed and integration levels of digital IC's. In order for these improvements to truly benefit the overall system performance, the communication bandwidth between IC's must scale accordingly. Rent, in his 1960 memorandum [1]-[3] related the gate count of a digital system (*Ng*) to its external interconnections (*Np*) in a formula which came to be known as Rent's rule:

$$Np = Kp\ Ng^{\beta} \qquad (1\text{-}1)$$

where β and *Kp* are empirically determined constants. Regardless of the absolute accuracy with which Rent's formula can predict the pin count of future IC's, its main implication is that *in order to maintain a balanced system the communication I/O bandwidth of IC's must scale with integration levels*. This thesis examines the problems associated with the design of high-bandwidth interfaces, and proposes techniques for increasing their speed and robustness while maintaining low latency and system cost.

Traditionally, system designers have addressed the increasing bandwidth demands by increasing the number of pins and wires interconnecting digital IC's. So it is not uncommon today, to have 128-bit wide busses in high-end workstation systems, and gate-array IC's packaged in 1000-pin ball grid arrays. However, this bandwidth improvement does

not come for free. Increased number of pins, printed-circuit-board (PCB) traces, connectors, and cables drive up the overall system cost. To minimize that cost, designers need to maximize the bandwidth of the data that can be transmitted per low-cost IC pin. Moreover, in many applications communication latency must be kept to a minimum, in order for the bandwidth increase to really benefit the system performance.

There are two main approaches to high-speed signalling. In serial interconnects, such as those used in local area networks, data is transmitted from one IC to another in a plesiochronous manner [4]. The receiving IC has to recover the clock encoded in the data transitions, and retime the data to its local clock. The main design goal in these systems is to increase the data transfer rate. The latency added into the system by the clock and data recovery circuits is a secondary concern, since the overall latency is usually dominated by the communication channel delay. Additionally, since serial links are not replicated in large numbers and usually employ a fiber optic based channel, the incremental circuit cost is not a major concern. So, this increased cost and latency imposed by the required clock recovery and data retiming circuits [5]-[10], make serial links more applicable to inter-system interconnects such as communication links between two computer systems.

High-speed parallel links are an alternative more amenable to interconnections within a single system, such as a workstation, a supercomputer, or a network switch [11]-[15]. The operation of these links derives from the conventional bus paradigm. Timing information is carried from one IC to another by means of a separate signal line, or alternatively, both IC's synchronize to a global system clock. The common timing is then used by the receiving IC to sample the data carried over a number of parallel channels. In this way, the cost of the extra timing line and the associated phase adjusting circuitry is amortized over a number of data lines. Since many of these parallel links need to be integrated within a single system, the overall overhead (area, power, latency) of the increased communication bandwidth is a key constraint. These constraints dictate a simpler design for the driver and receiver circuits and lower bandwidth per communication channel, when compared to serial links. It is these types of circuits that this thesis focuses on. Although the majority of this work was done with the particular application of multiprocessor interconnection networks in mind, the resulting techniques are general enough and can be applied directly to

other application areas, such as high bandwidth processor to memory interfaces [11], and high bandwidth communication switching systems [16].

1.2 Overview of this work

This thesis comprises six chapters of which this introduction is the first. Since system interface design has been addressed and studied extensively, Chapter 2 starts with an overview of parallel interface architectures, namely conventional multi-drop busses, and higher speed “source-synchronous” busses and parallel point-to-point links. Since some of the main limitations of the achievable bandwidth in existing designs are imposed by the system environment, the chapter continues by reviewing noise introduced both by transmission media and active circuits. The chapter concludes with an overview of signaling and synchronization methods.

A baseline high-speed interface design is the topic of Chapter 3. The trade-offs involved in the design of the signalling circuits, input receivers, and clocking circuits are described, along with the results obtained from a fabricated prototype [15]. The limitations of this simple design were the main motivation for the receiver and clocking circuits discussed in Chapters 4 and 5

One of the main limitations of existing parallel interface designs is that low swing, high speed signals have to be received in the noisy environment of a digital chip. The noise coupling becomes even worse in the most economical class of pseudo-differential interfaces. In these systems the maximum achievable bandwidth is limited by high frequency noise and by the fact that the high speed data is sampled only once per valid-bit period. Chapter 4 proposes a receiver design which integrates the incoming data over its valid time period, effectively filtering out high frequency noise [17], [18], [19]. The circuit design issues of the first stage integrator along with its associated biasing and sampling issues are discussed next. The chapter concludes with the description of a complete interface design utilizing current integrating receivers and the experimental results measured on a prototype fabricated in a 0.8-μm CMOS technology.

Another important issue is the design of clock phase alignment circuits which are necessary to generate the timing events used in the reception of the high speed signals. Chapter 5 addresses the design of these circuit blocks. A class of circuits known as Delay-Locked Loops (DLL's) offers many advantages over more conventional Voltage Controlled Oscillator (VCO) based Phase Locked Loops (PLL's). The main limitation of DLL's is their limited phase capture range. A new dual DLL architecture that eliminates this problem, while keeping the clock jitter and offset low is proposed. The implementation of the circuit building blocks and the results from a fabricated prototype are also discussed [20], [21].

The final chapter summarizes the contributions of this work and discusses areas of further development.

Chapter 2

Signalling and Clocking

This work focuses on circuits and architectures for high performance parallel links. In order to provide a framework for understanding the trade-offs and issues behind them, this chapter provides an overview of high speed interface design. The two main issues in extending the bandwidth of interconnections between system components are: *(i)* signalling - i.e., sending and receiving high speed data in the presence of digital system noise, and *(ii)* clocking - i.e., synchronizing the system so that the receivers and transmitters send and sample the data at the right time instant. These two issues are the topic of this chapter.

Section 2.1 discusses the structure and functionality of the ubiquitous bus-based systems, along with the signal integrity and timing uncertainty problems that impede the scaling of their transfer rates. Source synchronous systems mitigate the signalling and synchronization problems of conventional busses by constraining both the physical dimensions and the signal flow on the communication medium. The architecture of these systems is discussed in Section 2.2.

The signalling and clocking methods employed by source-synchronous interfaces is the subject of the rest of this chapter. The performance of these methods is often limited by their robustness in the presence of noise. The types and sources of noise present in digital systems are discussed in Section 2.3. The two main methods employed to send and receive signals in a transmission line environment, high and low impedance signalling, are the topic of Section 2.4. Section 2.5 concludes the chapter with an overview of the issues involved in synchronizing high-speed interfaces.

2.1 Limitations of Conventional Busses

The interconnects in a digital system have traditionally followed the bus paradigm [22]. In a bus system (Figure 2.1) a set of parallel wires is used to interconnect a number of IC's. In order to synchronize the transmission and reception of data, a global bus clock is distributed to all the IC's. The operation of the bus is divided into bus transaction periods - each period can be a multiple of the bus clock cycles. Arbitration for the use of the shared bus medium is usually performed by a "bus-master" component. Based on the decision of the master, a given bus cycle is allocated so that a single IC in the system ("bus-slave") transmits data on the bus, while one or more receiver IC's capture the transmitted data. This system provides a shared communication resource, through which each IC can communicate with all the other IC's in the system.

Although this communication paradigm has been adequate in the past, increasing speeds accentuate transmission line effects limiting the performance of conventional bus systems. In the past the electrical length of the bus conductor was short compared to the rise time of the signals. In this case transmission line effects are insignificant - the interconnect can be modeled as an equipotential lumped capacitor or a distributed RC line. However, increasing signal speeds magnifies the effect of the final propagation velocity of the signal energy on the line, creating both signal integrity and timing uncertainty problems. When signal rise times are comparable to the round-trip time of flight of the signal through the line, distributed transmission line characteristics become important and the line cannot be modeled as a single equipotential node [3], [23]. In this case, assuming the

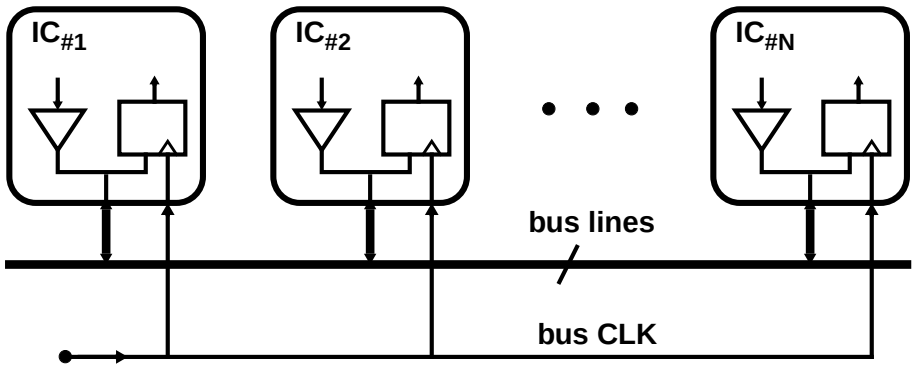

Figure 2.1: Conventional bus block diagram

conductor resistance is very small, the line can be modeled as a ladder network of infinitesimally small inductive and capacitive elements. The signal-wave propagation velocity[1] is then $u = 1/(\sqrt{L \cdot C})$ and the characteristic impedance that the line presents to a fast driver is $Z_0 = \sqrt{L/C}$ where *L*, *C* are the line inductance and capacitance per unit length respectively. For example, the propagation velocity of a signal on an 8-mil wide trace of a typical FR-4 PCB is approximately 7-in/ns (i.e., ε_r*=4.7, L=14nH/in, C=1.5pF/in*). Therefore, any trace longer than approximately 3-in will exhibit transmission line behavior, when driven with sub-nanosecond rise time signals.

From a circuit design perspective, the first problem that a designer has to face is that of matching the transmission line impedance to that of its load. In general, the interface of any transmission line with impedance Z_0 to a load or transmission line with impedance Z_L will reflect back a portion Γ of the incident wave. The reflection coefficient Γ is given by:

$$\Gamma = \frac{Z_L - Z_0}{Z_L + Z_0} \tag{2-1}$$

The reflection problem is usually addressed by terminating the bus conductors at both ends with resistors whose nominal value matches the line characteristic impedance Z_0. Although this increases the power dissipation of busses with TTL or CMOS signal swings, it has helped to extend the speed of conventional busses up to 80-100 MHz.

Increasing the bus signal speeds further can make the signal rise times comparable to the propagation delays through the “stubs” that tie the IC’s to shared bus medium (i.e., connecting PCB traces, board connectors and ultimately IC package traces and bond-wires). If the electrical length of these stubs is longer than the signal rise times, then the transmission line effects of the stubs become significant as well. This creates impedance discontinuities on the main bus lines and degrades the signal quality through reflections whose magnitude governed by Equation (2-1). The conventional approach for solving this

1. The propagation velocity *u* of a signal in a homogeneous loseless transmission line, is determined by the speed of light c_0 and the relative dielectric constant ε_r of the line insulating material $u = c_0/\sqrt{\varepsilon_r}$. In addition to ε_r and c_0 propagation velocities in realistic (i.e., non-homogeneous) interconnects are determined by the specific geometry of interconnect [3].

problem has been to limit the maximum signal edge rate, thus imposing an upper bound on the data bandwidth achievable by conventional bus systems.

Timing offset, or skew, creates the second most important limitation on system level interconnections. Skew results both from variation on the electrical characteristics of the bus IC's, and from the finite propagation speed of the signals through the bus conductors. In an ideal situation the bus clock events would occur at exactly the same time on every bus IC, causing data to be driven to or sampled from the bus simultaneously. To satisfy this requirement, bus designers try to equalize the distance from the bus clock source to every component. However, even when the clock distribution scheme is successful in minimizing skew, the bus clock still needs to be buffered internally on every bus IC. The unavoidable variation of the process and operating environment of the IC components introduces variations in the delay between the bus clock and the on-chip clocks. This variation degrades timing margins, and limits the maximum achievable transfer rate. Even when skew is compensated by using clock phase alignment circuits, the more fundamental uncertainty introduced by the propagation delay of the data through the interconnect transmission lines imposes the ultimate limit on the maximum transfer rate achievable by conventional busses. For example, the signal time of flight between $IC_{\#1}$ and $IC_{\#2}$ in Figure 2.1 is different from that between $IC_{\#1}$ and $IC_{\#N}$. This means that the minimum bus clock cycle will be ultimately limited by the maximum distance difference between any two bus components. To evade this problem, several bus designs adopt an asynchronous clocking paradigm where no global clock is used, and the data transfer is based on source-asserted strobe signals. However, asynchronous signalling suffers from the increased overhead of the required request-acknowledge protocol and does not scale well to higher speeds. For this reason high speed system interconnects have increasingly adopted the signalling methods discussed in the next section.

2.2 Source Synchronous Interfaces

The signal integrity and skew problems of conventional busses are a result of both the physical dimensions, and the unconstrained data-flow from every IC to every other IC in the system. If these two characteristics are constrained in a manner that does not limit the

system performance, then higher data rates can be achieved. High speed busses achieve this by limiting the flow of data on the bus, and constraining the physical dimensions and electrical characteristics of IC components. High performance parallel links take a more radical approach, by completely eliminating the shared medium and using point-to-point interconnections. These two types of systems are the subject of this section.

2.2.1 High Speed Busses

High speed busses solve signal integrity problems, based on the fact that connections on a shared transmission line create reflections if the connecting “stubs” are long compared to the signal wavelength components. If the stubs are short enough so that their inductance is insignificant, they behave just as capacitive discontinuities. Moreover, if the electrical distance between them is short, the interconnection appears as a uniform distributed transmission line. More specifically, this condition holds if the rise time t_r of the disturbance caused by the capacitive discontinuity C_D is larger than the round-trip time of flight $2t_f$ between two discontinuities, roughly: $2 \cdot t_f \leq 2.2 \cdot Z_0 \cdot C_D$. Under this constraint, the interconnect transmission line capacitance per unit length increases by the amount introduced by the stub capacitance. Consequently, the propagation velocity of the signals through the line and the line characteristic impedance decrease. Based on these constraints, the signal integrity problems of conventional busses can be solved, if the spacing between the bus components, the component capacitance, and the physical characteristics of the supporting PCB are carefully controlled [24], [25].

In order to alleviate timing uncertainty problems, high speed busses take advantage of the fact that in many interconnects, such as in memory subsystems, data transfers occur only between a single master and a bus component (or vice-versa). So the solution is to make the data travel the same electrical distance as the bus clock [11]. This idea is illustrated in Figure 2.2. The single bus clock travels in two directions on the bus, corresponding to the two ways of communication: master-to-slave corresponds to $\mathsf{CK}_{\mathsf{MS}}$ and slave-to-master corresponds to $\mathsf{CK}_{\mathsf{SM}}$. Each of the bus components synchronizes its signal transmission and reception with these two clocks. When, for example, one of the slaves transmits data to the master, its output data pins switch precisely aligned with the edges of

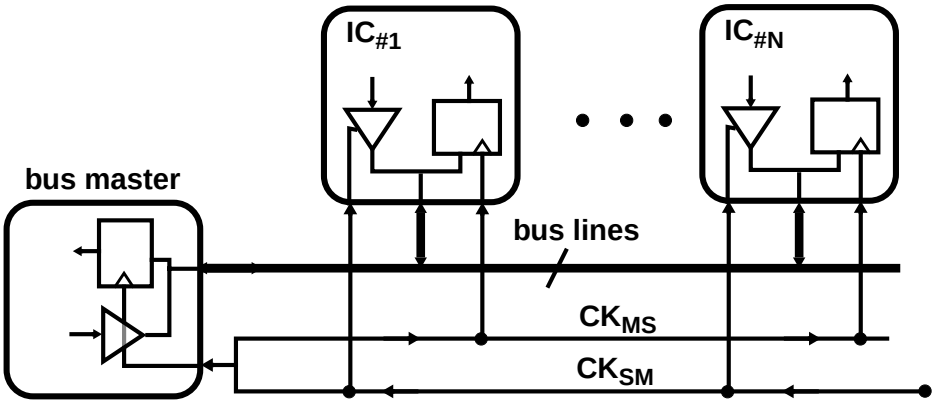

Figure 2.2: High speed bus block diagram

clock CK_{SM}. The data and clock arrive synchronized at the master which can then use the timing information carried on the clock line to receive the incoming data.

2.2.2 Point to Point Links

The physical constraints of high speed busses make that approach viable only in small scale systems, such as memory or peripheral busses. In larger scale systems, e.g., multiprocessors or communication switches, a more attractive approach is to completely abandon the bus paradigm and use point-to-point links. This approach has advantages both from a circuit design and an architectural point of view. From a circuit design perspective, the use of point-to-point transmission lines offers greater flexibility in the physical construction of the system. Moreover, a point-to-point link has potential for higher communication bandwidth than a bus, due to its reduced signal integrity problems. From an architectural perspective, the bandwidth demands of high speed systems make the shared bus medium the main performance bottleneck. For this reason, distributed interconnection networks and cross-bars have been gradually replacing busses in most large scale multiprocessors [12, 13, 14, 26], while the architecture of most high performance communication switches is inherently based on point-to-point interconnections [16], [27].

A simplified diagram of a high-speed point-to-point interface is illustrated in Figure 2.3. The synchronization scheme used in this system is similar to that used in high speed busses. Each of the two IC's sends data to the other through a dedicated channel of parallel transmission lines. A clock is transmitted along with the data, usually with its

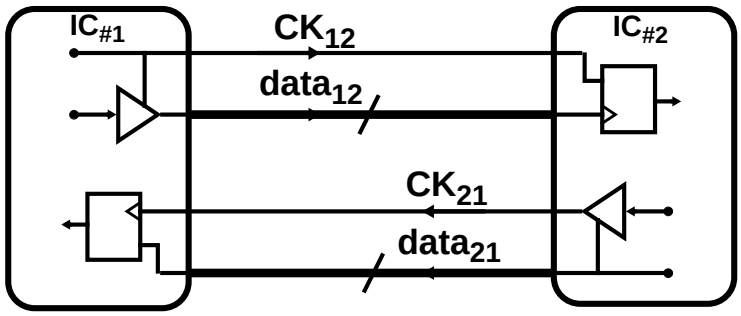

Figure 2.3: Point to point parallel link

edges synchronized with the potential transitions of the data. If the time of flight through the data and clock transmission lines are equal, the data and clock arrive synchronized at the receiving chip. The receiving chip can use the timing information embedded in the clock transitions to sample the incoming data. To maintain signal integrity, the parallel transmission lines need to be terminated to eliminate reflections. The uniformity of the transmission channel minimizes discontinuity related problems. The only inherent discontinuities in such a system are those introduced by the IC packages and the potential connectors.

Variations of this simple architecture which reduce the cost associated with increasing numbers of parallel data lines are possible. For example a source-synchronous point-to-point system might utilize a single set of parallel data lines, which can be used to carry data in both directions. This sort of sharing can be achieved either in a time division multiplexing fashion, or by using the data lines in a full duplex mode [28], [29].

The architecture of both the bus and point-to-point source-synchronous interconnects inherently solves many of the fundamental problems of conventional busses, thus enabling Gbit/sec/pin inter-chip data-rates. Nevertheless, the performance of source synchronous interconnects still depends on the performance of the signalling and clocking circuits employed by particular implementations. The design of these circuits is the subject of the rest of this chapter. First, however, the next section reviews digital system noise, which affects both signalling and clocking circuit performance.

2.3 Noise Considerations

The primary goal when selecting a particular signalling or clocking scheme is to transmit data between system components with maximum bandwidth, while keeping the associated costs low. These costs include the power dissipated and the area occupied by the signalling and synchronization circuits, as well as the cost of the required external components. One of the most important obstacles in achieving these objectives is the noise present in all digital systems. Noise alters the amplitude and timing of transmitted signals, thus impeding their correct reception. This additive noise can be either related to, or independent from the originally transmitted signal amplitude. Independent noise can be easily overpowered by increasing the amplitude of the signals. Dealing with a proportional noise source, though, requires minimizing or cancelling it. This is a more difficult goal, since it can only be accomplished through careful design of the signalling circuits and transmission environment. The most important proportional noise sources are reflections, cross-talk and self-induced power supply noise. Independent noise sources include thermal noise and unrelated power supply noise. Both of these types of noise are discussed in this section, along with methods commonly used to deal with them.

Reflection-induced intersymbol interference is the most common type of proportional noise. As was discussed in Section 2.1, to avoid reflections in a transmission line environment, signal lines need to be terminated. This can be accomplished by placing termination circuits on either the transmitter or the receiver end of the line. The termination circuit impedance absorbs the transmitted signal energy, and prevents it from being reflected back into the transmission medium. However, mismatches between termination and line impedances create reflected waves on the transmission line. These reflected waves add to the subsequent signals, thus resulting in a form of intersymbol interference. For example, Equation (2-1) shows that a 20% mismatch between the termination and line impedances gives rise to a reflected wave, whose approximate amplitude is equal to 10% of the original signal. Terminating both at the source and destination ends of the transmission medium can be used to alleviate this problem at the expense of increased power dissipation. Dynamic termination matching techniques can also be used to precisely match the termination impedance to that of the transmission line [30], [31].

Another source of reflections is manifested at higher speeds, when signal transition-times become comparable to the propagation delays through the package traces, bond-wires, and board connectors. These components can create inductive or capacitive discontinuities which again degrade the signal quality by generating reflections. The magnitude of the reflection V_{LD}, V_{CD} created by an inductive or capacitive discontinuity on a line with nominal impedance Z_0 is:

$$V_{LD} = \frac{L_D}{2 \cdot Z_0 \cdot t_r} \cdot V_I, \qquad V_{CD} = -\frac{C_D \cdot Z_0}{2 \cdot t_r} \cdot V_I \tag{2-2}$$

where L_D, C_D is the discontinuity inductance or capacitance, V_I the incident voltage magnitude and t_r the signal rise time. High quality connectors (e.g., Augat-EII, Teradyne-MBC), behave as distributed transmission lines with typical impedances in the range of 45-55 Ω. On the other hand, the pin inductance of typical ceramic packages varies between 15-30 nH. Better package designs reduce that inductance to 4-10 nH through the use embedded ground planes. Still, however, applying Equation (2-2) reveals why packages are one of the major limitations of high speed signalling today: a 5-nH discontinuity on a 50-Ω impedance line creates a reflection of approximately 12% with a 500-ps signal rise time.

Another problem created in a transmission line environment is that of cross-talk. The transmission line model used so far assumes that the capacitance and inductance of the line exist only between the forward and the return signal paths. While this is a safe assumption for a coaxial cable or an isolated PCB trace, it is not true for connector pins, package leads, and closely spaced PCB traces. The inter-conductor mutual inductance and capacitance couples noise between adjacent traces, connector pins, package leads, and bondwires. In such an environment, a wave propagating in a transmission line induces capacitive or inductive currents on adjacent and nearby lines. These currents create forward and backward propagating waves. The duration of the backward propagating wave is equal to twice the electrical length of the coupling and is proportional to the amplitude of the inducing wave. The governing proportionality constant K_R depends on the speed of

propagation u, the line impedance Z_0, and the mutual inductance L_M and capacitance C_M:

$$K_R = \frac{u}{4} \cdot \left(Z_0 C_M + \frac{L_M}{Z_0} \right) \tag{2-3}$$

The forward propagating wave has an amplitude proportional to the length of the coupled portion of the line, the inducing wave rise time, and the proportionality constant K_F which is given by:

$$K_F = \frac{1}{2} \cdot \left(Z_0 C_M - \frac{L_M}{Z_0} \right) \tag{2-4}$$

Equations (2-3) and (2-4) show that minimizing crosstalk entails minimizing the mutual inductance and capacitance by placing unrelated signal lines far apart and interleaving high speed signal pins with ground pins in connectors and packages. Furthermore, Equation (2-4) shows that forward crosstalk can be completely eliminated if the inductive and capacitive coupling between adjacent lines are matched (i.e., $C_M/C=L_M/L$), as is the case with embedded PCB traces. The reverse crosstalk reflection problem (i.e., crosstalk noise arriving to the receiver end after being reflected by the transmitter end) can be eliminated by terminating the transmission lines both at their source and destination ends. This way the reverse crosstalk is absorbed at the transmitter end by the corresponding termination resistor. However, the cost of these coupling minimization methods restricts their application in high-end systems, thus often forcing designers to accommodate large crosstalk components in the system noise budget.

Self-induced power supply noise is a result of the finite power supply pin impedance in semiconductor packages. When an output driver switches, the current drawn from the external supply of the chip changes at a rate equal to *dI/dt*. The inductance *L* of the chip's supply network will then cause the on-chip power supply voltage to drop by a voltage $\Delta V = L\ dI/dt$. For example, a 1-V amplitude signal transmitted in a 50-Ω line with 500-ps rise time requires a 20mA/500ps *dI/dt*. If the inductance of the on-chip power supply is 5nH,

driving the signal will result in a 200-mV voltage drop on that supply. This on-chip power supply voltage drop can appear as additive noise both on the switching and the quiescent signals. Since on-chip decoupling capacitance does not have an effect on the noise generated by switching output drivers [3], the two alternatives for minimizing this type of noise are: *(i)* minimizing the power supply network inductance, and *(ii)* using a signalling method which draws constant current from the external supply. The decision on which of these two methods is adopted depends mainly on the cost of increased number of power supply pins versus increased power dissipation.

The second class of noise sources is independent of the transmitted signal amplitude: thermal noise, process variation induced offsets, and unrelated power supply noise. Independent noise sources can be overpowered by enlarging the signal amplitude. Since thermal noise typically has very small amplitude, this method can be applied without any significant increase in power dissipation. For example a 50-Ω termination resistor driving a 1-pF load results in a thermal noise power of approximately $4.15\ 10^{-9}\ V^2$ over a 3.3-GHz bandwidth. This corresponds to a 64-μV RMS value of Gaussian thermal noise. The probability of this noise amplitude exceeding 0.5-mV is less than 10^{-26}. Since typical signal amplitudes are well above 100-mV, the Gaussian thermal noise can be safely ignored in a typical signalling system. On the other hand, offsets induced by process variations typically have larger amplitudes, and depend on the design and layout of the driver and receiver circuits. Still, however, their magnitude can be easily bounded to below 50-mV, so they are not a large concern. Unrelated power supply noise is created by digital circuits integrated on the same die as the signalling circuits. This type of noise can be either minimized by using on-chip decoupling capacitors, overpowered by using increased signal swings, or canceled by appropriate signalling. As will be seen in Section 2.4, overcoming proportional and unrelated power supply noise is the main challenge when designing a robust signalling method.

2.4 Signalling Methods

A signalling method provides the means by which binary information is reliably sent between IC's over a given transmission medium. A signalling system consists of various components, the design of which comprises several trade-offs affecting the performance and cost of the overall system. The binary information on the transmitter IC is first converted to a signal on the transmission medium by the transmitter output driver. The most important characteristics of this driver are its output impedance and the resulting signal levels on the line. The signal levels are chosen so that the signal can be distinguished from additive noise, according to considerations discussed in Section 2.3. The output impedance of the driver affects the noise rejection and power dissipation properties of the signalling system. Tightly coupled with the transmitter design is the topology and placement of the termination circuits which, as discussed in Section 2.3, absorb the transmitted signal energy and prevent it from being reflected on the transmission medium. The termination circuits may be placed on both the transmitter and receiver ends of the line. Moreover, depending on the impedance of the output driver, termination can either be in series or in parallel with the flow of the signal.

The signal sent over the transmission medium needs to be decoded back to binary information at the receiver end. This decoding function is essentially a comparison of the incoming signal to an explicit or implicit "reference" value. In the conventional approach of purely single ended signalling, the reference value is implicitly set relative to the supply by the threshold voltage of the receiver circuit. Despite the minimal design complexity of this method, its main drawbacks, when compared to low-swing differential or pseudo-differential signalling, are increased power dissipation and reduced noise immunity. In fully differential signalling both the true and complementary value of the signal are sent over the transmission medium, creating an implicit "reference" value and improving noise immunity. In pseudo-differential signalling the reference value is explicitly generated by a dedicated circuit and shared by a number of parallel receivers. In this way, pseudo-differential signalling trades-off noise immunity for reducing the system power dissipation and number of required pins and wires.

Signalling methods can be categorized mainly by whether the impedance of the transmitter output-buffer is comparable to, or much higher than the impedance of the transmission medium. Both types of signalling can be implemented either in a differential or a single-ended form. These two main methods of signalling – high and low impedance – are discussed in the following two subsections.

2.4.1 Low Impedance Signalling

In a low impedance signalling environment, the impedance of the output buffer is equal to or less than the impedance of the transmission medium. Thus, the buffer can be best approximated as a switching voltage source in series with a resistor driving the transmission line – also commonly referred to as “voltage-mode” driver. The traditional implementation of low-impedance signalling in bipolar or BiCMOS technologies uses an emitter follower driving a parallel-terminated transmission line. Since the CMOS equivalent of a parallel-terminated emitter follower is both difficult to implement and consumes a lot of power, low-impedance signalling systems in CMOS usually employ a simpler series-termination scheme with different characteristics. This type of signalling is the main topic of rest of this section

A model for a point-to-point series-terminated low-impedance signalling system is depicted in Figure 2.4. The transmitter buffer behaves as a time varying voltage source

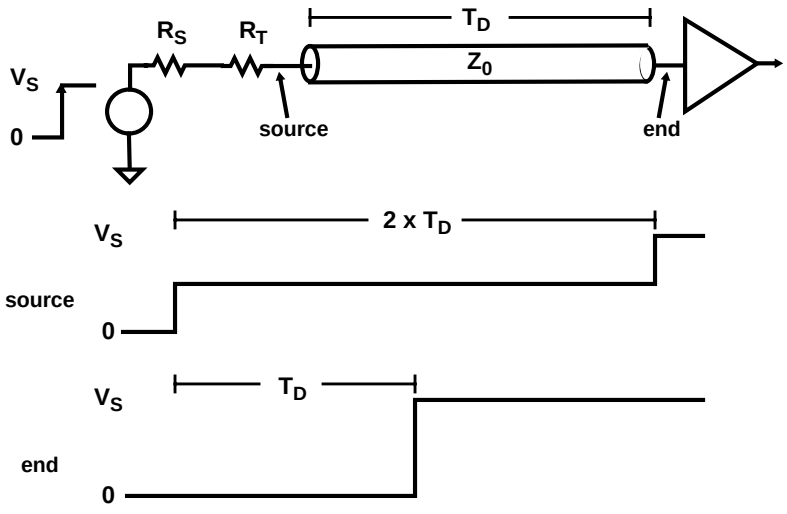

Figure 2.4: Model of a low impedance series-terminated signalling system

with a source impedance R_S. To avoid standing waves on the transmission line the value of the series termination resistor R_T plus the output impedance of the voltage source R_S is equal to the characteristic impedance of the transmission line Z_0. This way any wave propagating towards the source end of the transmission line is absorbed by the combined series impedance of R_S+R_T.

In the signalling system of Figure 2.4 the transmitter injects a voltage step in the transmission medium. In the ideal case where $R_S+R_T=Z_0$, a source voltage step of amplitude V_S is initially divided by 2 before it begins propagating in the transmission line. At the open far-end of the line where the reflection coefficient Γ is 1, the propagated wave is doubled, resulting in a receiver incident wave amplitude equal to the initially transmitted step V_S. In case the load of the far end termination is purely capacitive the incident wave is an exponential with time constant $Z_0 \times C$, where C is the capacitive load. The reflected wave has an amplitude $V_S/2$ and propagates back towards the source end of the line. When this reflection arrives at the source end, it is absorbed by the series combination of R_S+R_T. At this point the voltage throughout the system stabilizes at a value equal to the initially transmitted step V_S and the source driver current drops to zero. If the round-trip delay through the line $2 \times T_D$ is larger than the signal bit time T_B, the transmission of the next symbol starts before the reflection of the previous symbol returns to the source. In this case, assuming the source resistance is linear, the transmission line and transmitter-end voltages are a result of the continuous superposition of the forward and the reverse propagating voltages of the transmitted symbols and their half amplitude reflections. However, the resulting voltage at the receiving end is indistinguishable from the case when $T_B > 2 \times T_B$ regardless of the magnitude of symbol times.

The power dissipation of a series-terminated low-impedance signalling system depends on the relation of the bit time to the round-trip delay through the line. When the delay through the line is less than the bit time, the system dissipates power only during the initial round-trip of the signal through the transmission line. The resulting worst-case power dissipation is $T_D/T_B \times {V_S}^2/Z_0$. When the bit duration is shorter than the line round-trip delay, the driver is supplying continuous current through the line resulting in a worst case power dissipation of $V_S^2/(2 \times Z_0)$. However, the power dissipation can be less

than this absolute maximum, when the signal transition density is low enough to allow the line voltage to settle between transitions. This zero static power dissipation is the main advantage of low-impedance series-terminated signalling.

Decoding the signal at the receiving end, requires a way of referencing the incident wave to some predetermined voltage standard. For this purpose a low impedance signalling system can implement either purely single-ended, differential, or pseudo-differential signalling. A purely single ended system can be implemented by using conventional CMOS inverters both at the transmitter and receiver end of the line [32]. This straightforward implementation, shown in Figure 2.5-(a), is susceptible to common mode noise. The low impedance of the transmitting buffer causes power supply noise on the transmitter chip to appear unattenuated on the transmitted signals. Additionally, since the threshold voltage of the receiver is implicitly set by the power supply of receiving IC, any noise on either the receiver or transmitter IC directly subtracts from the signal noise margins. Another disadvantage, discussed in more detail in Chapter 3, is that the impedance of the transmitting CMOS inverter changes during the signal transition. Therefore, even in the presence of an external termination resistor this type of system might suffer from multiple

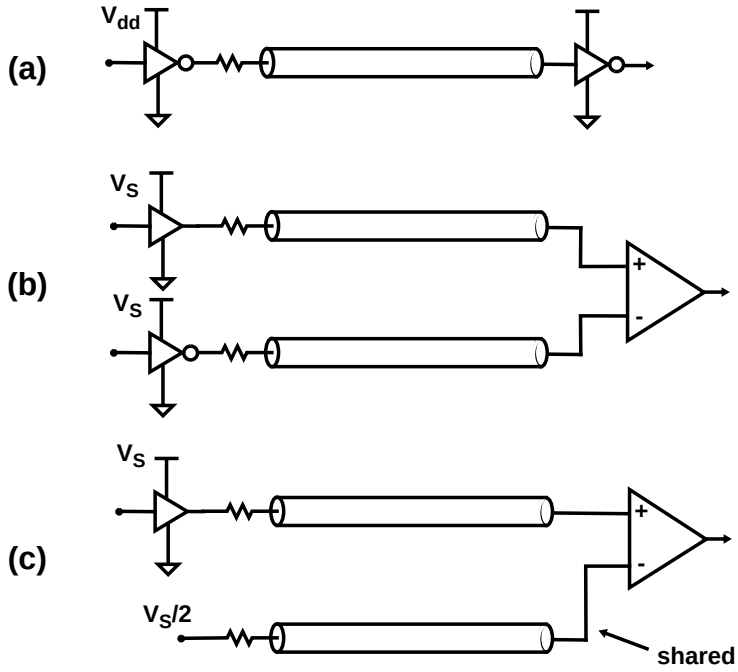

Figure 2.5: Alternative implementations of low impedance, series-terminated signalling

reflections. These disadvantages, along with the increased power dissipation and self induced *dI/dt* noise resulting from driving a full-swing signal, make this simple realization of low impedance signalling unattractive in high-speed systems.

Many of the problems outlined above can be mitigated if the signalling system uses lower signal amplitudes. Additionally, since, as described in Section 2.3 a large portion of the noise is proportional to the signal swing, reducing the signal swing does not result in a proportional reduction in signal noise margins. In fact, the noise margin as a fraction of the signal swing can remain unchanged as long as the system noise is dominated by proportional noise. An additional advantage of smaller signal swings is that the impedance of the transmitting buffer varies less over the signal swing, making the series termination more effective in absorbing reflections that arrive at the source end while the buffer changes state. A reduced output swing buffer can be implemented using a push-pull buffer with a supply equal to the signal swing [15], [29], [31], [33]. Another alternative is to implement the driver with open-drain MOSFETs operating in the linear region [12]. In the latter case the driver impedance becomes infinite when a high voltage is transmitted on the channel, so the transmission line needs to be also parallel-terminated to a voltage that determines the high end of the signal swing. The active driving impedance in both alternatives can be set to be equal to that of the transmission line, either by using external termination resistors [15], or by a dynamic impedance matching scheme [30], [31].

Reducing the signal swing means that the reference cannot be set implicitly by the IC supply voltages. A fully differential signalling scheme (Fig. 2.5-(b)) can be employed to provide an implicit reference, and simultaneously maximize the noise robustness of the system. In this case the transmitter chip sends both polarities of the signal, and the receiver considers only the difference of the two incident waves. In this way the bit decoding at the receiver is, to the first order, independent of the supply voltages, thus improving the noise tolerance of the system at the expense of power dissipation and increased number of pins and wires. Alternatively, pseudo-differential signalling can be used to reduce system cost. In pseudo-differential systems, such as the one shown in Figure 2.5-(c), a reference voltage in the center of the signal swing is generated on the transmitter side. This voltage is shipped to the receiver, and can be shared among a number of signals in a parallel inter-

face, thus reducing the number of lines and the overall power dissipation of the system. This form of pseudo-differential low impedance signalling is the most common in parallel point-to-point interfaces.

The series-termination method described above, is commonly used in CMOS implementations of point-to-point low-impedance signalling systems. In many applications however, low-impedance drivers are combined with other termination schemes. For example in some implementations, the backwards reflection of the source-terminated open-ended drivers in Figure 2.5 creates multiple reflections on the transmission medium, especially when combined with the non-linear driver impedance. In these cases, transmitter series termination can be combined with receiver-end parallel termination, in order to eliminate multiple reflections at the expense of increased power dissipation [14]. In bus environments, low impedance drivers are implemented as open-drain FETs operating in the low-impedance linear region, while the bus lines are terminated on both ends to a voltage that sets the high level of the signals [34]. Despite the differences of these alternative implementations, the characteristic they share with the most common series-terminated case is that of relatively low on-chip driver impedance. This reduced driver impedance creates the main disadvantage of low-impedance signalling systems, by not allowing isolation of the signal on the transmission medium from noise on the transmitter chip. High impedance signalling systems, discussed in the next section, eliminate this problem through the use of current source drivers.

