Precise Timing of Digital Signals:

Circuits and Applications

by

Muhammad A. Nummer

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2007

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

With the rapid advances in process technologies, the performance of state-of-the-art integrated circuits is improving steadily. The drive for higher performance is accompanied with increased emphasis on meeting timing constraints not only at the design phase but during device operation as well. Fortunately, technology advancements allow for even more precise control of the timing of digital signals, an advantage which can be used to provide solutions that can address some of the emerging timing issues. In this thesis, circuit and architectural techniques for the precise timing of digital signals are explored. These techniques are demonstrated in applications addressing timing issues in modern digital systems.

A methodology for slow-speed timing characterization of high-speed pipelined datapaths is proposed. The technique uses a clock-timing circuit to create shifted versions of a slow-speed clock. These clocks control the data flow in the pipeline in the test mode. Test results show that the design provides an average timing resolution of 52.9ps in 0.18μm CMOS technology. Results also demonstrate the ability of the technique to track the performance of high-speed pipelines at a reduced clock frequency and to test the clock-timing circuit itself.

In order to achieve higher resolutions than that of an inverter/buffer stage, a differential (vernier) delay line is commonly used. To allow for the design of differential delay lines with programmable delays, a digitally-controlled delay-element is proposed. The delay element is monotonic and achieves a high degree of transfer characteristics' (digital code vs. delay) linearity. Using the proposed delay element, a sub-1ps resolution is demonstrated experimentally in 0.18μm CMOS.

The proposed delay element with a fixed delay step of 2ps is used to design a high-precision all-digital phase aligner. High-precision phase alignment has many applications in modern digital systems such as high-speed memory controllers, clock-deskew buffers, and delay and phase-locked loops. The design is based on a differential delay line and a variation tolerant phase detector using redundancy. Experimental results show that the phase aligner's range is from -264ps to +247ps which corresponds to an average delay step of approximately 2.43ps. For various input phase difference values, test results show that the difference is reduced to less than 2ps at the output of the phase aligner.

On-chip time measurement is another application that requires precise timing. It has applications in modern automatic test equipment and on-chip characterization of jitter and skew. In order to achieve small conversion time, a flash time-to-digital converter is proposed. Mismatch between the various delay comparators limits the time measurement precision. This is demonstrated through an experiment in which a 6-bit, 2.5ps resolution flash time-to-digital converter provides an effective resolution of only 4-bits. The converter achieves a maximum conversion rate of 1.25GSa/s.

Acknowledgements

All praise is due to God, Most Gracious, Most Merciful, Whose bounties and blessings are ever dominating throughout my life.

I would like to express my deep gratitude and appreciation to my supervisor, Prof. Manoj Sachdev. Prof. Sachdev has always been an invaluable source of guidance and encouragement. I am extremely grateful to all his support throughout the years I have spent in Waterloo.

Throughout the period I spent doing my Ph.D., I received tremendous support from many people. I am grateful to Prof. David Nairn for his insights on testing and debugging the TDC chip. I am also grateful to Prof. Mark Aagaard and Prof. Cathy Gebotys for allowing me to use their lab equipment for testing my chips. Through many discussions, my colleagues Mohamed El Said, Ayman Hassan, Mohamed Maymandi-Nejad, and Mohamed Hassan shared many ideas and insights related to my research work. David Li helped me design the ROM encoder for my TDC chip. I am grateful to Wendy Boles for her help on administrative issues and to Phil Regier for his great computing resources support.

My appreciation to all my colleagues in the VLSI research group who were of great help and support. We spent wonderful times discussing all kinds of issues, technical and otherwise. In particular I would like to thank Mohamed Elgebaly, Hassan Hassan, and Ahmed Nour.

My wife, Rasha, shared with me every day throughout the course of this work. Her support, patience, and understanding played a major role in helping me finish this thesis. I am also grateful to my sister, Rasha, for her love, support, and encouragement. My daughter, Yumna, and son, Suhayb, brought so much joy to my life which has been and continue to be a great source of encouragement.

My deepest gratitude to my mother and father for their ever continuous support, encouragement, and prayers. No words of appreciation could ever reward them for all they have done for me. I am, and will ever be, indebted to them for all achievements in my life.

To my parents,

Gamalaat Elmasry and Ahmed Nummer,

with love and appreciation.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ADC	Analog-to-digital converter.
ADPA	All-digital phase-aligner.
ATE	Automatic Test Equipment.
BIST	Built-in-self-test.
CDFF	Controlled-Delay Flip-Flop.
CMOS	Complementary Metal Oxide Semiconductor.
DCDE	Digitally-controlled delay-element.
DCDL	Digitally-controlled delay-line.
DFT	Design-for-Testability.
DLL	Delay-locked loop.
DUT	Device Under Test.
IC	Integrated circuit.
INL	Integral non-linearity.
ITRS	International Technology Roadmap for Semiconductors.
PD	Phase detector.
PLL	Phase-locked loop.
PSD	Phase splitter and delay.
PVT	Process, temperature, and (supply) voltage.

TDC	Time-to-digital converter.
VCDE	Voltage-controlled delay-element.
VCDL	Voltage-controlled delay-line.
VDL	Vernier delay line.
VLSI	Very Large Scale Integration.

Chapter 1

Introduction

1.1 Motivation

Over the past 40 years, scaling of CMOS devices allowed for a steady improvement in performance and increase in the number of transistors integrated on the same die. High-speed microprocessors operating at multi-gigahertz are the norm of our time. Recently, a 9GHz integer execution unit in 65nm CMOS technology was reported [2]. The success of the CMOS processes in improving performance and increasing the level of integration compels manufacturers of high-volume devices to constantly adopt the new "digital" CMOS processes as soon as they are introduced.

The drive for higher performance is accompanied with increased emphasis on meeting timing constraints both at design time and during actual device operation. Timing uncertainties due to random variations and fabrication defects require additional design effort for testing and mitigation. This is becoming even more important with the increased delay variability due to process parameters, voltage, and temperature variations in modern deep-submicron technologies [1].

Fortunately, the advances in the performance of CMOS devices allow for the development of circuit and architectural techniques that can provide precise control of the timing of digital signals. Such techniques can be used to address and to provide solutions to some of the emerging timing issues, either on the design front or on the characterization and testing fronts.

1.2 Goals and Contributions

The goals of the work presented in this thesis are to propose circuit and architectural techniques that can be used to achieve precise control of the timing of digital signals. These techniques need to be demonstrated through applications in timing characterization and control for high-speed state-of-the-art digital systems.

The key contributions of this thesis can be summarized in the following.

1. Proposed a methodology for slow-speed timing characterization of high-speed pipelined datapaths. The methodology uses a clock timing circuit to adjust the phases of the pipeline clocks in the test mode. An implementation of the clock timing circuit that achieves a timing resolution of 52.9ps in 0.18μm CMOS technology is presented. The proposed implementation allows for the testing of the clock timing circuit itself. The basic methodology and the performance and feasibility of the specific implementation have been verified experimentally.

2. Presented a comprehensive review of circuit techniques for implementing digitally-controlled delay-elements (DCDEs). A DCDE is used as the basic building block in many techniques that require precise timing of digital signals.

3. Proposed a DCDE design that has a monotonic and linear code-delay characteristics. The proposed DCDE maintains its linearity with process, temperature, and supply voltage

(PVT) variations. It also has other features that are considered desirable in many application such as possibility of achieving infinitesimally small delay steps and independently adjusting the delays of rising and falling inputs. Detailed analysis of the DCDE design tradeoffs has been presented. Two variations of the DCDE have been implemented on a test chip. Experimental results verify the linearity of the DCDE characteristics and the possibility of achieving infinitesimally (sub-1ps) small delay steps.

4. Proposed a design technique for achieving PVT tolerance in phase detectors using redundancy. The technique has been used to design an array phase detector (APD) with redundancy based on a basic phase detector (PD) structure.

5. Implemented an all-digital phase-aligner (ADPA) using the proposed DCDE with 2ps nominal resolution and an APD. The ADPA is designed for clock deskewing as the target application. The measured phase alignment range is from -264ps to +247ps which corresponds to an average DCDE step of 2.43ps. Phase alignment results demonstrate the effectiveness of the ADPA in achieving sub-2ps precision phase alignment.

6. Proposed a true flash architecture for implementing high-speed time-to-digital converters (TDCs) for real-time on-chip time measurement. The design uses the proposed DCDE with a delay step of 2.5ps. The architecture uses a self-timed approach for generating the internal clock used by the converter. Using this architecture, a conversion rate of 1.25GSa/s has been demonstrated experimentally in 0.18μm CMOS technology.

1.3 Thesis Organization

Following this introduction, the thesis has five chapters. Chapters 2, 3, 4, and 5 are dedicated to presenting the circuit and architectural techniques proposed by this work. Each of these four chapter starts with a review of relevant background and previous work on the topic presented in the chapter and ends with a set of relevant references.

Chapter 2 presents a clock timing technique for slow-speed timing characterization of high speed pipelined datapaths. Design details of the clock timing circuit are presented. The viability of the proposed methodology is verified by performing delay fault simulations on a 16-bit pipelined multiplier. The chapter ends with presenting chip implementation details and experimental results.

Chapter 3 starts by reviewing DCDE application areas and previous implementations. The proposed DCDE implementation is then presented, along with detailed design methodology. The chapter ends with simulation as well as experimental results demonstrating the linearity and precision of the delay step obtainable using the proposed DCDE.

In chapter 4, a high-precision all-digital phase aligner design that uses the proposed DCDE and a PVT tolerant phase detector is presented. PVT tolerance is achieved using redundancy in the phase detector followed by majority voting. Simulation results are presented, followed by experimental results.

A TDC architecture based on the flash architecture used for analog-to-digital (ADC) converters is proposed in chapter 5. The chapter presents details of the proposed implementation and chip design followed by experimental results.

The thesis ends with a concluding chapter followed by a comprehensive bibliography of all references cited throughout the thesis.

References

[1] T.C. Chen. Where CMOS is Going: Trendy Hype vs. Real Technology. In *International Solid-State Circuits Conference Digest of Technical Papers*, pages 1–7, February 2006.

[2] S.B. Wijeratne et al. A 9-GHz 65-nm Intel Pentium 4 Processor Integer Execution Unit. *IEEE Journal of Solid-State Circuits*, 42(1):26–37, January 2007.

Chapter 2

Slow-Speed Timing Characterization and Testing of High-Speed Pipelined Datapaths

2.1 Introduction

Modern microprocessor and digital signal processor designs rely on high-speed pipelined datapaths to achieve performance goals. With the rapid advances in process technologies, the complexity of start-of-the-art integrated circuits is increasing steadily. Moreover, the drive for higher performance requires greater emphasis on cost-effective performance characterization and delay-fault testing. The purpose of testing a VLSI device is to ensure, with reasonable confidence, that the device functions according to the design specifications. This testing must be achievable within certain economic constraints to keep the cost per device as low as possible.