2.4.2 High Impedance Signalling

A model of a high impedance signalling system is shown in Figure 2.6. The transmitter buffer behaves as a time varying current source, generating current pulses of magnitude I_S. The resulting voltage pulses of magnitude $I_S \times Z_0$ propagate at full intensity through the transmission line. At the receiver end of the line, the termination resistor with a value ideally equal to the characteristic impedance of the line Z_0 absorbs the propagated voltage wave. Similar to the case of low-impedance series-terminated signalling, the incident wave is an exponential waveform with a time constant of $Z_0 \times C$. Following the transition at the receiver end, the system stabilizes until the transmission of the next bit through the

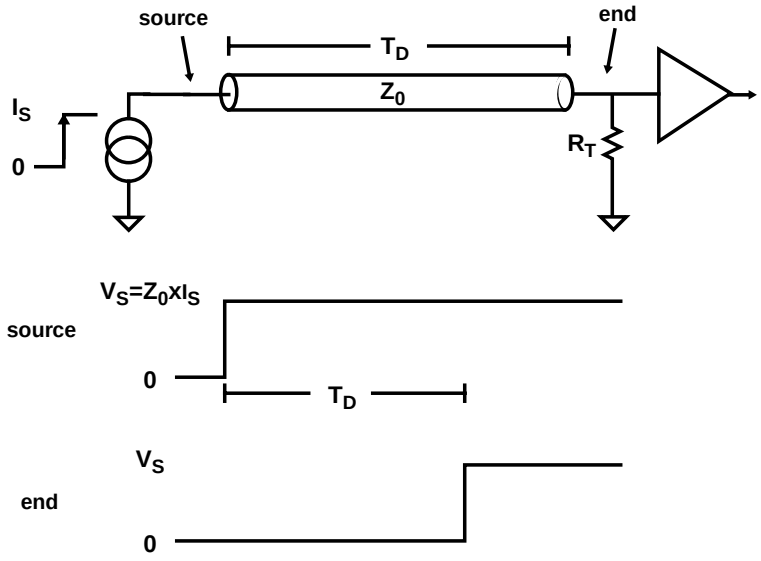

Figure 2.6: Model of a high impedance parallel-terminated signalling system

line. Similar to the case of low-impedance signalling, the transmission of the next bit can start before the pulse corresponding to the previous bit arrives at the receiver. The power dissipated in a high impedance signalling system depends on the pattern of the transmitted data. The worst case power dissipation of $I_S^2 \times Z_0$ in the model of Figure 2.6, occurs when the data is a stream of 1's, causing the transmitter current source to continuously supply current to the transmission line.

The main advantage of the signalling system in Figure 2.6 is that the current source driver isolates the line signal from noise on the transmitter IC power supply, thus minimizing a major source of proportional noise. Its main disadvantage however, is that the absence of termination on the transmitter end causes backwards propagating noise, such as reverse crosstalk, to be reflected towards the forward signal direction and add to the noise seen at the receiver end. Eliminating this problem requires terminating both ends of the transmission line at its characteristic impedance, which results in reduced signal swing or increased power dissipation.

High impedance signalling systems are usually implemented in fully differential or pseudo-differential form. Figure 2.6-(a) shows a typical implementation of low-swing fully-differential signalling [35], [36], [37]. The open-drain differential pair approximates the current source driver of Figure 2.6. The tail current is steered on the branches of the

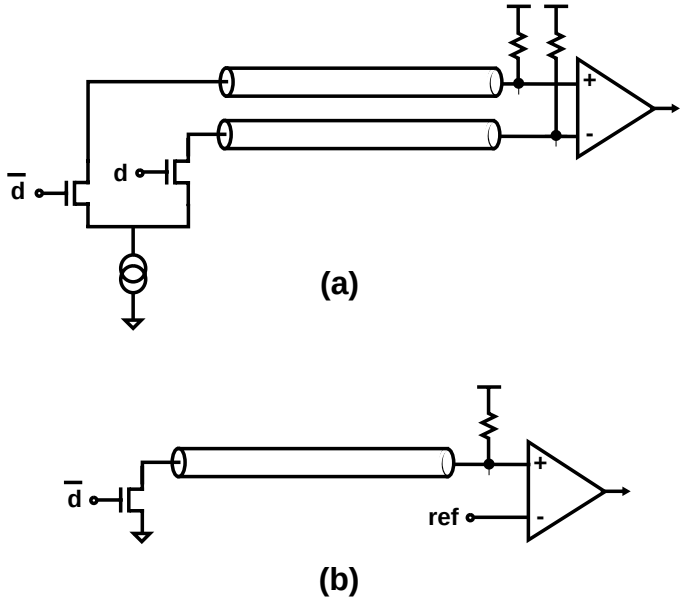

Figure 2.7: Alternative implementations of low impedance, series-terminated signalling

differential pair, creating a differential voltage on the two equal-length transmission lines. The receiver amplifies the voltage difference across the matched termination resistors, decoding the transmitted bit. The fully differential operation of the transmitter and receiver rejects common mode noise. Additionally, the constant current drawn by the transmitter buffer minimizes the noise induced on the transmitter chip. Although this form of high impedance signalling is the most noise immune, it is also the most expensive in terms of power dissipation and required transmission lines and package pins. As with low impedance signalling, a compromise is to implement the pseudo-differential signalling system shown in Figure 2.7-(b), by using an open drain NMOS driver [11], [13], [38]. In contrast to low-impedance open-drain drivers, in these implementations the driver transistor must remain in the saturation region of operation, where it best approximates a current source. Therefore the signal on the line must not fall more than a threshold voltage below the gate of the driver transistor. Additionally, to maintain a constant driver current and isolate noise on the transmitter chip supply, the gate voltage of the driver transistor must track its source voltage. The reference voltage of the receiver can be generated either externally or internally to the transmitter chip, and can be amortized over a number of parallel signals to reduce the cost of the system. The particular choice depends on the system constraints: point-to-point systems can easily use an internally generated reference voltage, while

multi-drop busses use a global reference and adjust the buffer current to maintain correct signal swings around the reference voltage.

Although the design space of low and high impedance drivers seems relatively large, the differences mainly concern trade-offs between power-dissipation and noise rejection, and the final choice depends on the constraints of the particular system. However, a robust signalling method is not the only requirement for achieving high bandwidth interchip communication. In addition to be able to distinguish between the different values of a single data item, the system needs to be synchronized, so that it can distinguish the boundaries of data items in time, and reliably transfer all of them from one IC to another. Methods that address that problem are discussed in the next section.

2.5 Clocking Methods

A clocking or timing discipline synchronizes a signalling system, by dictating when a driver circuit places a new data item on the interconnection line, and when the receiver samples that data item at the other end of the line. The main problem that a clocking method has to address is that of timing uncertainty. Timing uncertainty in interchip signalling comes from various sources, and can be distinguished into two main categories. Fixed timing uncertainty, or skew, is caused by unequal line lengths and the delay variation of IC components due to manufacturing. Time-varying uncertainty, or jitter, is caused mainly by signal amplitude and power supply noise. Signal amplitude noise can translate to jitter by altering the time at which the value of a signal changes. Power supply noise introduces timing uncertainty by affecting the delays through the on-chip signal paths. A third source of time varying timing uncertainty is temperature variations which also affect the delays through on chip signal paths.

In point-to-point signalling systems, synchronization is a problem that can be mainly addressed at the receiver end: every transmitted symbol needs to be sampled off the line at the time instant during which is most unlikely to change. This requirement usually dictates implementing data pin receivers which sample the data at the center of its “eye”, as illus-

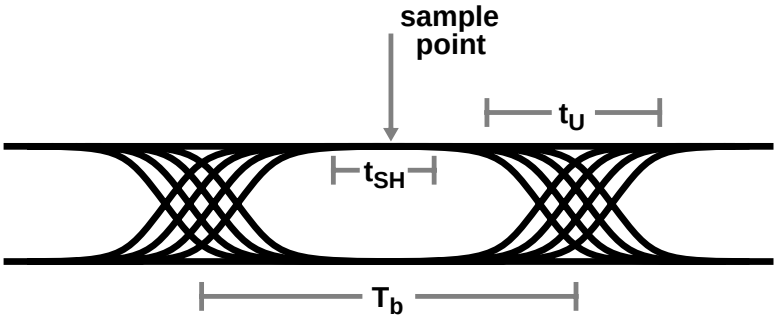

Figure 2.8: Optimal sampling point and timing margins

trated in Figure 2.8. This way the maximum data transfer rate is determined by the following two parameters:

- t_U: the incoming signal timing uncertainty. This is the sum of the rise/fall time of the signal plus the uncertainty in the total signal delay.
- t_{SH}: the receiver “setup-and-hold” uncertainty window - i.e., the time-zone around the sampling time during which a changing input signal can result in an undefined receiver output. Both the sampling time uncertainty and the receiver apperture contribute to t_{SH}.

The timing margin of the clocking scheme t_M, which can be viewed as the tolerance to additional delay uncertainty, is given by: $t_M=T_b-t_{SH}-t_U$, where T_b is the bit time.

Positioning the sampling event at the center of the data eye requires knowledge about the potential data transition points. This requirement, along with the fact that data is generally aperiodic, suggests that either encoding of the data, or providing an explicit timing reference signal is necessary. The first option can be implemented either through a completely asynchronous protocol [39], or by encoding the data to guarantee some level of transition density and recovering the transmitter clock [5]-[10]. Due to the increased overhead associated with both of these methods, intra-system interconnect designers usually rely on supplying an explicit reference clock signal, thus realizing a mesochronous timing environment [4]. The fixed timing uncertainty, introduced by the variation of the delay from the system clock source to the interface components, is minimized by using a source

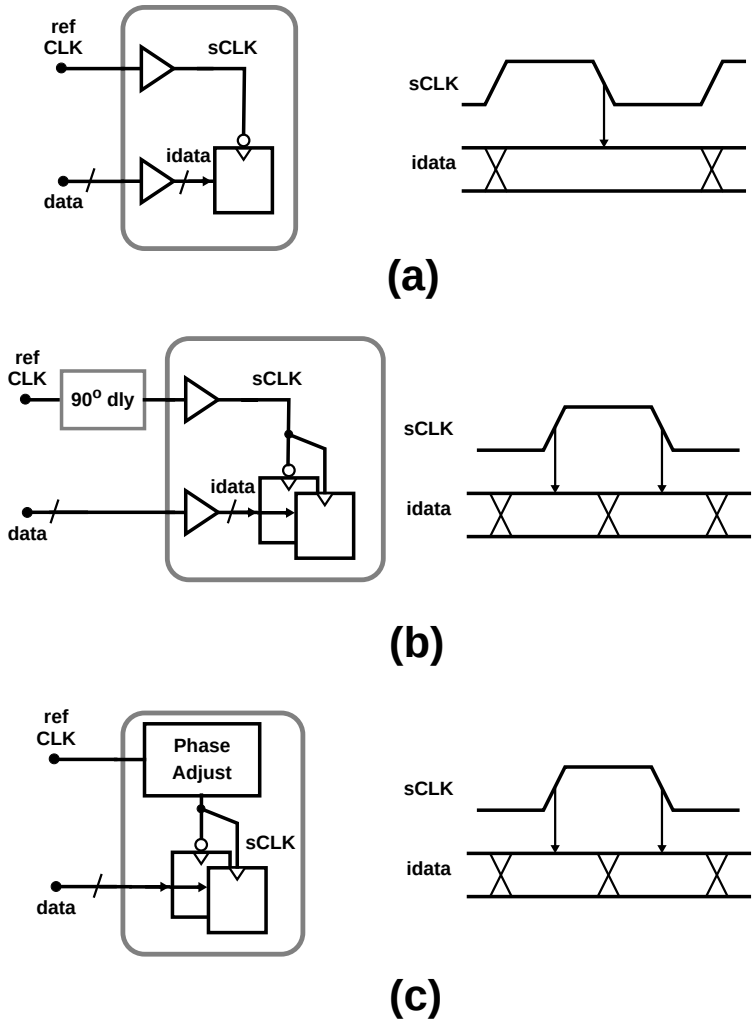

Figure 2.9: Receiver clocking alternatives

synchronous signalling architecture such as those described in Section 2.1.

Having minimized the major component of fixed timing uncertainty through architectural changes, the remaining problem of source-synchronous interfaces is to position the on chip sampling event in the center of the data eye. Figure 2.9 illustrates three alternative ways of accomplishing this goal. The simplest method is depicted in Figure 2.9-(a). The reference clock transitions at twice the rate of the data. Both the reference clock and the parallel data items are amplified and buffered on the receiver chip through matched delay buffers. The negative edge of the on chip clock is then used to sample the on chip data. The simplicity of this method is its main advantage. However, setting the data rate equal to the clock frequency utilizes poorly the available interconnect bandwidth. In an alternative

method, depicted in Figure 2.9-(b), data bits are sent during both of the half-periods of the clock [12]. Although the reference clock is again sent in phase with the data, it is also delayed by 90^o through an external transmission line whose electrical length is half a bit time longer than the electrical length of the parallel data lines. The on-chip data and clock buffers have matched delays, resulting in a 90^o phase displacement between the on-chip sampling clock and data. Therefore, both clock edges are positioned in the center of the data eye, and both of them can be used to sample the on chip buffered data. While this simple method can potentially double the data rate, its main disadvantage is that the receiver's timing margin is fixed by an external component, and does not improve at slower clock frequencies. Moreover, the external 90^o delay, usually created by a longer PCB trace, occupies board space and is not always accurate. For these reasons, the most reliable solution is to use an on-chip phase adjustment circuit to phase shift the on-chip sampling clock by 90^o relative to the external reference clock, and simultaneously cancel the potential amplification and buffering delay of the data as illustrated in Figure 2.9-(c). This phase adjustment circuit can be implemented either as a phase locked loop, or a delay locked loop, the design of which is the topic of Chapter 5.

The timing margin of the methods described above is mainly determined by the jitter of the sampling clock relative to the transmitted data. Since external coupling to the data signals and the reference clock can be minimized by careful system design, the main component of this jitter is introduced by on-chip power supply noise. At higher transmission speeds three additional factors of timing uncertainty need to be compensated. First, the potential offset in the timing uncertainty window of the input pin receiver results in the optimal placement of the sampling clock being slightly offset from the ideal 90^o point. Moreover, variations in the duty cycle of the sampling clock result in timing margin degradations, since the data is sampled on both clock edges. Finally, the skew introduced between the data signals and the reference clock by variations in the delay of nominally identical transmission lines might need to be compensated. Techniques for dealing with these sources of timing uncertainty will be discussed in the following chapters.

2.6 Summary

Increasing the interchip communication bandwidth in digital systems requires improving signal integrity and minimizing timing uncertainty. Both of these requirements lead system designers to abandon the traditional multi-drop bus architectures, and use source synchronous busses or point-to-point parallel links. Further bandwidth improvements require dealing effectively with digital system noise, which affects both the amplitude and the timing of interchip signals.

The designer can either overpower amplitude noise with large signals, or minimize its impact by using a noise insensitive signalling method. Since a large fraction of the noise is proportional to the signal amplitude, minimizing or canceling amplitude noise is the main goal of a signalling scheme. The driver circuit design largely determines the characteristics of signalling schemes. Low impedance signalling utilizes voltage mode drivers, and has the potential for zero static power dissipation. High impedance signalling uses current mode drivers, and has the advantage that it isolates the transmitter's power supply noise from the signal. While both methods can be implemented in a fully-differential fashion, cost considerations usually dictate pseudo-differential implementations, in which a shared reference voltage or current is used by the receiver to decode the signal levels. Similarly, cost and latency considerations dictate the use of a shared reference clock signal to designate the symbol/bit boundaries of the transmitted signals. The receiver uses the timing information embedded in the reference clock transitions, to sample the data on the interconnect lines. To position the sampling event at the optimal point in time, and cancel the timing uncertainty introduced by potential amplification and/or buffering, a phase alignment circuit is necessary, although less robust clocking schemes relying on external delay components are also used.

Chapter 3

Source Synchronous Interface Design

As discussed in Chapter 2, the inherent timing uncertainty problems of conventional multi-drop busses lead designers to adopt source synchronous signalling systems in order to increase interchip communication bandwidth. Increasing that bandwidth further requires careful consideration of the signalling and clocking issues of source synchronous systems. To elaborate on these issues, this chapter discusses the architecture and circuit design of a 500 Mbits/sec/pin point-to-point link parallel interface [15], intended for use in multiprocessor interconnection networks. This design shares the same basic approach with others described in recent literature [12], [13], [14]. The limitations of this baseline approach form the main motivation for the development of the signalling and synchronization circuits discussed in Chapters 4 and 5.

A high-level block diagram of the interface is shown in Figure 3.1-(a). To minimize system cost and power consumption this interface uses a low impedance pseudo-differential signalling scheme. To achieve synchronization, a reference clock is transmitted in phase with the parallel data signals, as was outline in Section 2.5. Given that the electrical lengths of the data and clock lines are equal, the signals arrive in-phase at the receiver end. The receiver can then use the timing information embedded in the transitions of the reference clock to position its on-chip sampling clock CLK in the center of the valid bit time, and sample the incoming data twice per clock period as depicted in Figure 3.1-(b).

This chapter will begin by discussing in Section 3.1 the design of the signalling circuits of the interface. In particular, the trade-offs considered in the design of the line driver

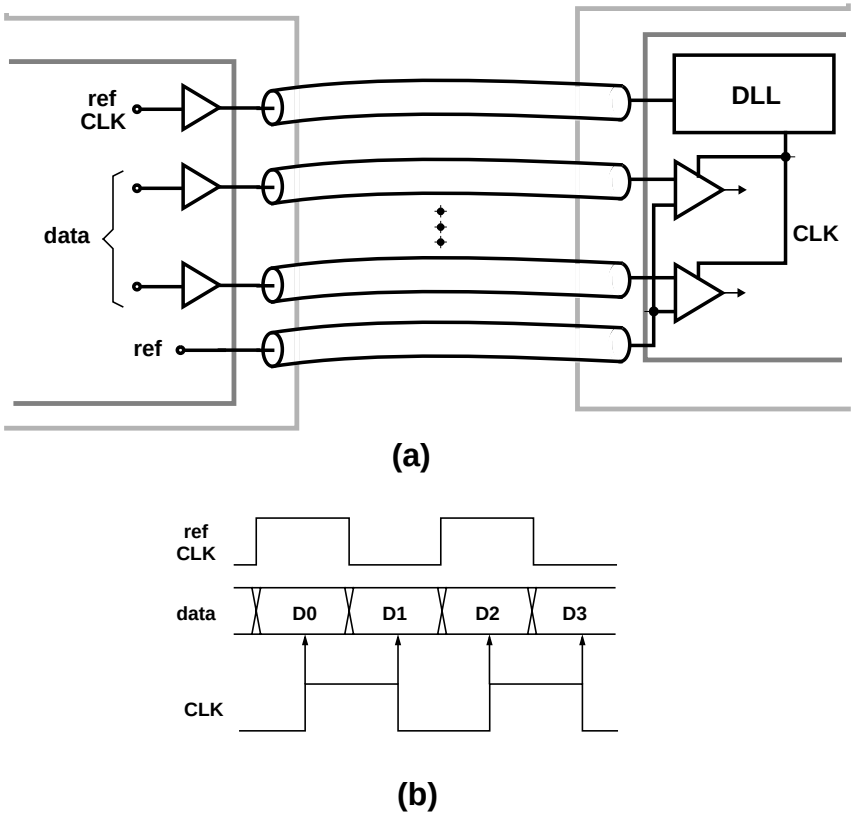

Figure 3.1: Interface block diagram (a), and timing (b)

circuits will be discussed along with the design of the input pin receiver. Section 3.2 will then address the design of the key clocking circuits: the DLL used to offset the on-chip sampling clock by 90^{o} relative to the transmitted data, and the duty cycle adjuster circuit used to restore the duty cycle of the sampling clock to its nominal 50% value. The experimental results measured on a fabricated prototype will be discussed in Section 3.3. Having reviewed the details of this baseline design, the chapter will conclude with a discussion of its main limitations.

3.1 Signalling Circuits

The configuration of the interface signalling circuits is depicted in Figure 3.2. This interface uses push-pull, series-terminated drivers, with low-swing 1-V output signals. To implement pseudo-differential signalling, a 0.5-V reference voltage is generated on the transmitter board using a resistive voltage divider. This reference voltage is sent to the

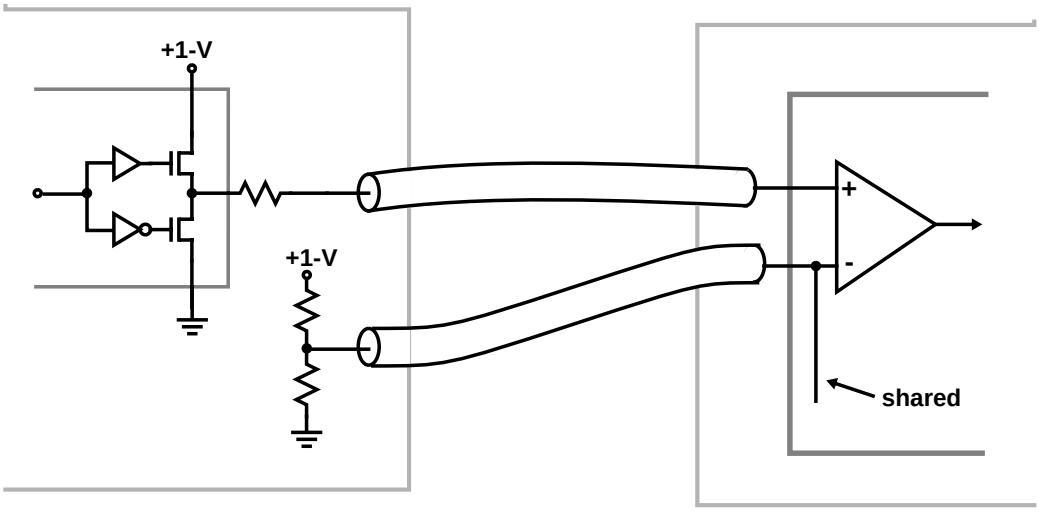

Figure 3.2: Interface signalling scheme.

receiver which uses a self-biased differential amplifier to decode the line signals. The following two sections will discuss in detail the design trade-offs considered in the design of the line driver and input receiver circuits.

3.1.1 Line Driver Design

As shown Figure 3.2 this interface uses a reduced-swing, low impedance driver with external series termination. This signalling scheme offers the advantage of zero static power dissipation for low signal transition rates. Moreover, in practical CMOS implementations, the push-pull series terminated driver used in this design will always result in a lower power dissipation compared to the alternative of an NMOS open-drain high impedance driver. Assuming a worst case transition density of 1, this signalling scheme dissipates a total (i.e. on-chip and off-chip) power of $P_{LI} = V_S^2/(2 \cdot Z_0)$, where V_S is the 1-V signal swing, and Z_0 the 50-Ω line impedance. In contrast, a signalling system using a high-impedance parallel terminated driver operating under the same conditions would dissipate a total power of $P_{HI} = (VDS + V_S) \cdot V_S/(2 \cdot Z_0)$, where V_{DS} is the voltage across the open-drain NMOS transistor. While V_{DS} can be made theoretically very small, practical considerations, such as the pre-driver voltage swing and the driver output capacitance, dictate the voltage across the NMOS open drain transistors to be at least equal to the voltage swing V_S. Thus, in point-to-point link environments, a series-terminated voltage-mode driver is more attractive than current mode drivers, in terms of power dissipation.

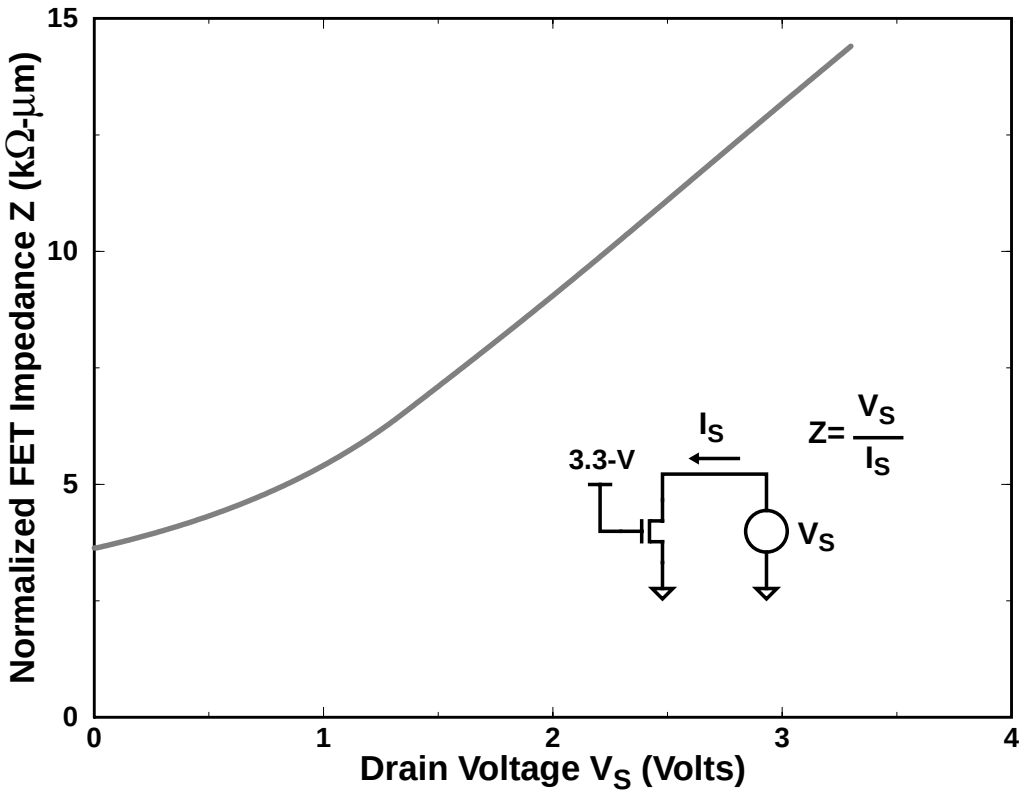

Figure 3.3: Large signal impedance of full-swing driver

Having selected a low impedance driver, the second parameter that needs to be determined is the signal swing. A lower signal swing is preferable to a full 3.3-V swing for several reasons. The first is power dissipation, since using a 3.3-V line swing increases the power dissipation of the signalling system by approximately ten-fold, when compared to a 1-V swing. Moreover, a larger signal swing does not necessarily result in increased noise margins. Given that a large fraction of the noise is self-induced *dI/dt* noise which scales linearly with the signal levels, reducing the swing means that the noise margins, as a fraction of the signal amplitude, stay approximately the same. The third reason dictating the use of 1-V swings in this design is the nonlinear nature of MOS transistors. Figure 3.3 shows the large signal impedance of an NMOS device with its gate tied to the 3.3-V supply while its drain voltage is varied between 0 and 3.3-V. The normalized large signal impedance of the driver varies between 3.7 kΩ-μm and 14.7 kΩ-μm. Ideally, the impedance of a voltage mode series terminated driver is required to be constant and equal to the small signal impedance over the output voltage range. This ensures that the magnitude of the initial voltage transition on the line is exactly equal to half the signal swing, and that

the half amplitude reflection from the far end of the line is completely absorbed by the line driver impedance, without causing standing waves on the line. Therefore a design alternative would be to size the driver so that its large signal impedance is approximately equal the line impedance at the half signal amplitude point. This would satisfy the requirement of a half swing initial transition, but it would result in standing waves on the line in case the far-end reflection returns to the source while the driver output has switched to the opposite value. Additionally, the MOS transistor nonlinearity results in a varying small signal impedance that affects the behavior of the driver with smaller amplitude reflections. For example, the small signal impedance of the driver in Figure 3.3 varies between 2.7 kΩ–μm and 223 kΩ–μm, resulting to potential standing waves from reflections caused by, e.g., transmission medium discontinuities. These effects suggest that if full swing signals are required, the only robust alternative is to size the driver so that its impedance is very low, and rely on an external termination resistor to make the source impedance linear and equal to that of the line. However the required large size of the driver transistors would result in prohibitively large dynamic power dissipation.

Using a push-pull NMOS driver with lower output swing is an attractive alternative, since it both increases the driver linearity and reduces the power dissipation without significant noise margin loss. The large signal impedance of a 1-V push-pull buffer, sized so that its mid-swing output impedance is approximately 50-Ω, is illustrated in Figure 3.4 (R=0-Ω). Although the pull-up impedance varies by only 2-Ω over the signal swing, the impedance of the pull-down varies by approximately 20Ω. Although the resulting percentage impedance variation is much less than that of a full swing driver, it would still result in a worst case 20% reflection, in case the source voltage happens to be pulled to one of the rails when a half amplitude reflection returns to the driver side. Additionally the small signal impedance of the pull-down driver varies between 41-Ω and 120-Ω, resulting in potential small-amplitude standing waves on the line.

To ameliorate the effects of nonlinear transistor impedance, this driver does not rely completely on the impedance of the push-pull transistors, but also uses an external series termination resistor. The push-pull transistors are sized to have a nominal impedance of 20-Ω, so a 30-Ω series termination resistor is required to drive the 50-Ω line. In this way,

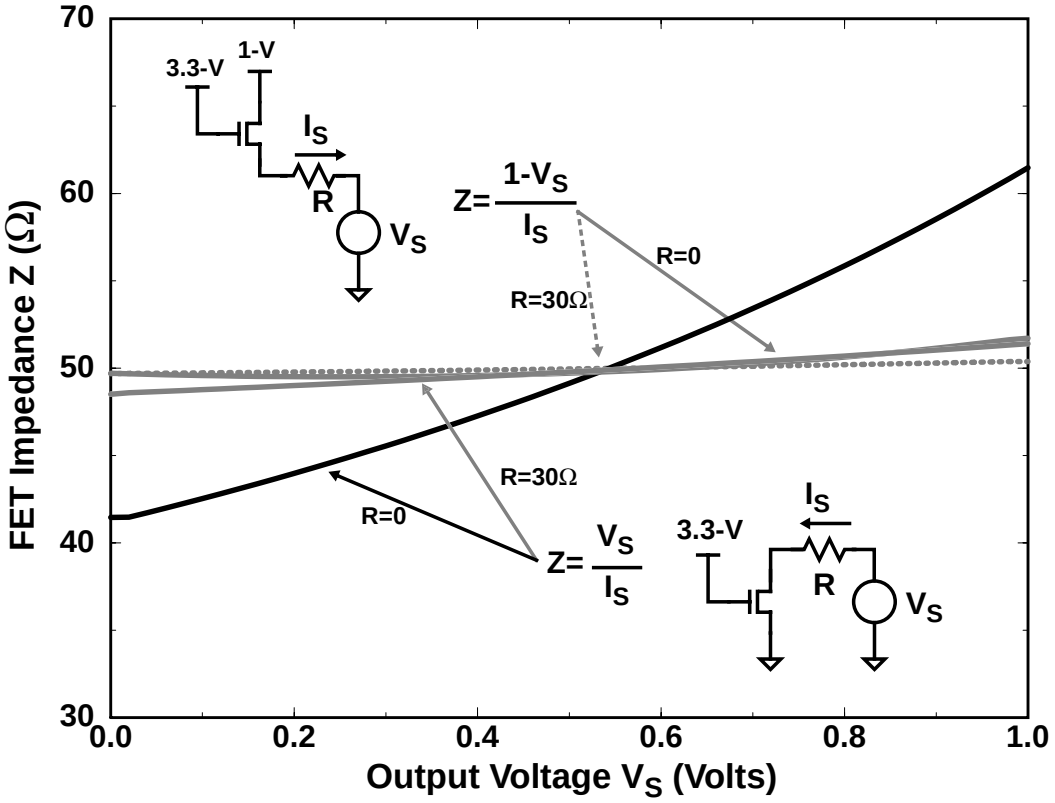

Figure 3.4: Large signal impedance of low-swing driver

the source impedance is largely determined by the fixed external resistor, minimizing the effect of transistor nonlinearity. Figure 3.4 shows the impedance of this alternative design (R=30 Ω). Using an external resistor minimizes the voltage variation across the driver transistors, reducing the maximum reflection due to transistor non-linearity from 20% to 0.75%. Additionally, the small signal impedance of this driver varies between 49-Ω and 50-Ω over the signal swing, minimizing the amplitude of discontinuity-induced reflections. Using an external termination resistor also mitigates the effects of process and environmental variations that cause the push-pull transistor impedance to deviate from its ideal value of 20-Ω. Simulation results indicate that the driver impedance can change from 14-28 Ω, over variations of the fabrication process and operating environment. The external termination resistor dampens the maximum source impedance variation to 16%. Although this method proved effective in this particular prototype, an active impedance matching scheme may be necessary if larger process, supply, and temperature variations need to be tolerated [30], [31]. Even in this case however an external resistor should be used to minimize the effects of transistor nonlinearity.

The power dissipation of this scheme depends on the electrical length of the wire and the switching frequency. When the time of flight through the wire is very short compared to the bit time, the buffer only dissipates dynamic power (1.22mW/100MHz). When the time of flight through the wire is longer than the bit time, the buffer sources 10 mA of current until the signal returns. This results in a worst case total power dissipation of 10mW+1.22mW/100MHz, while the on-chip worst case power dissipation is 4mW+1.22 mW/100MHz.

3.1.2 Receiver Design

The input receiver converts the 1-V swing input signal to a CMOS full-swing logic signal which is subsequently captured by a flip-flop triggered from the sampling clock of the receiver. To achieve high speed operation, the receiver must provide high gain for small signal deviations around the reference voltage. This design uses a variation of the “double mirror compensated” ECL to CMOS converter introduced in [40]. In order to accommodate the low common mode of the input signal, the design shown in Figure 3.5 uses a PMOS input differential pair with NMOS current mirror loads.

The biasing of the current source from the current mirror node enhances the amplifier gain. The mirror node d carries an attenuated version of the input signal, which, through the self biasing configuration, drives the tail current source transistor in a direction which increases the transconductance of the differential pair. Simultaneously, the self-biasing

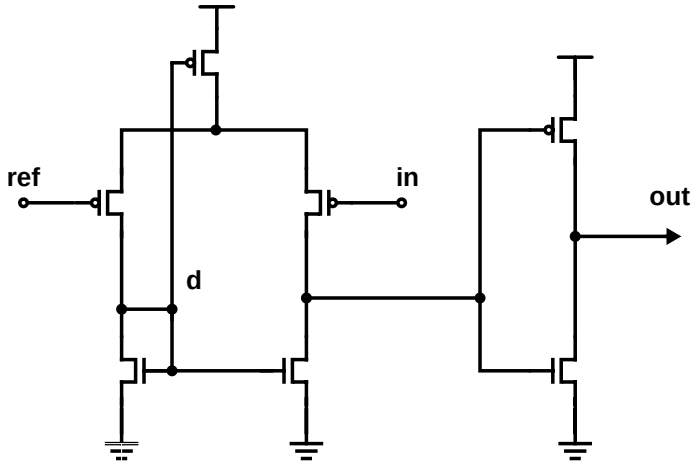

Figure 3.5: Input receiver schematic diagram

configuration provides common mode negative feedback, compensating for process shifts which would normally result in large input referred offsets. For example, a process with more conductive P relative to N devices that would cause the common mode of node d to rise would also increase the current through the amplifier causing that node to be pulled back down. Simulation results verify that the maximum DC amplifier offset over process and environmental variations is less than 30 mV. Moreover, this offset remains less than 72 mV even with additional 20-mV threshold voltage mismatches between nominally identical devices. The worst case small signal bandwidth of the amplifier was simulated to be 360-MHz, well beyond the design target operating frequency of 250-MHz.

3.2 Clocking Circuits

A simplified block diagram of the receiver is shown in Figure 3.6. The in-phase reference clock and data are first amplified to full swing CMOS signals by the input pin receivers. Subsequently, the amplified version of the reference clock is fed to the delay locked loop (DLL) of the receiver which phase shifts it by 90^o, and simultaneously buffers it up to drive the data receiving flip-flops. This configuration requires a minimum phase shift of the clock through the receiver's DLL of 90^o, which translates to a minimum delay require-

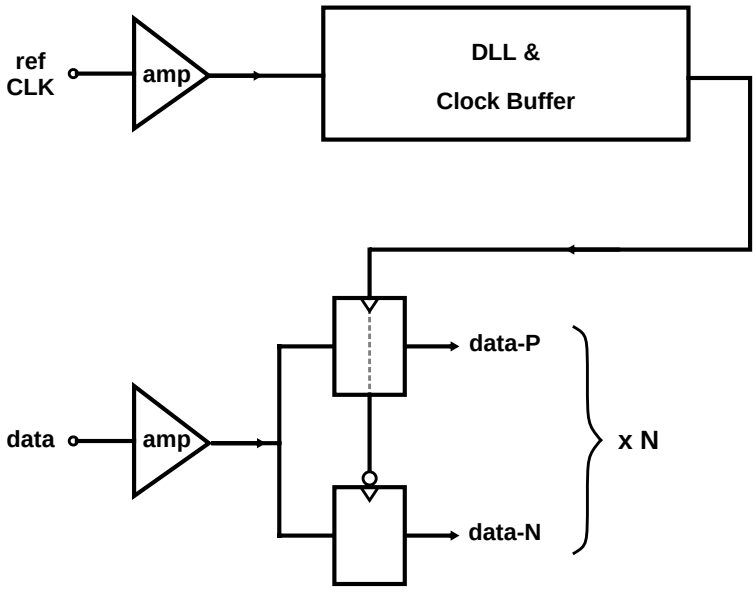

Figure 3.6: Simplified receiver block diagram

ment of 1 nsec for the 250 MHz operation target. Additionally, since the incoming data is sampled on both clock edges, the output clock duty cycle must be 50%.

3.2.1 Delay Locked Loop Design

The DLL consists of a delay line, a 90^o phase detector, and a single pole loop filter which consists of a charge pump and a 10-pF MOSFET gate capacitor. This design integrates the delay line with the clock buffer chain as shown in Figure 3.7. The delay line is implemented with shunt capacitor delay elements [41], [42]. By varying the control voltage V_{CTL}, the loop changes the inverter output time constant, thus affecting the phase shift of the input clock. The delay line comprises six shunt-capacitor delay elements, and a pair of conventional CMOS inverters which sharpen the final sampling clock edge. In order to achieve simultaneous phase shifting and buffering of the sampling clock, the delay line elements are gradually scaled up to drive the final output load. Thus, the delay line is also the clock buffer chain – the final buffer can drive the load presented to it by the 26 sampling flip-flops plus the wire capacitance across the receiver pad frame.

The delay line transfer function, simulated for typical process and operating conditions, is depicted in Figure 3.8. The total delay through the line varies between a minimum of 2-ns corresponding to control voltage values lower than an NMOS threshold, and a maximum of 7.9-ns corresponding to a control voltage equal to the 3.3-V supply. This voltage-to-delay transfer function does not satisfy the 1-ns/90^o minimum phase shift

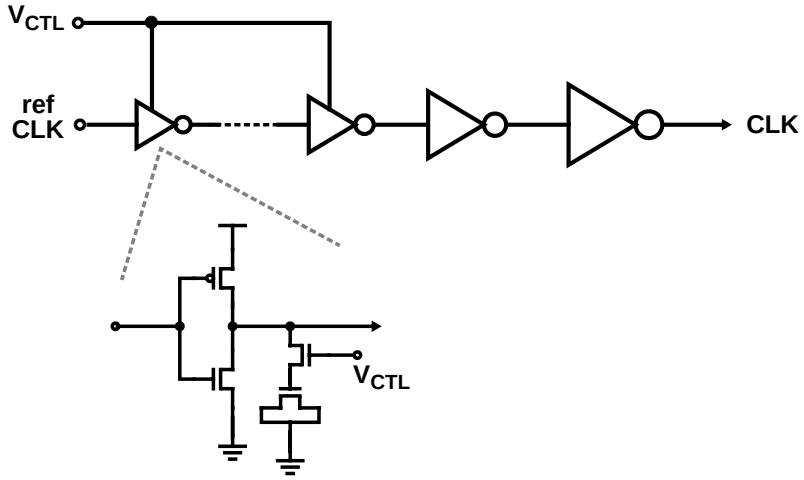

Figure 3.7: Delay line schematic

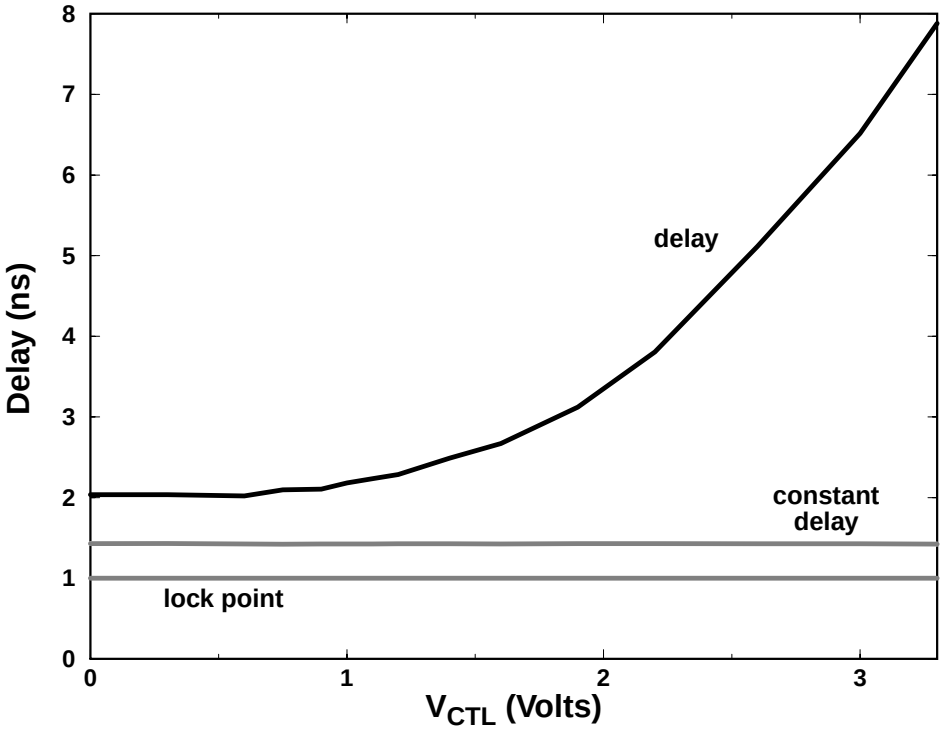

Figure 3.8: Simulated delay line transfer function

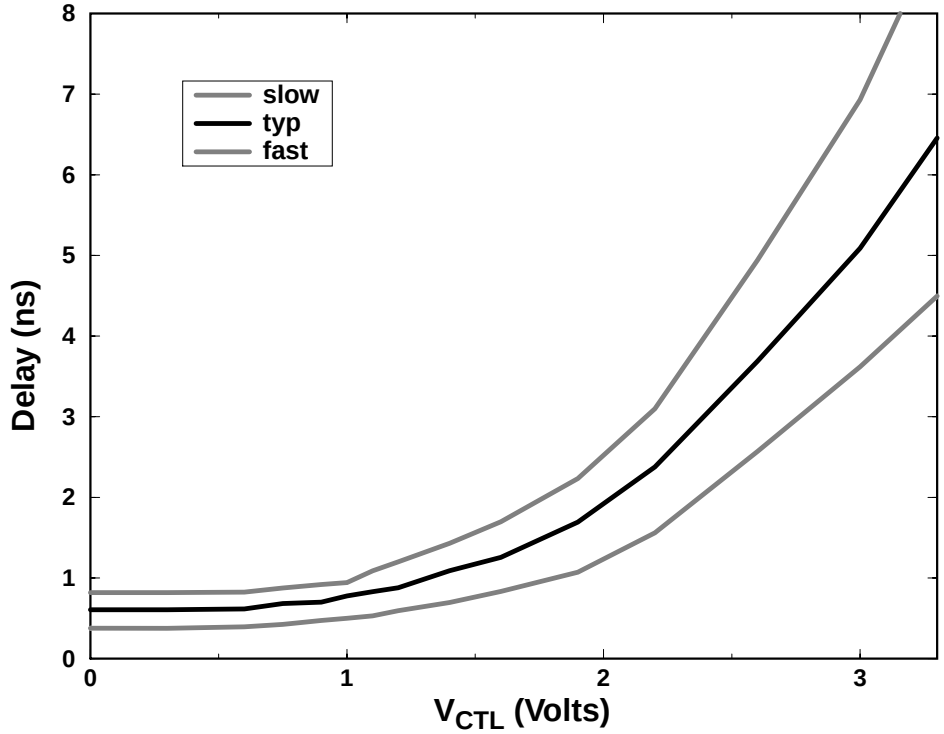

Figure 3.9: Simulated delay line effective transfer function

requirement for a 250-MHz sampling clock. The equivalent of 90^o phase shift with this design can only be achieved if the delay through the line is 5-ns – i.e. corresponding to *1+1/4* cycle of the 250-MHz target clock speed. However, in order to minimize clock jitter and duty cycle distortion it is desirable that the delay line operaties at a locking point closer to its minimum delay.

The straightforward solution to this problem would be to reduce the number of delay line stages, so that the minimum delay under the worst process and operating conditions is below 1-ns. Unfortunately, this solution would also decrease the delay line range. A solution that decreases the minimum delay through the line without affecting its tuning range is to subtract a constant delay from the delay line, effectively shifting down the curve of Figure 3.8 so that the minimum delay for low values of V_{CTL} is below the 1-ns requirement. This can be accomplished by delaying the reference clock and the input data through a buffer chain consisting of four inverters. The resulting delay line transfer function, simulated for three different operating and process conditions is depicted in Figure 3.9. Since the delay of the extra buffer chain tracks the delay of the delay line, this

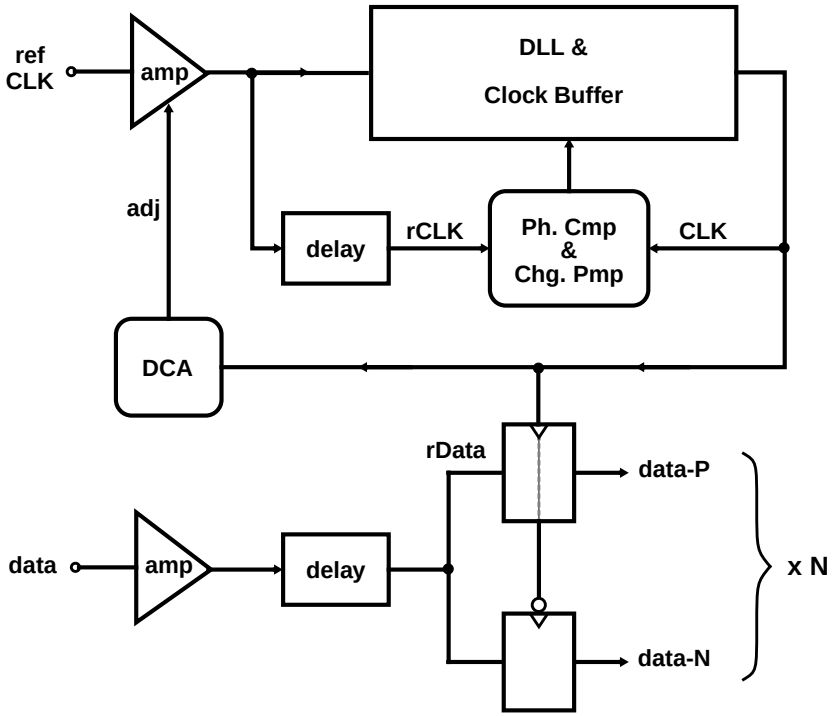

Figure 3.10: Detailed block diagram of the receiver

configuration achieves a 90^o locking point close to the line minimum delay for all the simulation corners.

A more detailed block diagram of the receiver incorporating these delay subtracting circuits is depicted in Figure 3.10. The delay subtracting buffer chain is placed both on the reference clock path and the incoming data path, in order to maintain their 0^o phase relationship. The output of the data-delaying buffer chain drives a pair of sampling flip-flops per input pin, while that of the reference clock drives the 90^o phase detector. The phase detector compares the delayed reference clock with the sampling clock, forcing them to be 90^o out of phase. This way the sampling clock is phase shifted by 90^o with respect to the input of the data sampling flip-flops.

A conceptual diagram of the 90^o phase detector and the associated charge pump is shown in Figure 3.11-(a). When the sampling clock CLK is phase shifted by 90^o relative to the reference clock rCLK, the output of the XOR gate is a clock waveform with double the frequency and 50% duty cycle. This causes the cycle-average current output of the charge pump to be zero, therefore maintaining a constant value of V_{CTL}. Any deviation in the relative phase of the two clocks results in a variation on the duty cycle of the detector output, which causes the loop to move the sampling clock towards 90^o point.

Since the output of the XOR gate in Figure 3.11-(a) needs to be twice as fast as the system clock, the phase detector and charge pump were implemented as shown in Figure 3.11-(b). Essentially this design integrates the XOR gate within the charge pump switching network, therefore reducing the capacitance that needs to be driven from the XOR pull-up or pull-down paths and increasing the maximum operating frequency. A common problem in all charge pump designs is the phase offset resulting from charge injection errors induced by the switch and current source transistor parasitic capacitances. To mitigate this problem in this implementation, the current source transistors are connected to the output node V_{CTL}. In this way, the control voltage is isolated from the switching noise induced by the gate-to-drain overlap capacitance of the switch transistors. Additionally the charge pump intermediate nodes charge towards the output voltage only by the amount allowed by gate overdrive of the current source devices, thus reducing

another source of phase error.

3.2.2 Duty Cycle Adjuster Design

In this design the receiver samples the input data on both clock edges. Therefore, any variations of the sampling clock duty cycle from its nominal 50% value reduces the interface timing margins. Since the DLL derives its output sampling clock from the upstream transmitter clock, any variations on the duty cycle of that clock will directly propagate to the data sampling clock. Additional duty cycle imperfections can be introduced by offsets in the clock receiving amplifier and the delay line. To alleviate this problem the DLL employs a duty cycle adjusting circuit, the block labeled DCA in Figure 3.10. This circuit is depicted in Figure 3.12. The voltage on node adj is generated by a charge pump similar to the one used to generate the main control voltage V_{CTL}. If the clock has any duty cycle

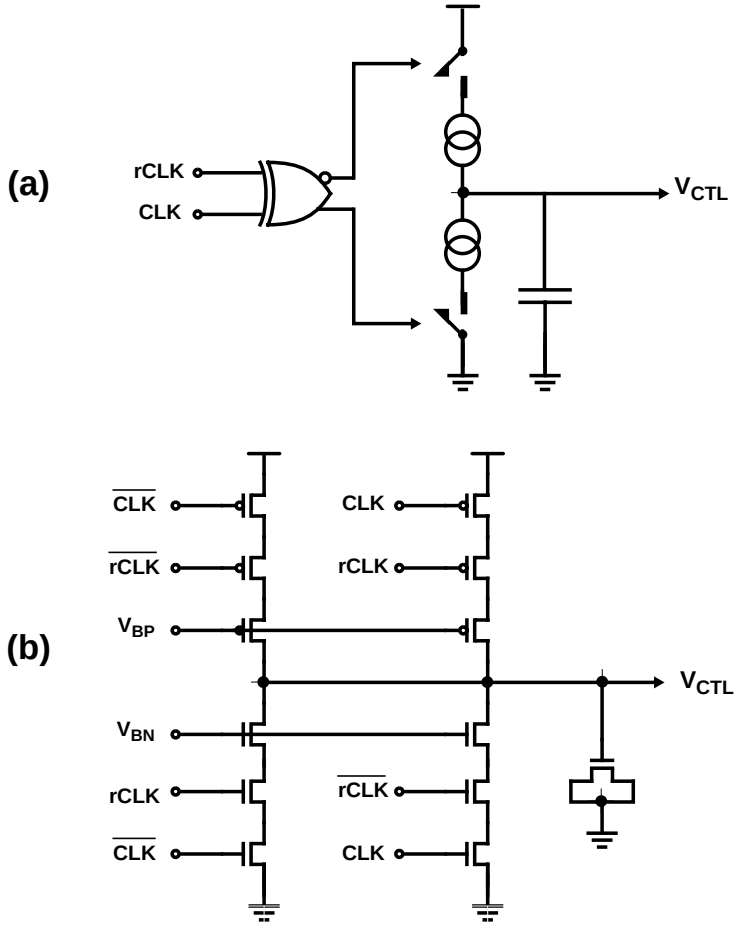

Figure 3.11: Phase detector: (a) conceptual diagram, and (b) implementation

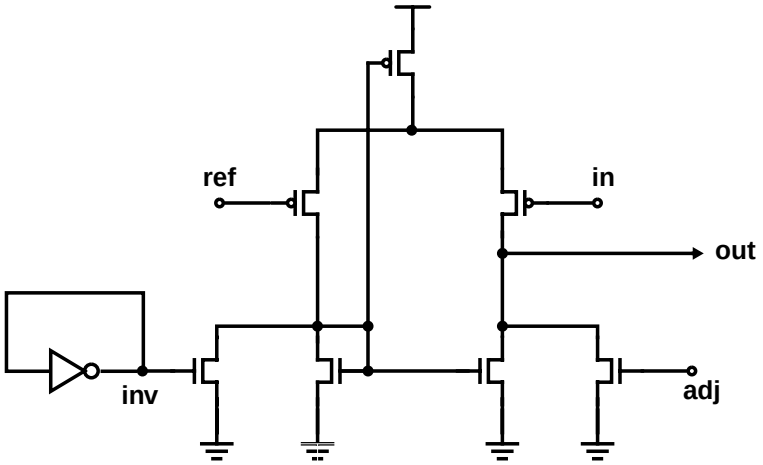

Figure 3.12: Duty cycle adjuster schematic

distortion, the extra charge deposited per cycle on the charge pump capacitor cause the voltage on node adj to slew linearly. The self-biased differential amplifier is identical with the input receiver discussed in Section 3.1.2, but it incorporates two additional transistors. These transistors are matched, so that the amplifier switches when the voltage in the input crosses the reference (given that V(adj)=V(inv)). If the duty cycle of the sampling clock is not 50%, moving V(adj) relative to V(inv) introduces an amplifier offset. Thus, given that the input has a finite transition rate, the negative feedback will drive the duty cycle of the output so that the sampling clock duty cycle is 50%. To match the delay in the data receiving paths with that of the reference clock path, the same circuit is used as the data receiving amplifier as well. However, in the data pins, node adj is connected to node inv so that no adjustment is made but the data delay tracks the reference clock delay.

The main weakness of this duty cycle adjuster circuit is that under changing process conditions it may introduce a systematic duty cycle error. The source of this error is that the voltage on node inv is not necessarily equal to that on node adj, even when the duty cycle of the clock input to the charge pump is 50%. This results because voltage V(inv) is mainly determined by the PMOS to NMOS threshold and device transconductance ratios. However, when the sampling clock duty cycle is 50%, the voltage on node V(adj) is determined by the PMOS to NMOS output conductance ratio. These two ratios do not necessarily track over varying process conditions, which might cause the DCA to introduce a systematic offset on the sampling clock. This problem can be easily eliminated if V(adj) is

generated by a charge pump replica in which both pull-up and pull-down current sources are permanently turned on. An alternative solution would be to drive the offset introducing nodes from the outputs of a differential charge pump such as that described in [43].

3.3 Experimental Results

To assess the performance of this design, a prototype chip was fabricated in the HP-CMOS26B process using the MOSIS scalable design rules. The 2×2 mm^2 die shown in Figure 3.13, contains two alternative DLLs, two data receivers and twelve line drivers, seven of which can transmit externally configurable patterns. The chip was packaged in a 40-pin ceramic dual-in-line (DIP) package. Although this package design is certainly not oriented for high speed applications, it was chosen since it simplified the bonding procedure and the test board construction. To alleviate the effects of the large pin inductance,

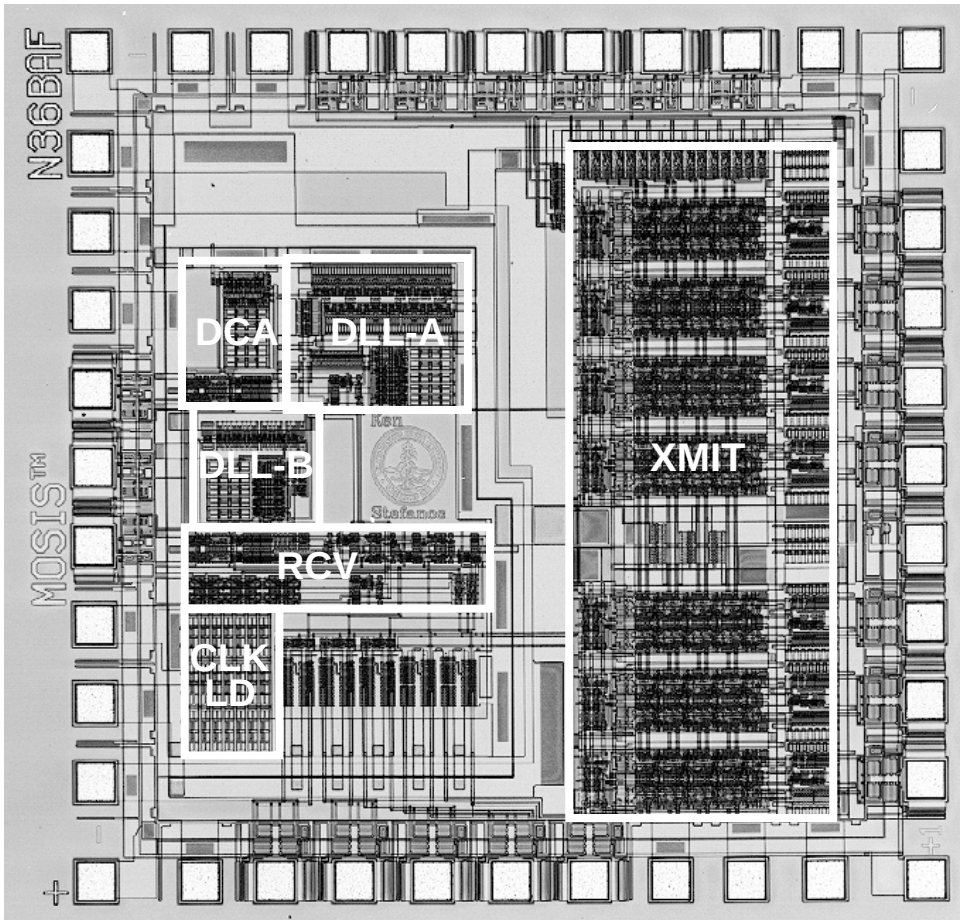

Figure 3.13: Chip photomicrograph

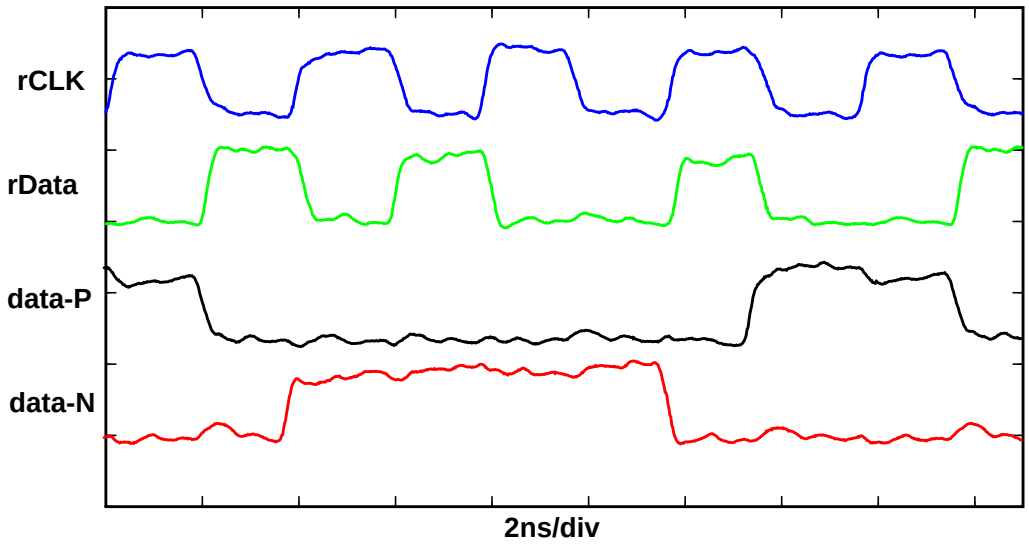

Figure 3.14: Received signal waveforms at 500 Mbps/pin

the high speed receiver signals were routed through the lower inductance (~10 nH) pins at the center of the package. To assist the testing process, the fabricated chips can be configured through a scan chain to use either the circuit of Figure 3.5, or a conventional CMOS inverter as the input data or clock receiver[1]. The test set-up used two printed circuit boards for the transmitter and the receiver chips. The boards were connected through 1-m long coaxial cables, which carried the data and reference clocks, as well as the reference voltage generated on the transmitter board. Since no active bit error rate measurement system was available, the bulk of the of the testing was done by relying on a 30-GHz sampling rate oscilloscope.

Figure 3.14 shows the waveforms of the reference clock, the incoming data, and the sampled data at the 500 Mbps/pin operating point. In this experiment the transmitter sends an 8-bit long “10100100” pattern, which is sampled correctly by the receiver flip-flops. The operating frequency was increased up to 340 MHz without any indication of error appearing, while the oscilloscope was in continuous accumulation mode. The same experiment was repeated with different data patterns at various operating voltages and transfer

1. The chips contained a layout flaw, which caused the scan chain to always select the inverter as the input receiver. After the error was repaired with laser-cutting, by setting the scan chain flip-flop values so that the self biased amplifiers were always selected, the testing of the complete interface was possible.

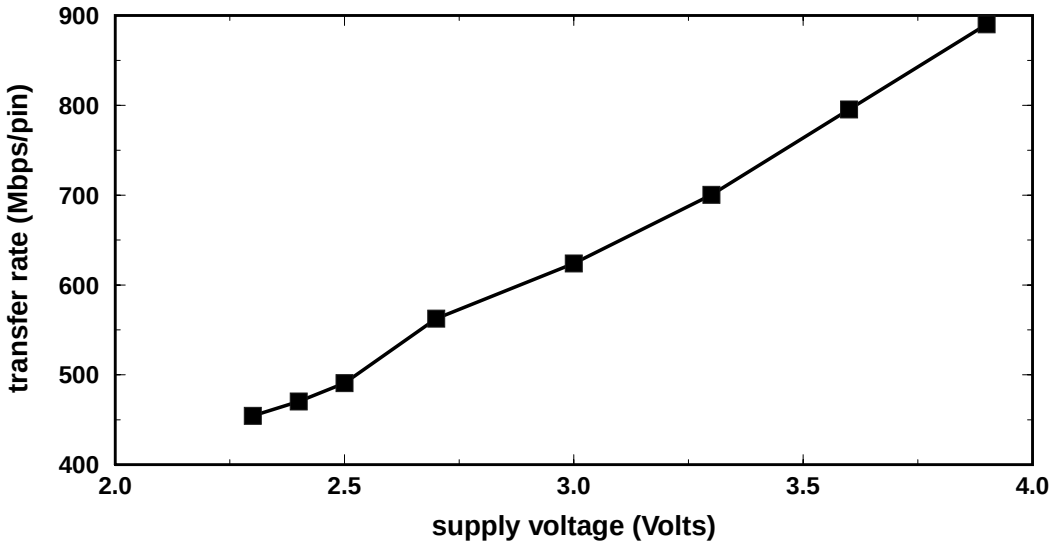

Figure 3.15: Prototype operating range

rates. The resulting curve of maximum “error-free” transfer rates versus the corresponding supply voltage is depicted in Figure 3.15.

The sampling uncertainty window of the receiver was evaluated by altering the relative skew between the reference clock and the data through variable transmission lines, and observing the point of failure on the oscilloscope screen. With this procedure the sampling uncertainty window (i.e., the bit time minus the timing margin) was measured to be 400 ps around the center of the bit.

The effectiveness of the DCA circuit was also measured by varying the duty cycle of the input clock and measuring the duty cycle of the sampling clock. Fortunately this particular fabrication process did not expose the design weakness of the circuit described in Section 3.2.2. The experimental results show that reference clock duty cycles of 40%-60% can be corrected by the circuit, resulting in a sampling clock duty cycle close to 50% [15].

The sampling clock jitter under quiet conditions was measured to be 120 ps. When nine line drivers on the receiving chip are switching simultaneously, but non-synchronously with the clock pattern, this jitter increases to 240 ps.

3.4 Summary

This chapter examined the design of a high-speed interface intended for use in multiprocessor interconnection networks. This interface demonstrates that adopting a source-synchronous point-to-point architecture can help achieve high data-rates at relatively low circuit complexity. The simple pseudo-differential signalling scheme used by this design can reduce system interconnect cost and minimize system power dissipation. Additionally, the push-pull series terminated drivers discussed in Section 3.1.1 address effectively problems associated with transistor nonlinearity. Finally, many receiver synchronization problems can be easily addressed by using a narrow range DLL, which integrates the delay line within the clock buffer chain, as discussed in Section 3.2.1.

This relatively simple interface has some important limitations which are common among similar design approaches. The first limitation stems from the input pin receiver design. Although using an amplifier and delaying the data along with the clock helps extend the DLL range, it simultaneously increases the receiver's uncertainty window and limits the interface maximum speed. This is because the limited bandwidth of the front-end amplifier and the subsequent data-delaying buffer chain increases the inter-symbol interference and timing uncertainty. Sense amplifier designs can achieve higher speeds by combining the amplification and latching operation in a single stage, as has been demonstrated in A/D converter designs [44]. Even in this case, however, the interface would have to address a more fundamental limitation stemming from the fact that data is sampled only once per bit-period. Undoubtedly, the quadrature positioning of the sampling clock relative to the incoming data maximizes the timing margin of the input pin receiver. However, noise that might occur during the sampling time can cause the sampling and amplifying circuit to resolve the wrong value. As will be discussed in Chapter 4, one of the largest sources of noise comes from the asymmetric nature of pseudo-differential signalling. The increased capacitive coupling of the shared reference voltage transforms high frequency noise on the receiver's supplies to differential-mode noise at the inputs of the receiving amplifiers. A method that filters this high frequency noise would enable pseudo-differential signalling to be used even in environments more hostile than the experimental prototype described in this chapter.

Another limitation of this approach emanates from ignoring many second order effects in the design of the clocking circuits. Although, as will be seen in Chapter 5, the use of a DLL minimizes long-term jitter accumulation, the delay sensitivity of the simple shunt-capacitor delay element used in this design can significantly decrease the receiver's timing margins. The delay of a shunt capacitor delay element can change by as much as 15% with a 10% change in the supply voltage. This means that in large digital chips with significant amounts of supply or substrate noise the sampling clock jitter can be the main limitation of the maximum transfer rate. Moreover, additional sampling clock jitter can be introduced by the fact that the DLL's output clock is derived directly from the noisy reference clock. Thus, the on-chip sampling clock carries the delay uncertainty introduced on the reference clock by the transmitter supply variations, plus any additional delay uncertainty introduced by amplifying and buffering the reference clock on the receiver chip. Employing differential delay elements, along with using a separate clock as the input to the delay line is an effective solution employed by many designs. However, it introduces additional problems which stem from the finite delay range of conventional DLL architectures and the fact that the two clocks might have an unknown phase relationship. In order to eliminate this problem, a DLL with unlimited delay range is necessary.

The following two chapters present receiver and clocking circuits which effectively deal with the limitations described above. Chapter 4 will discuss a receiver design which averages the data during its valid-time period by employing a current-integrating front-end. This integration of the data makes the reception of the signals insensitive to high frequency supply or reflection noise, and allows pseudo-differential signalling to be used even with hostile on-chip noise conditions. Chapter 5 describes a DLL which uses a dual-loop architecture to achieve unlimited phase shift capability and low jitter. In addition, the use of digital control enables the implementation of complicated phase acquisition algorithms which can be effectively used in higher speed interfaces to remove intra-bit skew.

Chapter 4

Current Integrating Receivers

Receiver structures play a crucial role in determining the performance and robustness of high speed interfaces. As discussed in Chapter 3, the most common approach employed in the reception of source synchronous signals is to position the receiver clock in quadrature with the transmitted data, and resolve the value of a single sample of that data per transmitted bit. This quadrature positioning of the clock maximizes the timing margin of the input pin receiver, and compensates for the skew that might be present between the reference clock and the data. The receiver structure may either consist of a front-end amplifier followed by a latch (similarly to the design discussed in Section 3.1.2), or it may alternatively combine the amplification and latching operation into a single stage sense amplifier. Regardless of the specifics of the receiver implementation, this conventional single sampling approach has a significant drawback: noise that might occur during the sampling time can cause the receiver to resolve the wrong data value, degrading the interface performance. As discussed in Chapter 2, the sources of noise in the environment of a digital chip are numerous. Additionally, their effect becomes more pronounced in pseudo-differential interfaces in which high frequency on-chip noise couples more heavily to the reference voltage, degrading the receiver noise margins.

This chapter presents a receiver design which overcomes the noise sensitivity problem of conventional single-sampling receivers. By integrating current on capacitors based on the incoming data and resolving the received data value based on the integrated voltage polarity, this design filters high frequency noise and improves the reception robustness.

This chapter begins by a detailed discussion of the problem of reference noise in pseudo-differential interfaces. The next two sections focus on the proposed current integrating receiver. Section 4.2 presents the concept and operation of an ideal current integrating receiver, and discusses its performance in the presence of high frequency noise. Section 4.3 focuses on the circuit implementation details of current integrating receivers, by discussing various current integrator topologies along with their associated biasing circuits, and the succeeding amplifying and latching stages. The design of a complete source synchronous interface using current integrating receivers is presented in Section 4.4, while the experimental results measured on a fabricated prototype transceiver chip are discussed in Section 4.5.

4.1 Reference Noise in Pseudo-Differential Signalling

As described in Chapters 2 and 3, pseudo-differential signalling is a compromise between the increased cost of fully differential signalling and the higher power dissipation and poor noise properties of conventional full swing single ended signalling systems. The use of a reference to indicate the common mode value of the transmitted signals enables pseudo differential signalling systems to reduce the transmitted signal amplitude, therefore decreasing both the power dissipation and the noise induced by the interface circuits. Since the reference voltage is shared among all the signals of a parallel interface, these benefits come at no significant interconnect cost increase. However, using a shared reference introduces a fundamental impedance asymmetry, which reduces the common mode rejection of the system at higher noise frequencies and limits the achievable interface bandwidth.

Figure 4.1 shows a simplified model of the reference and a single signal line on a parallel pseudo-differential interface. The transmission line and driver impedances are modeled with the lumped resistor R_D. This impedance is in series with the package pin, which is modeled with the parasitic inductance L_P. Both the data and reference inputs are capacitively coupled to the substrate of the receiver chip (V_{SS}). This simple model remains valid even when the inputs are directly coupled to some other on-chip node. The reason is that all nodes on a chip are directly or indirectly coupled to the chip's supply rails and during

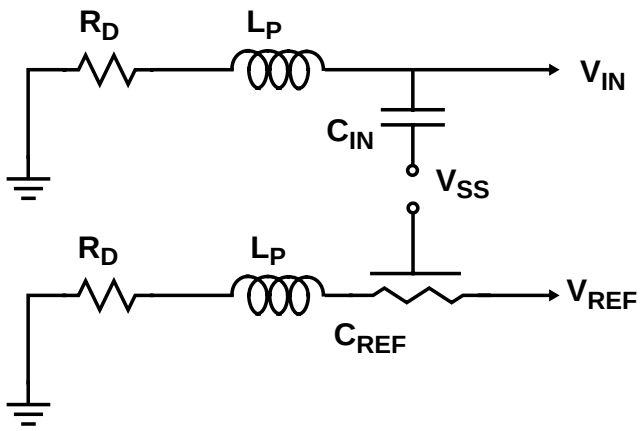

Figure 4.1: Simplified noise injection model in a pseudo-differential interface.

on-chip noise transients they move relative to the off-chip power lines. The noise coupling path of both the input and reference lines consists mainly of the capacitances of the input pad, the electrostatic discharge protection (ESD) circuitry, and the input receiver circuit. Since the reference line drives a multitude of input receivers along with a longer routing wire, its coupling capacitance is significantly larger than the capacitance of the input line. Moreover, depending on the length and the resistivity of the reference routing wire, the additional capacitance of the reference line may behave as a distributed *RC* line.

This capacitive coupling results in noise injection from the chip's power supply rails to the input and reference lines. When the capacitive coupling of the input and reference lines is equal, as is the case in a fully differential interface, the noise injection is common mode and does not affect the reception of the signals. However, in a pseudo-differential interface the noise injection on the reference line is fundamentally larger than that on the input lines. This results in a reduction of the common mode rejection for high noise frequencies. This effect can be quantified by analyzing the simplified model of Figure 4.1. The transfer function from node V_{SS} to nodes V_{REF} or V_{IN} can be shown to be

$$\frac{V_X(s)}{V_{SS}(s)} = \frac{s^2 \cdot L_P \cdot C + s \cdot R_D \cdot C}{s^2 \cdot L_P \cdot C + s \cdot R_D \cdot C + 1} \tag{4-1}$$

where V_X can be substituted by V_{IN} or V_{REF} and C can be correspondingly substituted by C_{IN} or C_{REF} (for simplicity Equation (4-1) ignores the effects of the distributed reference

wire resistance). The transfer function of Equation (4-1) exhibits a high-pass peaked behavior which is illustrated in Figure 4.2. The difference on the peaking frequency of the input and reference transfer functions is a result of the capacitive coupling mismatch between the two inputs. For the slightly optimistic values used in Figure 4.2 (R_D=50Ω, L_P=2nH, C_{IN}=0.5pF, C_{REF}=1-2pF) the magnitude and phase difference is maximized for V_{SS} noise frequencies around 2-GHz. In an interface operating at 0.5-1 Gbps/pin, this frequency band is occupied by noise associated with the transitions of the on-chip clocks. This type of high frequency noise might prove to be detrimental for a receiver design which phase-shifts the clock by 90° and samples the data once per valid bit period. The reason is that a reference voltage "glitch" can coincide with the sampling time instant, causing the receiver to resolve the wrong value of the input signal.