Fundamentally, there are two ways in which a VLSI device can fail. It can fail *parametrically* in that the technology dependent electrical parameters such as voltage, current, capacitance, speed, or gain are out of the specifications. Or, it can fail *functionally* such that the device, independent of its electrical characteristics, performs an incorrect logic function. The objective of most functional and parametric test techniques is to detect chips that are not working according to design specifications. If an IC passes both functional and parametric tests, it is assumed to be functional.

However, some of the "functional" ICs may be in fact marginal ICs. Marginal ICs contain flaws; defects in a chip that do not cause failures at normal operating conditions but degrade chip performance, reduce noise immunity, or draw excess supply current [2, 4]. Marginal ICs cause problems with reliability and must be detected before they are shipped. These chips can cause intermittent failures in the sense that even if they may pass production tests, they can fail to work in the field at different operating conditions. They may cause problems if the supply voltage changes during operation due to IR drops or simultaneous switching noise. If these chips are used for low-power applications, and if the defect inside the chip is causing abnormal static current flow, this may have serious consequences. The excess static current can cause early-life failures and accelerated chip wearout.

The most noticeable effects of marginal ICs on chip performance are timing failures in the form of delay flaws. Timing failures occur when the delay of the manufactured component is different from the designed delay. If the timing failure is such that the circuit fails to work at the designed speed but continues to be functional at a lower speed, it is called a delay fault. On the other hand, a circuit has a delay flaw (non-operational delay failure) if there is a timing failure but the circuit continues to work at the designed speed [3].

There are many test techniques that can be used for detecting marginal ICs. The common procedure in most of these techniques is to change the operating environment so as to provoke the flaws within the chip. A review of such techniques was presented in [4].

2.2 Review of Techniques for Testing High-Speed Circuits Using Slow-Testers

Following Moore's Law for the past three decades, the silicon die cost of integrated circuits has decreased as the number of transistors per die has continued to increase. In contrast, during the same period, the cost of testing integrated circuits in high-volume manufacturing has been steadily increasing. It is predicted that the cost of testing transistors may actually surpass the cost of fabricating them within the next one or two decades [8]. As ICs become more highly integrated, the job of diagnosing failures becomes increasingly difficult. Marching into the deep submicron regime poses many challenges to the testing problem. With smaller geometries, higher speeds, and increased interconnects, it is more likely to have an increase in the tendency for small imperfections in the fabrication process to result in actual failures. That is why it is necessary to use advanced test techniques that enable the testing process to cope with advances in semiconductor technology.

Using DFT/BIST techniques that allow testing high-performance circuits with slow-speed testers is one way to tackle the problem of high-performance circuit testing. The creation of a low frequency test mode in digital circuits was first introduced by Agrawal and Chakraborty [1]. In their proposal, a quantifiable, externally controlled delay is added such that high-performance testing can be carried out with relatively slow-speed testers. They used a pulse-triggered flip-flop in which a dynamic latch is introduced inside a traditional master-slave flip-flop as shown in Figure 2.1. The resulting three-latch structure has two modes of operation; normal mode and test mode. In normal mode, the intermediate latch must hold data for most of the clock period while the other two latches remain transparent. In test mode, flip-flop delay can be modulated by changing clock's pulse width. This allows for testing combinational logic and interconnects for delay faults with a lower clock frequency. Although the concept of adding delay in test mode is

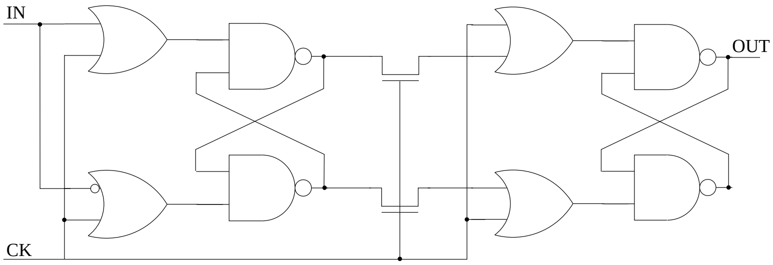


Figure 2.1: Pulse-triggered flip-flop [1].

elegant, this implementation has some important shortcomings as the dynamic latch makes the flip-flop operation sensitive and timing critical.

Shashaani and Sachdev proposed the controlled delay flip-flop (CDFF) as an alternative to the pulse-triggered flip-flop [9]. In this technique an additional test mode clock is used to control the delay of the flip-flop. The main advantages of the CDFF over the pulse-triggered flip-flop are the stable operation and improved performance in normal mode.

Figure 2.2 illustrates a gate level implementation of the CDFF. The transfer of data from the master latch to the slave latch is controlled through a control logic and depends on the relative timing of the clock (CLK) and the test clock (TCLK). To illustrate the operation of the CDFF, a simple model of digital VLSI circuits is depicted in Figure 2.3(a). In this model, a combinational block is sandwiched between two sequential blocks (registers, flip-flops,...etc). In normal mode, TCLK is kept high ensuring normal flip-flop operation (Figure 2.3(b)). Under this condition, the normal mode clock period (T_{NM}) is given by:

$$T_{NM} = t_{prop} + t_{comb} + t_{setup} \tag{2.1}$$

where t_{prop} is the propagation delay of the flip-flop, t_{comb} is the time window allowed for the combinational block to evaluate its input, Q_1, and produce the input of the next sequential block,

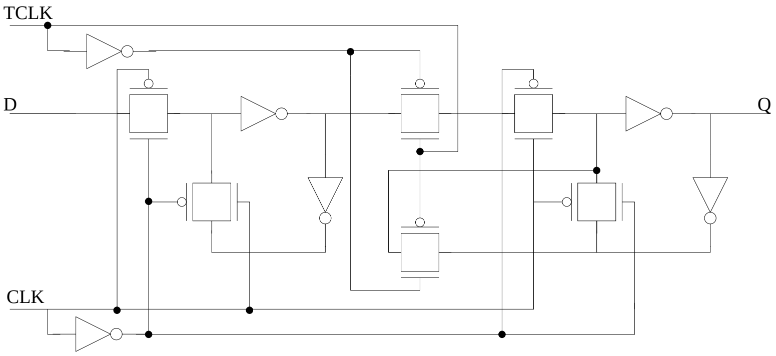


Figure 2.2: Controlled delay flip-flop [9].

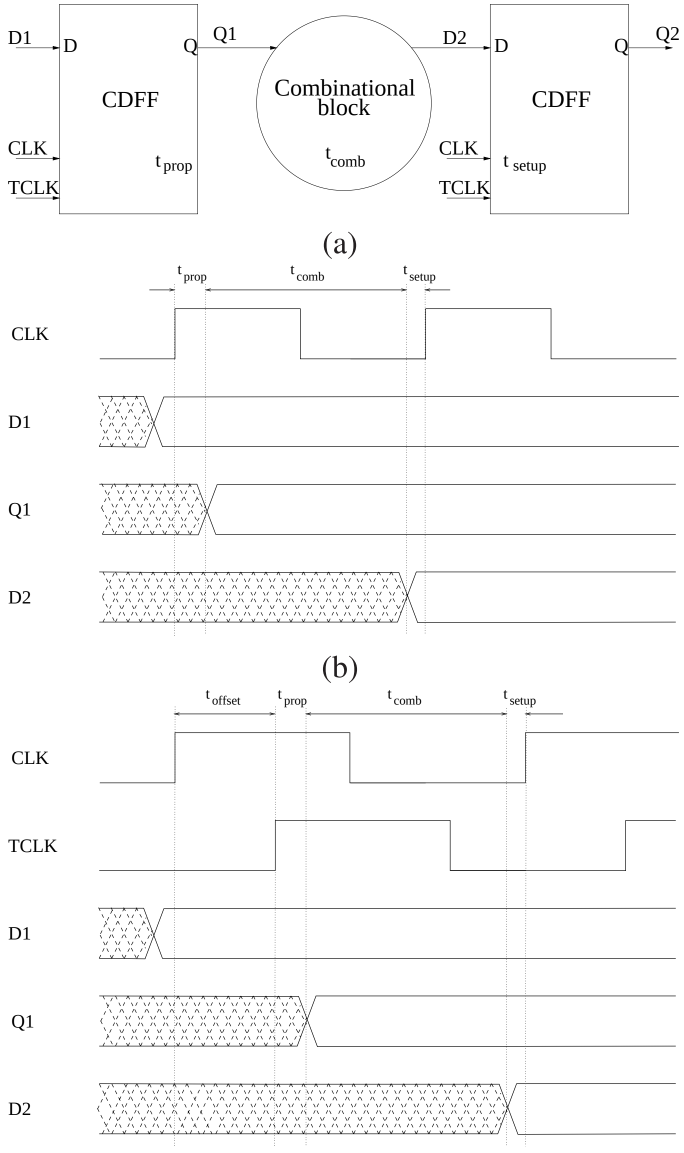
D_2, and t_{setup} is the setup time of the flip-flop.

In test mode, a tester programmed time offset of the clock is used to generate TCLK. Consequently, flip-flop output, Q_1, appears after an additional delay equal to the time offset between the two clocks. This scenario is illustrated in Figure 2.3(c). Under this condition, the test mode clock period is given by:

$$T_{TM} = t_{prop} + t_{comb} + t_{setup} + t_{offset} \tag{2.2}$$

where t_{offset} is the time offset between the clock and the test clock. The test mode clock period should be large enough to accommodate all delay terms in Equation 2.2. It is clear from this equation that increasing t_{offset} allows the circuit to be tested at a frequency lower than the normal mode frequency. In other words, clock frequency can be reduced while the combinational circuit delays are tested with the same delay margins.

In 2001, Nummer and Sachdev [6] proposed a methodology for generating the clock and the test clock for a device using CDFFs in a way that allows the test mode clock frequency to be reduced arbitrarily. This is done through an on-chip clock generation circuit. When generating the clock and the test clock for a circuit using CDFF to improve testability, one has to take into consideration the timing requirements for correct operation. For the CDFF to function properly, the timing of the clock and the test clock must be carefully adjusted to accommodate both the



(a)

(b)

(c)

Figure 2.3: CDFF operation. (a) Circuit model. (b) Normal mode. (c) Test mode.

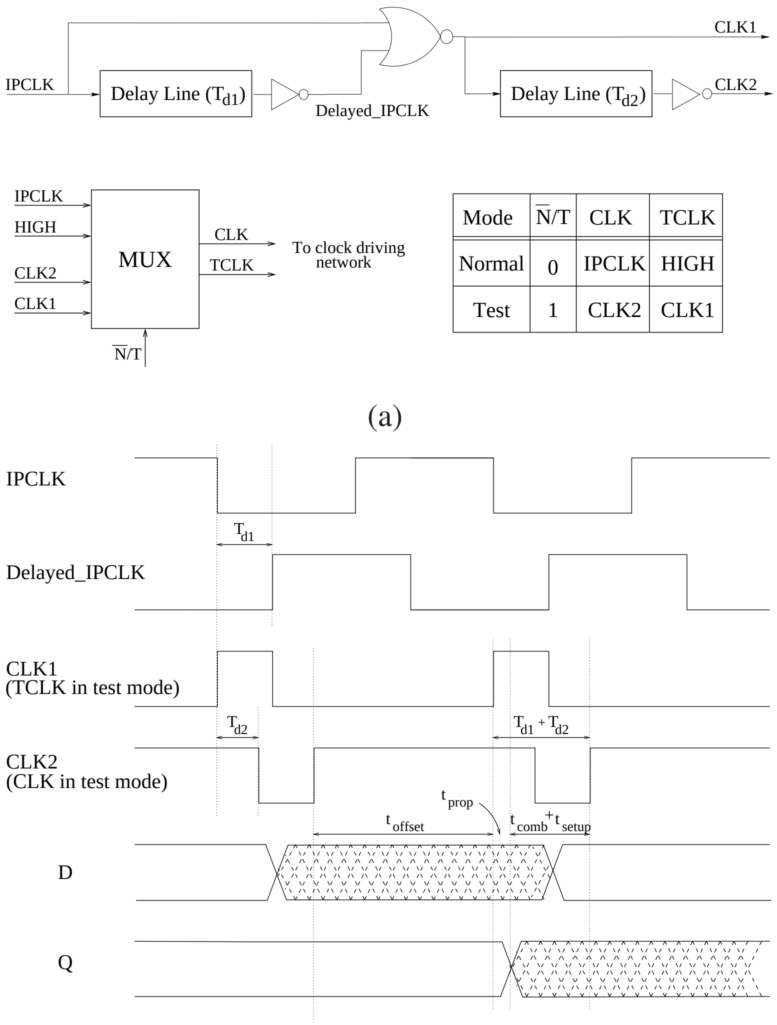
setup time (t_{setup}) and the propagation delay (t_{prop}) of the flip-flop. For the combinational block, it is necessary to have the flexibility to change the value of t_{comb} so as to determine, with reasonable accuracy, the delay through this block and test the circuit for delay faults. This is also important in order to enable performance binning[1] to know how well the circuit meets its timing specifications.

In test mode, reducing clock frequency while maintaining correct timing operation for all parts of the circuit means that, if the clock frequency becomes very low, t_{offset} has to be extremely large. As suggested in [9], the test clock can be generated as a delayed version of the clock with a delay of t_{offset}. The problem with this approach is that a slow tester is a low specification device. It is normally difficult for such a device to provide very large time offset with state-of-the-art timing accuracy. As an alternative, H. Speek et al. [10] suggested the use of two programmable duty-cycle controllers and a programmable delay line to generate the clock and the test clock in test mode. Using their design, reducing the test mode clock frequency to a very small value requires a large delay line to generate the required delay with appropriate timing resolution.

Careful examination of the timing diagram in Figure 2.3(c) shows that, instead of generating the test clock by delaying the clock in test mode, the clock can be generated by delaying and inverting the test clock. Generating the clock this way makes t_{offset} (which is the key factor in reducing the test mode clock frequency) independent on the relative timing of the two clocks and allows its value to be increased arbitrarily. Increasing t_{offset} while keeping all the other terms in Equation 2.2 unchanged implies a reduction in test mode clock frequency without affecting the time window allowed for the evaluation of the combinational block. It is clear that by doing this, the test mode clock frequency can be reduced with no lower limit.

Figure 2.4(a) depicts a block diagram of a system for generating the clock and the test clock. The input clock, IPCLK, is a rated frequency signal in normal mode and a low frequency, 50%

[1] Performance binning is a test procedure used to segregate devices in a number of bins depending on their maximum operating frequency.



Mode	$\overline{N}$/T	CLK	TCLK
Normal	0	IPCLK	HIGH
Test	1	CLK2	CLK1

(a)

(b)

Figure 2.4: Generating clock and test clock. (a) Block diagram. (b) Timing diagram.

duty cycle signal in test mode. A multiplexer (MUX) is used to select the mode of operation through the mode select input ($\overline{N}/T$). For normal mode operation ($\overline{N}/T$=LOW), IPCLK passes through the MUX to the CLK driving network while TCLK is kept high. In test mode ($\overline{N}/T$=HIGH), two delay lines are used to generate both CLK and TCLK. This is illustrated by the timing diagram in Figure 2.4(b). A delay line is used to generate a clock with pulse width T$_{d1}$ (CLK1). This clock is selected by the MUX to be the test clock, TCLK. CLK1 passes through the second delay line (with delay T$_{d2}$), resulting in CLK2. The MUX selects CLK2 to be CLK in test mode. Figure 2.4(b) also shows the D and Q signals of a CDFF to illustrate the relationships amongst the various timing parameters of the system in Figure 2.3(a) on one side and the delays T$_{d1}$ and T$_{d2}$ and the frequency of IPCLK, f, on the other side . These relationships can be expressed by the following two equations.

$$T_{d1} + T_{d2} = t_{prop} + t_{comb} + t_{setup} \tag{2.3}$$

$$1/f = T_{d1} + T_{d2} + t_{offset} \tag{2.4}$$

Assuming fixed f, t_{prop}, and t_{setup}, these equations suggest that a change in either T_{d1}, T_{d2}, or both, leads to an equal change in t_{comb}. This allows the combinational block to be tested for delay faults by changing the time slot allowed for the evaluation of its inputs. For constant T_{d1} and T_{d2}, changing f causes only t_{offset} to change without affecting the operation of neither the flip-flop or the combinational block.

Although using CDFFs can considerably facilitate the testing process, there are two main concerns associated with this technique.

1. The need for two clocks requires the creation of two clock propagation networks. Both networks need to have the same high-performance characteristics (small propagation delays, minimum clock skew, etc.). This is important to maintain the delay between the two clocks. The need for two high-performance clocks can result in an increase in the cost of the chip and complicates the process of clock generation and propagation.

2. Careful examination of the structure of the CDFF in Figure 2.2 shows that the path of dataflow is different in test mode when compared to normal mode operation. This means

that the CDFF propagation delays being tested in test mode is not the actual normal mode propagation delay.

In this chapter, a methodology for testing high-performance pipelines using slow-speed testers is proposed. The technique depends on creating shifted versions of a low-speed clock in test mode. The technique adds no extra hardware in the data path of the pipeline and therefore has virtually no performance penalty.

2.3 Testing Pipelines by Clock Shifting

The basic idea in the proposed technique is demonstrated in Figure 2.5. In this study, it is assumed that the registers in the pipeline use a single phase common clock and are built using flip-flops as the storage element. A clock timing circuit is used to control clocks driving all registers in the pipeline. The idea is to use shifted version(s) of the input clock, IPCLK, to control the timing of data flow through the pipeline in the test mode. In order to achieve that, each register in the pipeline has to have a separately routed clock, as shown in Figure 2.5. This results in higher complexity of clock generation and propagation. The clock net of a pipelined circuit normally consists of a tree of buffers. Special care should be given to balance the load of the different clock tree branches in order to keep the skew between the different clock signals within acceptable limits. In the proposed technique, only those buffers close to the clock timing circuit require extra design and layout effort. Due to the small number of these upstream buffers, the extra effort imposed by the technique would not be substantial.

The normal and test mode operation of the circuit is illustrated in Figure 2.6, showing stage i in the pipeline ($0 \leq i \leq n$). In normal mode, a single phase high-frequency clock is used for all registers in the pipeline. This is shown in Figure 2.6(b). As a result, the operation of the circuit

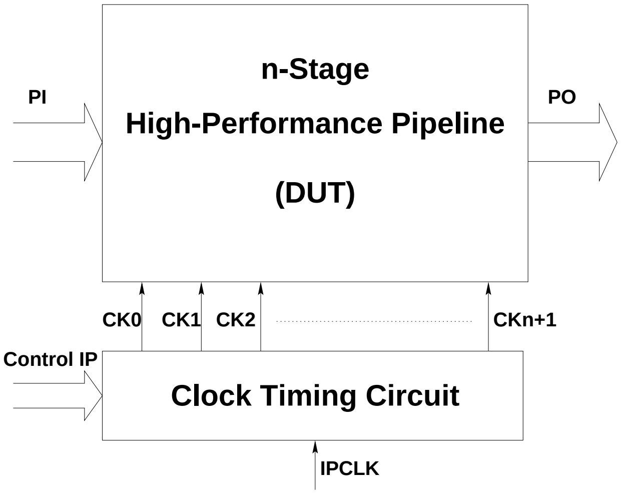


Figure 2.5: Block diagram of the DUT and the clock timing circuit.

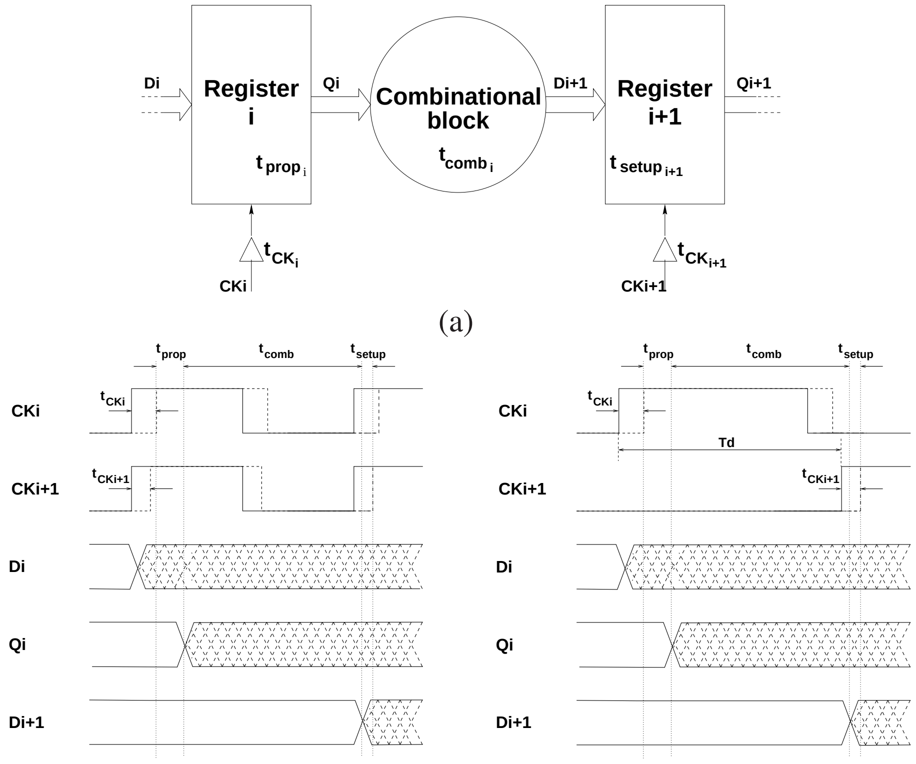
depends on the period of this clock. The delay of stage i, t_{d_i}, can be expressed using

$$t_{d_i} = t_{prop_i} + t_{comb_i} + t_{setup_{i+1}} + (t_{CK_i} - t_{CK_{i+1}}) \tag{2.5}$$

where t_{prop_i} is the propagation delay of register i, t_{comb_i} is the delay of the *ith* stage combinational block, $t_{setup_{i+1}}$ is the setup time of register $i + 1$, and $(t_{CK_i} - t_{CK_{i+1}})$ is the difference between the delays through the clock driving networks of registers i and $i + 1$. For the pipeline to function correctly, the normal mode clock period, T_{NM}, has to be at least equal to the largest stage delay, i.e.