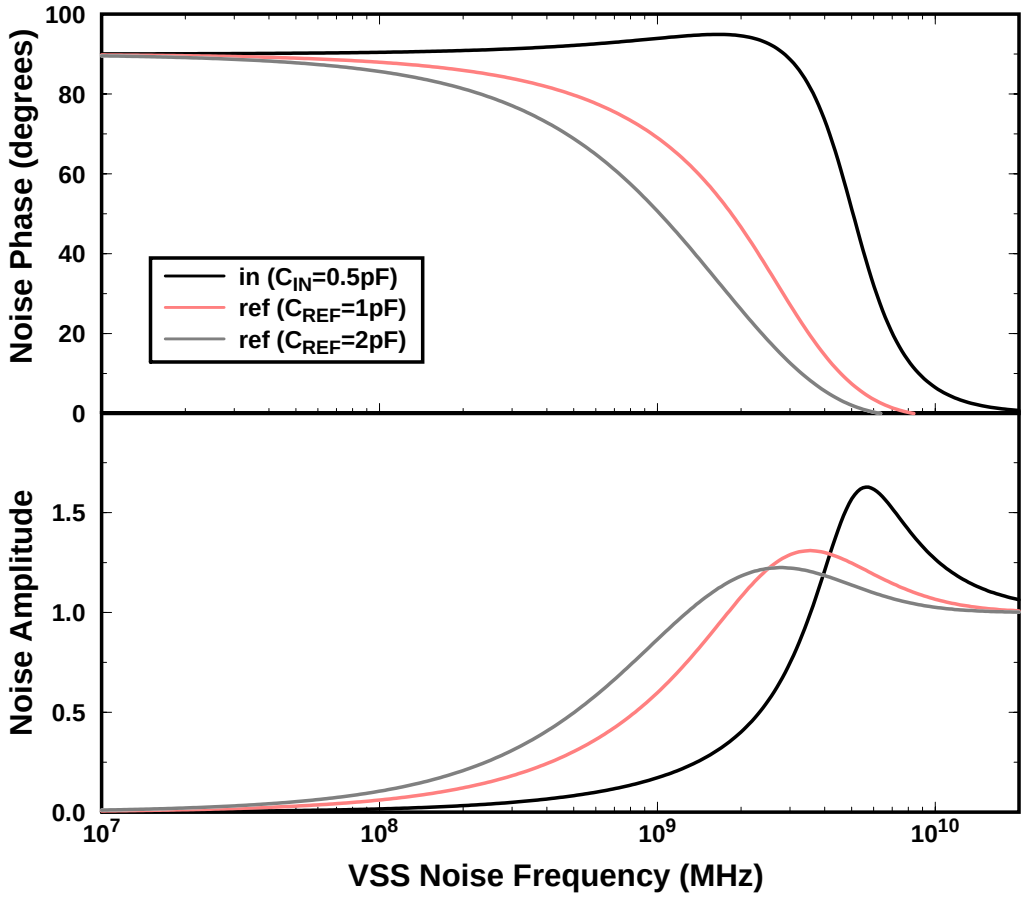

Figure 4.2: Reference and input signal frequency response to on-chip supply noise

The reference noise problem becomes worse in the case of a bidirectional interface which superimposes the transmit and receive signals on the same transmission line [28], [29], [45]. In bidirectional interfaces the decoding of the resulting ternary signal on the transmission line is accomplished by multiplexing two different reference voltages at the input of the receiver. This reference multiplexing is controlled by the local transmitter output. In this type of design, in addition to the noise injected on the reference line by its larger capacitive coupling to the chip's supply rails, the receiver circuit has to cope with potential glitches on its reference input. These glitches can be introduced by imperfect tracking of the delays through the output driver and the reference voltage multiplexer.

Solving the reference noise problem by increasing the signal swing is only partially effective, since the magnitude of V_{SS} noise is largely proportional to the signal swings. The alternative solution of employing fully differential transmission comes at the expense of additional interconnect pins and wires. Additionally, fully differential signalling does not address the reference voltage glitching problem in bidirectional links, and therefore it is effective in eliminating noise only in unidirectional designs. A cost effective solution to the high frequency noise problem would be to ensure that noise does not occur during the receiver sampling time, e.g., by restricting the switching of the output drivers to occur at a time instant different from the input pin sampling instant. In bidirectional interfaces this solution can be implemented by restricting the line delay to be a very small fraction or a fixed multiple of a bit time. In general, achieving quiet conditions during the input signal sampling time might be impossible to implement in systems in which switching activity is not entirely related to the high speed interface transmit and receive clocks. For example this is a common situation in IC's where power supply noise due to internal circuits dominates, or in IC's where more than one high speed interfaces with unconstrained phase relationships between their clocks are integrated on the same die. Despite these disadvantages designers are often forced to implement fully differential interfaces or restrict the line delays [28], [36], [46]. As the next section argues, however, an effective solution can be achieved if one realizes that in intra-system signalling interfaces, where line bandwidth limitations iare not a concern, the input signals are valid for more than the brief sampling period of the input receiver.

4.2 Concept of Operation

As was described in the previous section, the reference noise frequency is higher than the bandwidth of the interface signals or the transmission medium. This suggests that the noise problem can be mitigated by employing some form of filtering. This filtering can be performed either in the digital or analog domain.

A digital filtering implementation would require sampling the incoming data more than once per bit period, and subsequently using a digital majority voting scheme to determine the value of the transmitted data. Since high frequency noise would only affect a minority of the sampled values, its effects would be greatly attenuated, thus improving the signalling system performance. The main disadvantage of such a solution is that the required power and area grow linearly with the required number of samples: a minimum of three samples per bit period would increase the power dissipated on the sampling clock and the area occupied by the input pin sampler by at least a factor of three. An additional disadvantage is that it requires the multiple clock sampling edges to be positioned evenly across the incoming bit time and exhibit low intra-edge jitter. Therefore noise that would affect the value of the sampled data can also potentially affect the relative position of two sampling clock edges, thus degrading the effectiveness of the majority voting strategy.

A straightforward analog filtering scheme would be to use a single pole *RC* filter to dampen the noise on the reference line before driving a conventional sampling receiver. Unfortunately, in order for this scheme to be effective, the bottom plate of the filter capacitor should be isolated from on-chip noise transients. Since this requires a low impedance connection of that bottom plate node to the off-chip reference voltage line, this solution is essentially equivalent to fully differential signalling. An alternative and more economic solution is to implement the analog equivalent of majority voting using a receiver that integrates current on a capacitor based on the incoming signal polarity $sgn(V_{IN}-V_{REF})$. At the end of the integration period, which in this scheme should coincide with the input bit valid time, the receiver can determine the value of the transmitted datum based on the polarity of the voltage on the integrating capacitor. Figure 4.3 shows an idealized diagram of a current integrating receiver. This ideal receiver consists of a current switch, a pair of load

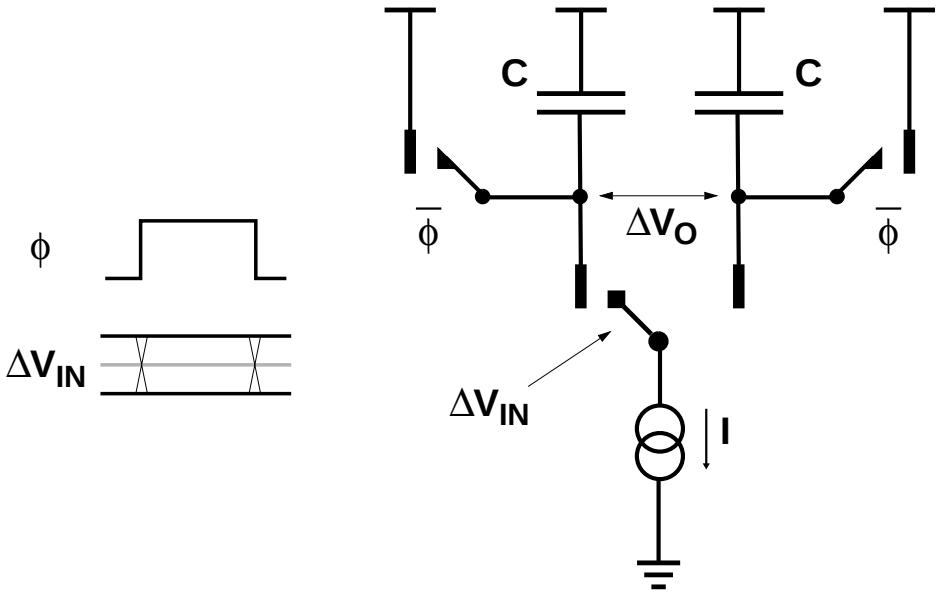

Figure 4.3: Ideal current integrating receiver

capacitors and two reset switches. The level of clock phase ϕ indicates the input data-valid period. When ϕ is low the switches are closed, equalizing the integrator output. When ϕ transitions to its high level, the switches open and the current switch steers current to the one branch of the integrator or the other, depending on whether the input is higher or lower than the reference voltage. At the end of clock phase ϕ the polarity of the output differential voltage ΔV_O indicates whether the input signal was mostly low or high during the integrating period. Its value is given by:

$$\Delta V_O(t_b) = \frac{I}{C} \cdot \int_0^{t_b} \mathrm{sgn}[V_{IN}(t) - V_{REF}(t)]\,dt \qquad (4\text{-}2)$$

Any noise transients that would cause the input to cross the reference voltage do not affect the correct reception of the signal, as long as the duration of the noise transient is less than half the integration period. Such noise transients affect only the magnitude but not the polarity of voltage ΔV_O. Therefore, as long as the final value of ΔV_O is larger than the offset of the amplifier following the first integrating stage, the correct reception of the input signal will not be affected.

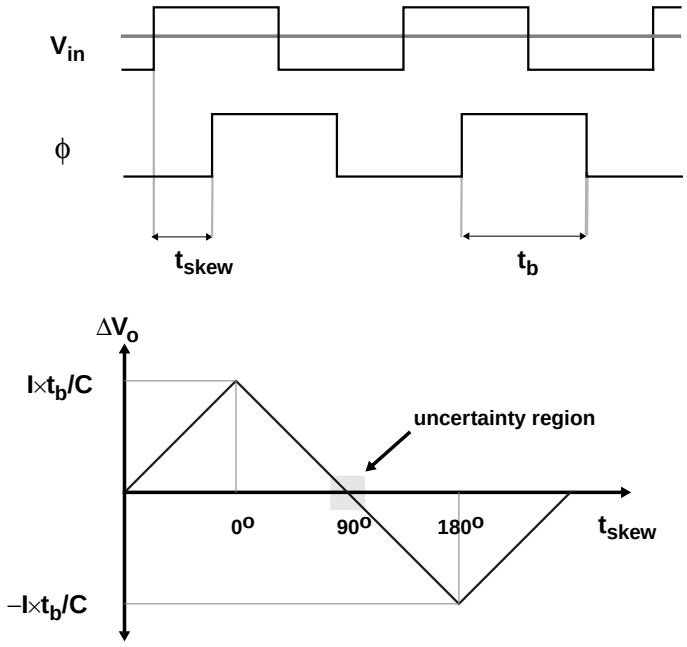

Figure 4.4: Ideal current integrating receiver phase characteristics

Figure 4.3 shows the behavior of the ideal integrator in the presence of a phase offset between the integrator sampling clock and the incoming data. In this conceptual experiment the input data is a clock waveform of the same frequency as the sampling clock. The phase shift between the data and the sampling clock varies from 0^o to 360^o and the value of ΔV_O is plotted versus the corresponding phase shift. The resulting curve has a triangular shape, which is identical to the phase-to-voltage gain curve of a quadrature phase detector. When the input signal and the sampling clock are in phase or 180^o out of phase, the switch current is dumped in only one of the integrating capacitors for the full bit duration. Therefore in that case the differential output voltage ΔV_O has its maximum or minimum value of $\pm (I \times t_b)/C$, where I is the switch current, t_b is the bit time (half the clock period), and C the integrating capacitor size. When the input signal and the sampling clock are in quadrature, the switch current is dumped evenly on each of the two integrating capacitors for half the bit time and ΔV_O ends up being zero.

For the ideal integrating receiver of Figure 4.3 the zero crossing on the phase-shift axis in Figure 4.4 is equivalent to the center of the sampling uncertainty window of a conventional sampling receiver. In the ideal receiver this center of the *setup+hold* time is located

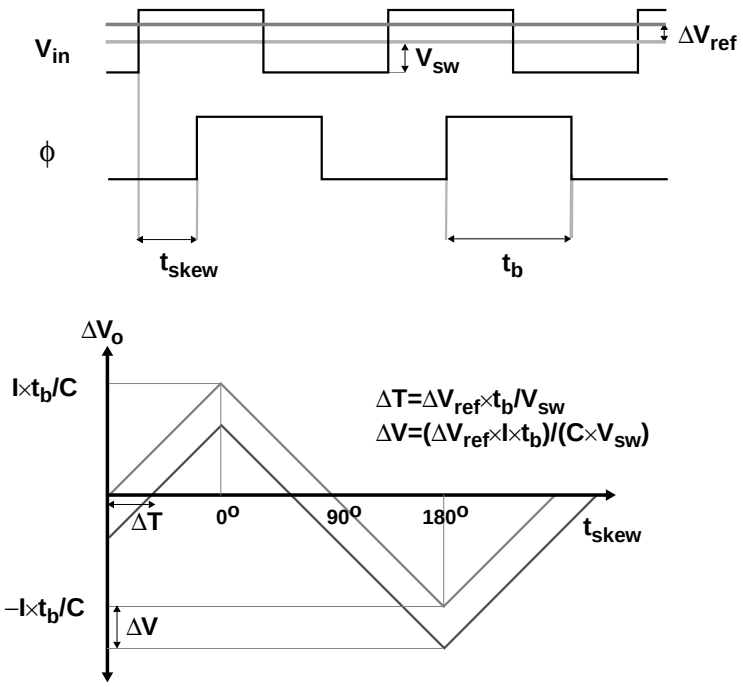

Figure 4.5: Integrate-and-dump filter phase characteristics

in the center of the bit time. However, in a real implementation the center of the sampling uncertainty window deviates from the ideal 90^o point due to offsets in the current switch and the following amplifying stage. Any such deviation of the sampling uncertainty window center subtracts from the signalling system timing margin and should be minimized.

Minimizing the *setup+hold* timing uncertainty is the main reason that the integrating receiver described in this section differs from an integrate-and-dump filter (i.e., a square pulse matched filter detector)[47], [48]. A current integrating implementation of an integrate-and-dump filter would integrate a current proportional to the actual input differential voltage V_{IN}-V_{REF}, rather than a current proportional to the input voltage polarity *sgn*(V_{IN}-V_{REF}) as is the case with the receiver described above. This results in an output voltage:

$$\Delta V_O(t_b) = \frac{g_m}{C} \cdot \int_0^{t_b} [V_{IN}(t) - V_{REF}(t)]\,dt \qquad (4\text{-}3)$$

where g_m is the transconductance transforming the input voltage to an integrating current. In order to compare this design with a current integrating receiver, we assume that the

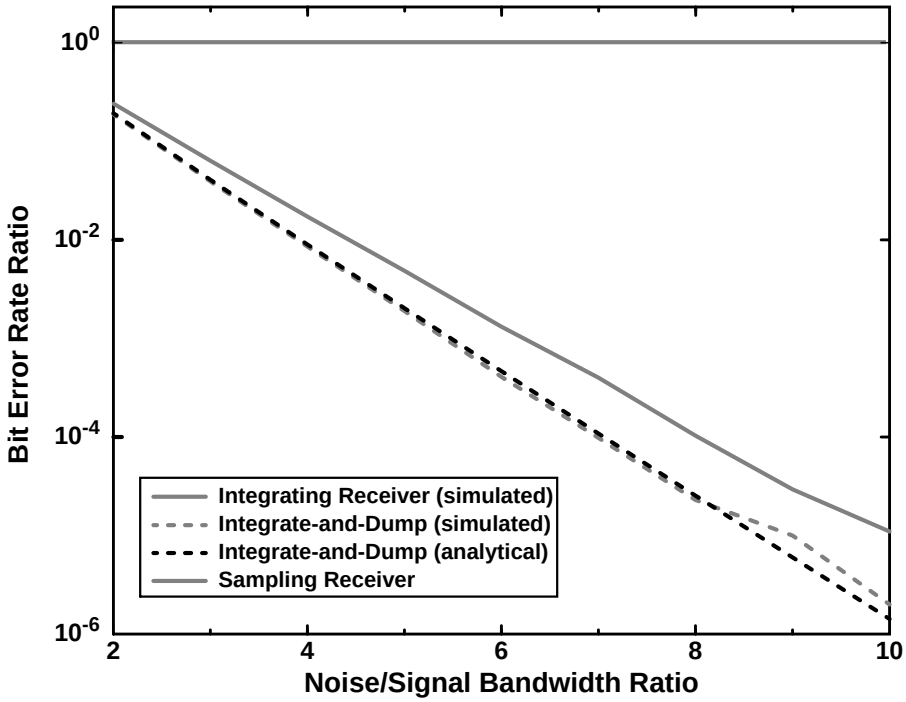

Figure 4.6: Performance comparison of integrating receiver and integrate-and-dump filter

maximum output voltage of the integrate-and-dump is identical with that of the current integrating receiver of Figure 4.3 (i.e., $g_m=I/V_{sw}$ where V_{sw} is the input signal swing). In case the DC value of the reference voltage is in the center of the input signal swing, the phase characteristics of the two designs are identical. However when the DC value of the reference voltage deviates from its nominal value, this deviation will create a proportional displacement of the zero crossing on the phase shift axis as illustrated in Figure 4.5. Since this displacement subtracts directly from the receiver timing margins, the current integrator of Figure 4.3 is preferable in pseudo-differential implementations, especially when the phase offset and jitter of the sampling clock are a concern.

The current integrating receiver, being a modification of the integrate-and-dump detector, will exhibit suboptimal performance in the ideal case where there are no deviations of the DC value of the reference voltage. Figure 4.6 quantifies this performance penalty under the assumption of white Gaussian amplitude noise with a bandwidth higher than the signal bandwidth (as is the case with reference noise in pseudo differential interfaces), and no sampling clock offset or jitter. In the plot of Figure 4.6 the horizontal-axis is the ratio of the noise bandwidth to the signal data rate (the noise bandwidth is scaled such that the

total noise power, and consequently the signal to noise ratio, remain constant). The vertical axis is the ratio of the bit error rate of the current integrating receiver or matched filter detector to the bit error rate of a single sampling receiver. Due to the non-linear nature of the current integrating receiver no accurate analytical model can be developed, and thus only simulation results are being plotted. To verify the simulator accuracy, both analytical and simulation results are plotted for the integrate-and-dump detector. Figure 4.6 shows that, due to the averaging effect of the integration, both the integrate-and-dump and the current integrating receiver exhibit superior performance to the single sampling receiver with increasing noise bandwidth. When the bandwidth of the noise is approximately ten times larger than that of the transmitted signal, the integrate-and-dump detector has approximately a decade lower bit error rate than the current integrating receiver. When the noise bandwidth is in the neighborhood of four times that of the signal, as might be the case for a 500 Mbps/pin interface, the performance difference is much less pronounced. These results indicate that an implementation closer to the integrate-and-dump detector might be the preferable solution in systems which do not suffer from unequal high/low pulse amplitudes, such as fully differential interfaces.

The following section discusses the implementation of a current integrating receiver which approaches the behavior of the model in Figure 4.3. While that section does not specifically address the design of an integrate-and-dump filter, this is a much easier design problem. Moreover, due to the finite transconductance of MOS transistors, the receiver described in the following section approaches the behavior of an integrate-and-dump filter with reduced input signal swing.

4.3 Circuit Design

In a high speed parallel interface the output of the input pin receiver needs to be a full swing CMOS signal which is held stable for a full clock cycle. Since a CMOS implementation of the rudimentary current switch described in the previous section does not satisfy the above requirements, a complete current integrating receiver utilizes some additional circuits. Figure 4.7-(a) shows the block diagram of such an implementation. The complete receiver consists of a front-end current integrating stage followed by a sample-and-hold

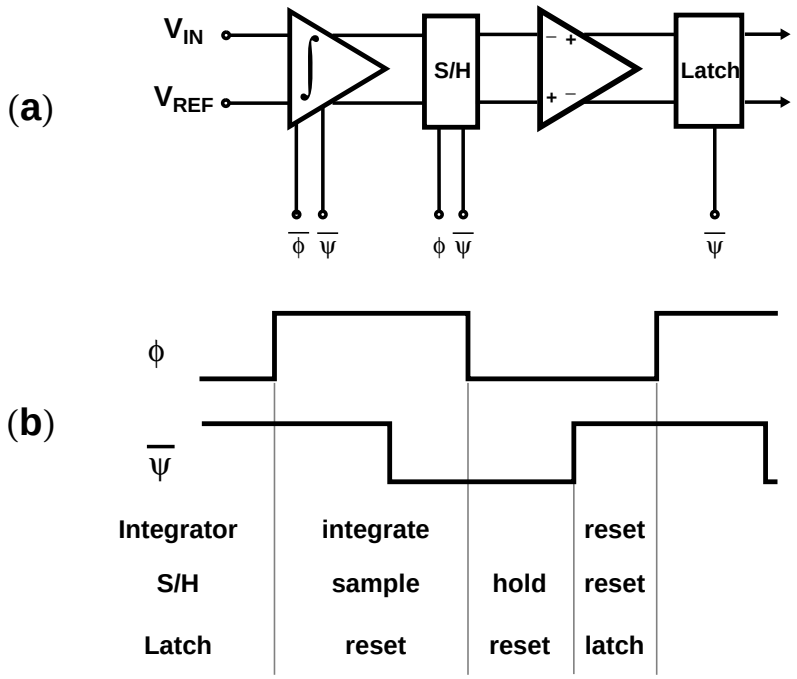

Figure 4.7: Receiver block diagram and timing

circuit, a buffering amplifier, and a final regenerative latching stage. To achieve a data transfer rate equal to the clock rate two receivers are used in parallel per input pin. The two parallel receivers sample the incoming data on both the half-periods of the clock. The receiver inputs are compatible with the interface described in Chapter 3: the input signal swings nominally between 0 and 1 volts, and the receiver compares the input against a 500-mV reference voltage. The timing operation of the receiver is illustrated in Figure 4.7-(b). During the integration period, indicated by the high level of phase ϕ, the differential output voltage of the integrator is tracked by the sample-and-hold circuit, while the amplifier and the latch are being reset. At the end of phase ϕ the sample-and-hold enters the hold state, while the second stage amplifies integrated voltage. At the positive edge of phase ψ the latch is triggered and the integrator and the sample-and-hold network are reset. Phase ψ is generated locally at the receiver, by delaying phase ϕ through two inverters. This self-timed three phase operation facilitates equalization of all the intermediate nodes in the circuit, thus minimizing a potential source of intersymbol interference. The rest of this section describes in detail the implementation of all the receiver stages, and the auxiliary circuits used to establish the current integrator bias.

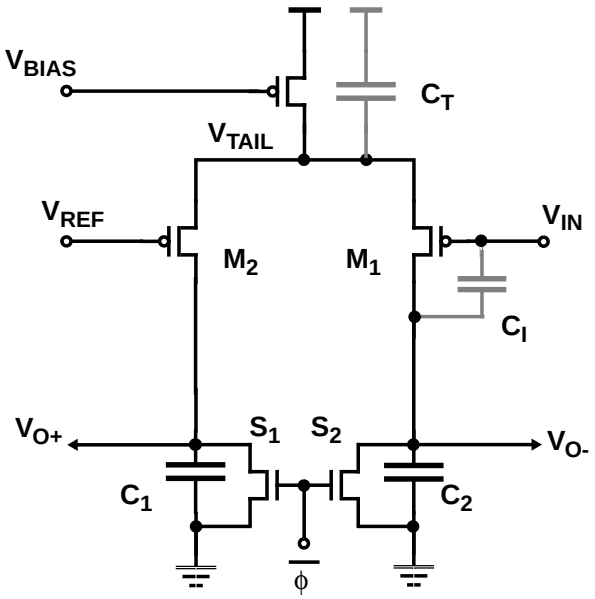

Figure 4.8: CMOS current integrator schematic

4.3.1 Current Integrator Design

A straightforward CMOS implementation of the ideal current integrator of Figure 4.3 is shown in Figure 4.8. Since the common mode of the input signals in this design is close to the bottom supply rail, the current switch is implemented as a PMOS source coupled pair consisting of devices M_1-M_3. The reset switches are implemented as the NMOS devices S_1, S_2 and the load capacitors are shown here as the linear elements C_1, C_2. In order for the source coupled pair to achieve a behavior close to the current switch of Figure 4.3, the input devices M_1 and M_2 must be able to steer all of the tail current with only a fraction of the input voltage swing. A MOS differential pair satisfies this requirement only when the input differential voltage is large compared to the gate overdrive of the source coupled devices needed for their particular operating tail current. We can define the operating margin V_{OM} of the current integrating receiver to be the difference between the nominal input differential voltage ΔV_{IN} and the voltage required to completely steer the source coupled pair tail current. Thus, the operating margin V_{OM} is equal to the excess input signal swing, for which the source coupled pair still behaves as a current switch. In case the input swing is equal to the operating margin voltage *($\Delta V_{IN}=V_{OM}$)*, the integrating receiver operation will be identical to the integrate and dump filter of Equation (4-3). When the input swing is less than the operating margin, but still larger than the source coupled pair input referred

offset ($V_{OM} > \Delta V_{IN} > V_{offs}$), the receiver still operates as an integrate and dump filter but with reduced output swing – i.e. the correct reception of the signals is not affected as long as the DC value of V_{REF} is in the middle of the input signal swing.

In order to make the receiver operation less sensitive to the offsets of the subsequent stages and on-chip noise, the output swing of the first stage integrator $V_O = (I \times t_b)/C$ should be maximized. To maximize the output swing one needs to minimize the load capacitance C_1, C_2. Thus, as long as M_1-M_2 can be kept in the saturation region (meaning that the output voltage V_O is less than a body-affected V_{TP}), one can set $C_1 = C_2 = 0$ and let the differential pair integrate its tail current on its parasitic drain junction capacitance. Assuming a quadratic MOSFET model and setting the tail current $I = (V_O \times C)/t_b$ and the integration capacitor $C = c_d \times W$, the operating margin of the integrator implementation can be derived:

$$V_{OM} = \Delta V_{IN} - \sqrt{\frac{2 \times V_O \times c_d \times L}{t_b \times \mu_p \times C_{OX}}} \tag{4-4}$$

where ΔV_{IN} is the input voltage swing (usually set by system constraints), V_O is the integrator output swing, c_d is the drain junction capacitance of M_1 and M_2 per unit of their width, L is the length of M_1 and M_2, t_b is the bit time, μ_p is the hole mobility, and C_{OX} the MOS transistor gate oxide capacitance. Equation (4-4) shows that, given the input and output swing requirements and a target operating speed, the operating margin of the current switch depends only on process technology parameters. Moreover, the operating margin increases as the process transconductance and the parasitic junction capacitance are improved. Essentially, for any given requirement of V_O and t_b, there is a set of pairs of widths of M_1-M_2 and tail currents which determine the operating margin voltage. In a practical design a designer must pick a tail current and set the device widths, based on a trade-off between minimizing power and making the integrating capacitors large enough to minimize coupling effects. A secondary requirement is that the tail current has to be large enough so that noise coupling will not introduce a significant integration error. In the implementation described in this section, where t_b=2nsec and ΔV_{IN}=500 mV, V_O was set

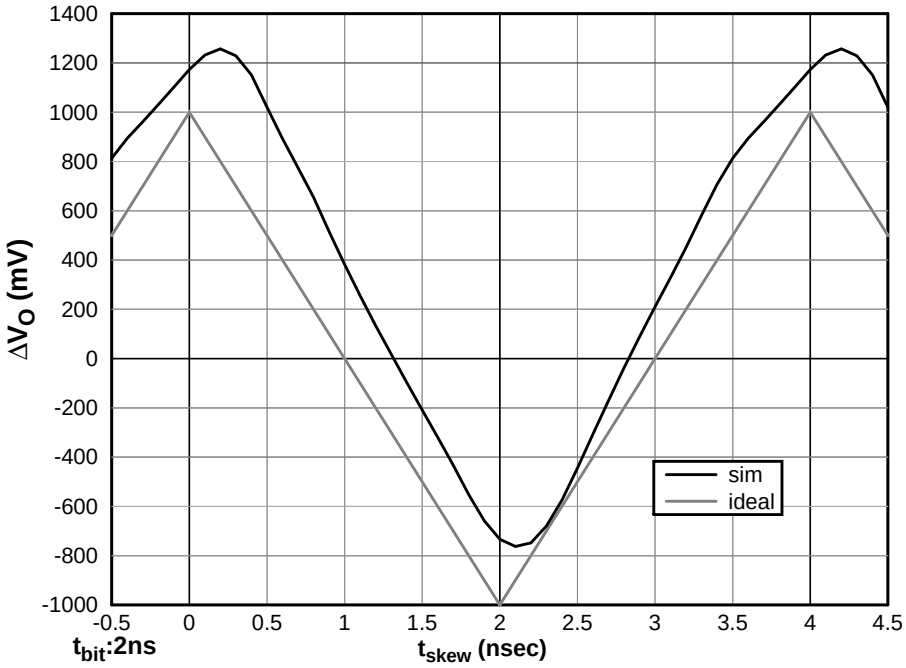

Figure 4.9: Effect of charge injection in the integrator characteristics

to 800 mV, while the width of M_1-M_2 was set to 100 µm for 200 µA of nominal tail current. This resulted in a minimum V_{OM} of 280 mV over process and temperature variations. Note, however, that these results correspond to the actual implementation described below in which the charge injection error cancellation circuits double the parameter c_d.

For high current gain M_1-M_2 must run at low current per unit width, which increases the effect of parasitic capacitances C_I and C_T of Figure 4.8. The gate to drain overlap capacitance C_I of transistor M_1 couples the input transitions on the output integrating capacitor, introducing a systematic integration offset. The significant tail capacitance of the source coupled pair C_T creates another systematic offset. Due to the fact that the input to the receiver is pseudo-differential, when the input transitions the tail node must settle at a gate overdrive above $min\{V_{IN},V_{REF}\}$ before the tail current is steered to one of the branches of the source coupled pair. The significant tail capacitance C_T consumes or sources some charge to or from the output node, and therefore introduces another systematic offset into the integrator. Figure 4.9 shows the simulated effect of these offsets in an integrator operating with a 2-ns bit time. The offsets result in an increase of the integrator output voltage when the input is low. Thus, the time interval where the data is detected as

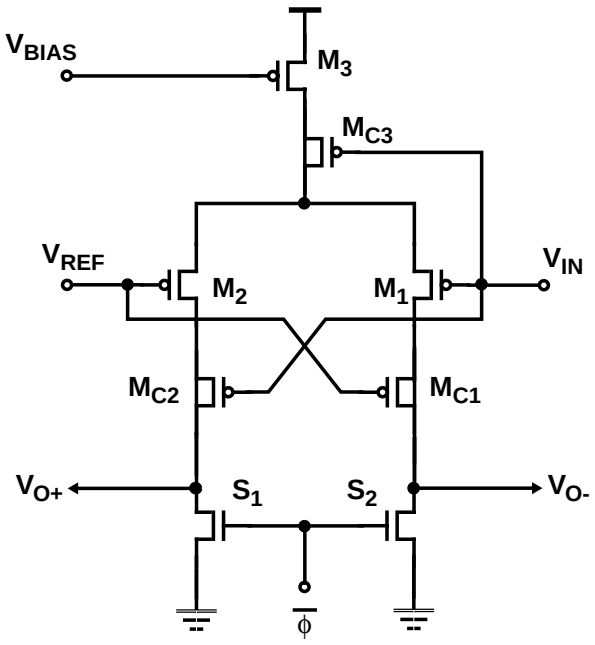

Figure 4.10: Initial current integrator implementation

high is increases resulting in a wider sampling uncertainty window. The net effect of these offsets is that the simple integrator of Figure 4.8 has a 500-psec sampling uncertainty window – i.e. 25% of the 2-nsec operating target bit time.

4.3.2 Charge Injection Offset Cancellation

In order to compensate for the charge injection errors and decrease the sampling uncertainty window width, the circuit of Figure 4.10 was used in an initial design which is described in more detail in [17]. The source coupled pair consists of devices M_1-M_3 and the reset network consist of devices S_1-S_2. The source coupled pair is augmented by three capacitor-connected devices M_{C1}-M_{C2}. Transistors M_{C1} and M_{C2} are employed to cancel the systematic offset introduced by the gate-to-drain capacitance of M1. These two devices have half the width of the input devices and they always remain off. Therefore, the transitions of the input node are coupled on both the outputs of the integrator. Thus, the differential mode error introduced by the gate-to-drain capacitance of M1 is converted to a common mode variation of the integrator output voltage. The offset introduced by the tail capacitance of the differential pair is canceled by transistor M_{C3}. When the input transitions, M_{C3} boosts the tail capacitance to its quiescent voltage level, thus minimizing the

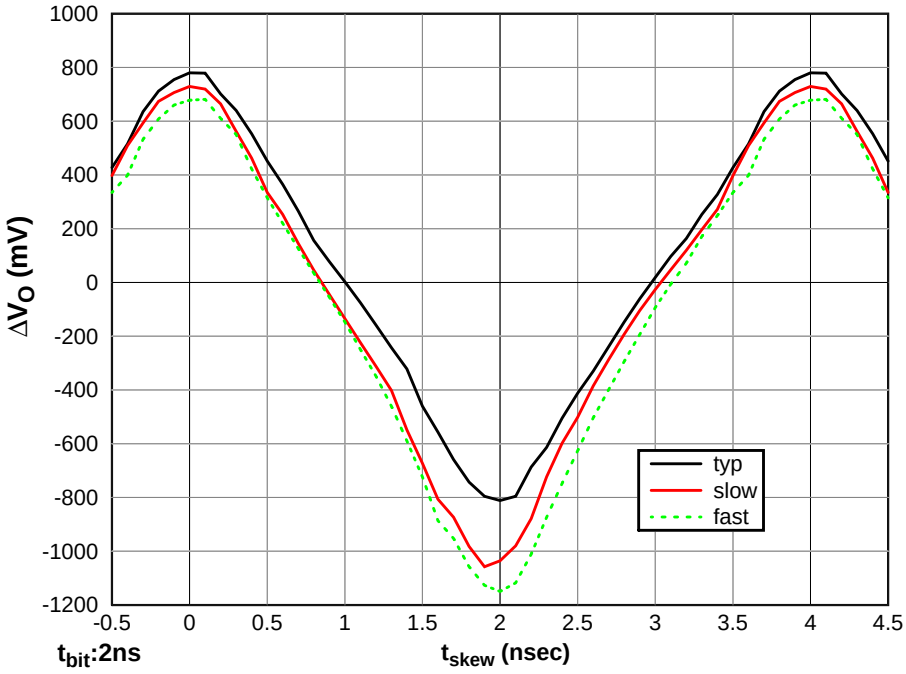

Figure 4.11: Phase characteristics of the initial integrator implementation.

charge injection error if the capacitances C_T and M_{C3} are properly ratioed. The phase characteristics of this design, simulated over three different process and environmental condition corners at an operating frequency of 250 MHz, are illustrated in Figure 4.11. It can be seen that by employing the cancellation techniques, the worst-case sampling uncertainty window width decreased from 500-psec to 180-psec. The imperfect cancellation seen in the "fast" and "slow" simulation corners is mainly introduced by the mismatch in the ratio between gate and drain capacitances in the tail boosting circuit. These two parameters can change disproportionately over process variations and increase the sampling uncertainty window width.

In order to decrease the process sensitivity an improved integrator design was developed. Figure 4.12 shows the design of the improved current integrator along with the associated sample and hold circuit. The current integrator consists of two differential pairs with cross coupled outputs. The main differential pair consists of transistors M_1-M_3 and is operated at a nominal current level of I_M=*200*-μ*A*. The auxiliary differential pair, comprises devices M_{C1}-M_{C4}, is operated at a fraction of the current main differential pair tail current (I_C≈*50mA*) and cancels the charge injection induced offsets. The auxiliary pair

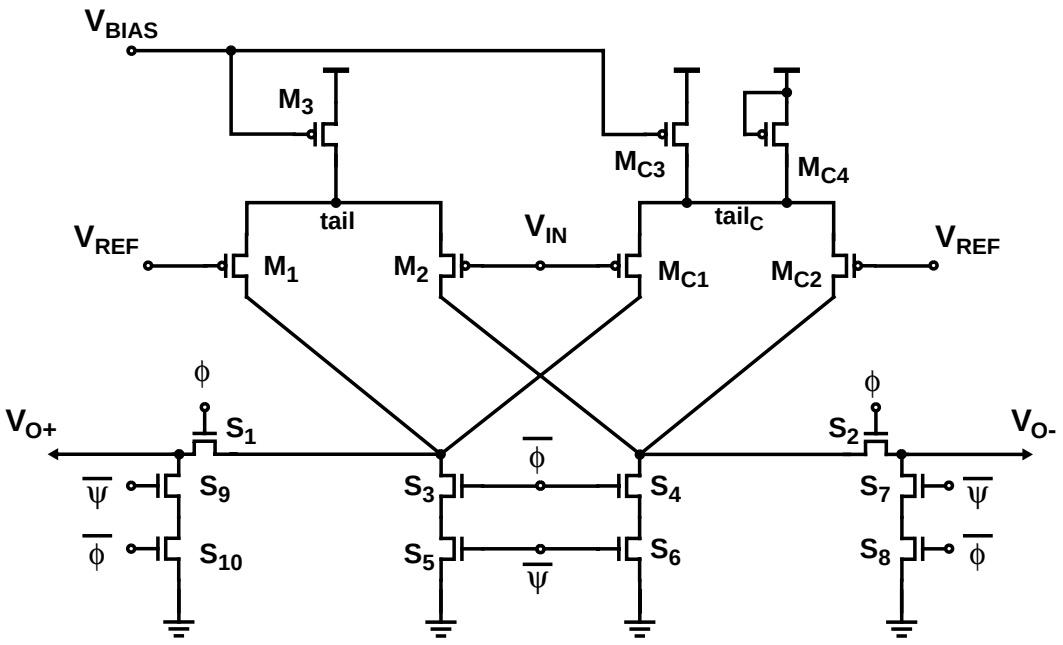

Figure 4.12: Improved integrator implementation.

devices M_{C1}-M_{C2} are identical in size to M_1-M_2, while the total width of M_{C3} and M_{C4} is equal to the width of M_3. This results in identical parasitic capacitances in the two source coupled pairs. Therefore, the cancellation differential pair provides a parasitic induced error that is equal in magnitude and opposite in polarity to that induced by the main differential pair. In this way, all the parasitic induced differential errors are converted to a common mode variation which is small enough not to affect the operation of the second stage. Using the auxiliary differential pair implies that the output differential voltage ΔV_O of the integrator is reduced by a fraction equal to the ratio of the currents I_C and I_M. This fact suggests that the ratio of these two tail currents should be minimized. However, making this current ratio arbitrarily small affects the canceling action of the auxiliary differential pair. The requirement that bounds the lowest level of the offset canceling current I_C is that the node $tail_C$ should reach its quiescent voltage within the integration period, in order for the parasitic induced error of the two differential pairs to be equal. Simulation results indicate that choosing the ratio of I_M to I_C to be 4 results in a worst case 20% margin over all process corners for a 1.6-nsec minimum bit time.