$$T_{NM} \geq MAX_{i=0}^{n}(t_{d_i}) \tag{2.6}$$

In the test mode, a delayed version of the clock is used to test the pipeline. This is illustrated in Figure 2.6(c). In order to test stage i, a delayed version of the input clock, IPCLK, with delay T_d is applied to register $i + 1$, while the original clock is used for all other registers. The test mode clock period, T_{TM}, has to be at least equal to the normal mode clock period. It is clear that



(a)

(b)

(c)

Figure 2.6: Proposed technique. (a) Circuit model. (b) Normal mode. (c) Test mode.

using a larger value of T_{TM} means that it is possible to test the pipeline in the test mode using a slow-speed tester. Setting T_d to be equal to t_{d_i} allows the *ith* stage to operate within its normal mode timing constraints while the whole circuit is running at a lower frequency. As a result, a slow-speed tester can be used for performance binning and delay-fault testing. In the test mode, the tester supplies a slow-speed input clock as well as test vectors for the target stage. These vectors are supplied at the rate of the slow-speed clock. After a pre-determined number of clock cycles, the tester reads the results at the same rate of the slow-speed clock. This technique does not require any changes in the design of the registers or the combinational blocks and therefore has virtually no performance penalty.

For the proposed technique to achieve its goals, it is essential for the value of T_d to be controllable within state-of-the-art timing accuracy. Furthermore, the design should provide the ability to program T_d in order to bin the device under test (DUT) for performance. This feature also allows different stages in the pipeline to be tested even if they have unbalanced delays. For these reasons, an on-chip clock timing circuit is used to generate the delayed version of the input clock and to control the clocks propagated to the different stages in the pipeline.

2.4 Test Vehicle: 16-bit Pipelined Multiplier

In order to verify the benefits of the proposed technique, a 16-bit pipelined multiplier is designed and utilized as a test vehicle. Parallel multiplication is done in three steps. In the first step, the two operands are used to generate partial products whose number depends on the type of encoding algorithm used. The second step is to add the partial products together in a summation network which reduces the partial products to two operands. The product is generated in the final step by adding the resulting two operands using a carry propagate adder.

In our design, no encoding is used to generate the partial products. This results in a number

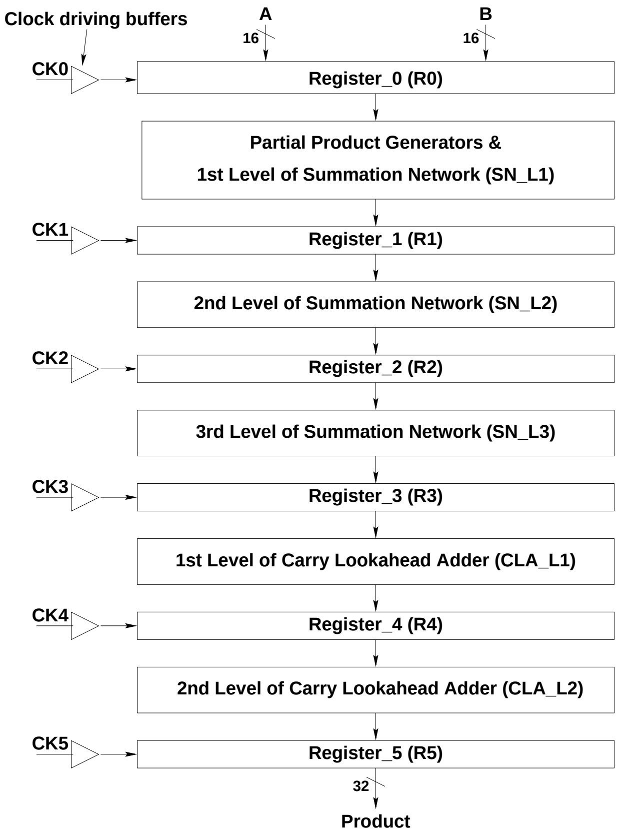

Clock driving buffers
A
16
B
16
CK0
Register_0 (R0)
Partial Product Generators &
1st Level of Summation Network (SN_L1)
CK1
Register_1 (R1)
2nd Level of Summation Network (SN_L2)
CK2
Register_2 (R2)
3rd Level of Summation Network (SN_L3)
CK3
Register_3 (R3)
1st Level of Carry Lookahead Adder (CLA_L1)
CK4
Register_4 (R4)
2nd Level of Carry Lookahead Adder (CLA_L2)
CK5
Register_5 (R5)
32
Product

Figure 2.7: Block diagram of the pipelined multiplier used as a test vehicle.

of partial products equal to the size of the multiplier (16). These partial products are added in the summation network using 4-2 compressors as the main component. A 4-2 compressor accepts 4 partial sums and reduces them to two [5]. In order to reduce the 16 partial products to 2 operands, this has to be done in three levels of 4-2 compression. A carry-lookahead adder with conditional sum select [7] is used to generate the product from these two operands.

The multiplier is implemented with five pipelined stages, as shown in Figure 2.7. The first stage is used to generate the 16 partial products and reduce them to 8 partial sums after the first level of the summation network. The second and third levels of the summation network are implemented in the second and third stages of the pipeline. The final addition is done in the last two stages.

Pipeline stages are separated by registers to control the timing of data flow through the multiplier. Static flip-flops are used as the storage elements in all registers. As shown in Figure 2.7, each register is controlled by a separate clock provided by the clock timing circuit. Details of the design of this circuit are given in the next section.

Performance characterization of the multiplier is carried out in order to find its maximum operating frequency and the critical path through each stage of the pipeline. These results are shown in Table 2.1. The delays shown include the propagation delay of the register feeding the stage, the setup time of the register accepting the output of the stage, and the difference in delays of the clock driving networks of the input and output registers (refer to Equation 2.5). As shown in Table 2.1, the first stage (SN_L1) has the largest delay and the operating frequency of the multiplier is determined by this stage. This delay is equal to 715ps which translates to a maximum operating frequency of 1.4GHz.

Pipeline stage	Critical path delay (ps)
SN_L1	715
SN_L2	655
SN_L3	655
CLA_L1	665
CLA_L2	615

Table 2.1: Critical path delays through multiplier stages.

2.5 Design of the Clock Timing Circuit

This section provides design details of the clock timing circuit used for the test vehicle. A number of issues have to be taken into account in the design of the clock timing circuit in order to ensure proper delay-fault testability of the pipeline.

1. **Programmability and resolution:** The delay, T_d, should be programmable for two reasons: (i) to allow DUT performance binning , and (ii) to allow paths of different delays to be tested for delay faults. The resolution with which T_d is varied should be high enough to allow the detection of small delay faults.

2. **Timing accuracy:** Process, temperature, and supply voltage variations have a large effect on the delays of the circuit. The design should compensate for these effects such that the value of T_d will be as desired regardless of these variations.

3. **Tester edge placement inaccuracy (jitter):** A slow-speed tester is generally a low specification device. It has a low timing resolution and high edge placement inaccuracy, or jitter. The design should have the ability to suppress the tester's jitter in order to minimize its effect on the timing resolution and accuracy of the clock timing circuit.

4. **Testability:** Adding hardware to enhance testability begs the question: how can we ensure the functionality of the extra hardware? is it testable? Answering these questions is very

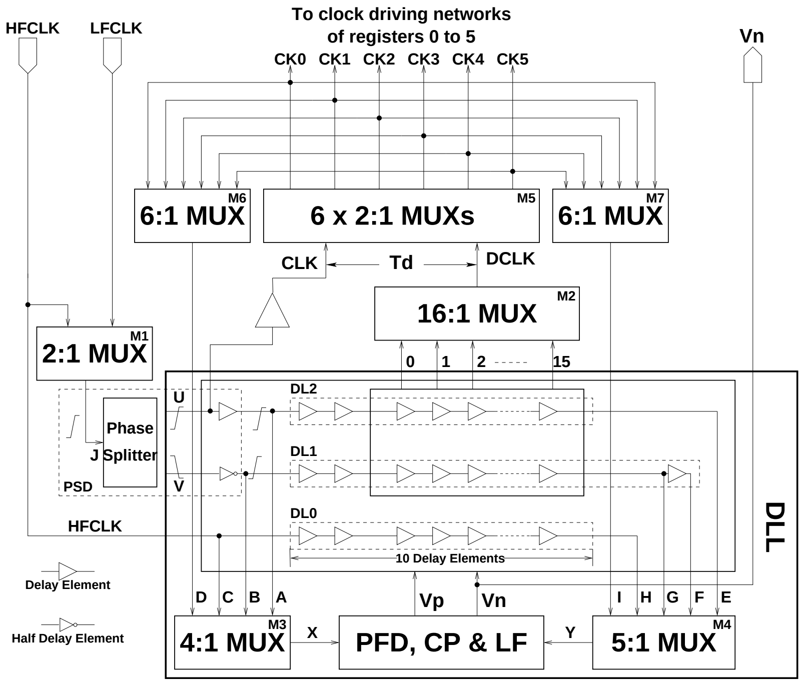


Figure 2.8: Clock Timing Circuit (multiplexer select inputs are not shown).

important and is a challenge to most design-for-testability techniques.

Figure 2.8 shows the clock timing circuit used in this design (multiplexer select inputs are not shown). This circuit is designed using CMOS 0.18μm technology provided through TSMC. It comprises three delay lines, a phase splitter & delay circuit, a DLL, and a number of multiplexers. The design allows T_d to vary between 250ps and 1000ps with 50ps increments (resolution). These are design variables and it is up to the designer to choose the values suiting a specific circuit or application. For our test vehicle, the range of T_d is reasonable to allow us to do performance binning and test for delay faults. A 50ps timing resolution is high enough considering the maximum operating frequency of the multiplier. The design and operation of the different blocks used in the clock timing circuit is described below.