Transistors S_1-S_{10} in Figure 4.12 form the sample and hold network and the integrator reset switches. To compensate for any overlap that might be present between the reset-

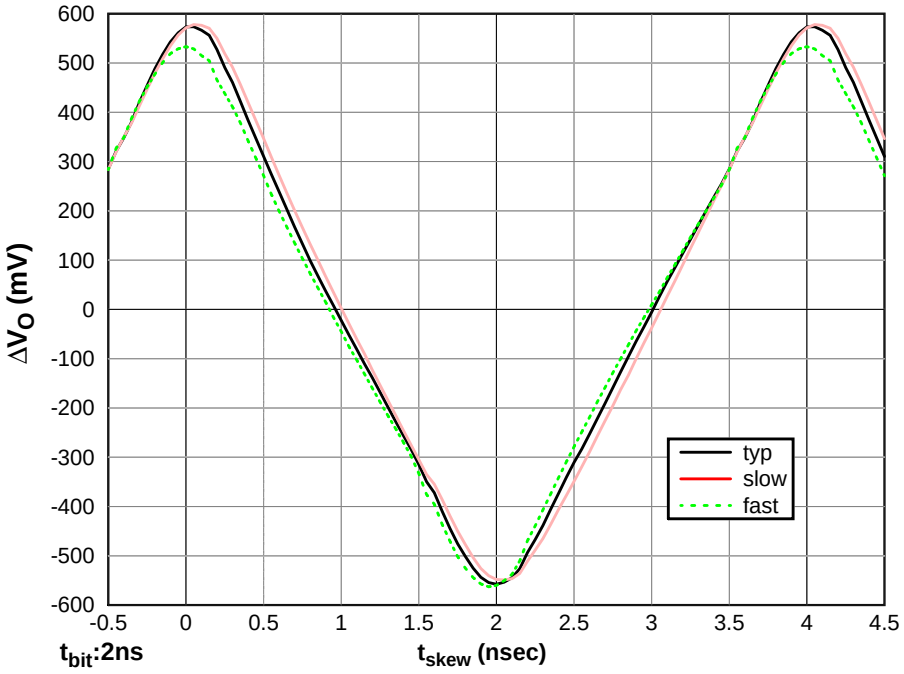

Figure 4.13: Phase characteristics of the improved integrator implementation.

ting phase ϕ and the sampling phase ϕ, the reset network is formed as a stack qualified by a delayed version of ϕ. Phase ψ generated locally by delaying ϕ through two inverters. This way the first stage is reset only after the sample and hold switches S_1-S_2 have been completely shut off. Using the overlap of signals ϕ and ψ to reset the sample and hold network through an NMOS stack, rather than resetting the sample and hold through a single transistor driven by a logic gate, has the additional advantage that shorter signal overlaps can be achieved. This enables the receiver to operate at higher speeds without introducing intersymbol-interference.

Figure 4.13 shows the phase characteristics of this improved design. Employing the auxiliary differential pair, reduces the systematic errors to below 50 psec (2.5% of the target bit time). The remaining small error is due to the nonlinear nature of the tail junction capacitance: since the offset canceling differential pair runs at a lower current than the second differential pair, node tail settles at a higher voltage than node tail_C. This higher voltage makes the non-linear junction capacitance component of node tail slightly higher, and thus introduces an imbalance between the charge injected by the parasitics of the two source coupled pairs.

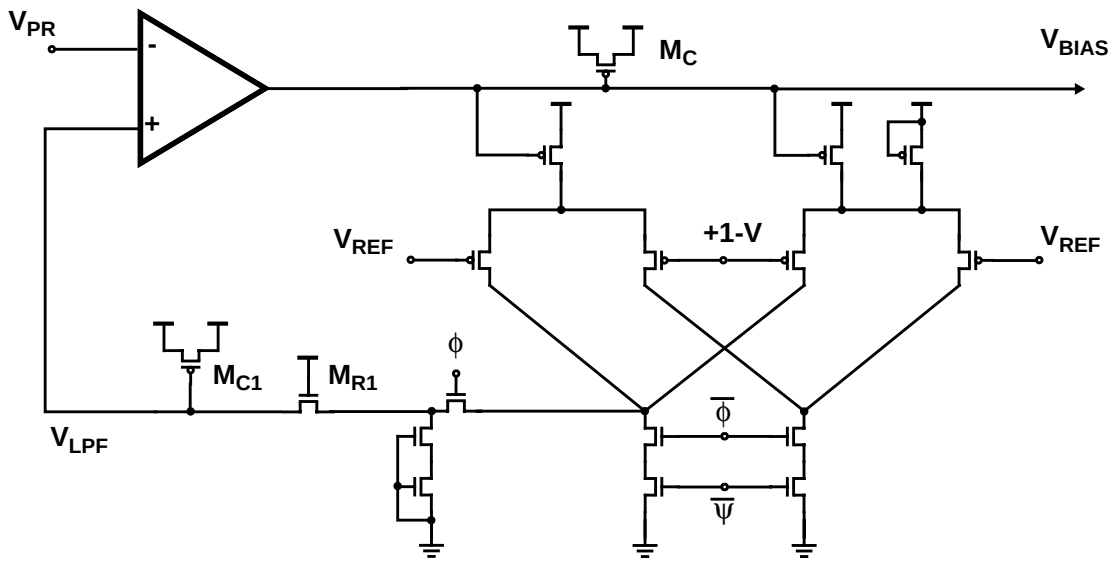

Figure 4.14: Current integrator replica-feedback biasing circuit.

4.3.3 Current Integrator Biasing

As described in the previous section, the current integrator does not use explicit load capacitance, but it instead integrates current on the parasitic drain junction capacitors of its source coupled pair transistors. This design choice maximizes both the output swing and the integrator operating margin for a given process technology. However, it also has the effect that the integrator output swing (being inversely proportional to the load capacitance) is sensitive to process variations that affect the PMOS drain junction capacitance. Additionally, since the integrator swing is proportional to the bit integration time, using a constant tail current would result in a very narrow interface operating frequency range. Interface operating frequencies higher than the upper limit of this range would result to a diminishing output swing, while operating frequencies lower than the lower limit of the range would drive the integrator source coupled pair transistors out of their saturation region of operation.

In order to extend the receiver operating frequency range and mitigate the effects of the uncontrolled junction capacitance, the circuit of Figure 4.14 is used to generate the bias voltage for all the on-chip integrator circuits. This replica-feedback biasing circuit dynamically adjusts the current through the PMOS current sources so that the maximum differential output voltage of the integrator ΔV_O is held constant and independent of the

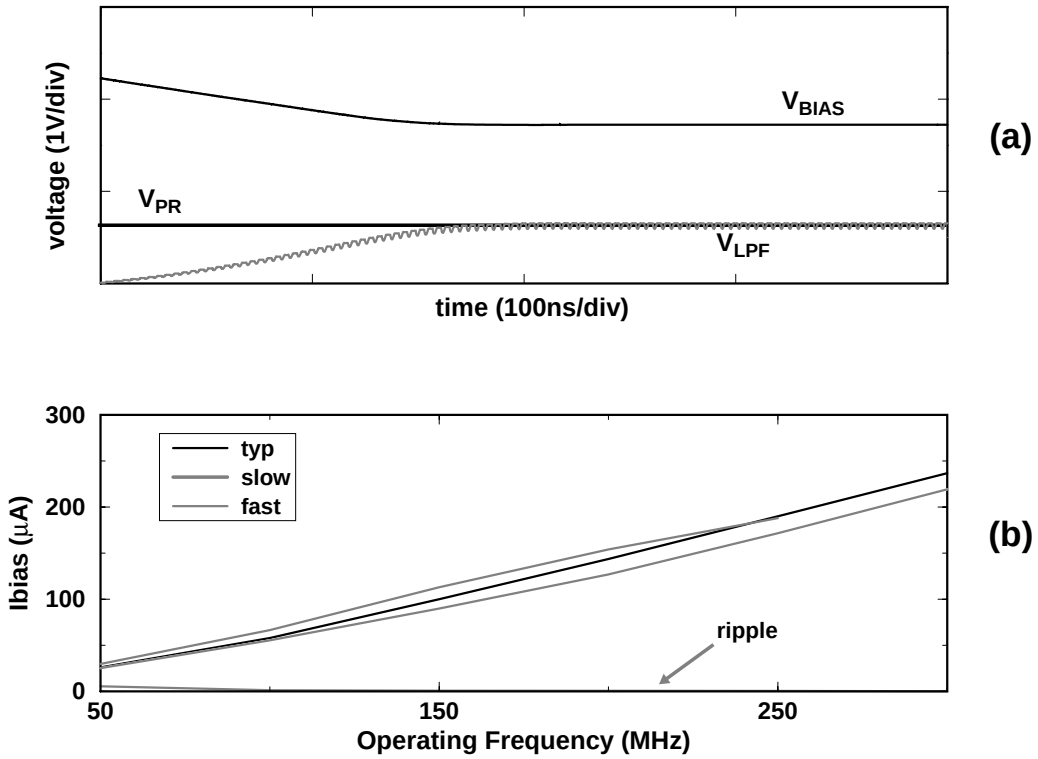

Figure 4.15: Bias loop start-up (a), and output current vs. operating frequency (b)

parasitic integration capacitor and the operating clock frequency. The output of a replica of the first-stage integrator is low pass filtered through M_{R1}-M_{C1} and subsequently drives an operational amplifier which compares it with a preset voltage V_{PR}. The amplifier adjusts the current through the replica circuits, such that the low-pass filtered output voltage V_{LPF} remains equal to the preset voltage V_{PR}. Compensation for the feedback biasing loop is accomplished with the explicit capacitor formed by transistor M_C. The dominant pole of the circuit is set to 1-MHz, well below the minimum operating frequency of the receiver clock and the first non-dominant pole formed by M_{R1}-M_{C1}.

Figure 4.15-(a) shows the simulated waveforms of the replica-feedback loop start-up under typical process conditions at an operating frequency of 250-MHz (t_b=2 ns). The stability of the loop is evident from the overdamped settling behavior of both V_{LPF} and V_{BIAS}. The circuit is initially reset to reproduce a minimum biasing current and settles in less than 200-nsec. Figure 4.15-(b) is a plot of the resulting integrator bias current over operating frequency and varying process and environmental conditions. The integrator bias current varies linearly with operating frequency and with a slope proportional to the

junction capacitance of the corresponding simulation condition. The switched-capacitor nature of the circuit results in a bias current ripple. However, this ripple is negligible at high operating frequencies, while its maximum value of 2.5-μA (about 10% of the average bias current value) can be observed at an operating frequency of 50-MHz.

4.3.4 Amplifier and Latch Design

Figure 4.16 shows the schematic diagram of the amplifier and the latch used in the last two stages of the current integrating receiver. Although the output differential voltage of the integrator is large enough to reliably drive the final regenerative stage, the use of the amplifier improves the overall receiver robustness by buffering the output of the sample and hold network. This buffering isolates the high impedance nodes of the sample-and-hold from potential latch kick-back. Additionally, the amplifier's modest gain and low input referred offset improve the setup-and-hold timing uncertainty of the receiver by increasing the slope of the phase characteristic curve as seen at the latch input.

The amplifier uses a combination of cross-coupled and diode connected loads which simultaneously provide high differential and low common mode output impedance [43], [49]. The high differential impedance improves the small signal gain of the amplifier, while the low common mode impedance limits the common mode voltage variations at the latch inputs. Simultaneously the diode clamps limit the large signal swing, facilitating

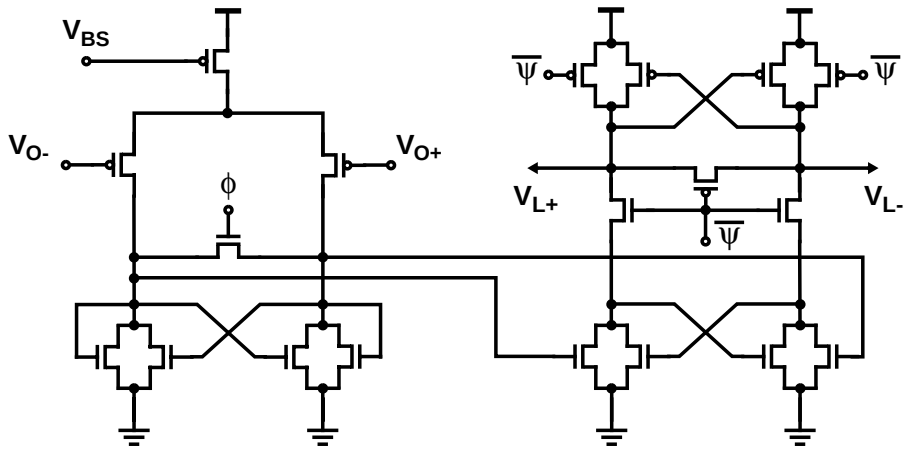

Figure 4.16: Amplifier and latch schematic

faster reset and reducing coupling at the sample-and-hold output. In order to prevent variations of the amplifier's bandwidth to introduce intersymbol interference, an explicit reset switch is used at its output.

The differential latch [50] converts the low swing output of the amplifier to a full swing CMOS signal. The output of the precharged latch is held stable for a full clock cycle by a nand-gate based S-R latch. The width of the sampling uncertainty window of the receiver is affected not only from systematic charge injection offsets in the first stage but also by mismatches between nominally identical devices in the amplifier and the following latch. Simulation results assuming 25-mV threshold voltage mismatches between nominally identical devices in all the integrator stages indicate that random offsets increase the receiver's sampling uncertainty window by less than 80-ps.

4.4 An Interface Using Current Integrating Receivers

In order to evaluate the performance of the current integrating receiver design described in the previous section, an IC implementing a complete point-to-point parallel interface was designed in a 0.8-µm CMOS technology. Figure 4.17 shows the block diagram of the

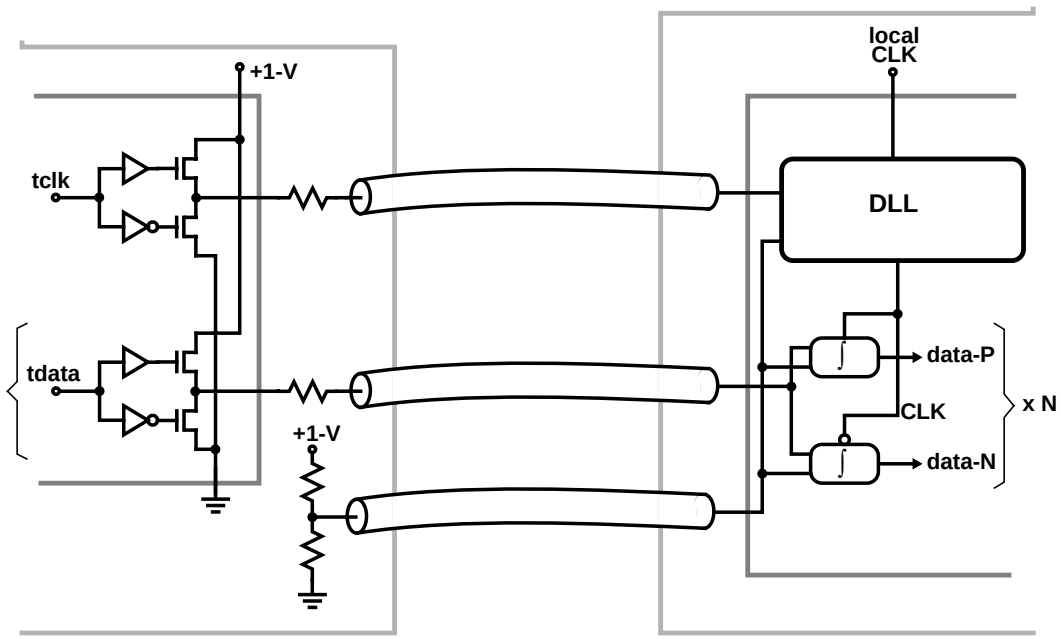

Figure 4.17: Interface block diagram

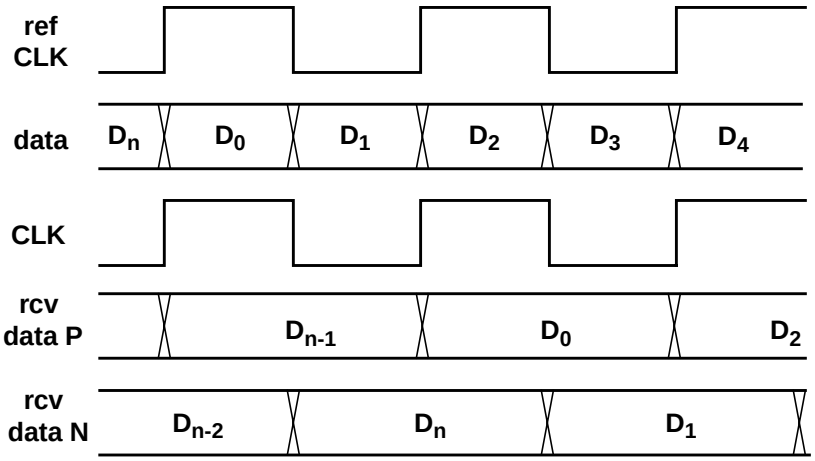

Figure 4.18: Timing of the interface signals

interface. Externally the design is compatible with the one described in Chapter 3. The transmitter utilizes the push-pull output driver described in Section 3.1, resulting in compatible signal voltage levels. In addition, the same source synchronous clocking scheme is used, i.e., the reference clock is transmitted in phase with the parallel data signals. On the receiver side a DLL uses the phase information of the reference clock to position the on-chip clock (CLK) at the optimal point for sampling the transmitted data.

The timing of the interface signals is illustrated in Figure 4.18. While the conventional design of Chapter 3 positions the receiving clock in quadrature with the reference clock, this design positions the receiving clock (CLK) in phase with the reference clock (ref-CLK) in order to maximize the timing margins for the current integrating input receivers. The two integrating receivers, attached on each input pin, utilize the level of the receiving clock to integrate the incoming data during the appropriate clock phase. Compared to a conventional design, using a current integrating receiver implies an inherent latency penalty of half a bit time. Thus, in this design the incoming data (rcv-dataP, rcv-dataN) is available to the internal circuits approximately half a clock cycle after the end of the integration period. To evaluate the interface performance, the chip contains both a “pseudo random bit sequence” (PRBS) generator on the transmitter side. A corresponding PRBS decoder was integrated on the receiver side. The rest of this section briefly describes the receiver clocking circuits, and the peripheral bit error rate testing circuits used in this experimental prototype.

4.4.1 Clocking Circuits

The task of the delay locked loop is to position the receiving clock at an optimum point for integrating the incoming data independent of variations in process, voltage, temperature, and the position of the receiver's center of timing uncertainty window. Since the current integrating receivers exhibit very small variation of their timing uncertainty window, the optimum point of the receiving clock is exactly in phase with the incoming data. Therefore, the need for a DLL at the receiver side would diminish if the reference clock were a full swing signal that could directly drive the on chip receivers. However, since the reference clock is a low swing signal that needs to be amplified and buffered up before driving the receiving circuits, a DLL is crucial in canceling this amplification and buffering delay.

Figure 4.19 shows a block diagram for the DLL used in this design. The DLL consists of a conventional core (delay line, charge pump and phase detector) along with a controlling finite state machine and a pair of duty cycle adjusters (DCA-I and DCA-O). The design described in Chapter 3 followed the conventional approach of using an amplified version of the reference clock as the delay line input [42]. This approach has the main disadvantage that any reference clock jitter is inherently propagated to the on-chip receiving clock in an all-pass filter fashion. Therefore, if the reference clock edges deviate from their nominal position due to noise, the on-chip receiving clock edges will deviate from the optimal sampling point a time instant t_D later, where t_D is the delay through the delay line and the clock buffers. Although the magnitude of the receiving clock deviation remains the same as the deviation of the reference clock and data, the phase of the deviation peaks at 180^o at jitter frequencies which are odd multiples of $1/(2\times t_D)$, directly subtracting from the receiver's timing margins. In addition to the phase noise induced on the reference clock at the transmitter side, the reference clock in a pseudo-differential interface can acquire additional phase noise at the receiver side from high frequency amplitude noise on the reference voltage line. To minimize this effect this design uses a differential ECL level clock as the delay line input. Although this clock has the same frequency as the reference clock it carries much less phase noise, since it is not buffered on the noisy transmitter chip. Additionally, its differential nature minimizes jitter induced from the receiver on-chip common mode noise. Although this phase noise rejection advantage comes at the cost of

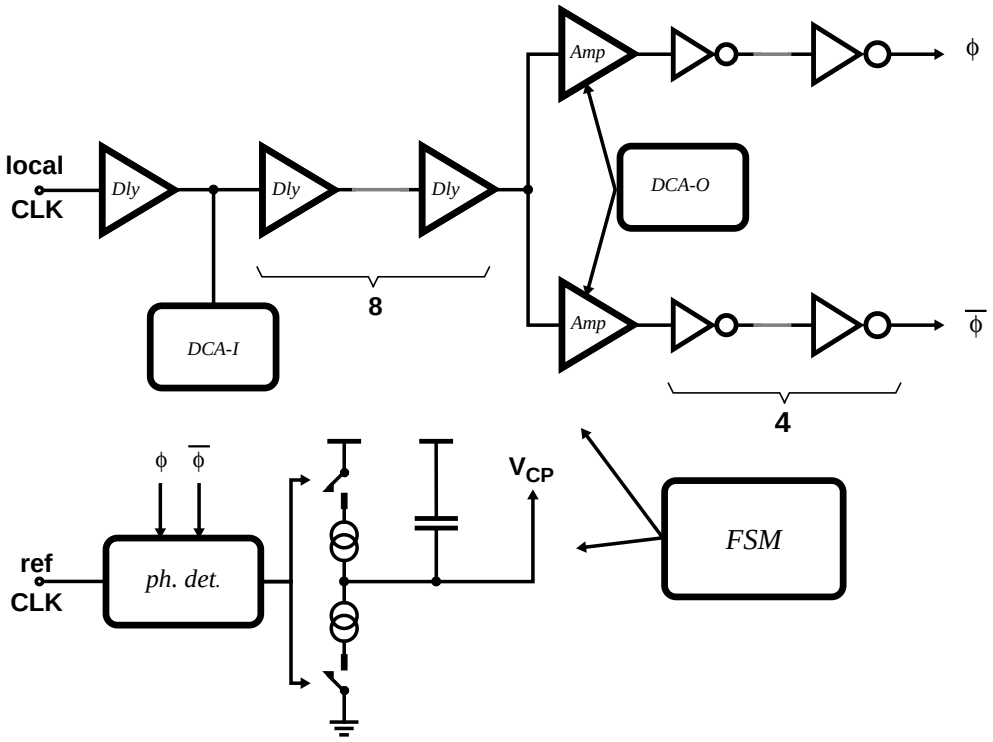

Figure 4.19: DLL block diagram

two additional pins, in a large IC such as a multiprocessor router the extra pin cost can be amortized over the multiple interfaces integrated on the same die.

A phase detector compares the phase of the reference clock with that of the DLL output clock. This phase detector is implemented as a sampling variation of the integrating receiver. A pair of NMOS pass transistors sample the input reference clock at the rising edge of the receiving clock and drive the sampled value to a regular integrating receiver. In order to compensate for the offset introduced in the phase detector by the sampling MOSFETs, all the regular integrating receivers are augmented with the same sampling transistors with their gates tied to the positive supply. The output of the phase detector is integrated by the charge pump generating the control voltage V_{CP}. The output of this type of phase detector is simply a binary “up/down” phase error indication, rather than being proportional to the phase error as is the case with the phase detector described in Section 3.2.1. Thus, when the DLL is in lock it will dither with some amplitude around the desired locking point. The magnitude of the dithering amplitude is proportional to the loop delay and the operating clock cycle. In order to limit the amount of dithering, especially at lower operating frequencies, the phase detector output is qualified by a fixed width pulse

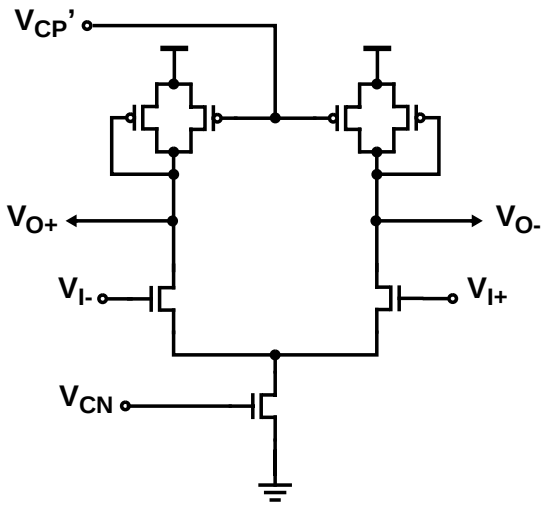

Figure 4.20: Delay element schematic

[42]. In this way, a fixed charge packet is delivered to the filter capacitor every clock cycle, keeping the dithering of the control voltage around the locking point fixed and independent of the operating frequency.

The delay line is implemented as a series of eight delay elements. In order to improve noise sensitivity, the DLL uses differential delay elements with symmetric controlled-impedance loads [51]. Figure 4.20 shows the delay element schematic diagram. The control voltage V_{CP}' is a buffered version of the charge pump control voltage, while V_{CN} is generated by a replica feedback biasing circuit which keeps the delay through the elements constant and independent of supply variations.

In this design the differential clock input to the delay line and the reference clock have an arbitrary phase relationship with each other. Therefore, there is no guarantee that after the initial system reset, the DLL will not be trying acquire lock at a point which is close to or below the minimum delay that can be generated by the delay line. In order to evade this problem, a finite state machine controls the DLL acquisition process. Initially, voltage V_{CP} is reset to a value that generates a delay which is at least 500-ps more than the minimum of the delay line. When the reference clock is activated, the finite state machine ignores the phase detector output for the first 8 cycles of operation. If during the subsequent 16 clock cycles the phase detector generates a “down” signal, the finite state machine phase-shifts the sampling clock by 180^{o} - a multiplexing differential delay ele-

ment in the delay line is used for this purpose. Thus, the maximum required range of the delay line is *T/2+500* ps (where *T* is the operating clock period). Resetting the delay line above its minimum delay point gives the DLL margin across temperature induced drifts of the reference clock. In addition to this reset sequencing the FSM improves the loop acquisition time by controlling the charge pump current. After the decision on the relative phase between the input and reference clocks, the FSM increases the charge pump current by a factor of 5 for 50 clock cycles. Subsequently the charge pump current is scaled back to the normal operating levels to obtain low dither jitter [43]. Driving the FSM directly with the phase detector output might compromise its robustness due to propagation of metastable states. Although in this implementation no particular care was taken to mitigate this problem, it can be easily solved by delaying the phase detector output through a flip-flop chain before driving the FSM.

Since data is transferred on both the half-clock periods, any variation of the sampling clock from the optimal 50% duty cycle point would cause one of the two input samplers to integrate the wrong data item, thus degrading the timing and noise margins of the input pin samplers. Variations in the clock duty cycle can either be inherited from the ECL clock input to the delay line, or induced by the amplifiers and the clock buffers at the delay line output. To compensate for these effects two duty cycle adjusters are used. Input clock duty cycle variations can be detrimental to the operation of the loop since, in combination with the inherent bandwidth-limitation at delay line buffer outputs, they can result in a loss of the receiving clock. Thus, the first duty cycle adjuster is used at the input of the delay line. This DCA uses two differential delay elements connected in a feedback loop with NMOS capacitors that remove the AC component of the voltages [51]. The output of the input DCA is tied to the output of the first delay element, compensating for both duty cycle variations and common mode offsets of the input clock. The second DCA is embedded in the final stage which converts the low swing clock output of the delay line to a full swing CMOS signal. The schematic diagram of the this converter is shown Figure 4.21. Two amplifiers with current mirror load and an extra port are employed to generate the signals c and $\overline{c}$. The extra port of the amplifier is connected to the output of a band limited delay element which low pass filters the sampling clock signals. Thus, variations of the duty

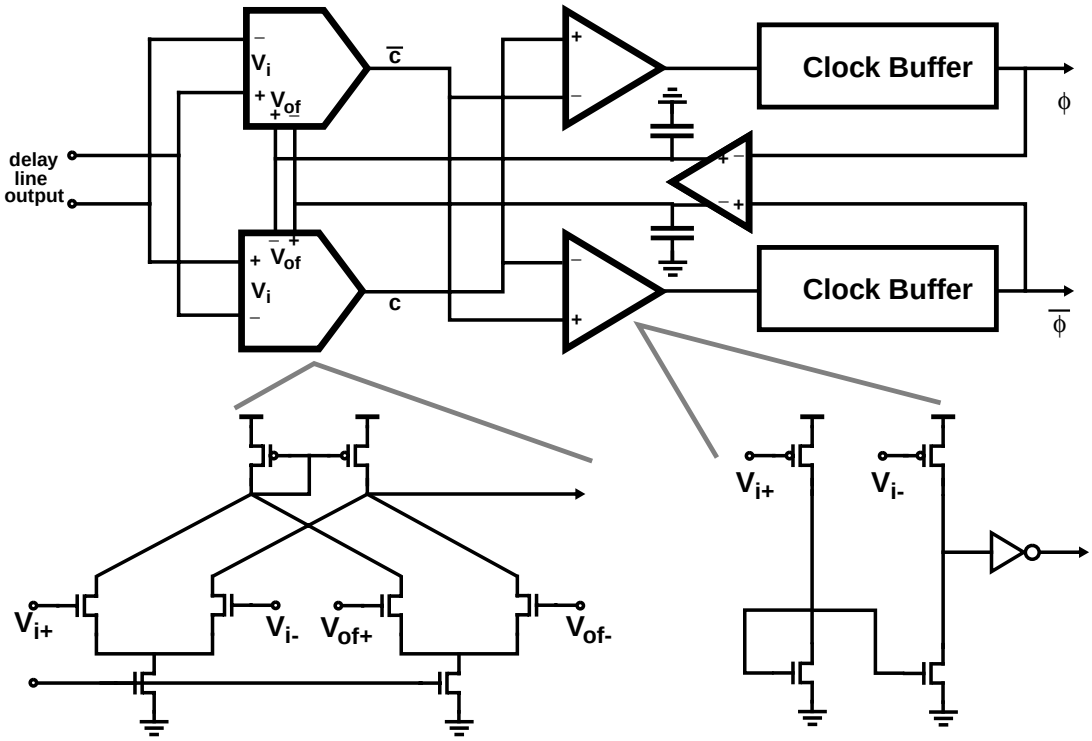

Figure 4.21: Output DCA schematic

cycle of the receiving clock from 50% induce an input referred offset to the clock amplifier. This negative feedback causes the duty cycle of the sampling clock to be adjusted. A second stage current mirror amplifier increases the output swing before driving the first stage of the clock buffer inverters.

4.4.2 Peripheral Circuits

A number of peripheral testing circuits were integrated on the prototype transceiver die. To assist in the debugging of the prototype, the transmitter can continuously transmit an 8-bit long pattern or a pseudo-random bit sequence (PRBS), while the corresponding decoder is integrated in the receiver side. The receiver contains “monitor” output drivers, which can be configured to drive to a scope the outputs of the current integrating receivers, the receiver internal clock signals, or the PRBS decoder circuit output. Alternatively, the “monitor” outputs can drive to the scope buffered versions of the signals on the receiver input pads. For this purpose the differential amplifier described in Section 3.1.2 was used in the receiver data and reference clock inputs.

The transmitter fixed pattern generator consists simply of two sets of 4-long shift registers. The contents of these shift registers can be externally configured through a scan chain. Since the frequency of the on-chip clock is equal to half of the external data-rate, the PRBS circuit is implemented as two linear feedback shift registers (LFSR) with corresponding characteristic polynomials: X^7+X+1 and X^7+X^6+1. The two LFSR's are clocked by different on-chip clock edges and since the second terms of their characteristic polynomials are symmetric, they generate two palindrome sequences of length 2^7-1 [52]. The two palindrome sequences are multiplexed on the output driver, generating a PRBS with a period of $2\times(2^7-1)$ on the inter-chip transmission line. The composite sequence is demultiplexed on the receiver side and the outputs drive the two PRBS decoders.

Multiplexing two palindrome sequences results in a sequence with slightly less random characteristics, i.e., the power spectral density of the composite sequence is not as flat as that of the individual palindrome sequences. However, when judged from an intersymbol interference perspective, the composite sequence has minimum and maximum run-lengths of 0's and 1's which are comparable to the run-lengths of the original palindrome sequences. The minimum run length of the composite sequence is 1 in all cases. The maximum run length of 0's is 5, while the maximum run length of 1's is 14 (contrasted to 6 and 7 for the original palindrome sequences). Normally a longer PRBS would be desirable to expose any bandwidth limitations. In our case however, where the input receivers are reset every clock cycle and the transmission medium has a bandwidth far exceeding the bit rate, the use of a shorter PRBS does not affect the validity of the observed bit-error-rates.

4.5 Experimental Results

The prototype transceiver chip was fabricated in the HP-CMOS26B process using the MOSIS scalable design rules (1.0/0.8-µm drawn/effective channel lengths). The chip die occupies 2.5×2.5 mm^2 and is shown in Figure 4.22. The chips were packaged in a 40-pin ceramic dual-in-line (DIP) package. Similar to the design described in Chapter 3, to ameliorate the effects of the large pin inductance of the package, the high speed signals were routed through the pins with the lowest inductance (≈10 nH). Additionally a total of 14

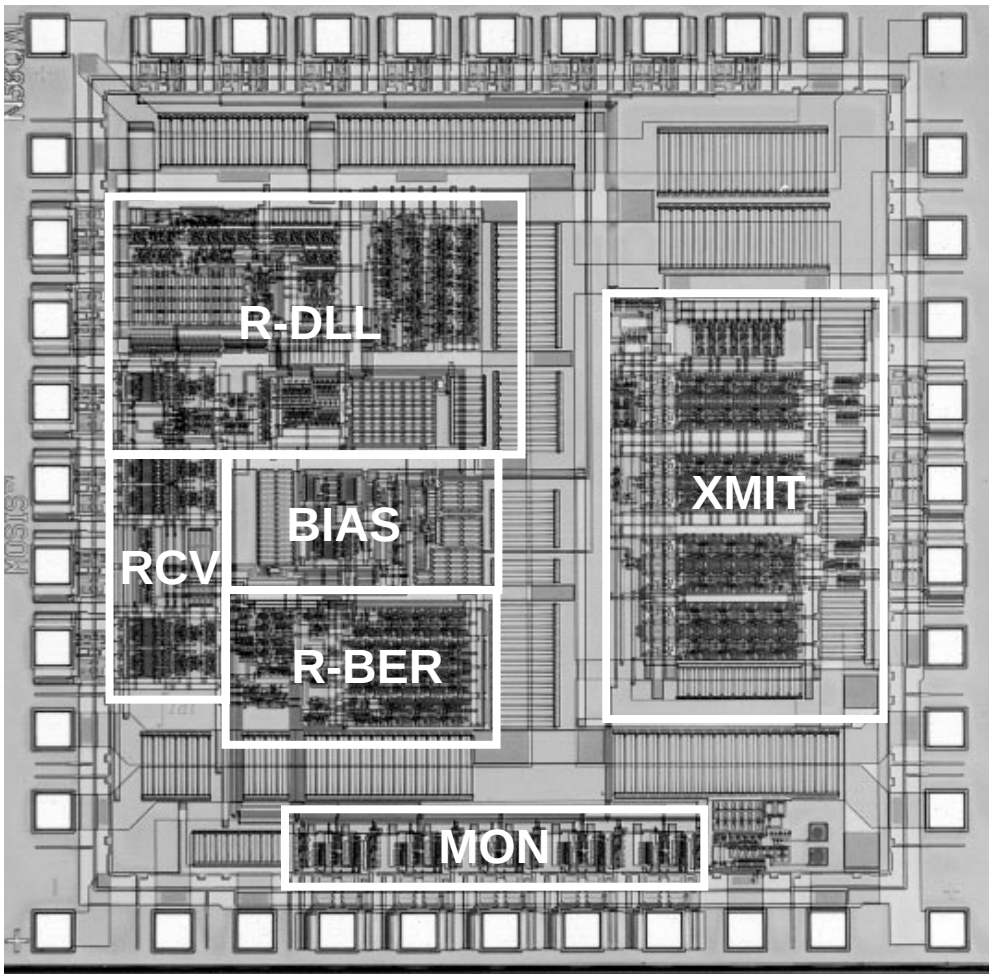

Figure 4.22: Transceiver chip photomicrograph

pins were dedicated to the chip power: six for the ground/substrate node, four for the 3.3-V positive supply and four for the 1-V output driver supply.

In the experimental set-up we used two-sided printed circuit boards for the transmitter and receiver chips. The packages of the transmitting and receiving chips were mounted on the PCB through zero-insertion-force sockets, while the high speed signals were carried through 1-m long coaxial cables. Figure 4.23 shows the measured maximum transfer rate versus the corresponding operating supply voltage. The bit error rate for this measurement is less than 10^{-11} for all cases (at least 5 minutes of continuous operation did not yield a single error). At the low end of supply voltages (2.7 volts) the interface achieves an operating speed of 540 Mbps/pin. At the high end of the operating voltage range the achievable speed was limited from the ability of the pulse generator used in the set-up to generate a

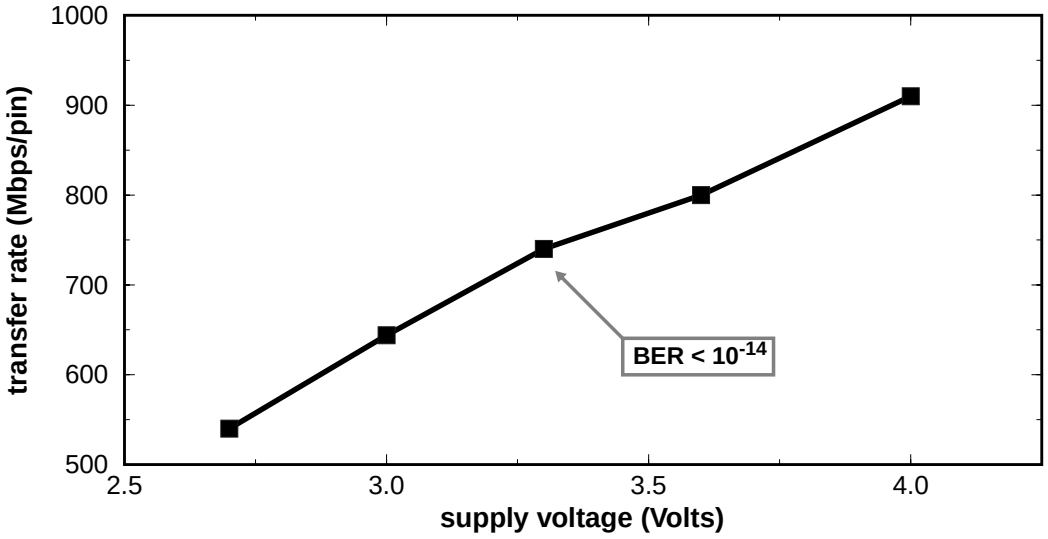

Figure 4.23: Prototype operating range

clean reference clock beyond 455 MHz. As a result, although the fabrication process can withstand a 5-V nominal supply voltage, the performance of the prototype transceiver was evaluated up to an operating point of 4-volts. Since the design of the integrating receiver and the clocking circuits was performed with a 3.3-V nominal supply, a more persistent bit error rate measurement was performed at this nominal operating point. This measurement revealed that the actual BER at the 740 Mbps/pin transfer rate is less than 10^{-14} (three days of continuous operation did not yield a single error).

Figure 4.24 shows an eye diagram of the received data output of one the receiver's samplers (after being buffered and driven off-chip). In this experiment the chip operates in a loopback mode at a 740 Mbps/pin transfer rate, while the peak-to-peak on-chip supply noise is 200 mV. The pseudorandom data eye has a measured jitter of 180-ps peak-peak (28-ps RMS). Furthermore, simulation results indicate an overall DLL static supply sensitivity of 0.7 ps/mV.

The sampling uncertainty window of the interface was measured by keeping the reference clock at a fixed position and varying the skew between the reference clock and the data. In this experiment we assumed that the sampling uncertainty window was violated if the bit error rate exceeded 10^{-9} (i.e., a transmission error occurred within the first 5-seconds of operation). The worst case sampling uncertainty window width of the system was

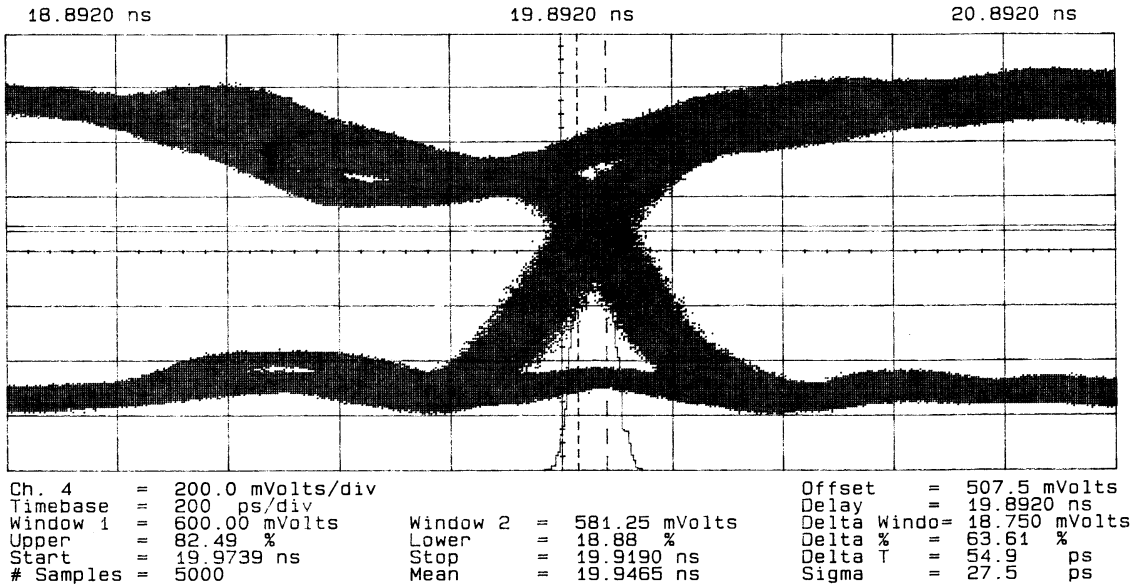

Figure 4.24: Received data eye diagram

found to be 280 psec, with its center located 80 psec from the center of the DLL locking point. Referring back to Figure 4.4, the uncertainty window has a width of 25.2^{o} and its center is offset by -6.3^{o} from the ideal 90^{o} point. Note however, that this is the composite uncertainty window of the system, since along with the inherent receiver offset it includes offsets introduced by the transmitter and the DLL. To evaluate the inherent receiver sampling uncertainty window, the experiment was repeated with the input data being a clock waveform and all the major on-chip noise sources turned off. In that case, the window width was found to be 50 psec. This result shows that the improved design of Figure 4.12 has a sampling uncertainty window width which is a factor of 3 smaller than that of the initial design of Figure 4.10 [17]. The sensitivity of the integrating receiver was measured by decreasing the 1-V output driver supply and simultaneously scaling the reference voltage. This experiment revealed that the receiver can still operate with a BER of 10^{-11} when the input pin voltage is 50-mV around the reference. As a comparison, the received PRBS data eye at the output of the conventional receivers integrated on the same die shows signs of failure with a differential input as high as 200-mV. Moreover, this data-eye collapses completely at 100-mV, despite the fact that the measured DC sensitivity of the conventional receivers is only 50-mV. This comparison illustrates the effectiveness of the current integrating receiver in filtering high frequency noise.

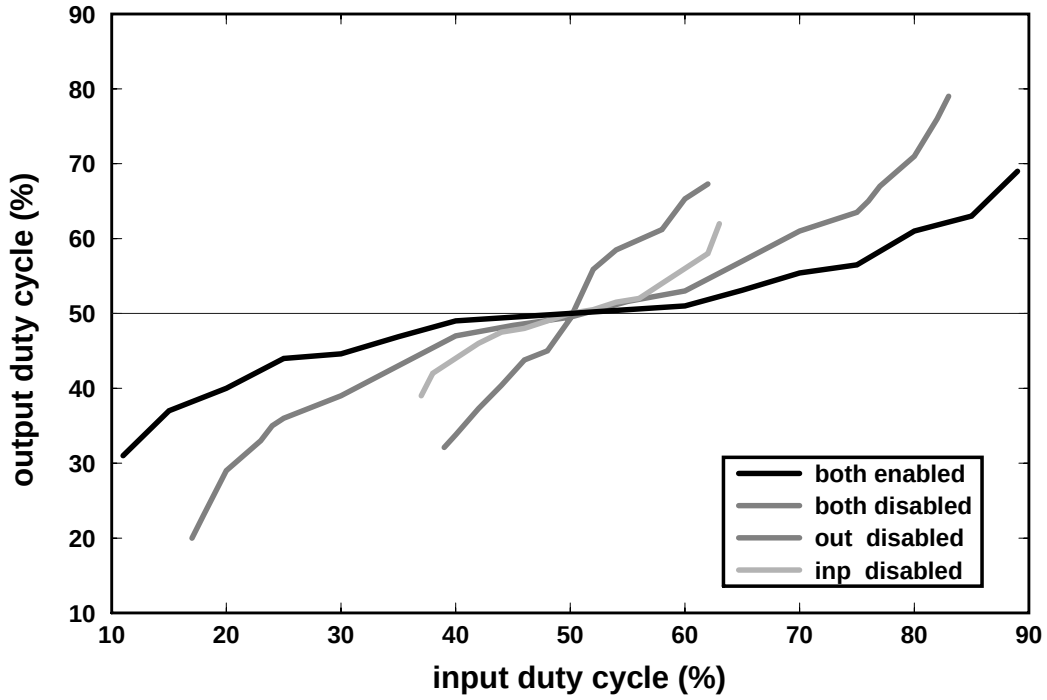

Figure 4.25: Duty cycle adjuster effectiveness

To evaluate the effectiveness of the duty cycle adjuster circuits, we varied the duty cycle of the input clock and measured the duty cycle of the sampling clock with the two DCA's selectively enabled and disabled. The results of this experiment are illustrated in Figure 4.25. It can be seen that with both DCA's enabled the chip can accommodate up to 10% duty cycle distortion, generating a sampling clock duty cycle within 1% of its nominal value. Since a distorted duty-cycle clock accumulates more duty cycle error as it passes through the band-limited buffers of the delay line, the effectiveness of the input DCA is more pronounced in this experiment.

The maximum power dissipation of the chip operating in loopback mode at 740 Mbps/pin from a 3.3-V supply was measured to be 300 mW. The performance of the prototype transceiver chip is summarized in Table 4-1.

4.6 Summary

The performance of pseudo-differential interfaces is compromised by high frequency noise injected on their reference line through the large capacitive coupling of the reference

Transfer Rate	740 Mbps/pin
Bit Error Rate	10^{-14}
System setup+hold	280 ps
Receiver Sensitivity	50 mV
Maximum Power Dissipation	300 mW (loopback mode)
Int. Receiver Power	2 mW (simulated)
Fabrication Technology	1.0/0.8 μm (drawn/effective)
Package	40 pin ceramic DIP

Table 4-1: Performance summary of the prototype transceiver

to the receiver's noisy supplies. This chapter has shown that employing a relatively simple integrator as the front end stage of the input pin receiver substantially reduces the adverse effects of high frequency noise, resulting in more robust interface operation. The front-end integrator/filter can be implemented as a differential pair integrating its tail current on a pair of output load capacitors. Implementing such a current integrator in a CMOS technology is possible, but requires addressing the issues of charge injection induced timing errors, and process and operating condition independent biasing.

A transceiver chip using the proposed current integrating receiver has been designed and fabricated in a 0.8–μm CMOS technology. The experimental results measured on the fabricated prototype demonstrate that the proposed receiver is a more robust alternative to conventional single-sampling receivers.

Chapter 5

Dual Interpolating Delay Locked Loop

Implementing a high speed interchip signalling system is more than simply a problem of realizing robust transmitter and receiver circuits. For example, the previous chapter has shown that irrespective of its inherent amplitude-noise robustness, a current integrating receiver is still limited by the offset and phase noise of its input clock. Hence, the second major design component of interchip signalling systems is the design of the phase alignment circuits, which cancel the on-chip clock amplification and buffering delays, and improve the I/O timing margins. A multitude of techniques exist to implement this phase alignment, ranging from passive off-chip delay circuits [12], to active on-chip phase-adjusting blocks [13], [14], [38]. Static methods, despite their simplicity, decrease the flexibility and increase the cost of the overall system. As a result, active on-chip phase aligning circuits have gained widespread use. The interface designs presented in Chapters 3 and 4 use a class of phase aligning circuits known as delay locked loops (DLL's). The topic of this chapter is an improved DLL architecture, which uses a combination of techniques to achieve better performance and offers great flexibility in implementing complex synchronization algorithms.

Section 5.1 focuses on architectural issues. It begins with a brief comparison between DLL's and the more commonly used VCO-based phase locked loops (PLL's). After outlining some of the disadvantages of conventional DLL architectures, the section introduces the dual interpolating DLL architecture. Section 5.2 focuses on the implementation details of the architecture, by discussing the design of the two loops, the phase interpolator, and

finally presenting the issues involved in the design of the loop-controlling digital FSM. Section 5.3 presents the experimental results measured on the prototype fabricated in a 0.8 μm CMOS technology and concluding remarks follow in Section 5.4.

5.1 Loop Architecture

As was discussed in Chapter 2, a reliable and flexible method dealing with the high speed interface synchronization problem is to use on-chip active phase aligning circuits. Generally, these circuits fall within a class of control systems known as phase locked loops (PLL's). These systems rely on the use of negative feedback to align the phase of the on-chip receive or transmit clock to the phase of an external reference clock. By appropriately embedding the on-chip clock buffer delay in the feedback path, phase locked loops can cancel out the on chip clock amplification and buffering delay. However, in order to truly improve the system timing margin, the additional fixed and time-varying timing uncertainty (i.e., offset and jitter) introduced by the phase aligning blocks must be minimized. The main obstacle in achieving this goal of timing margin maximization is again the supply and substrate noise caused by the switching of digital circuits integrated on the same die as the phase aligning blocks.

Figure 5.1 shows the two alternative control loop topologies that can be used in high speed signalling systems: VCO-based phase locked loops (PLL's), and delay line based phase locked loops (DLL's). The basic idea behind the operation of these two types of circuits is quite similar: they both try to drive the phase of their periodic output signal (clk) to have a fixed relationship with the phase of their input signal (ref-clk). A PLL employs a voltage controlled oscillator (VCO) to generate its output clock. The phase of that clock is compared with that of the reference clock by the phase detector. The output of the phase detector is filtered by the loop filter (LF), generating the loop control voltage (VC) which drives the control input of the VCO. Since a VCO integrates frequency to generate the phase of its output clock, a PLL is inherently a higher order control system. The transfer function of the system contains two poles at the origin: the first due to the phase integrating nature of the VCO, and the second due to the integrator usually embedded in the loop filter to achieve zero static phase error. To counteract the effect of these two poles, the loop

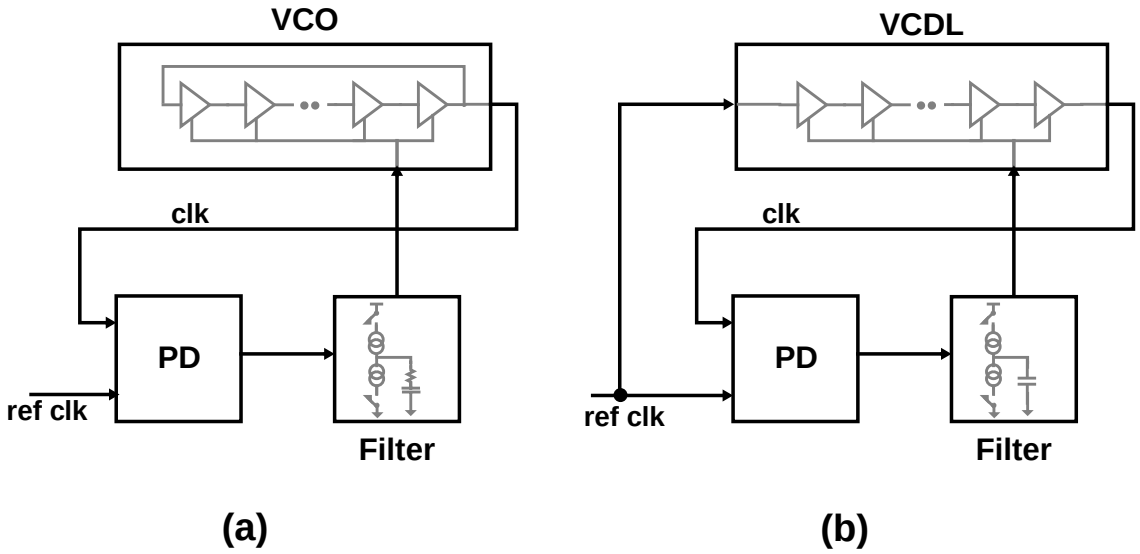

Figure 5.1: Phase locked loops (a), and Delay locked loops (b)

transfer function must contain a stabilizing zero. This zero is usually implemented in the loop filter by employing a series resistor with the integrating capacitor. This higher order nature of the PLL creates some design challenges. For example, the effects of process and environmental conditions variations on the stabilizing zero position might be detrimental on the loop stability [53], [54], [55]. On the other hand, however, using a VCO has some important advantages. First, the output clock jitter is only indirectly affected by the jitter of the reference signal, since the loop acts as a low-pass filter. Second, the output clock period can be a fraction of the reference clock period if a frequency divider is implemented in the loop feedback path. This frequency multiplication property is the main reason for the widespread adoption of PLL's in applications such as microprocessor clock generation [56], [57], [58]. Moreover, since the VCO inherently generates a periodic clock signal, PLL's utilizing appropriate phase detector designs are commonly used in clock and data recovery applications [59].

Delay locked loops on the other hand, make use of the fact that in many applications the reference clock already has the correct frequency [42]. Hence, instead of generating their output clock with a VCO, DLL's use a voltage controlled delay line (VCDL) which generates the output clock by delaying its input clock by a controllable time delay. The phase of the VCDL output clock (clk) is compared by the phase detector with the phase of

the reference clock. The output of the phase detector is filtered by the loop filter generating the control voltage VC. This control voltage drives the VCDL control input closing the negative feedback loop. In this system the VCDL is simply a delay gain element. Thus the loop can employ a single-pole filter without a stabilizing zero, which can be implemented by a single integrator (e.g., a charge pump and a capacitor). This control system is unconditionally stable resulting in a much easier design. In addition a DLL can be more easily implemented as a bang-bang control system, in which the phase detector output is simply a binary up-down phase error indication, rather than a voltage proportional to the instantaneous phase error. The significance of this property for high speed interface applications is that the DLL phase detector can be a replica of the input pin receiver, resulting in an optimum placement of the receiving clock edge. In contrast, PLL's due to frequency acquisition constraints usually rely on a specific state-machine based phase-frequency detector design, thus resulting in a suboptimal placement of their output clock edge.

In the noisy environment of a digital IC, the most important difference between PLL's and DLL's is in the way they react to supply or substrate induced phase noise. Typically, a PLL will have higher supply or substrate noise sensitivity than a DLL comprising identical delay elements [43]. An intuitive explanation for this difference in performance is that a change on the supply or substrate voltage of a VCO results in a change on its operating frequency. This frequency difference results in an increasing phase error, which keeps accumulating until the corrective action of the loop feedback takes effect. In contrast, the change on the supply of a VCDL results just in a delay change. Since the VCDL does not recirculate its output clock, the resulting phase error does not accumulate, and starts to decrease immediately with a rate proportional to the loop bandwidth.

The performance difference between PLL's and DLL's is illustrated in Figure 5.2 which shows the simulated phase error transient of a PLL and a DLL under a supply voltage step. Both the PLL and the DLL are locked to a reference clock with a frequency of 250 MHz. The VCO and the VCDL comprise six voltage controlled delay elements with a supply sensitivity of 1.8^{o} per volt each (i.e. a 1-volt change in the supply of the VCO or the VCDL changes the delay through each element by 20-ps). A 300-mV supply step is applied on both the VCO and the VCDL 200 nsec after the start of the simulation. As can

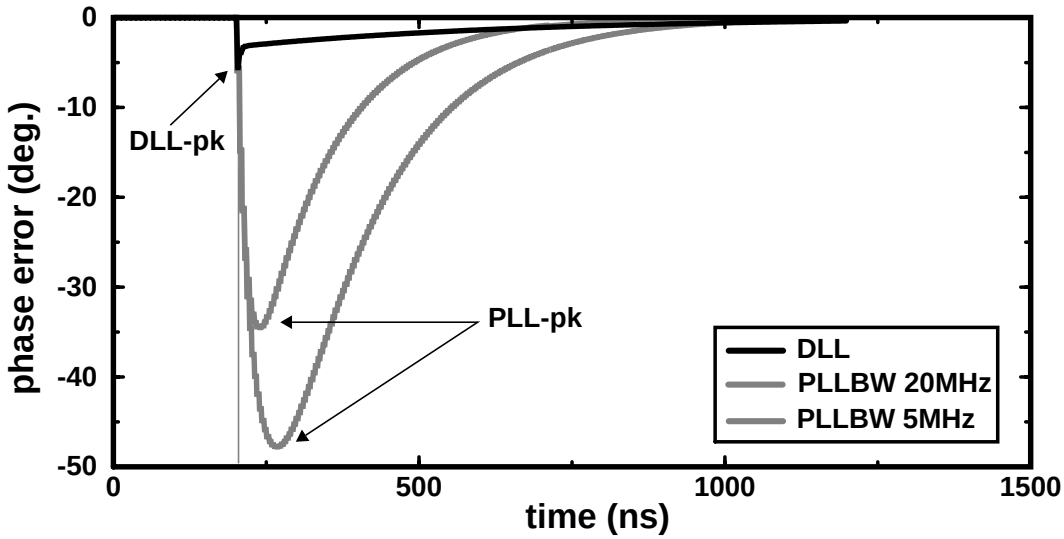

Figure 5.2: Simulated supply step response of a PLL and DLL

be seen in Figure 5.2, the PLL peak phase error is generally much larger than the initial VCO period error (i.e., $6.5^{o} = 12 \times 0.3 \times 1.8^{o}$). The magnitude of this error depends both on the delay element supply sensitivity and the loop bandwidth. A larger loop bandwidth expedites the loop correcting action, thus resulting in less phase error accumulation and minimizing the peak phase error. In contrast, the DLL phase error depends only on the supply sensitivity of the delay elements, and its peak occurs during the first clock cycle after the supply step. Even in the best case where the PLL bandwidth is 20-MHz the difference on the peak phase error is approximately a factor of 6 larger than that of the DLL (increasing the PLL bandwidth further that 1/10 of the operating clock frequency compromises the loop stability).

It should however be noted, that other factors such as the quality of the system clock and the supply sensitivity of the final on-chip clock buffer can affect design trade-offs. For example, the above comparison does not include the supply sensitivity of the final on-chip clock buffer, which typically comprises CMOS inverters. Since the supply sensitivity of inverters is approximately 5× worse than that of well designed delay elements, a long buffer chain can contribute to a significant fraction of the total jitter. This narrows the performance gap between a PLL and a DLL. However, in applications such as high speed intra-system interconnects, where a good quality system clock of the right frequency is

available, using a DLL still maximizes the timing margins both because it exhibits lower supply and substrate induced phase noise, and because it can more readily use the input pin receiver as a phase detector.

Despite their advantages over VCO based PLL's, conventional DLL architectures suffer from two important disadvantages. In a design in which the VCDL input is the reference clock, jitter on that clock directly propagates to the DLL output. As discussed in Section 4.4.1, this all-pass filter behavior with respect to the frequency of the jitter of the reference clock results in reduced I/O timing margins. The solution outlined in Section 4.4.1 is to use a separate differential clock as the input to the delay line. In this way jitter induced on the reference clock, by the noisy transmitter chip or by coupling on the shared reference line, does not propagate to the output clock.

A more important problem is that a VCDL does not have the cycle-slipping capability of a VCO. Therefore, at a given operating clock frequency, the DLL can delay its input clock by an amount bounded by a minimum and a maximum delay. As a consequence, extra care must be taken by the designer, so that the loop will not attempt to lock towards a delay which is outside these two limits. The typical solution, presented in Section 4.4.1, is to extend the VCDL range and use a FSM that controls the loop start-up. However, DLL's relying on quadrature phase mixing completely eliminate this problem [43], [62]. The quadrature mixing approach is based on the fact that two clocks with a phase shift of 90^o can be easily generated, given a clock of the correct frequency. The quadrature clocks are then fed to a controllable phase interpolator, which can generate a clock whose phase can span the full 0^o-360^o phase interval. This approach eliminates the limited phase range problem of conventional DLL's since the phase mixer can essentially rotate the output clock phase infinite times providing seamless switching at the quadrant boundaries. The main disadvantage of quadrature mixing is that the output of the phase mixer is a clock with a slew rate inherently limited by $4 \cdot V_{SW}/T$ where V_{SW} is the is the output swing of the phase mixer, and T the operating clock period. This slow clock exhibits increased dynamic noise sensitivity, thus degrading the jitter performance of quadrature mixing DLL's [63], [64].

Another approach, relying on phase interpolation, avoids the jitter sensitivity problem by interpolating between phases that are spaced by 30^o instead of 90^o [65]. These tighter spaced phases are generated by locking a 6-stage VCO-based PLL at the reference clock frequency, and tapping out the true and complementary outputs of the VCO buffers. Even though this approach minimizes the slew-rate induced jitter sensitivity of the interpolator output, it still exhibits increased supply sensitivity because of phase-error accumulation in the VCO. The architecture presented in the following section eliminates that problem, based on the observation that a DLL can be used in place of a VCO-based PLL to generate tightly spaced clock phases.

5.1.1 Dual Delay Locked Loop Architecture

Figure 5.3 shows a high level block diagram of the proposed architecture. This architecture is based on cascading two loops. A conventional first-order core DLL so that its line delay spans 180^o. Assuming that the delay line of the core DLL comprises six buffers, their outputs are six clocks which are evenly spaced by 30^o. The peripheral digital loop selects a pair of clocks, ϕ and ψ, to interpolate between. Clocks ϕ and ψ can be potentially inverted in order to cover the full 0^o-360^o phase range. The resulting clocks, ϕ' and ψ', drive a digitally controlled interpolator which generates the main clock Θ. The phase of this clock can be any of the N quantized phase steps between the phases of clocks ϕ' and ψ', where 0-N is the interpolation controlling word range.

The output clock of the interpolator (Θ), drives the main loop phase detector which compares it to the reference clock. The output of the phase detector drives the peripheral loop finite state machine (FSM), which controls the phase selection, the selective phase inversion, and the interpolator phase mixing weight. The FSM moves the phase of the clock Θ according to the phase detector output. In the more common case this means just changing the interpolation mixing weight by one. If, however, the interpolator controlling word has reached its minimum or maximum limit, the FSM must change the phase of clock ϕ or ψ to the next appropriate selection. This phase selection change might also involve an inversion of the corresponding clock if the current interpolation interval is adjacent to the 0^o or 180^o boundary. Since the phase selection changes happen only when the

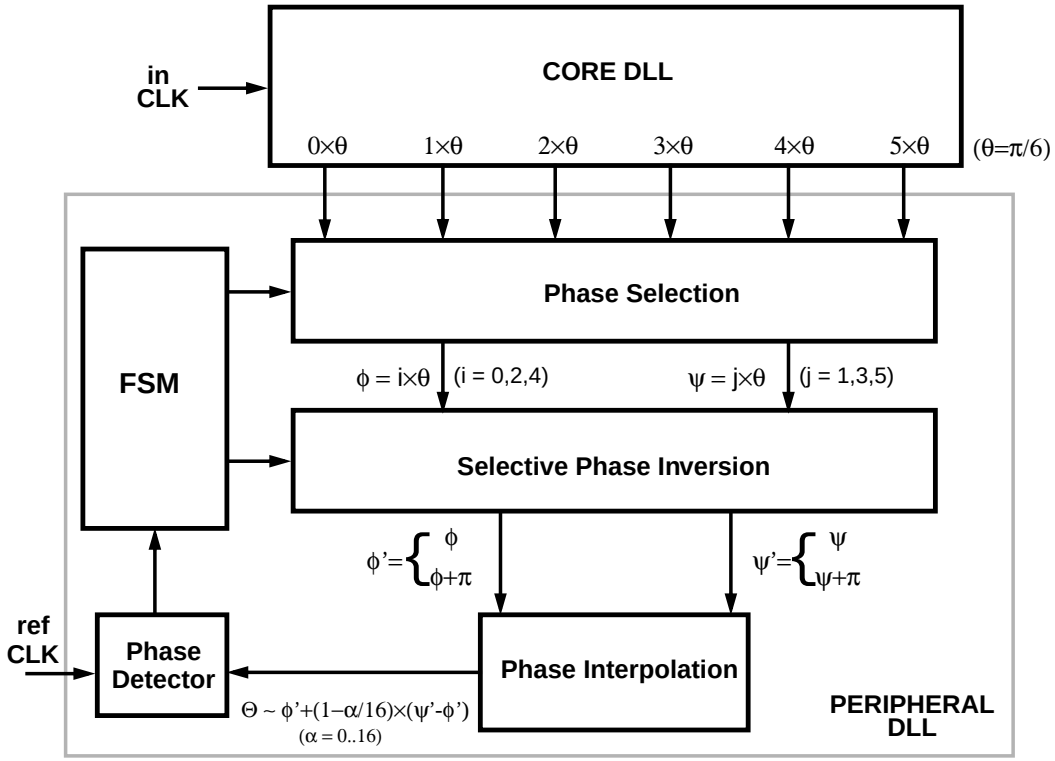

Figure 5.3: Dual interpolating DLL architecture

corresponding phase mixing weight is zero, no glitches occur on the output clock. The digital “bang-bang” nature of the control loop results in dithering around the zero phase error point when the loop is in lock. The dither amplitude is determined by the interpolator phase step and the delay through the peripheral loop.

In this architecture the output clock phase can be rotated, so no hard limits exist in the loop phase capture range: the loop provides unlimited (modulo 2π) phase shift capability. Unlike more conventional DLL designs (e.g., the DLL described in Section 4.4.1 or the designs discussed in [42], [60], [61]), noise transients can never result in a condition in which the loop has exhausted its locking range. Thus, this architecture eliminates boundary conditions and phase relationship constraints. The only requirement is that the DLL input clock and the reference clock be plesiochronous (i.e., their frequency difference is bounded [4]), making this architecture suitable even for clock recovery applications. Since the system does not use a VCO, it does not suffer from the phase error accumulation problem of conventional PLL’s [65]. Moreover, since the phase interpolator input clocks are spaced by just 30^{o}, the output of the phase interpolator does not exhibit the noise sensitiv-

ity of the approaches relying in quadrature mixing. Finally, the fact that the phase capture algorithm can be completely implemented in the digital domain provides for larger flexibility in its implementation.

5.1.2 Dual Loop Dynamics

Cascading two loops can compromise the overall system stability and lead to undesired jitter peaking effects. However, as the analysis in this section will show, this dual loop architecture does not exhibit any jitter peaking irrespective of the dynamics of the two loops. The behavior of the DLL can be analyzed with respect to two types of perturbations: *(i)* input or reference clock delay variations, and *(ii)* delay variations resulting from supply and substrate noise. The frequency response of the dual loop can be analyzed with a continuous time approximation, in which the sampling operation of the phase detectors and the digital nature of the peripheral loop are ignored. This approximation is valid for core and peripheral loop bandwidths at least a decade below the operating frequency. This constraint needs to be satisfied anyway in a DLL in order to eliminate the effects of higher order poles resulting from the delays around loop.

Figure 5.4 shows the a linearized model of the dual loop. The model includes both the loop clocks $D_{IN}(s)$ and $D_{REF}(s)$, as well as delay errors introduced by supply or substrate noise $D_N(s)$. Each of the two loops is modeled as a single pole system, in which the input, output, and error variables are delays, similarly to the single loop analysis discussed in [66]. For example, the output delay of the core loop $D_{OC}(s)$ (in seconds) is the delay established by the core loop delay line, while the input delay $D_{IN}(s)$ is the delay for which the core loop phase detector and charge pump do not generate an error signal. Since the core loop VCDL spans half a clock cycle, $D_{IN}(s)$ is equal to half an input clock period. By using these loop variables the input to output transfer function of the core loop can be easily derived:

$$\frac{D_{OC}(s)}{D_{IN}(s)} = \frac{1}{1 + s/p_c} \tag{5-1}$$

where p_c (in rads/sec) is the pole of the core loop as determined by the charge pump current, the phase detector and delay line gain, the loop filter capacitor, and the operating clock frequency. Similarly, the supply induced delay-error to overall delay error transfer function of the core loop can be shown to be:

$$-\frac{D_{EC}(s)}{D_N(s)} = \frac{s/p_c}{1 + s/p_c} \tag{5-2}$$

where $D_N(s)$ is the additional delay introduced in the core loop from supply or substrate noise, and $D_{EC}(s)$ is the delay error seen by the core loop phase detector. This transfer function indicates that noise induced delay errors can be tracked up to the loop bandwidth and that the response of the loop to a supply step consists of an initial step followed by a decaying exponential with a time constant equal to $1/p_c$.

Before proceeding to analyze the response of the dual loop, it should be noted that the linearized model of Figure 5.4 uses a simplifying assumption. The assumption is that the delay error $D_N(s)$ introduced by supply or substrate variations is identical in both loops and does not depend on the state of the phase selection multiplexers. Since the supply and substrate sensitivity of the peripheral loop depends on the phase selection and will be typically higher due to the presence of the final CMOS system clock buffer, this assumption is not necessarily accurate. However, that assumption does not affect the conclusions drawn

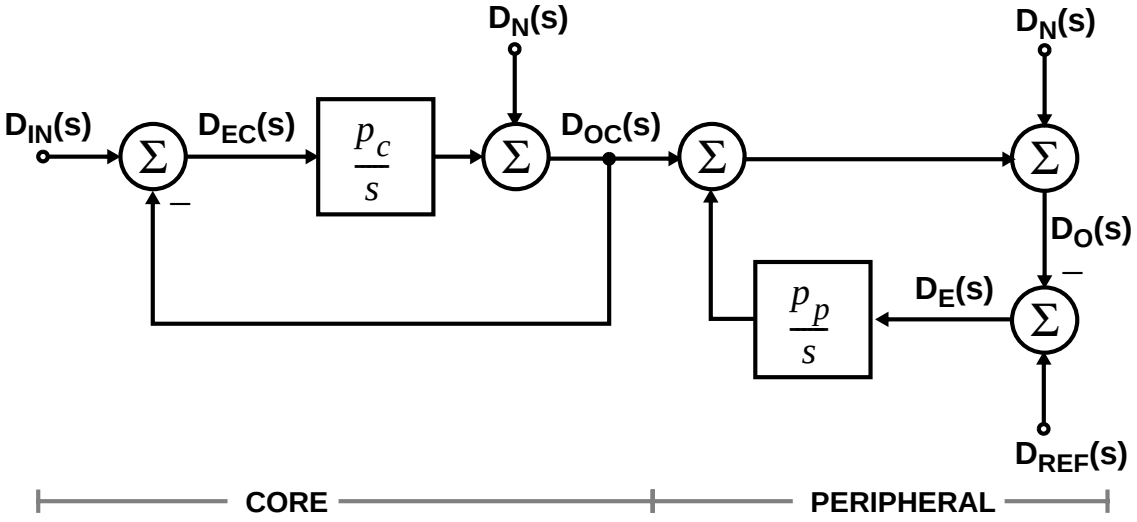

Figure 5.4: Linearized dual DLL model

below about the stability of the loop, since it only removes a modifying constant, which is equal to the ratio in the delay sensitivities of the two loops. This constant only affects the relative location of the poles and zeros of the resulting transfer function and, as it will be shown below, the loop is unconditionally stable irrespective of the relation between the individual poles and zeros.

Using the model of Figure 5.4, it is straightforward to show that the transfer function $D_O(s)/D_{REF}(s)$ of the peripheral loop is identical in form to that of the core loop. This result agrees with intuition, since reference clock perturbations do not affect the core loop. More interesting is the transfer function of the input clock $D_{IN}(s)$ to dual loop error $D_E(s)$, since changes in the period of the input clock will cause both the core and peripheral loop to react. Based on Equations (5-1) and (5-2) this transfer function can be shown to be:

$$-\frac{D_E(s)}{D_{IN}(s)} = \frac{s/p_p}{(1+s/p_c)\cdot(1+s/p_p)} \tag{5-3}$$

This bandpass transfer function exhibits no peaking at any frequency regardless of the relative magnitudes of p_c and p_p. The step response of the system, shown in Figure 5.5-(a), reveals that unit-step changes in $D_{IN}(s)$ (i.e., step changes in the input clock period) will initially peak to a value less than unity. This initial value is determined by the ratio of the poles of the core and peripheral loops. Moreover, as the magnitude of p_p increases, the disturbance on the output is reduced since the peripheral loop compensates quickly for disturbances at the output caused by changes of the input clock phase.

As was discussed in the previous section, in this particular architecture the DLL input and reference clocks are split in order to avoid clock jitter propagation from the reference to the output clock. If, however, the design is altered such that in-CLK and ref-CLK are identical the resulting transfer function is:

$$\frac{D_O(s)}{D_{IN}(s)} = \frac{1+s\cdot(p_c+p_p)/(p_c\cdot p_p)}{(1+s/p_c)\cdot(1+s/p_p)} \tag{5-4}$$

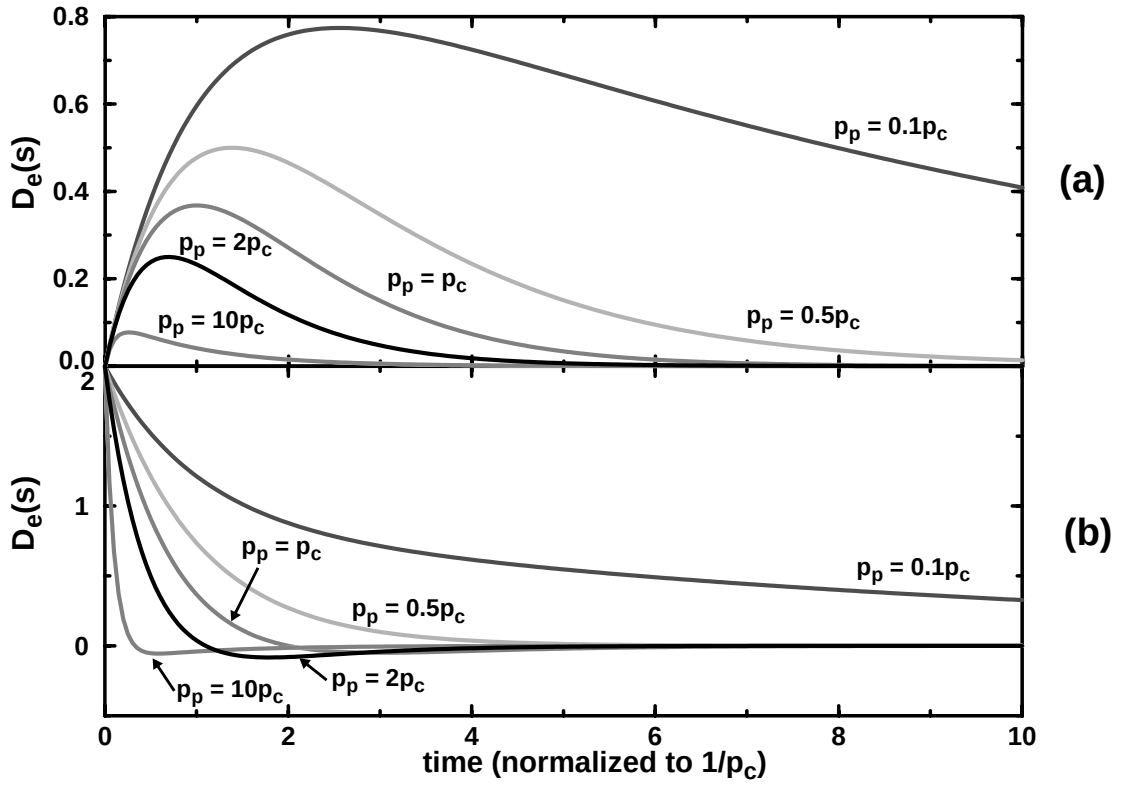

Figure 5.5: Dual loop step-response to: change in clock period (a), and supply noise (b)

The transfer function of Equation (5-4) exhibits a low-pass peaked behavior. However, the resulting frequency domain peaking is small, exhibiting a maximum of 15% when p_p=p_c. This peaking is less than 5% as long as p_p and p_c are an order of magnitude apart in frequency. Nevertheless, as long as the phase noise of in-CLK and ref-CLK are uncorrected, the non-peaking transfer function of Equation (5-3) governs the system response.