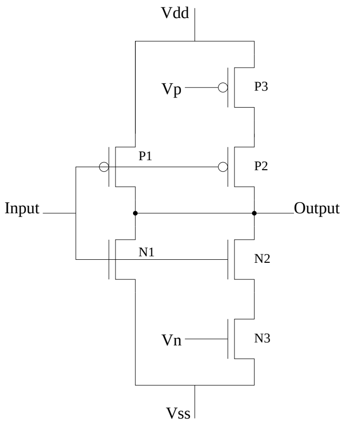


Figure 2.9: Schematic diagram of half the delay element.

The delay element used in the delay lines is built using static inverters connected in parallel with current-starved inverters. Delay line DL0 and the high-speed clock, HFCLK, are used to calibrate delay lines DL1 and DL2 in order to generate the required values of T_d. The calibration process ensures correct test mode timing regardless of PVT (process, temperature, and/or supply voltage) variations. Ideally, HFCLK should be generated on-chip through a phase-locked loop. For the purpose of this study, HFCLK is supplied from an external source which allows us to adjust the frequency of this clock in order to gain insight into the performance and the functionality of our design.

The design goal is to achieve a timing resolution of 50ps. Using the DLL to calibrate the delay lines requires a relatively balanced rising- and falling-edge delays, otherwise the clock might disappear before reaching the end of the delay line. Using the target technology (0.18μm CMOS), it was not possible to design a balanced delay element that achieves 50ps delay regardless of PVT variations (i.e. for all possible extreme cases). We achieved a timing resolution of 50ps using the PSD circuit and the two delay lines DL1 and DL2. The delay element is designed to have a delay of 100ps and the PSD circuit adjusts the delay between the inputs to DL1 and DL2 to 50ps (half delay-element delay) to achieve a timing resolution of 50ps.

With 10 delay elements in DL0, setting the period of HFCLK to 1ns and closing the DLL loop using this delay line results in control voltages V_n and V_p that adjust the total delay across DL0 to one clock period. Since these voltages control all three delay lines, the delays of all delay elements in the circuit are thus adjusted to 100ps. DL1 and DL2 consist of 11 and 10 delay elements, respectively. The main function of these two delay lines is to generate a delayed clock signal, DCLK, with programmable delays with respect to the reference clock, CLK (Fig. 2.8). Programmability is achieved through multiplexer M2.

Multiplexer M5 consists of six 2:1 multiplexers. It is used to control the clocks feeding all registers in the pipeline (CK0 to CK5). Depending on the mode of operation and the pipeline stage to be tested, M5 sets the clock of each register to either CLK or DCLK.

Closing the DLL loop using various combinations of the inputs to multiplexers M3 and M4 allows great flexibility for testing the clock-timing circuit itself. As a result, most of the extra hardware in our design can be tested simply by comparing values of the DLL output, V_n, when different signals are used to close the loop. This is achieved using multiplexers M1, M3, M4, M6, and M7 as is illustrated in the next section. The number of stages that can be tested using the proposed clock-timing circuit depends only on the design of multiplexer M5. As a result this design can be easily scaled depending on the pipeline to be tested. Depending on the size of M5 and in order to facilitate its testing, M6 and M7 need to be designed accordingly.

The proposed clock timing circuit takes into account all the issues mentioned at the beginning of this section. Programmability is achieved through DL1, DL2, and the multiplexer M2. Using a DLL allows the design to achieve the same timing accuracy regardless of process, temperature, and/or supply voltage variations. The DLL with a number of multiplexers help ensure the functionality of the clock timing circuit itself. With the help of the PLL, tester clock jitter is not allowed to propagate to DL0. It also has no effect on the timing accuracy of signals generated from DL1 and DL2. The area overhead due to this design is estimated to be 100 gates per pipeline stage. This should be acceptable for medium to large pipelined circuits. It is important to note that matching between the different components in the design is essential to ensure correct timing even with small local process, temperature and/or supply voltage variations. This can be achieved through circuit layout techniques similar to those used for analog circuits.

2.6 Modes of Operation

The clock timing circuit operates in three modes. In *normal mode* the pipeline is used to perform the function it is designed for. In the *DUT test mode*, the clock timing circuit is used to verify the performance of the DUT and to test it for delay faults. In the *clock timing circuit test mode*, the clock timing circuit is tested to ensure its ability to give a correct image about the performance of the pipeline.

2.6.1 Normal Mode

In normal mode, the control inputs of multiplexers M1 and M5 are set such that IPCLK propagates to all registers in the pipeline. As a result, the whole pipeline runs using the same clock ensuring normal operation. The remaining part of the clock timing circuit serves no function in normal mode, and can be turned off. This is important to reduce the effect the clock timing circuit has on normal mode power consumption of the chip.

2.6.2 DUT Test Mode

This mode is used to verify the performance of the DUT and test it for delay faults. The pipeline is tested one stage at a time. For every target path, two vectors are used to test the circuit. The first vector initializes the DUT, while the second vector activates the target path of the stage to be

Nodes	# of delay elements
A and E	10
B and G	10
C and H	10
A and F	$10\frac{1}{2}$
B and E	$10\frac{1}{2}$

Table 2.2: Number of delay elements between different nodes in the delay lines.

tested. Using multiplexer M2, T_d is set according to the expected delay of that path. Depending on the stage to be tested, multiplexer M5 is configured to set one of the register's clocks to DCLK and all other clocks to CLK. For example, to test stage 3 in the pipeline, clocks CK0, CK1, CK3, and CK5 are set to CLK while CK4 is set to DCLK. As a result, data flows between registers 3 and 4 within normal mode timing constraints. For all other stages, the low frequency clock allows operation under relaxed timing. This is important to ensure that delay faults in these stages do not affect the target path and hence the stage under test. This procedure is repeated for every path to be tested until the DUT is tested completely.

2.6.3 Clock Timing Circuit Test Mode

In this mode, the DLL, with the help of multiplexers M1, M3, M4, M6, and M7, is used to characterize and verify the performance of the clock timing circuit. Multiplexer M1 is set to allow the high-frequency clock, HFCLK, to propagate through DL1 and DL2. The clock timing circuit is tested in three phases as shown in Figure 2.10. In the first phase, all three delay lines (excluding the last delay element in DL1) are tested. The second phase is used to test the PSD circuit. This is done by verifying that the delay between nodes A and B is equal to half delay-element delay. The third phase is used mainly to test multiplexer M5. This phase also covers multiplexers M6, M7, and a small part of M2. Each of the three phases cover parts of multiplexers M3 and M4, such that these two multiplexers are fully testable. In this design, only M1 and most of M2 are not testable. Compared to the rest of the clock timing circuit and the DUT, this is a very small area of the circuit. Additionally, delay faults in M1 do not impact the operation of the circuit. Fully testing M2 would require additional hardware which itself may not be testable.

Phase 1: Testing DL0, DL1, and DL2

In this phase, delay lines DL0, DL1, and DL2 are tested between nodes C and H, B and G, and A and E, respectively. Table 2.2 gives the number of delay elements between the different nodes in the clock timing circuit. The first three entries in the table have the same number of delay elements. As a result, closing the DLL loop with these node pairs one at a time, should result in very close values of V_n (monitored off-chip) for all three configurations. As shown in Figure 2.10, the test in this phase is done in four steps:

1. Close the DLL loop using C and H ($M3 : C \rightarrow X; M4 : H \rightarrow Y$). Measure V_n and record it as V_{n_0}.
2. Close the DLL loop using B and G ($M3 : B \rightarrow X; M4 : G \rightarrow Y$). Measure V_n and record it as V_{n_1}.
3. Close the DLL loop using A and E ($M3 : A \rightarrow X; M4 : E \rightarrow Y$). Measure V_n and record it as V_{n_2}.
4. Compare V_{n_0}, V_{n_1}, and V_{n_2}. If the differences between the three values are within acceptable limits, the three delay lines are considered to be free of delay faults.

Ideally, the three values should be equal. Mismatch between the delay elements and the different components in the DLL would result in some differences. Characterization is necessary to define how much difference due to mismatch and process variations is acceptable. In this context, it is assumed that only a single delay fault exists in the circuit. It is worth noting that it

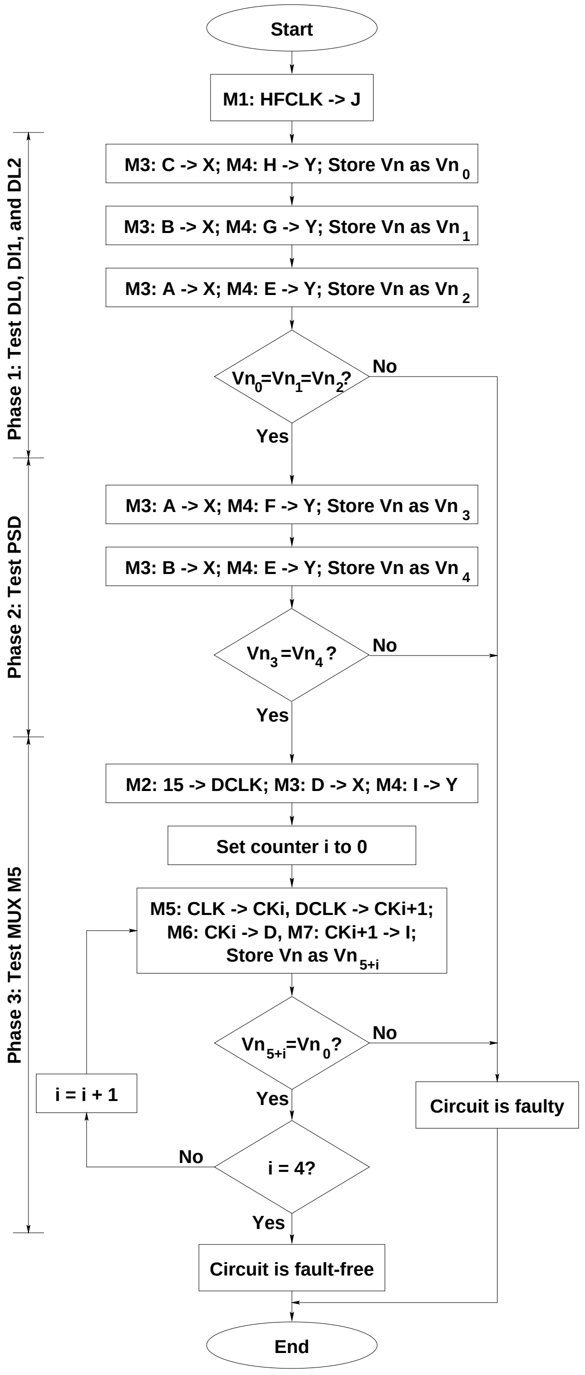

Start
M1: HFCLK -> J
M3: C -> X; M4: H -> Y; Store Vn as Vn_0
M3: B -> X; M4: G -> Y; Store Vn as Vn_1
M3: A -> X; M4: E -> Y; Store Vn as Vn_2
$Vn_0=Vn_1=Vn_2$?
No
Yes
M3: A -> X; M4: F -> Y; Store Vn as Vn_3
M3: B -> X; M4: E -> Y; Store Vn as Vn_4
$Vn_3=Vn_4$?
No
Yes
M2: 15 -> DCLK; M3: D -> X; M4: I -> Y
Set counter i to 0
M5: CLK -> CKi, DCLK -> CKi+1;
M6: CKi -> D, M7: CKi+1 -> I;
Store Vn as Vn_{5+i}
$Vn_{5+i}=Vn_0$?
No
Yes
i = i + 1
Circuit is faulty
No
i = 4?
Yes
Circuit is fault-free
End
Phase 1: Test DL0, DI1, and DL2
Phase 2: Test PSD
Phase 3: Test MUX M5

Figure 2.10: Procedure for testing the clock timing circuit.

is highly unlikely to have the exact same amount of delay fault in two of or all three delay lines. In addition to DL0, DL1, and DL2, the test in this phase covers paths through multiplexers M3 and M4 used to close the DLL loop (A-X, B-X, C-X, E-Y, G-Y, and H-Y).

Phase 2: Testing the Phase Splitter & Delay Circuit

Referring to Table 2.2, the delay between nodes A and F on one side and nodes B and E on the other are both equivalent to $10\frac{1}{2}$ delay elements. This is only true if the two outputs of the PSD circuit (A and B) are exactly half delay-element delay apart. These observations are used to test the PSD circuit for delay faults. As shown in Figure 2.10, the test is done first by closing the DLL loop using nodes A and F. When locked, DLL output V_n is recorded as V_{n_3}. Similarly, V_{n_4} is obtained by closing the DLL loop through nodes B and E. For fault-free PSD, the difference between V_{n_3} and V_{n_4} should be within acceptable limits (defined through characterization). The test in this phase covers the G-F-Y path as well.

Phase 3: Testing Multiplexer M5

Phase 3 of the clock timing circuit test procedure is used to test multiplexer M5. This is done with the help of multiplexers M6 and M7. The main idea is to set the delay between CLK and DCLK (T_d) to one cycle of HFCLK (1ns). The different paths in M5 are tested two at a time. As shown in Figure 2.10, to test the paths from CLK to CK_i and DCLK to CK_{i+1} (where i is any number between 0 and 4), M5 is set accordingly and M6 and M7 are used to close the DLL loop using nodes CK_i and CK_{i+1}. Under these conditions, if all signal paths are free of delay faults, when locked, the DLL output, V_n, should be equal to the values obtained in phase 1 of the test procedure (for fault free DL0, DL1, and DL2). In addition to M5, the test in this phase covers M6, M7, the 15-DCKL path in M2, the D-X path in M3, the I-Y path in M4, and the buffer used for CLK.

2.7 Delay Fault Simulation Results

Several simulations were carried out to test the proposed design and to verify the possibility of testing the DUT and detecting small delay faults at low frequency. Furthermore, delay fault simulations for the clock timing circuit are done in order to demonstrate the ability to test and characterize this circuit. In all such simulations, delay faults are inserted using buffers and delay elements. Using active elements to add extra delays ensures the existence of a pre-determined delay fault without degrading the quality (rise and fall times) of the delayed signals.

2.7.1 Delay Fault Simulation for the Test Vehicle

Delay faults are inserted in the test vehicle one at a time using buffers. Simulations are carried out on a number of paths in the DUT. For every target path, two vectors are used to test the circuit. The first vector initializes the DUT while the second vector activates the target path. The delay of a given path depends on the input vectors of the circuit. As a result, for every stage in the pipeline, the critical path and its delay might change depending on the applied vectors. For all paths tested in the simulations, activation vectors are chosen such that these paths are the critical paths in their pipeline stage for these vectors. As mentioned before, the design allows adjustment of T_d such that paths of different delays can be tested for small delay faults.

Delay fault simulation results for the test vehicle are shown in Table 2.3. The test is done at an input clock frequency of 100MHz. Delay faults of 50ps are inserted in each path one at a time. T_d from the clock timing circuit is set to the next higher value compared to the delay of the target path. Under these conditions, the multiplier gives incorrect output for all paths tested in our simulations, as shown in Table 2.3. The left half of the table gives the delays of the different paths and the value of T_d used to detect a 50ps delay fault in each path. The right half of the table gives the test vectors used for each path as well as the fault-free and faulty products of the

Path #	Pipeline stage	Path delay (ps)	T_d (ps)	Vector type	Input A	Input B	Product (fault-free)	Product (faulty)
1	SN_L1	715	750	Init.	0002	FFFF	0001 FFFE	0001 FFFE
				Activ.	0000	FFFF	0000 0000	0000 0008
2	SN_L1	685	700	Init.	000C	FFFF	000B FFF4	000B FFF4
				Activ.	0008	FFFF	0007 FFF8	0007 FFF0
3	SN_L2	655	700	Init.	0001	FFFF	0000 FFFF	0000 FFFF
				Activ.	0000	FFFF	0000 0000	0000 0040
4	SN_L2	620	650	Init.	0070	FFFF	006F FF90	006F FF90
				Activ.	0040	FFFF	003F FFC0	003F FF80
5	CLA_L1	665	700	Init.	0000	FFFF	0000 0000	0000 0000
				Activ.	FFFF	FFFF	FFFE 0001	FFFD 0001
6	CLA_L1	425	450	Init.	FFFF	FFFF	FFFE 0001	FFFE 0001
				Activ.	0000	FFFF	0000 0000	0001 0000
7	CLA_L2	615	650	Init.	0000	FFF0	0000 0000	0000 0000
				Activ.	FFE0	FFF0	FFD0 0200	FFC0 0200
8	CLA_L2	415	450	Init.	FFE0	FFF0	FFD0 0200	FFD0 0200
				Activ.	0000	FFF0	0000 0000	0010 0000

Table 2.3: Test vehicle delay fault simulation results.

multiplier. The extent of delay fault that goes undetected is a function of the slack between the delay of the path and the value of T_d used to test it. A path with larger slack will have a larger undetectable delay fault. For example, in the case of path #2, it is expected that delay faults as small as 15ps should be detectable using the same value of T_d. On the other hand, for path #3, it will take a delay fault of at least 45ps to cause the timing failure. The situation is worse if the target path is not the critical path for the applied vectors. In general, delay fault detection is dependent on the target path delay. Most of the delay fault testing techniques have similar limitations. Balancing path delays is the most commonly used method to alleviate this problem.

Fault #	Fault location	Faulty Path	Delay fault (ps)	V_{n_0} mV	V_{n_1} mV	V_{n_2} mV	$\triangle V_n$ mV
F1	DL0	C-H	60	**633**	611.5	611.5	21.5
F2	DL1	B-G	100	611.5	**649**	611.5	37.5
F3	DL2	A-E	200	611.5	611.5	**691.7**	80.2
F4	M3	A-X	60	611.5	611.5	**594.2**	-17.3
F5	M3	B-X	100	611.5	**583.3**	611.5	-28.2
F6	M3	C-X	200	**554**	611.5	611.5	-57.5
F7	M4	E-Y	60	611.5	611.5	**634**	22.5
F8	M4	G-Y	100	611.5	**648.7**	611.5	37.2
F9	M4	H-Y	200	**697**	611.5	611.5	85.5

(a) Faults detected in phase 1 (fault-free V_n is 611.5mV)

Fault #	Fault location	Faulty Path	Delay fault (ps)	V_{n_3} mV	V_{n_4} mV	$\triangle V_n$ mV
F10	PSD	J-A	60	609.8	651.5	-41.7
F11	PSD	J-B	100	669	597.9	89.1
F12	M4	F-Y	200	720	628.6	91.4

(b) Faults detected in phase 2 (fault-free V_n is 628.6mV)

Fault #	Fault location	Faulty Path	Delay fault (ps)	V_n mV	$\triangle V_n$ mV
F13	M2	15-DCLK	60	633.2	21.7
F14	M5	DCLK-CK4	100	647	35.5
F15	Buffer	U-CLK	200	554.8	-56.7

(c) Faults detected in phase 3 (fault-free V_n is 611.5mV)

Table 2.4: Delay fault simulation results for the clock timing circuit.

2.7.2 Delay Fault Simulation for the Clock Timing Circuit

Delay fault simulations for the clock timing circuit were carried out in order to verify the ability to characterize the performance of the circuit. 15 delay faults are inserted in different locations of the circuit one at a time. These fault are generated using the same delay element used in the design. The delay of the delay element is at its minimum value when $V_p = V_{ss}$ and $V_n = V_{dd}$. This value is 60ps, and hence the smallest delay fault used in the simulations is 60ps. Larger delays are achieved using larger values of V_p and smaller values of V_n. The test is done according to the procedure in Figure 2.10. The results are shown in Tables 2.4(a), (b), and (c), showing faults detected in each phase of the clock timing circuit test procedure.

A fault detectable in phase 1 or phase 3 of the test procedure causes only one value of V_n (V_{n_0}, V_{n_1}, or V_{n_2} for phase 1 or $V_{n_{5+i}}$ for phase 3) to deviate from the fault-free value. That is why for faults detected in phase 1 and phase 3, the last column in each table gives the difference in V_n compared to the fault-free value. This is different from faults detectable in phase 2, where the fault causes both V_{n_3} and V_{n_4} to deviate in opposite directions from the fault free value. For faults detected in phase 2, $\triangle V_n$ in Table 2.4(b) is computed as the difference between V_{n_3} and V_{n_4}.

A positive $\triangle V_n$ indicates that the delay seen by the DLL is larger than it should be. As a result, the DLL causes the voltage to increase in to order to compensate for the extra delay. The opposite is true for negative $\triangle V_n$. For a certain delay fault, the larger the value of $\triangle V_n$, the easier it is to observe the error due to the fault. As shown in Table 2.4, the smallest value of $\triangle V_n$ is 17.3mV which can be easily measured off-chip. These results demonstrate our ability to test the clock timing circuit for delay faults. This is important to ensure its ability to give a true image about the operation and performance of the DUT.