Perhaps more interesting is the transfer function from supply or substrate noise induced delay errors $D_N(s)$ to the delay error of the dual loop $D_E(s)$:

$$-\frac{D_E(s)}{D_N(s)} = \frac{(1 + 2s/p_c) \cdot s/p_p}{(1 + s/p_c) \cdot (1 + s/p_p)} \tag{5-5}$$

Equation (5-5) shows that the system does not amplify supply induced delay errors of any frequency, since the location of the last zero can never be above that of the poles. The step response of the system is plotted in Figure 5.5-(b) for various ratios of the core to peripheral pole frequencies. Under all conditions the initial delay error is equal to twice the

injected unity error $D_N(s)$ since this error added on both the core and peripheral loops. When the peripheral loop bandwidth is less than half that of the main loop, there is no overshoot in the dual loop step response. This result occurs because the core loop compensates for its delay error quickly, while the slower peripheral loop compensates for the output delay error later. When the pole frequencies of the two loops are very close, the system overshoots since the peripheral loop compensates for the output delay error at approximately the same rate as the peripheral loop. The worst case overshoot of approximately 4.5% of the initial disturbance occurs when the peripheral loop bandwidth is twice that of the core loop. As the peripheral loop bandwidth increases, the overshoot becomes progressively smaller since the peripheral loop corrects for both the peripheral and core delay errors. Subsequently the influence of the slower core loop correction on the output delay error is compensated by the peripheral loop. Therefore, even in the worst case the dual loop cascade exhibits only minor overshoot.

5.2 Circuit Design

A more detailed block diagram of the dual loop is depicted in Figure 5.6. As discussed in the previous section, this implementation uses a separate differential clock as the input to the delay line. To minimize the effects of input clock duty cycle imperfections and common mode mismatches, a duty cycle adjuster (DCA), is employed after the first clock receiving buffer. The DCA design is based on the differential offset cancellation circuit discussed in [51]. The 50% duty cycle clock drives the core DLL. The core delay line consists of six differential buffers. An extra pair of buffers B_0, B_Π generate two clocks which drive the core loop 180^o phase detector. The output of the phase detector controls the charge pump which forces clocks C_0 and C_Π to be 180^o out of phase. Since all the buffers in the core delay line (including B_0 and B_Π) have the same size, all the core VCDL stages have the same fan-out and delay. Therefore, forcing C_0 and C_Π to be 180^o out of phase generates six evenly spaced by 30^o clocks at the outputs of the core delay line.

The phase selection and phase inversion multiplexers are differential elements controlled by the core loop control voltage. In order to eliminate jitter sensitive slow paths, all buffers in the clock path need to have approximately the same bandwidth. For this reason

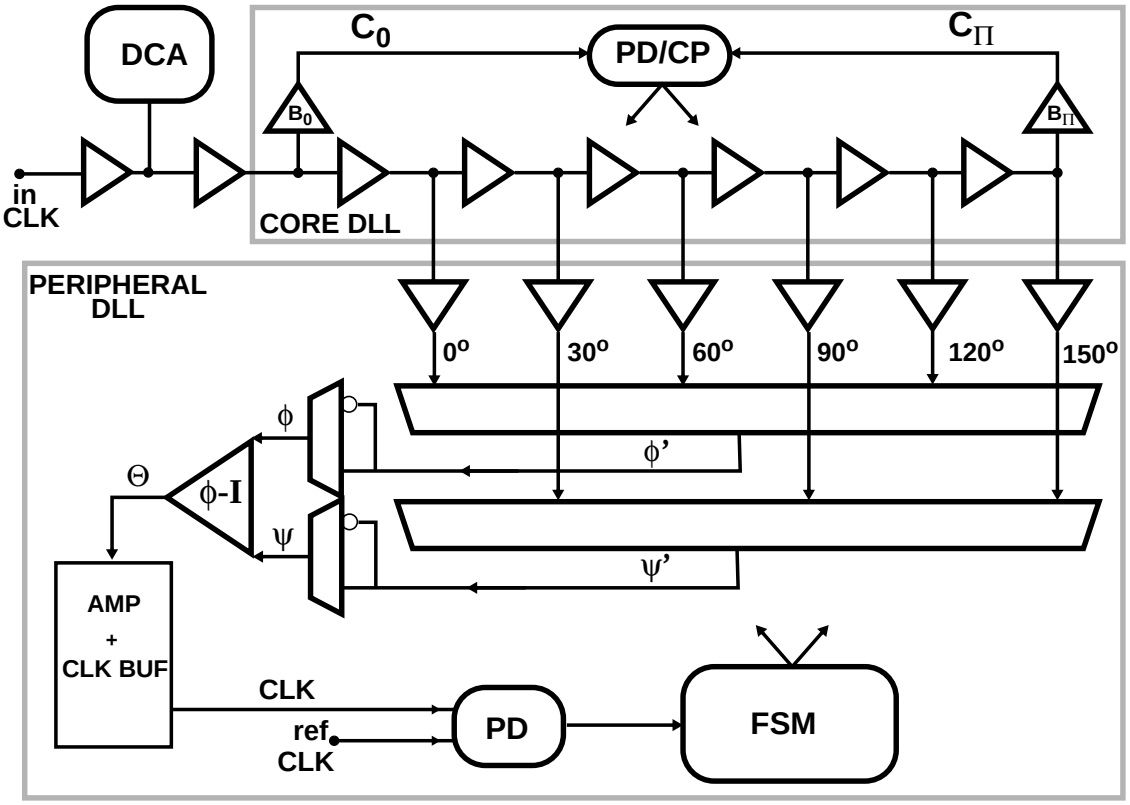

Figure 5.6: Dual DLL block diagram

the phase selection and inversion is implemented as a combination of a 3-to-1 and a 2-to-1 multiplexers, instead of a single 6-to-1 differential multiplexer with lower total power. Since the phase selection multiplexer can affect the phase shift of the core delay line through data-dependent loading, the six output clocks are buffered before driving the phase selection multiplexers. This way changing the multiplexer select does not affect the core delay line phase shift.

The outputs ϕ, ψ of the phase inversion multiplexers drive the phase interpolator which generates the low swing differential clock Θ. This clock is then amplified and buffered through a conventional CMOS inverter chain to generate the main clock (CLK). The peripheral loop phase detector (PD) compares that clock to the reference clock generating a binary phase error indication that subsequently drives the FSM. The FSM, based on the phase detector output, selects phases ϕ, ψ and controls the phase interpolation. In this particular implementation, in order to accurately measure phase offsets, the peripheral loop phase detector was implemented as a symmetric NAND-based circuit [42]. However, the

use of this particular circuit is not inherent in the loop architecture. A replica of the input pin receiver would be normally used in high speed interface applications to maximize the system timing margins.

5.2.1 Core Delay Locked Loop

To minimize the jitter supply sensitivity all the delay buffers in the design, from the input clock (in-CLK) to the output of the phase interpolator (Θ), use differential elements with symmetric impedance loads and replica feedback biasing [51]. Although the delay that can be generated by a VCDL comprising such symmetric load buffers has a very broad range, the overall gain of the VCDL has an inverse square-law dependence to the core loop control voltage. This effectively narrows the operating frequency range of the core loop, since the loop bandwidth increases with decreasing operating frequency. To counteract this effect the core-loop utilizes the self-biasing technique of [66]. As illustrated in Figure 5.7, the current of the core loop charge pump is scaled along with current of the VCDL buffers. Voltage V_{CN} is generated through the replica-feedback biasing circuit, while V_{CP}' is a buffered version of the charge pump control voltage V_{CP}. In addition to the core VCDL buffers, voltages V_{CP}' and V_{CN} control the differential buffer elements of the peripheral loop. This ensures that all the buffers in the design have approximately equal delays, and that the edge rate of the interpolator input clocks (ϕ, ψ) scale with the operating frequency of the loop. The significance of this edge rate scaling property is further discussed in the next section.

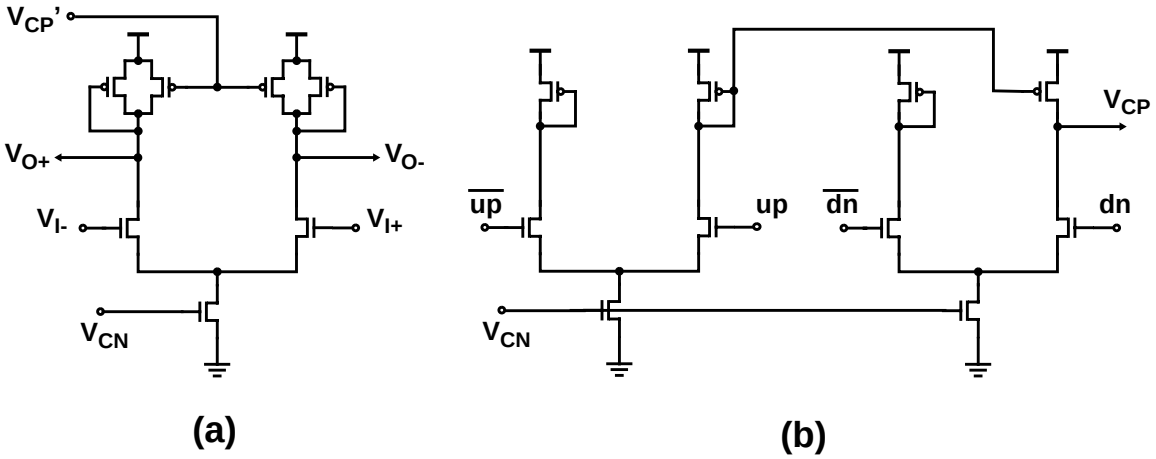

Figure 5.7: Core loop delay buffer (a), and charge-pump (b)

The sensitivity of the dual loop architecture to the core loop phase offset depends on the particular application. In applications where the dual DLL is used just to generate a clock whose phase is directly controlled by the phase detector output, the phase offset of the core loop does not affect the system phase offset. In this case the loop operation will not be affected as long the core loop phase offset is bounded. An absolute core loop offset less than 30^o ensures monotonic switching at the 0^o and 180^o interpolation boundaries, so the interpolating loop functions correctly, albeit with a larger than nominal interpolation phase step. Core loop phase offsets larger than 30^o will result in a hysteretic locking behavior at the 0^o and 180^o interpolation boundaries, which increases the peripheral loop dither jitter whenever the loop attempts to lock at this point.

The dual loop operation becomes more sensitive to core loop phase offsets in case the designer chooses to use this architecture to generate an additional clock which is offset by 90^o relative to the reference clock. In such an application the quadrature clock would be generated by using an extra pair of phase selection and inversion multiplexers. The select lines of these multiplexers need to be offset by three relative to the select lines of the multiplexers generating the main clock. This would create a 90^o offset between the corresponding interpolation intervals, resulting in the required quadrature phase shift. In this case, the core loop phase offset would impact the quadrature phase, if the select lines of the extra multiplexers happen to wrap around the 0^o or 180^o interpolation interval boundaries.

Even though the prototype described here does not implement quadrature phase generation, a low offset phase detector and careful matching of the layout were employed to ensure uniform spacing of the six clocks. A self-biased DLL requires a linear phase detector. The self-biased DLL design described in [66], uses a conventional state-machine based phase frequency detector (PFD) [67], [68]. The presence of the third state in the PFD creates some start-up problems: the phase detector can start driving the core DLL towards a locking point below the minimum achievable delay, if one of the C_0, C_Π pulses is lost during start-up. To avoid this problem, the core loop employs the simple design shown in Figure 5.8. The absence of extra state in this design eliminates any start-up false locking conditions: if the core loop is reset to the minimum delay point this phase detector

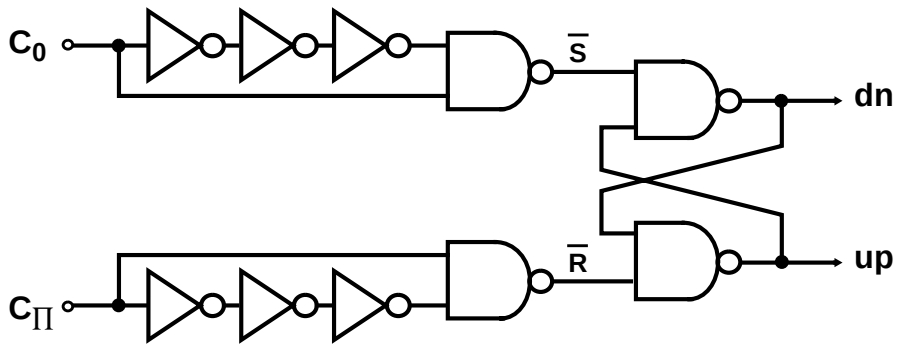

Figure 5.8: Core loop linear phase detector

will eventually drive towards a locking point of 180^o, regardless of whether some clock pulses are lost during the start-up transient. In addition, the symmetric structure and the pulse triggered nature of this design minimize the core loop phase offset.

The core of the design of Figure 5.8 is a conventional NAND based S-R latch phase detector [55]. The S-R latch phase detector, ensures a 180^o phase shift between the falling edges of its inputs only when the duty cycle of the two input clocks is identical. However, when the duty cycle of the two input clocks is different, this mismatch will propagate as a core loop phase locking offset. This happens because an unbalanced overlap of the two input clocks causes the output of the S-R latch to have a duty cycle deviating from 50%. To compensate for this effect, the design in Figure 5.8 augments the basic S-R latch with two pulse generators which generate a low pulse on the positive edges of the input clocks. Since potential overlaps are minimized, the design can tolerate large duty cycle imperfections and still provide an accurate 180^o lock in the core loop.

Figure 5.9 shows the simulated transfer characteristics of the phase detector and charge pump over three extreme process and environment conditions. The cycle time of the two input clocks is set at 4 ns, while their duty cycles are mismatched by 0.5 ns such that the duty cycle of C_0 is 37.5% while the duty cycle of clock C_Π is 62.5%. It can be seen that the transfer function is linear and has no offset or dead-band around the 2-ns (180^o) locking point. However, the combination of input pulsing and duty cycle imperfections result in nonlinear transfer function characteristics at the vicinity of the boundaries of the locking range (i.e., 0 and 4-ns). The only effect of this nonlinearity is that the core loop can exhibit an initial slew-rate limited reduction of its phase error, since the output

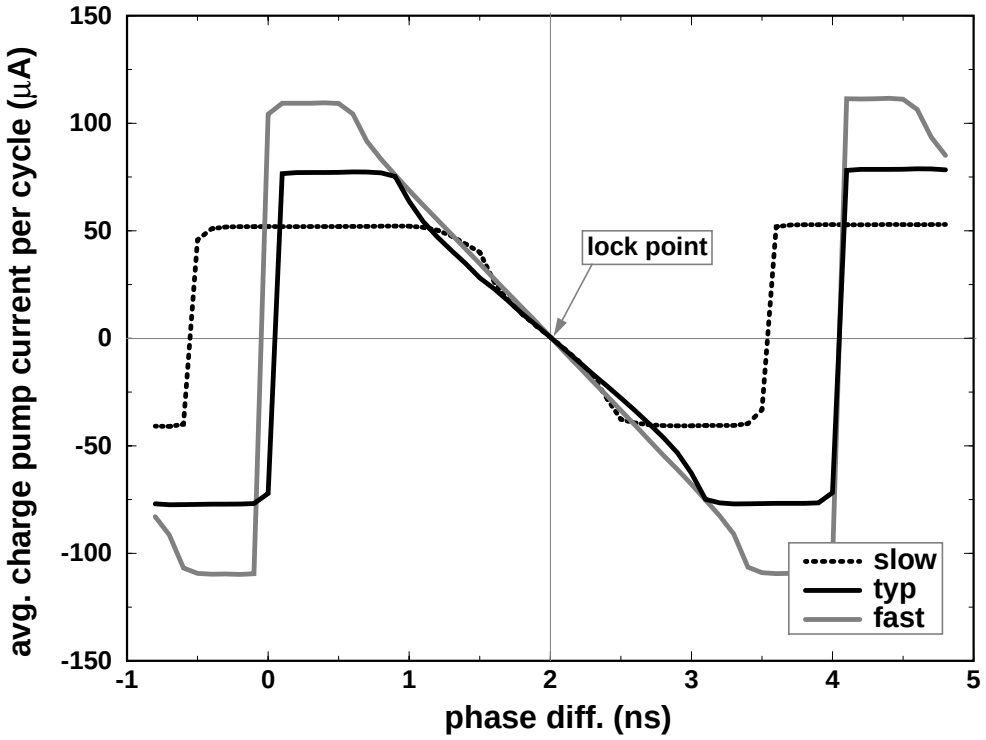

Figure 5.9: Simulated transfer function of phase detector and charge pump

current of the phase-detector/charge-pump block is constant in this range. Nevertheless, after the initial phase error has been reduced, so that the phase detector operates within its linear region, the core loop will exhibit a conventional single pole response. Harmonic locking problems, common in PLL's using S-R phase detectors, are eliminated in this design, since the core loop is reset to its minimum delay at system start-up.

5.2.2 Phase Interpolator

The most critical circuit in the design of the peripheral digital loop is the phase interpolator. The phase interpolator is a dual input delay buffer which receives two clocks, ϕ' and ψ', and generates the main clock Θ. Ideally, the phase of clock Θ is the weighted sum of the phases of clocks ϕ and ψ, which are delayed by a single buffer delay from the interpolator inputs (Figure 5.10). Interpolators with static phase mixing weights can be constructed by shunting the output of two half-sized CMOS or current mode delay buffers [51]. Variable weight designs employ current mode buffers in which the analog interpolation weight controls the ratio of the tail currents of the two buffers [69], [70].

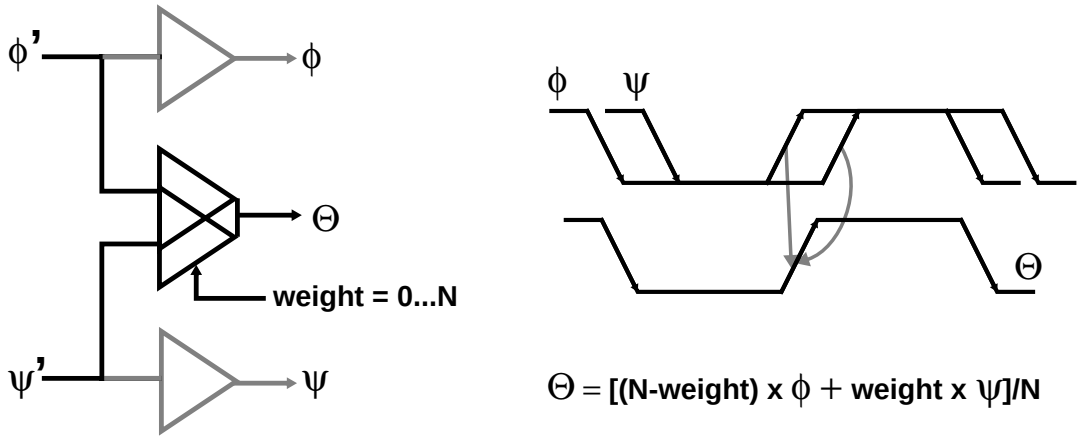

Figure 5.10: Timing generation using phase interpolators

The interpolator used in this DLL incorporates two D/A converters within the dual input buffer, converting the digital weight code generated by the FSM to two complementary buffer currents which affect the phase of the output clock Θ. A simplified model of the interpolator is depicted in Figure 5.11-(a). In this model the switching action of the two buffers is modeled by applying the corresponding current to the output at a time controlled by the timing of the two input edges ϕ' and ψ'. The delay of the interpolator is intrinsically controlled by its output *RC* time constant. However, as illustrated in Figure 5.11-(b), changing the currents of the two branches affects the overall delay by controlling the swing of the branch that switches first. The interpolator output voltage as a function of time is given by:

$$V_O(t) = V_{CC} + R \cdot I \cdot \left[(1-w) \cdot u(t) \cdot \left(e^{-\frac{t}{RC}} - 1 \right) + w \cdot u(t-\Delta t) \cdot \left(e^{-\frac{t-\Delta t}{RC}} - 1 \right) \right] \quad (5\text{-}6)$$

where *R* is the total interpolator resistive load, *C* the output capacitance, *w* is the interpolation weight, and Δt the time delay between the two input phases. Equation (5-6) shows that the interpolator delay depends not only on the interpolation weight but also on the time delay between the interpolator inputs. Using Equation (5-6) the interpolator transfer function (*w-to-delay*) can be derived. Figure 5.12 illustrates the transfer function, for varying values of Δt. Both Δt and the interpolator delay are normalized to the output *RC* time

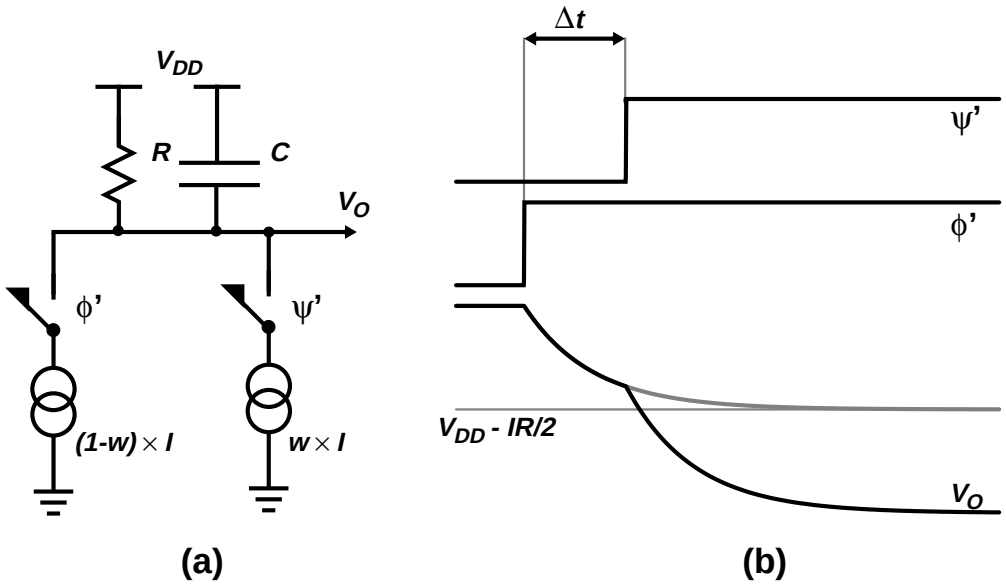

Figure 5.11: Simplified model of the phase interpolator

constant. Moreover, the interpolator delay in this figure is referenced to the delay of the circuit with w=0. Figure 5.12 shows that the interpolator transfer function becomes increasingly nonlinear, as the delay between the two step inputs becomes larger than the *RC* time constant of the circuit. Although in a real implementation this nonlinearity would be mitigated due to the finite slew-rate of the input phases, it is a strong argument for retaining approximately the same *RC* time-constant throughout the peripheral loop. This delay equalization not only increases the interpolator linearity, but it also ensures that the interpolator output does not settle to a value equal to half the final swing thus increasing the jitter sensitivity.

In this design, the interpolator is enclosed in the peripheral loop feedback. Thus, interpolator nonlinearity is not as important a requirement as in open-loop designs [10], [70], because it does not affect the loop static phase error. The most important requirement in this design is that the interpolation process be monotonic to ensure that no jitter-increasing hysteresis exists in the loop phase capture characteristics. Additionally, the phase step must be minimized since it determines the loop dither amplitude. Interpolation nonlinearities become a concern only when they overly increase the loop dither jitter above the 2^o nominal interpolation step (i.e., 1/16th of the 30^o interpolation interval). Another important requirement is that the design should provide for seamless interpolation-boundary

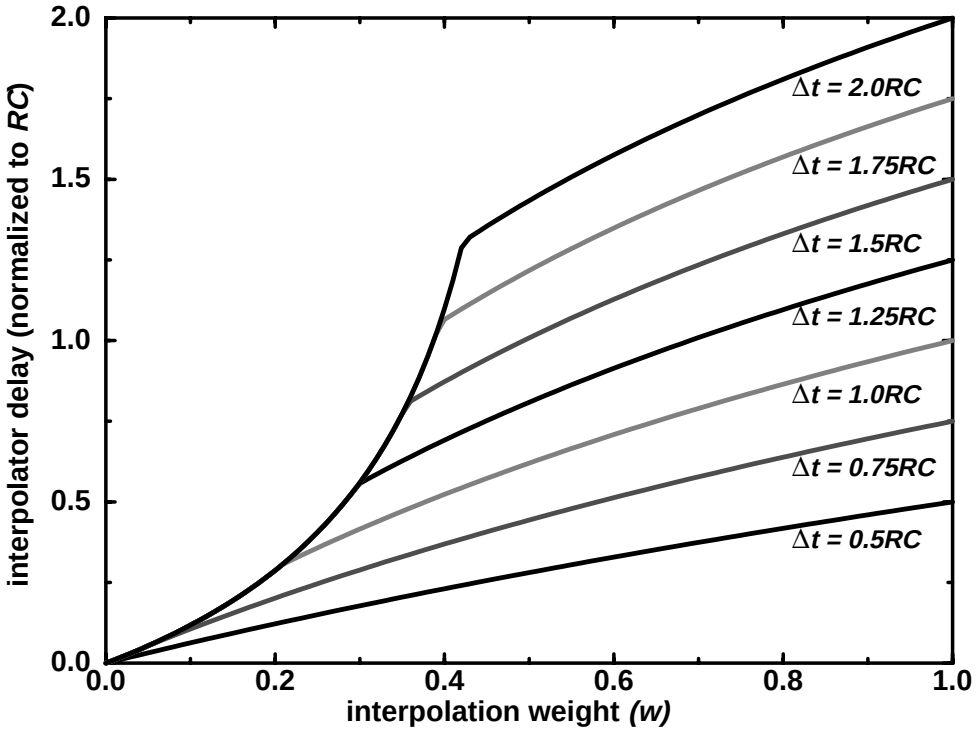

Figure 5.12: Interpolator transfer function with varying Δt

switching. This means that when the input code is such that the weight on one of the input clock phases is zero this clock should have no influence on the output.

The interpolator design used in the prototype DLL chip is shown in Figure 5.13. This design is a dual input differential buffer which uses the same symmetric loads as all the core VCDL buffers and peripheral loop multiplexers. The bias voltages V_{CP}' and V_{CN} are identical with those biasing the core loop. Moreover, the multiplexers driving the interpolator are differential elements biased by the same control voltages V_{CP}' and V_{CN}. Therefore the transition time of the interpolator input clocks is larger than the minimum delay through the interpolator and the two input transitions overlap, independently of the operating clock period. This increases the interpolator linearity and the loop operating range.

The current sources of the two differential pairs comprising the interpolator of Figure 5.13-(a) are thermometer controlled elements. The thermometer codes are generated by a 16-bit long up/down shift register which is controlled by the peripheral loop FSM. Essentially this design integrates two current mode single ended DAC's into the interpolator design. This design (type-I) does not completely satisfy the seamless bound-

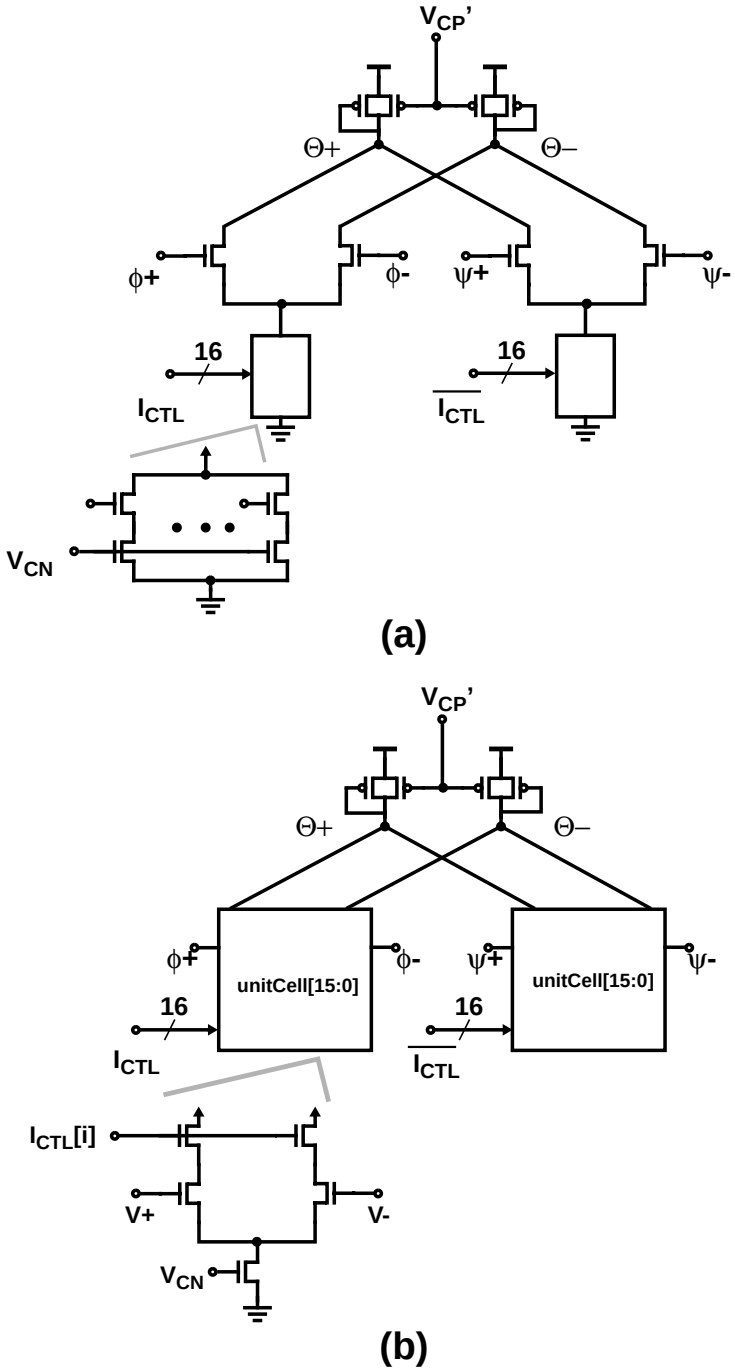

Figure 5.13: Alternative interpolator designs (a) type-I, and (b) type-II

ary-switching requirement. Even when the current through one of the differential pairs is zero, the input transition still influences the output of the interpolator. This influence is due to the capacitive coupling created by the gate to drain capacitance of the differential pair input transistors.

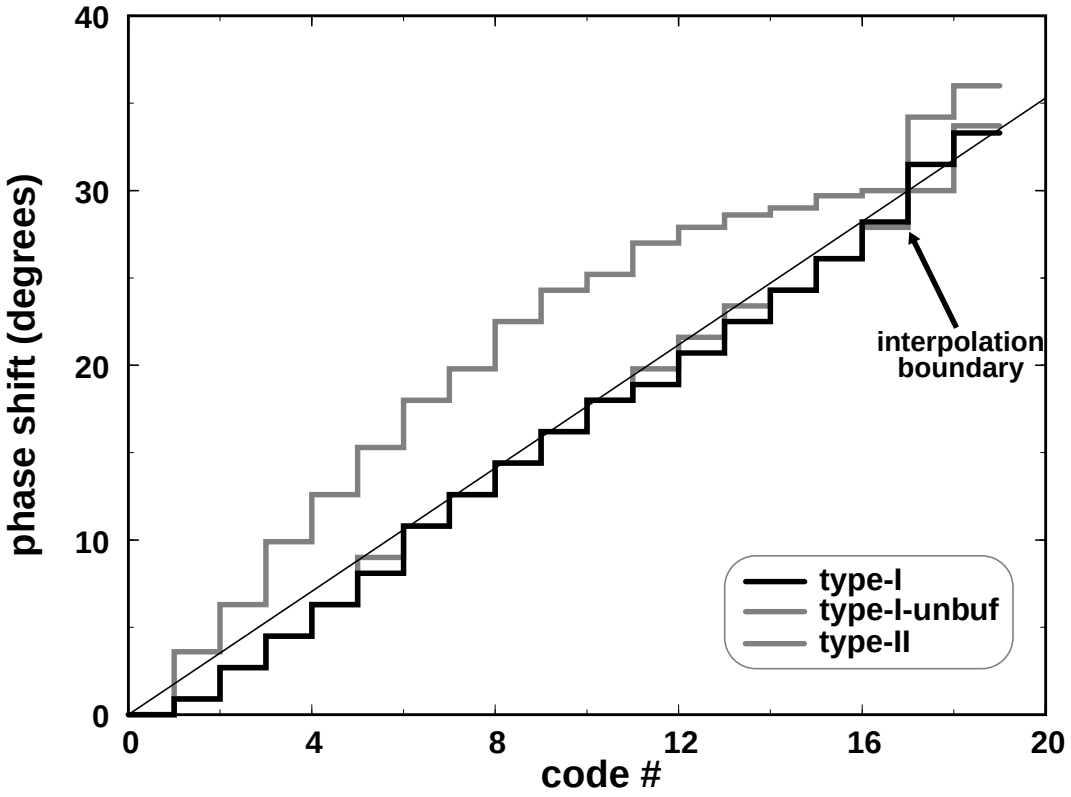

Figure 5.14: Simulated interpolator transfer function

Figure 5.13-(b) shows an alternative design which does not suffer from this problem. In this design (type-II) the interpolator differential pairs consist of unit cell differential pairs. Therefore, when one of the interpolation weight thermometer codes is zero the corresponding input is completely isolated from the output eliminating the gate to drain coupling capacitance. Figure 5.14 shows the simulated transfer function of the interpolator alternative designs. This simulation includes random (<20-mV) threshold voltage offsets in the thermometer code current sources. The type-I design exhibits a nominal step of approximately 2^{o}. Due to the gate-to-drain capacitive coupling effect the maximum step of 3.8^{o} occurs at the interpolation boundary when the input clock ϕ is switched to the next selection. In the lower power implementation where no buffering is used at the core delay line outputs (type-I-unbuf), the data-dependent loading on the previous stage results on a double phase step at the interpolation interval boundaries. Although the alternative design (type-II) does not exhibit a boundary phase step, it was not used, because it occupies more layout area and exhibits more nonlinear characteristics due to the data-dependent loading of the previous stage.

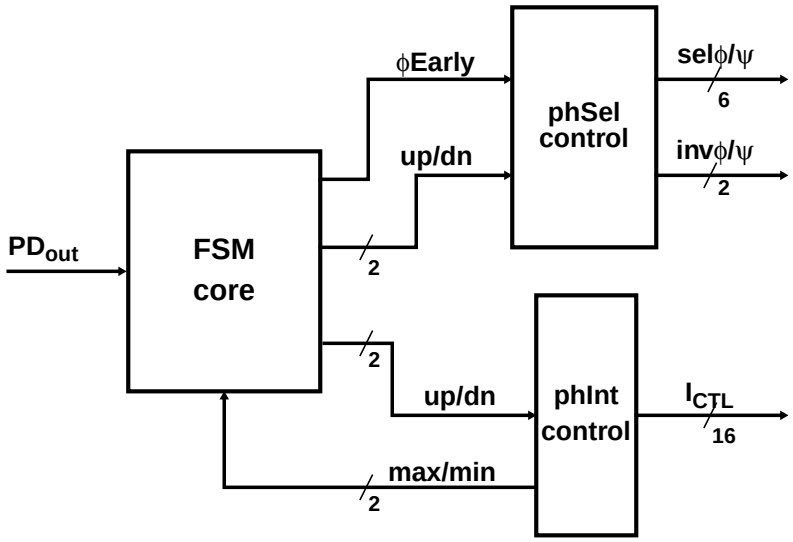

Figure 5.15: Peripheral loop controller block diagram

5.2.3 Peripheral Loop Control

A block diagram of the digital peripheral loop controller is shown in Figure 5.15. The core FSM, controls the phase selector (phSel), and the phase interpolator (phInt) control blocks, according to the phase detector output PD_{OUT}. The phase selector control block, consists of a 3-bit up/down shift register and associated combinational logic which drive the control lines of the phase selection and phase inversion multiplexers. The phase interpolator control block is a 16-bit up/down shift register which drives the interpolator thermometer-code current sources. All of the controller blocks operate at a clock rate equal to 1/4 of that of the loop output clock.

A simplified form of phase capture algorithm implemented by the dual loop prototype is outlined in Figure 5.16-(a). The single state ϕEarly of the FSM indicates the relationship of the two interpolator input clocks. During every cycle of its operation the FSM might take two actions:

- In the more frequent case of in-range interpolation (i.e., $I_{CTL} \neq 0$, and $I_{CTL} \neq N$) the FSM simply increments or decrements the interpolation weight by asserting the up or down signal of the phase interpolator controlling shift register. The direction of the shift is decided based on the phase detector output and the current value of the state ϕEarly.

- If the peripheral loop has run out of range in the current interpolation interval, the FSM seamlessly slides the current interpolation interval by switching phase ϕ or ψ to the next appropriate selection. This phase switch is accomplished by asserting the up or down signal of the phase selector control block, and toggling the value of the output state ϕEarly. Subsequently the "slave" phase selection control block adjusts appropriately the phase selection control lines to reflect the decision of the FSM. In case the current selection of phase ϕ or ψ is adjacent to the 0^o or 180^o interpolation interval boundary, switching to the next phase selection involves toggling the select of the second-stage phase inversion multiplexer. This phase toggling decision is made by the phSel-control block and is transparent to the core FSM. The fact that the interpolation has run out of range in the current interval is indicated to the FSM by a combination of the most or least significant bit of the thermometer register (max/min), and the output of the phase detector. Again the direction of the shift is decided based on the phase detector output and the current value of the state ϕEarly.