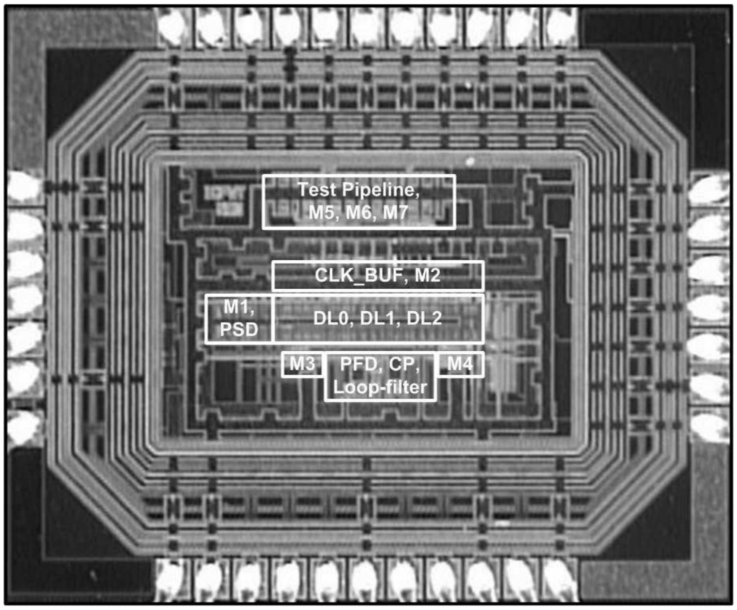


Figure 2.11: Chip micrograph.

2.8 Implementation and Test Setup

The clock-timing circuit in Fig. 2.8 has been implemented in 0.18μm CMOS technology. The chip micrograph is shown in Fig. 2.11 and Fig. 2.12 illustrates its block diagram. The layout of the chip is done such that the delay elements and the various components in the clock-timing circuit are closely matched. The core area of the chip is 700×500μm^2, out of which 500×300μm^2 is dedicated to the clock-timing circuit. At a supply voltage of 1.8V, a 1GHz HFCLK, and a 50MHz LFCLK, the clock-timing circuit consumes 13.6mW.

The design includes a simple 5-stage test pipeline used to verify the ability of the technique to perform slow-speed timing characterization. The schematic of the test pipeline is shown in Fig. 2.12. Each stage in the pipeline consists of a number of voltage controlled delay elements (VCDEs) whose control voltage, $V_{n_{PL}}$, is adjusted off-chip allowing us to modify the delay of the pipeline stages. The higher the value of $V_{n_{PL}}$, the smaller the delays of the VCDEs, and

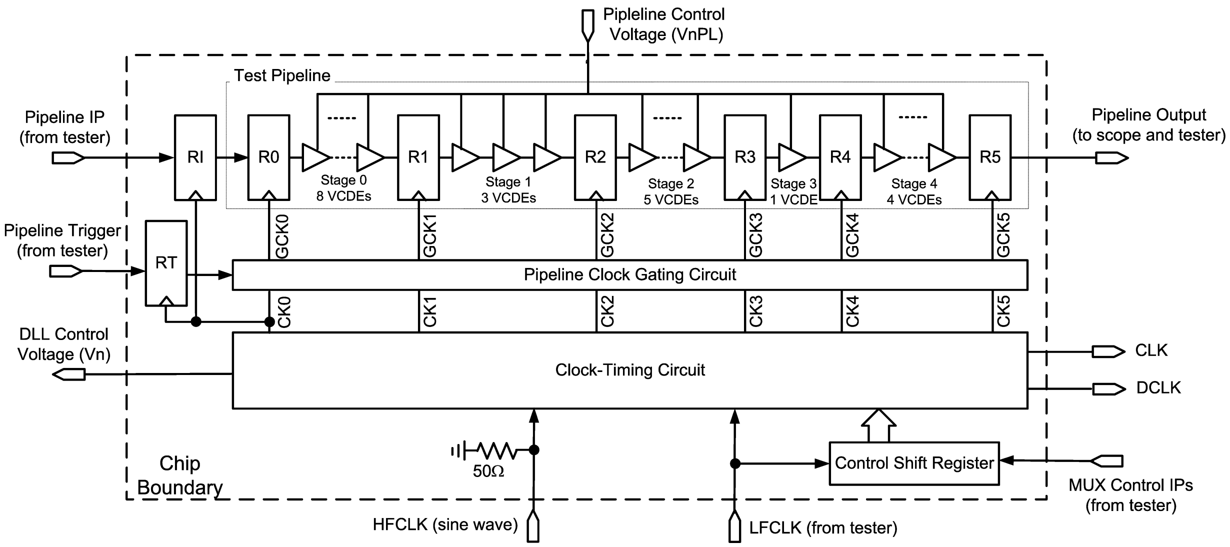


Figure 2.12: Chip block diagram showing the schematic of the test pipeline.

consequently the stage delays, will be. The input register, RI, is used to synchronize the input from the tester with CK0 generated from the clock-timing circuit. This clock is derived from LFCLK in the pipeline test mode and from HFCLK in the normal mode.

In the pipeline test mode, it is easy to observe the output of the pipeline using the slow tester. However, observing the at-speed normal mode output with the pipeline running at frequencies around 1GHz is not feasible because of the limitations of the output drivers and the package. Observing the normal mode output of the pipeline is important in order to verify the pipeline test mode results. In order to overcome this problem the test chip includes a pipeline clock-gating circuit as shown in Fig. 2.12. The tester supplies a pipeline trigger signal which is synchronized by CK0. This signal is then supplied to the clock gating circuit which produces gated versions of the clocks that run for the exact number of cycles needed for the pipeline to produce the correct output in the normal mode.

Normal mode testing is done by applying an initialization vector at least twice to ensure proper initialization of the pipeline. This is then followed by the activation vector. Because

Table 2.5: Test results for the clock-timing circuit

Phase	DLL Config.	Measured V_n (mV)
1	C-H	$V_{n_0} = 680.7$
	B-G	$V_{n_1} = 680.2$
	A-E	$V_{n_2} = 673.6$
2	A-F	$V_{n_3} = 712.2$
	B-E	$V_{n_4} = 707.8$
3	D-I	Minimum: $V_{n_5} = 701.9$ Maximum: $V_{n_9} = 705.8$

register RI uses a non-gated clock, it stores the input vector with the next edge of CK0. When the tester activates the pipeline trigger input, the clock-gating circuit will activate the gated clocks such that the input will propagate through the pipeline until the output is stored in R5. The gated clocks will then be disabled allowing this output to be easily observed.

HFCLK is generated through a sine wave supplied from an R&S SMT06 signal generator. This signal is biased to have a DC level of 0.9V and is terminated with an on-chip 50Ω resistor. The sine wave is then converted to HFCLK through a number of inverters. An Agilent VXI test system is used to supply LFCLK as well as the control inputs to the multiplexers. These inputs are provided from the tester serially and are then stored in a shift register. The tester is also used to supply the input of the test pipeline and to capture its output. The pipeline output is also monitored using an Agilent Infiniium 54853A digital storage oscilloscope. The scope is also used to monitor the CLK and DCLK signals. The control output of the DLL (V_n) is monitored using a Fluke 867B graphical multimeter. The control input for the test pipeline (V_{nPL}) is supplied to the chip using a variable resistor acting as a voltage divider.

2.9 Experimental Results

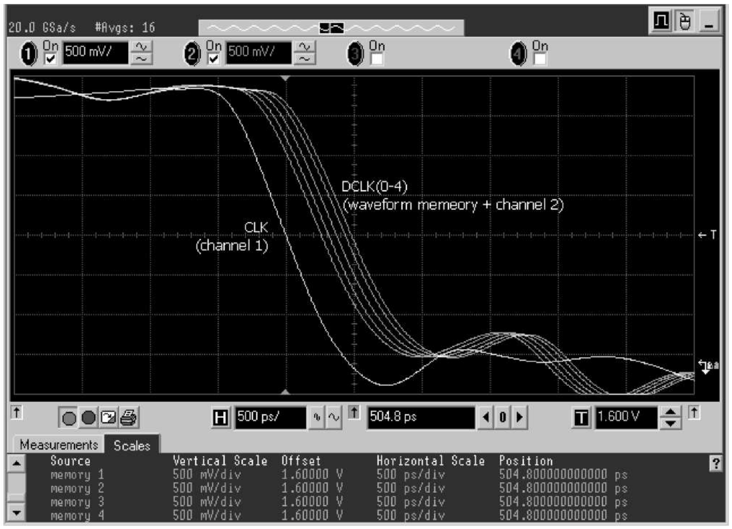
The chip has been tested in the three modes of operation detailed in section 2.6. The testing starts by testing the clock-timing circuit and verifying the results using the scope. Based on these results, the pipeline is tested using various configurations and the results are verified against the normal mode test results.

2.9.1 Clock-Timing Circuit Test Mode Results

The clock-timing circuit was tested according to the procedure in Fig. 2.10. The test results are presented in Table 2.5.

Phase 1 produces values of V_{n_0}, V_{n_1}, and V_{n_2} that are within 7.1mV of each other. Through characterization of the delay element, this difference corresponds to a deviation of less than 1ps per delay element. Hence, the three delay lines are assumed to be free of delay anomalies. Similarly, phase 2 reveals close values of V_{n_3} and V_{n_4} which indicates that the PSD circuit is also free of delay anomalies. For phase 3, the values of V_{n_5}-V_{n_9} are all within 3.9mV of each other. However, these measurements are, on average, 25.5mV higher than those of phase 1. This indicates the presence of unintentional delay in the DCLK path from tap 15 in Fig. 2.8 through multiplexers M2, M5, and M7 compared to the CLK path through the CLK buffer, M5, and M6.

In order to verify these results, the CLK and DCLK signals are observed off-chip. Fig. 2.13 shows the CLK and DCLK (taps 0-4) waveforms as observed using the scope. Fig. 2.14 presents the measured CLK-DCLK delays for all taps in the clock-timing circuit compared to the ideal and post-layout simulation delays. The measured average step is 52.9ps which verifies the conclusion that DL0, DL1, and DL2 are free of delay anomalies. From this average, the average delay deviation between the even- (from DL1) and the odd- (from DL2) numbered taps is 3.5ps. This explains the slight variation in V_{n2} compared to V_{n0} and V_{n1} and the small discrepancy between



Source	Vertical Scale	Offset	Horizontal Scale	Position
memory 1	500 mV/div	1.60000 V	500 ps/div	504.800000000000 ps
memory 2	500 mV/div	1.60000 V	500 ps/div	504.800000000000 ps
memory 3	500 mV/div	1.60000 V	500 ps/div	504.800000000000 ps
memory 4	500 mV/div	1.60000 V	500 ps/div	504.800000000000 ps

Figure 2.13: CLK versus DCLK for taps 0-4 (using scope waveform memory).

V_{n3} and V_{n4}. The discrepancies between the post-layout and the measured delays verify the presence of larger delays in the DCLK path. These delay anomalies explain the higher values obtained for V_{n_5}-V_{n_9}. However, part of the additional delays can be attributed to differences in the drivers, bonding wires, PCB traces, and/or scope probes in the path of CLK and DCLK.