The loop phase capture behavior, resulting from the execution of this simple algorithm is illustrated in Figure 5.16-(b). The phase error decreases at a linear rate until the system

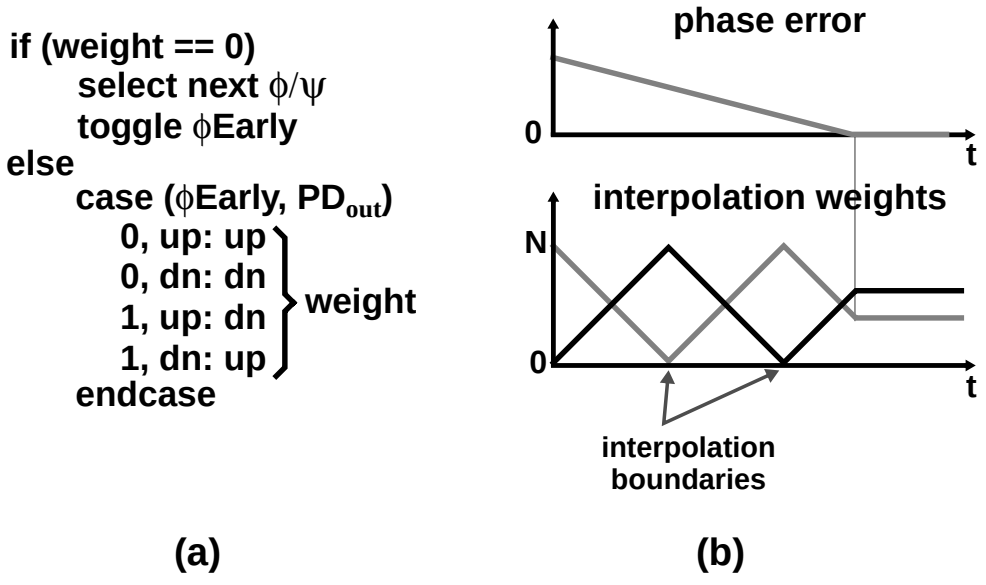

Figure 5.16: Simplified FSM algorithm (a) and resulting loop behavior (b)

achieves lock. Subsequently the loop dithers around the zero phase error point with a dither magnitude of one phase interpolation interval. The complementary interpolation weights slew linearly changing direction at the interpolation interval boundaries. Once the system finds lock, they either dither by one, or they stay constant in case the dither point happens to lie on an interval boundary.

This decoupled three block micro-architecture of the peripheral loop controller enabled the implementation of the core FSM through logic synthesis of a behavioral model. The more area demanding phase selection and interpolation control blocks were implemented through full-custom layout. A significant property of this implementation is that it can gracefully recover from transient faults. In a slightly simpler implementation that was briefly considered, the four phase selection/inversion multiplexers are driven by independent state-registers. In the case of a transient fault, these independent state-registers can select phases ϕ and ψ which are spaced by more than 30^o. This overall inconsistent state would result in the loop locking with increased dither jitter. In contrast, the current implementation of the controller will eventually force the peripheral loop to re-lock with minimal dither jitter, irrespective of the state that resulted from a transient fault. This is guaranteed by the fact that the phase selection block outputs depend on a single 3-bit shift register and the FSM state ϕEarly. This “one-hot” encoded shift register is initialized to a value of “001”, while embedded self correction circuits guarantee that the register will be re-initialized in the case of a transient fault. Finally, in order to block glitches from the phSel combinational logic to propagate to the loop output clock, the outputs of that block are driven by transparent latches controlled by the negative half period of the controller clock.

A fundamental difference of this digital controller from conventional analog DLL and PLL control loops is in the way it treats the phase detector output. In conventional analog PLL’s the output of the phase detector directly drives an analog circuit, e.g., the loop charge pump or an RC low-pass filter. Since the phase detector does not drive any digital circuits, metastable states at its output do not affect the correct operation of the loop. In this digital control implementation the output of the phase detector drives the FSM. Although as described above, the peripheral loop controller can recover from inconsistent

states caused by metastability, the design still must minimize the probability of metastability events because they would disrupt the normal operation of the loop. In addition, the metastability resolution requirements are more stringent than those in a conventional synchronizer [49], since the whole loop is driving the phase detector to its metastable point of operation. For this reason the output of the phase detector is delayed by three metastability hardened flip-flops. This increases the mean time between failures (MTBF) of the system to an estimated worst case of approximately 100 years.

Increasing the metastability robustness, however, has an adverse effect on the loop dither jitter. The magnitude of the peripheral loop clock dither is determined by the minimum interpolation step and the delay through the feedback loop. Increasing the system MTBF by inserting metastability hardened flip-flops at the phase detector output also increases the delay through the peripheral feedback loop. To compensate for that additional delay and decrease the loop dither, the FSM logic implements a front-end filter which counts eight continuous phase detector “up” or “down” outputs before making a phase adjustment decision. This causes the FSM to delay its next decision until the results of its previous action have been propagated to the phase detector output and reduces the inherent peripheral loop dither to one phase interpolation interval.

As outlined above, the digital nature of the peripheral loop control enabled the implementation of the FSM through synthesis of a behavioral model followed by standard cell place-and-route. The FSM behavioral-HDL model was verified by simulating it in conjunction with a behavioral core loop model. This automated methodology enables more complicated algorithms to be implemented, requiring minimal design effort. Faster phase acquisition can be achieved by disabling the front end counter/filter and changing the interpolation step by a larger amount while the loop is not in lock. The loop can also implement a periodic phase calibration algorithm. In this case, the FSM is activated initially to drive the loop to zero phase error. Subsequently, it is shut down to save power while it is periodically turned on to compensate for slow phase drifts. Since the FSM can run at a frequency slower than that of the system clock, the implementation of different algorithms is not in the system critical path.

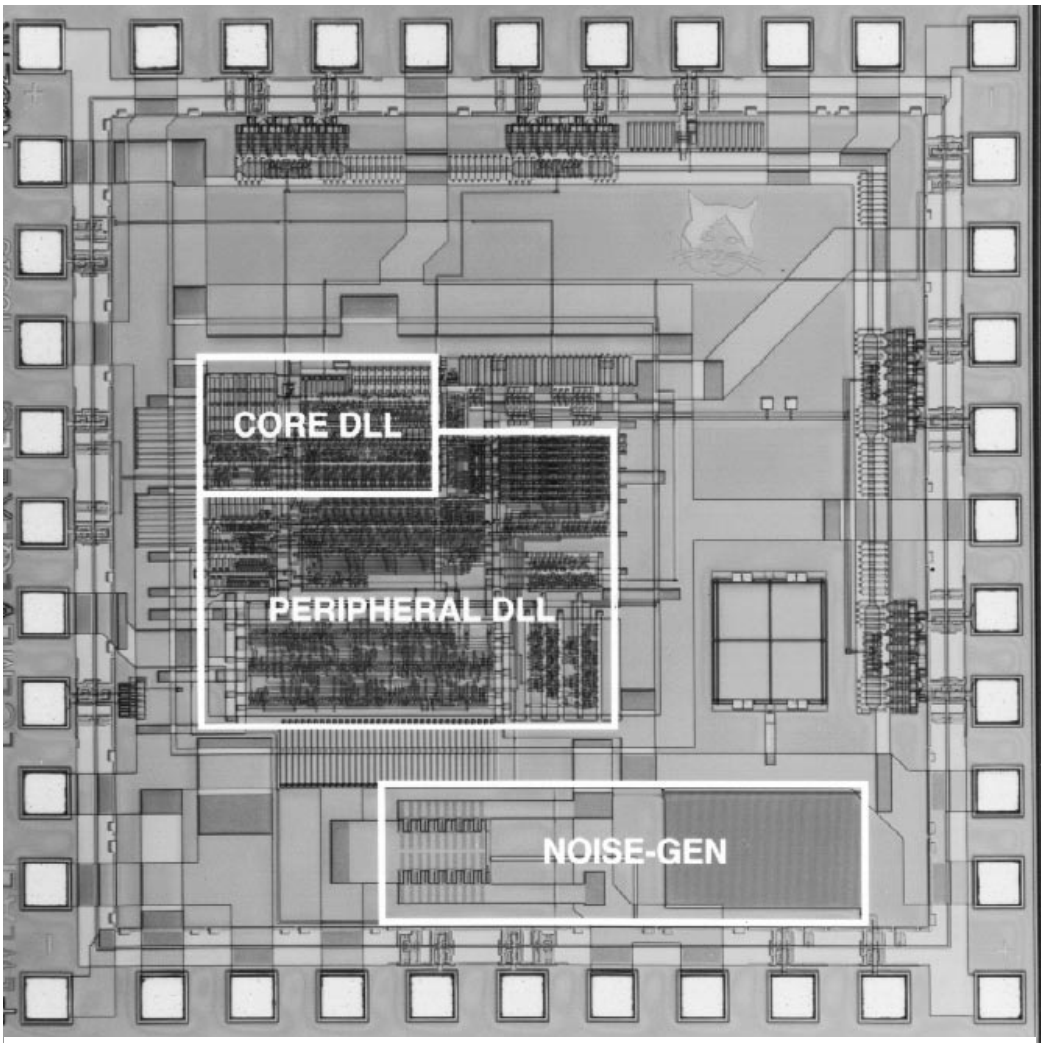

Figure 5.17: Prototype chip photomicrograph

5.3 Experimental Results

The prototype DLL was fabricated through MOSIS in the HP CMOS-26B process. As with the transceiver chips discussed in Chapters 3 and 4 the design and testing was performed with a 3.3-V nominal power supply voltage. Figure 5.17 shows a micrograph of the prototype IC. The chip integrates the dual DLL, along with noise injection and monitoring circuits and current-mode differential output buffers. The dual DLL occupies 0.8 mm^2 of silicon area, the majority ($\approx$ 60%) of which is devoted to the peripheral loop logic. This is mainly due to the relatively large standard-cell size of the library used in this implementation.

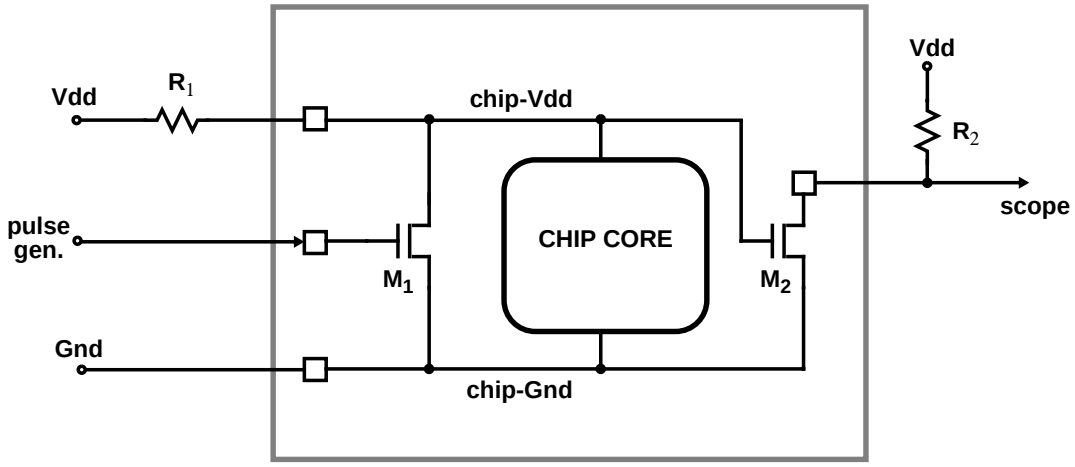

Figure 5.18: Noise generation and monitoring circuits

The block labeled NOISE-GEN in Figure 5.17 is used to inject and measure on chip supply noise. Figure 5.18 shows a schematic diagram of these circuits. The 1000-μm wide transistor M_1 shorts the on-chip supply rails to create a voltage drop across the off-chip 4-Ω resistor R_1. In order to monitor the droop on the on-chip supply, device M_2 and the external 5-Ω load resistor R_2 form a broadband attenuating buffer which drives the 50-Ω scope. The gain of the buffer is computed during an initial calibration step. The use of these circuits enables the injection and monitoring of fast (<1-ns rise time) steps on the on-chip supply.

The dither jitter of the loop with quiescent on chip supply varies according to the phase of the reference clock. This occurs because the offset of the interpolator and the phase selection multiplexers change according to the point of lock. Figure 5.19-(a) shows the worst case jitter (68-ps) with quiescent supply. The jitter histogram consists of the superposition of two Gaussian distributions resulting from the switching of the peripheral loop between two adjacent interpolation boundaries. The distance between the peaks of the two superimposed distributions is about 40 psec which is in fair agreement with the simulation results of Figure 5.14. Operating with the noise generation circuits injecting a 750-mV 1-MHz square wave on the chip supply, the peak to peak jitter increases to 400 psec, as shown in Figure 5.19-(b). It should be noted that simulation results indicate that approximately 50% of this jitter is not inherent to the loop, but is due to the supply sensitivity of the succeeding static CMOS clock buffer and off-chip driver.

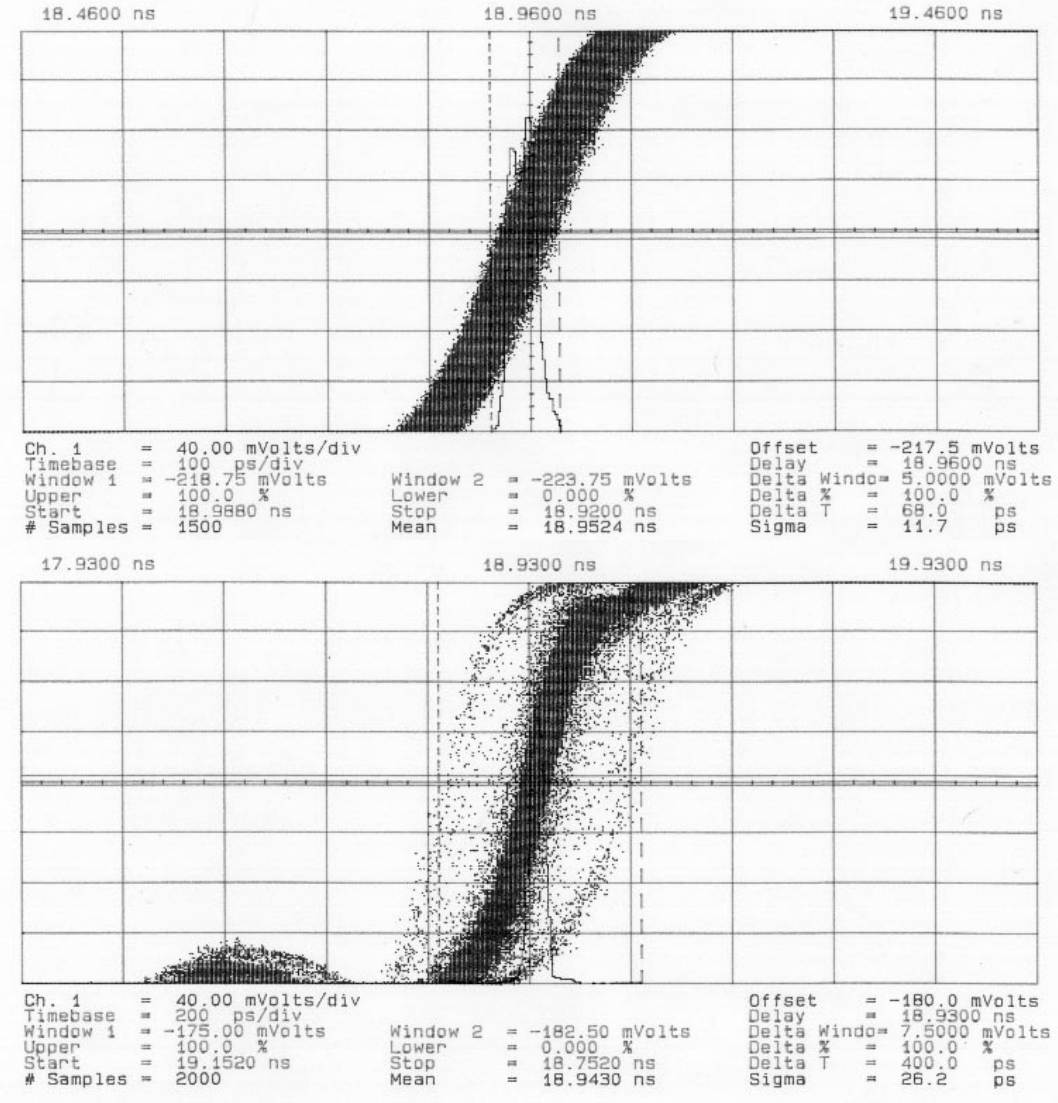

Figure 5.19: Jitter histogram with: (a) quiet, and (b) noisy supply

Figure 5.20 illustrates the linearity of the interpolation process in the peripheral loop under dynamic conditions. The histogram was generated by keeping the reference clock at a constant value, while running the input clock at its nominal frequency of 250-MHz. This causes the peripheral loop controller to move the phase of the output clock during every cycle of its operation, resulting in a continuous rotation of the output clock phase throughout the full 0^o-360^o interval. The histogram valleys correspond to the interpolation interval boundaries. The spacing of the valleys is within 10% of their nominal 333-ps distance indicating good matching of the delays of the core loop buffers. The absence of one valley at the 180^o interpolation boundary indicates a slight offset in the core loop. The average of

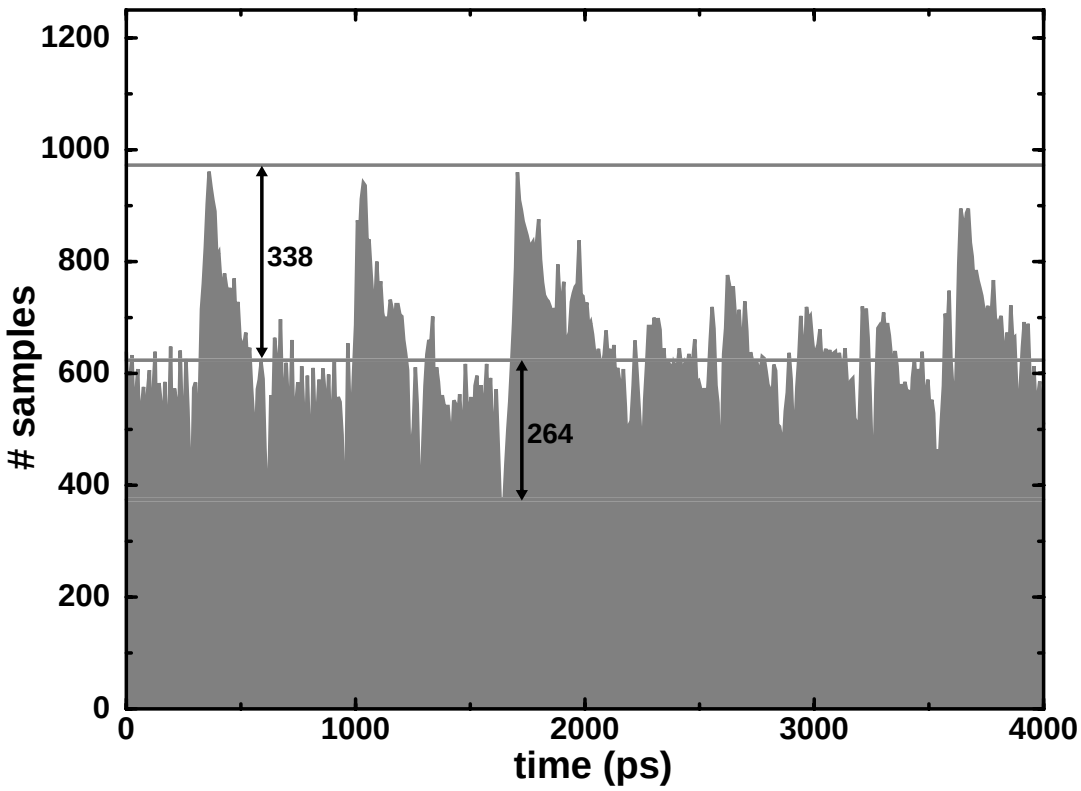

Figure 5.20: Clock histogram with continuously rotating clock

the histogram is 625 samples per bin. The distance of that average from the highest peak and the lowest valley is less than the average histogram value. This fact indicates that the interpolator achieves the target 4-bit linearity, and that the overall linearity of the DLL is limited by the steps at the interpolator interval boundaries. The linearity of the interpolator design was also verified through a static measurement performed on a transceiver IC capable of single-stepping the interpolator control word. The PLL integrated on this transceiver uses an identical interpolator scaled for a 0.25-µm CMOS process [71].

Table 5-1 summarizes the performance characteristics of the prototype DLL. With a 3.3-V supply the loop operates from 80 KHz to 400 MHz. The phase offset between the reference clock and the output clock of the loop is less than 40 psec. Operating at 250 MHz the dual DLL draws 31-mA of DC current from a 3.3-V power supply.

Fabrication Technology	1.0/0.8 μm (drawn/effective)
Active Area	0.8 mm^2
Supply Voltage	3.3 V
Power Dissipation	102 mW (@ 250 MHz)
Operating Range	80 KHz-400 MHz
Phase Offset	< 40-ps
Jitter	68 ps pk-pk
Supply sensitivity	0.4 ps/mV

Table 5-1: Performance summary of the prototype DLL

5.4 Summary

In high-speed interfaces delay locked loops are an attractive alternative to VCO-based phase locked loops, due to their better jitter performance, inherent stability and simpler design. The main disadvantage of DLL's is their limited capture range, which restricts their application to isochronous environments and requires error prone start-up controlling circuits. DLL's based on quadrature mixing alleviate this problem but suffer from inherent jitter sensitivity. This chapter discussed a dual DLL architecture which eliminates both the start-up and jitter sensitivity problems. This architecture relies on a core DLL to generate coarsely spaced clocks, which are then used by a peripheral DLL to generate the output clock by using phase interpolation. This dual loop has unlimited phase shift capability, thus removing boundary conditions and phase relationship constraints between the system clocks. In addition, the digital nature of the peripheral loop control eases the implementation of complicated phase capture algorithms.

Implementing a high performance dual DLL still requires robust circuit building blocks. Although simpler implementations are possible, the prototype described in this chapter uses differential elements improving supply noise rejection. Moreover, in order to extend the DLL operating frequency range, the biasing established by the core loop is also used by the interpolating peripheral loop, thus enabling uniform time-constant scaling throughout the design. The experimental results measured on this prototype clearly demonstrate the feasibility and performance advantages of this new DLL architecture.

Chapter 6

Conclusion

Increasing the performance of digital systems along with the performance and integration levels of individual IC components requires increasing the bandwidth of intra-system interconnects. In many applications the increase in the interconnection bandwidth must be achieved with minimal increase in communication latency. In addition, to keep the overall system cost attractive, the power and area overhead of the increased bandwidth circuits must be kept modest. To achieve these goals designers must effectively address the problems created by signal amplitude noise and timing uncertainty. This thesis presented circuit techniques for effectively dealing with both of these classes of problems, while maintaining low communication latency and modest circuit area and power overhead.

Dealing with signal amplitude noise becomes more difficult in the economical class of pseudo-differential interfaces. The performance of these systems is severely compromised by high frequency noise injected on-chip on the shared reference voltage. As was shown in Chapter 4, using an integrator as a front-end filter of the input receiver greatly attenuates the effects of high frequency noise. Chapter 4 also discussed the implementation issues of such an integrating receiver in a CMOS technology, and demonstrated techniques for effectively dealing with the issues of charge-injection and operating condition independent biasing. A prototype transceiver chip, fabricated in a 0.8-µm CMOS technology, confirmed the superior robustness of integrating receivers compared to more conventional approaches.

Synchronizing the signalling system, while maintaining low timing uncertainty, necessitates the use of active on-chip phase alignment circuits. In high speed interfaces, delay line based PLL's are an attractive alternative to VCO-based systems because of their inherently better jitter performance, stability and simpler design. Chapter 5 proposed a dual-loop DLL architecture, which uses a core loop to generate coarsely spaced phases, subsequently utilized by a peripheral loop which generates the output clock through phase interpolation. This way, the dual loop architecture eliminates the limited phase capture range of conventional DLL's, while maintaining all of their inherent advantages. Moreover, by relying on digital control it provides for straightforward implementation of different phase alignment algorithms. The implementation of a prototype dual loop based on a linear self-biased core loop was discussed along with issues pertaining to the implementation of the phase interpolator and the digital control of the peripheral loop. The results from the prototype dual loop demonstrate the merits of this architecture.

6.1 Future Work

The work presented in this dissertation can be extended in several ways. Bidirectional signalling is a promising technique for achieving high bandwidth in systems with limited number of interconnections. Current integrating receivers can be effectively used in these systems, to attenuate the inherent signal amplitude noise resulting from multiplexing the reference voltage. Designing a complete bidirectional transceiver would be an excellent application of the techniques discussed in Chapter 4.

Work on phase alignment circuits has up to now concentrated mainly on reducing the jitter resulting from the core PLL or DLL. However, the experimental results of Chapters 4 and 5 indicate that a significant portion of the timing uncertainty is due to the supply sensitivity of the final clock buffer. Wider parallel interfaces or more complicated circuit structures will further increase that portion of the jitter, since they increase the clock loading and the delay through the clock buffer. This fact suggests that significant performance gains can be realized by reducing the load and delay of the clock buffer, through micro-architectural changes. An alternative and more challenging way of achieving the same goal is to realize clock buffer circuits which achieve reduced noise sensitivity, while main-

taining the low power dissipation of static CMOS buffers.

Based on a simple technology-independent performance metric, it can be argued that the performance of the circuits presented in this thesis will continue to scale with improving fabrication technology [72]. This has been demonstrated by porting both the integrating receiver and clocking circuits of Chapters 4 and 5 to a 0.25 μm CMOS technology. The scaled version of the circuits achieve a transfer rate of 2-Gbps/pin even when operating in a pseudo-differential signalling mode [71]. At these transfer rates the circuits are approaching the fundamental limitations imposed by the interconnection media. The first limitation circuit designers need to overcome is mismatches of the electrical lengths in parallel interconnects and of on-chip delays. The effect of these mismatches will be accentuated with shorter bit-times and will eventually limit the transfer rate achievable by parallel interfaces. A promising area of work is to construct a parallel interface which compensates for these timing offsets through a calibration scheme. The digitally controlled interpolating DLL of Chapter 5 can be a useful building block in such systems.

A more fundamental limitation is created by the finite bandwidth of the off-chip electrical interconnects which does not scale along with IC fabrication technology. In order to address this problem, interchip signalling systems will need to employ channel equalization and coding techniques. Due to low latency requirements these techniques will be limited, at least initially, to simple transmitter-side equalization. However, extending the communication bandwidth even further may require more complicated techniques, such as phase amplitude modulation. Constructing such more complicated transmitters and receivers with minimum latency and cost overheads, while simultaneously dealing with the effects of increased noise, will certainly be a challenging problem.

The discussion above is mainly oriented towards purely technical issues. However, the author’s recent experience, created by participating in the design of a consumer-oriented high-speed interface, indicates that issues such as compatibility with existing infrastructure and hesitation to abandon time-tested approaches are dominant factors in “real-world” designs. Overcoming such obstacles may in the end prove more challenging than addressing purely technical problems.

References

[1] B.S. Landman and R.L Russo, “On pin versus block relationship for partitions of logic graphs,” *IEEE Transactions of Computers*, vol. C-20, pp. 1469-1479, Dec. 1971.

[2] D.C. Schmidt, “Circuit pack parameter estimation using Rent’s rule,” *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems,* CAD-1, pp. 186-192, Oct. 1982.

[3] H.B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*, Reading, MA: Addison-Wesley, 1990.

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

[5] B. Kim, et al., “A 30-MHz Hybrid Analog/Digital Clock Recovery Circuit in 2-µm CMOS,” *IEEE Journal of Solid-State Circuits,* vol. 25, no. 6, pp. 1385-1394, Dec. 1990.

[6] T. Hu and P. Gray, “A Monolithic 480Mb/s Parallel AGC/Decision/Clock-Recovery Circuit in 1.2um CMOS,” *ISSCC 1993 Dig. Tech. Papers*, pp. 98-99, Feb. 1993.

[7] M. Banu and A. Dunlop, “A 660Mb/s Clock Recovery Circuit with Instantaneous Locking for NRZ Data and Burst-Mode Transmission,” *ISSCC 1993 Dig. Tech. Papers,* pp. 102-103, Feb. 1993.

[8] M. Soyuer and H. Ainspan, “A Monolithic 2.3Gb/s 100mW Clock and Data Recovery Circuit,” *ISSCC 1993 Dig. Tech. Papers*, pp. 102-103, Feb. 1993.

[9] A. Pottbacker and U. Langmann, “An 8-GHz Silicon Bipolar Clock-Recovery and Data-Regenerator IC,” *ISSCC 1994 Dig. Tech. Papers,* pp. 116-117, Feb. 1994.

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

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

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

[13] A. Mu. et al. “A 285 MHz 6-port Plesiochronous Router Chip with non-blocking cross-bar switch,” *Proceedings of the 1996 Symposium on VLSI Circuits*, pp. 136-137, June 1996.

[14] E. Reese, et al. “A phase-tolerant 3.8 GB/s data-communication router for a multi-processor supercomputer backplane,” *ISSCC 1994 Dig. Tech. Papers*, pp. 296-297, Feb. 1994.

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

[16] N. McKeown et al., “Tiny Tera: a packet switch core” *IEEE Micro*, vol.17, no.1, pp. 26-33, Jan.-Feb. 1997.

[17] S. Sidiropoulos and M. Horowitz, “Current Integrating Receivers for High Speed System Interconnects,” *Proceedings of the 1995 IEEE Custom Integrated Circuits Conference*, pp. 107-110, May 1995.

[18] S. Sidiropoulos and M. Horowitz, “A 700 Mbps/pin CMOS Signalling Interface Using Current Integrating Receivers,” *Proceedings of the 1996 IEEE Symposium on VLSI Circuits*, pp. 142-143, Jun. 1996.

[19] 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.

[20] S. Sidiropoulos and M. Horowitz, “A Semi-digital DLL with unlimited phase shift capability and 0.08-400 MHz operating range,” *ISSCC 1997 Dig. Tech. Papers*, pp. 332-333, Feb. 1997.

[21] 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.

[22] R.P. Borril, “32-bit busses-An objective comparison,” *Proceedings BUSCON 1986*, pp. 138-254.

[23] R. Matick, *Transmission lines for digital and communication networks*, 3rd ed., IEEE Press, 1997.

[24] D. Pozar, *Microwave Engineering*, 2nd ed., Addison Wesley, 1993.

[25] S. Nimagadda et al., “Measurement modeling and simulation of a high speed digital system using a Vector Network Analyzer”, *Proceedings of the IEEE Technical Applications Conference*, Northcon/96, pp. 158-161.

[26] J. Kuskin et al. “The Stanford FLASH multiprocessor,” *Proceedings of the 21st International Symposium on Computer Architecture,* ISCA-94, pp. 274-284.

[27] F. Tobagi, “Fast Packet Switch Architectures for Broadband Integrated Services Digital Networks,” *Proceedings of the IEEE*, vol. 78, no. 1, pp. 133-167, Jan 1990.

[28] R. Mooney, C. Dike and S. Borkar, “A 900 Mb/s bidirectional signaling scheme” *IEEE Journal of Solid-State Circuits*, Dec. 1995. vol.30, no.12, pp. 1538-43.

[29] T. Takahashi et al. “A CMOS Gate Array with 600 Mb/s Simultaneous Bidirectional I/O Circuits,” *IEEE Journal of Solid State Circuits*, vol. 30, no. 12. Dec 1995.

[30] T. Gabara, “Digital transistor sizing techniques applied to 100K ECL CMOS output buffers,” *Proceedings of the Sixth Annual IEE International ASIC Conference*, pp. 456-459, Oct. 1993.

[31] A. DeHon, T. Simon and T. Knight, “Automatic impedance control,” *ISSCC 1993 Dig. Tech. Papers*, pp 164-165, Feb. 1993.

[32] C. Flaig, *VLSI Mesh Routing Systems*, Technical Report 35241/87, California Institute of Technology, May 1987.

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

[34] B. Gunning, L. Yuan, T. Nguyen, T. Wong, “A CMOS low-voltage-swing transmission-line transceiver,” *ISSCC 1992 Dig. Tech. Papers*, pp. 58-59, Feb. 1992.

[35] S. Kim, K. Lee and D.K. Jeong, “A pseudo-synchronous skew-insensitive I/O scheme for high bandwidth memories,” *Proceedings of 1994 IEEE Symposium on VLSI Circuits*. pp. 41-42, June 1996.

[36] 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.

[37] W. Dally and J. Poulton, “Transmitter Equalization for 4-Gbps signalling,” *IEEE Micro*, vol. 17, no. 1, pp. 488-56, Feb. 1997.

[38] K. Donnelly et al., “A 660 MB/s interface megacell portable circuit in 0.3mm-0.7mm CMOS ASIC,” *ISSCC 1996 Dig. Tech. Papers*, pp. 290-291, Feb. 1996.

[39] C. Svensson and J. Yuan, “A 3-level asynchronous protocol for a differential two-wire communication link,” *IEEE Journal of Solid-State Circuits*, April 1995. vol. 29, no. 9, pp. 1129-1132.

[40] B. Chappell et al., “Fast CMOS ECL receivers with 100-mV worst case sensitivity,” *IEEE Journal of Solid State Circuits*, vol. 23, no. 1, pp. 59-67, Feb. 1988.

[41] M. Bazes, “A Novel Precision MOS Synchronous Delay Line,” *IEEE Journal of Solid State Circuits*, vol. SC-20, no. 6, pp. 1265-1271, Dec. 1985.

[42] 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.

[43] T. Lee, et al., “A 2.5V Delay-Locked Loop for an 18Mb 500MB/s DRAM,” *ISSCC 1994 Dig. Tech. Papers*, pp. 300-301, Feb. 1994.

[44] J.T. Wu and B. Wooley, “A 100-MHz pipelined CMOS comparator,” *IEEE Journal of Solid-State Circuits*, vol. 23, no. 6, pp. 1379-85.

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

[46] D. Cecchi, M. Dina and C. Preuss, “A 1GB/s SCI link in 0.8-mm BiCMOS,” *ISSCC 1995 Dig. Tech. Papers*, pp. 326-327, Feb. 1995.

[47] M. Roden, *Digital Commuunication Systems Design*, Prentice Hall, 1988.

[48] J. Proakis, M. Salehi, *Communication Systems Engineering*, Prentice Hall, 1994.

[49] C. Portmann, *Characterization and reduction of metastability errors in CMOS interface circuits*, Technical Report CSL-95-671, Computer Systems Laboratory, Stanford University.

[50] A. Yukawa, “A CMOS 8-bit high speed A/D converter IC,” *IEEE Journal of Solid-State Circuits*, vol. 20, no. 3, pp. 775-779, June 1985.

[51] 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.

[52] A. Lempel and W. Eastman, “High speed generation of maximal length sequences,” *IEEE Transactions on Computers*, vol. 20, no. 2, pp. 227-229, Feb. 1971.

[53] F. Gardner, *Phaselock Techniques*, 2nd ed. New York, NY: John Wiley, 1979.

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

[55] R. Best, *Phase-Locked Loops: Theory, Design and Applications*, New York, NY: McGraw-Hill, 1984.

[56] D. Jeong, G. Borriello, D. Hodges, “Design of PLL-Based Clock Generation Circuits,” *IEEE Journal of Solid-State Circuits*, vol. 22, no. 2, pp. 255-261, Apr. 1987.

[57] I. Young, et al., “A PLL Clock Generator with 5 to 110 MHz of Lock Range for Microprocessors,” *IEEE Journal of Solid-State Circuits*, vol. 27, no. 11, pp. 1599-1607, Nov. 1992.

[58] J. Alvarez, et al. “A wide-bandwidth low-voltage PLL for PowerPC microprocessors,” *IEEE Journal of Solid-State Circuits*, April 1995. vol.30, no.4, pp. 383-91.

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

[60] A. Waizman, “A Delay Line Loop for Frequency Synthesis of De-Skewed Clock,” *ISSCC 1994 Dig. Tech. Papers*, pp. 298-299, Feb. 1994.

[61] A. Efendovich, Y. Afek, C. Sella and Z. Bikowsky, “Multifrequency Zero-Jitter Delay-Locked Loop,” *IEEE Journal of Solid-State Circuits*, vol. 29, no. 1, pp. 67-70, Jan. 1994.

[62] M. Izzard, et al., “Analog versus Digital Control of a Clock Synchronizer for a 3 Gb/s Data with 3.0 V Differential ECL,” *Proceedings of the 1994 Symposium on VLSI Circuits*, pp. 39-40, June 1994.

[63] B. Kim, et al., “PLL/DLL system noise analysis for low jitter clock synthesizer design,” *Proceedings of the IEEE International Symposium on Circuits and Systems*, ISCAS’ 94, vol. 4, pp. 31-34, June 1994.

[64] T. Weigandt et al., “Analysis of timing jitter in CMOS ring oscillators,” *Proceedings of the IEEE International Symposium on Circuits and Systems*, ISCAS’ 94, vol. 4, pp. 27-30, June 1994.

[65] Horowitz, et al., “PLL Design for a 500MB/s Interface,” *ISSCC 1993 Dig. Tech. Papers*, pp. 160-161, Feb. 1993.

[66] J. 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.

[67] J. Brown, “A digital phase and frequency-sensitive detector,” *Proceedings of the IEEE*, pp. 717-718, Apr. 1971.

[68] A. Sharpe, “A 3-state phase detector can improve your next PLL design,” *EDN Magazine*, Sept. 20, 1976.

[69] T. Knotts, D. Chu and J. Sommer, “A 500MHz Time Digitizer IC with 15.62ps Resolution,” *ISSCC 1994 Dig. Tech. Papers*, pp. 58-59, Feb. 1994.

[70] M. Zargari and B. Wooley, “A BiCMOS active substrate probe card technology for digital testing,” *ISSCC 1996 Dig. Tech. Papers*, pp. 308-309, Feb. 1996.

[71] K.Y. Chang, et al., “A 2Gbps assymetric serial link,” *Hot Interconnects 1997 Symposium Record*, Aug. 1997.

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