2.9.2 Pipeline Test Mode Results

Pipeline test mode operation is used to perform timing characterization of the test pipeline. This is achieved by adjusting $V_{n_{PL}}$ and finding the smallest DCLK tap in the clock-timing circuit that results in a consistently correct output of the pipeline. This process is repeated for all stages in the test pipeline. Each test is run 2M times and the pipeline output is observed using the "infinite display persistence" function of the scope. This function allows us to view the pipeline output for all test runs overlayed on the scope's display. The number of test runs is estimated based on test execution and output observation times. For correct operation, the pipeline output switches seven clock cycles after applying the input transition. Given that the clock-gating circuit provides

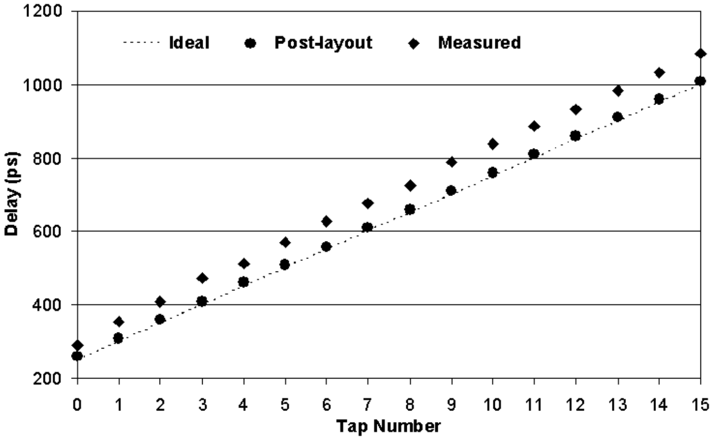


Figure 2.14: CLK-DCLK delays for all taps in the clock-timing circuit.

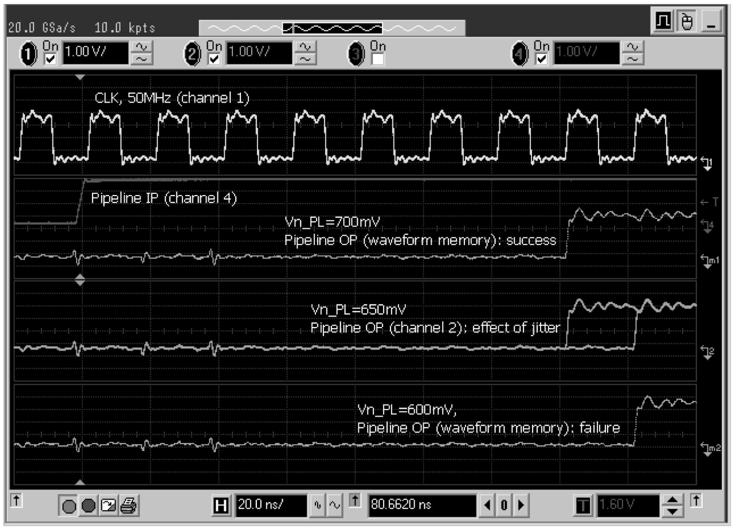
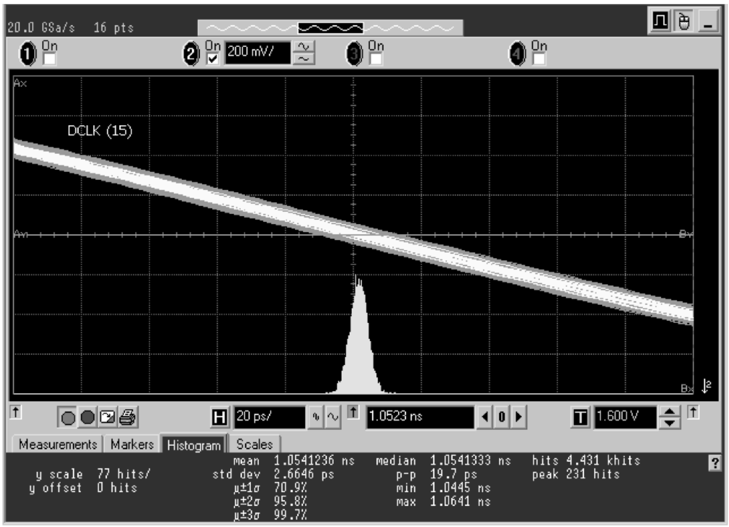


Figure 2.15: Pipeline test mode results for stage 3 (LFCLK: 50MHz).

an additional cycle than is needed for the test mode operation, if a stage fails due to timing constraints violation, the output switches one clock cycle later.

Fig. 2.15 presents an example for this procedure for stage 3 (one VCDE + flip-flop) in the pipeline at 50MHz LFCLK and with DCLK set to tap 3 (474ps CLK-DCLK delay). When $V_{n_{PL}}$ is set to 0.7V, the pipeline produces correct output for all test runs. When this voltage is reduced to 0.65V, the stage delay increases such that the pipeline starts to fail for some of the test runs. The fourth waveform in Fig. 2.15 shows that the output switches after seven clock cycles for some runs and after eight cycles for others. This behavior is attributed to timing uncertainty due to the CLK and DCLK jitter. The peak-to-peak jitter of DCLK has been measured to be equal to 19.7ps as shown in Fig. 2.16 (this value includes jitter due to the output driver). The fact that the circuit behavior changes with such small uncertainty in the CLK-DCLK delay further verifies that the delay margin allowed for the evaluation of the stage under test is within high-speed timing constraints. When $V_{n_{PL}}$ is further reduced to 0.6V, the pipeline fails for all test runs. Similar results were obtained for all stages in the test pipeline.

Table 2.6 shows the smallest DCLK tap that results in a consistently correct pipeline output for various values of $V_{n_{PL}}$. For example, when $V_{n_{PL}}$ is set to 0.6V, stage 0 in the pipeline (the critical path) fails for all DCLK taps. This indicates that, for this value of $V_{n_{PL}}$ the stage delay is larger than the largest possible value of T_d. For the same value of $V_{n_{PL}}$, stages 1 to 4 produce consistently correct output when DCLK is generated using taps 8, 12, 4, and 6, corresponding to T_d values of 727ps, 932ps, 512ps, and 628ps respectively. These results show the ability of the design to characterize the performance of the individual stages at frequencies much lower than the rated operating frequency.



```
                             mean  1.0541236 ns   median  1.0541333 ns   hits 4.431 khits
  y scale   77 hits/     std dev  2.6646 ps         p-p  19.7 ps          peak 231 hits
 y offset   0 hits          μ±1σ  70.9%             min  1.0445 ns
                            μ±2σ  95.8%             max  1.0641 ns
                            μ±3σ  99.7%
```

Figure 2.16: Jitter histogram for DCLK (tap 15).

Table 2.6: Test mode results for the test pipeline.

Stage	No. of	$V_{n_{PL}}$ and the corresponding DCLK tap				
No.	VCDEs	0V	0.6V	0.8V	1V	1.8V
0	8	Fail	Fail	14	14	13
1	3	8	8	6	6	6
2	5	13	12	10	9	8
3	1	4	4	3	3	3
4	2	6	6	5	4	4

2.9.3 Normal Mode Results

The pipeline test mode results are verified by performing at-speed testing of the pipeline in the normal mode. For a given value of $V_{n_{PL}}$, the frequency of the normal mode clock (HFCLK) is adjusted in 1MHz steps until the maximum normal mode frequency, determined by the delay of the critical stage in the pipeline (stage 0, 8 VCDEs), is identified. Fig. 2.17 provides a comparison between these results and the test mode results for the critical stage. As shown, the test mode operation tracks the normal mode behavior within a timing margin less than or equal to the timing

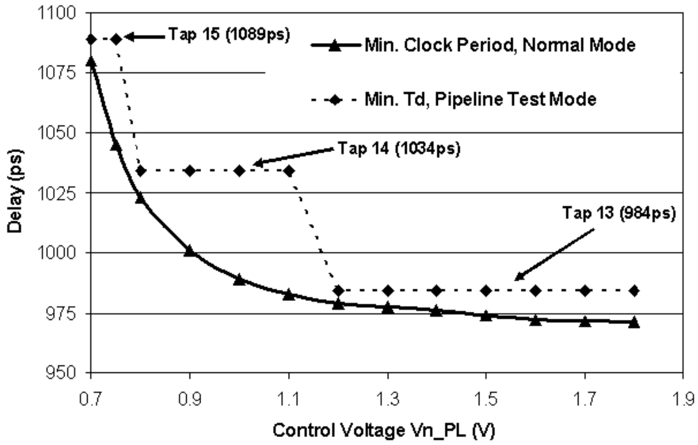


Figure 2.17: Timing results for normal and test modes.

resolution of the clock-timing circuit.

2.10 Conclusions

In this chapter, a methodology for testing high-performance pipelined circuits with slow-speed testers was presented. In this technique, each pipeline stage is clocked using a separate clock generated from an on-chip clock timing circuit in test mode. The technique adds no extra hardware in the data path of the pipeline and therefore has no performance penalty.

A design for the clock timing circuit capable of achieving a timing resolution of 52.9ps in 0.18μm CMOS technology was presented. The design provides the ability to test the clock timing circuit itself.

The effectiveness of the technique was demonstrated using a 16-bit pipelined multiplier as a test vehicle. Simulations show that it is possible to detect delay faults as small as 50ps at an input clock frequency of 100MHz. Simulations also prove the ability of the technique to test the clock timing circuit itself for delay faults.

Experimental results show that the design is capable of evaluating the performance of the clock-timing circuit and tracking the normal mode timing behavior of the test pipeline. These results demonstrate the effectiveness of the technique and its potential in reducing manufacturing test cost for high-speed pipelined datapaths.

Although the design requires observing V_n off-chip, using an all-digital DLL whose control word can be scanned out allows the test to be fully performed on an all-digital ATE.

References

[1] V.D. Agrawal and T.J. Chakraborty. High-Performance Circuit Testing with Slow-Speed Testers. In *Proc. of the International Test Conference*, pages 302–310, 1995.

[2] J.T.Y. Chang and E. McClusky. Quantitative Analysis of Very-Low-Voltage Testing. In *Proc of the VLSI Test Symposium*, pages 332–337, April 1996.

[3] P. Franco. Testing Digital Circuits for Timing Failures by Output Waveform Analysis. Technical Report 94-9, Center for Reliable Computing, Stanford University, September 1994.

[4] H. Hao and E.J. McCluskey. Very-Low-Voltage Testing for Weak CMOS Logic ICs. In *Proc. of the International Test Conference*, pages 275–284, October 1993.

[5] M. Mehta, V. Parmar, and Jr. E. Swartzlander. High-Speed Multiplier Design Using Multi-Input Counter and Compressor Circuits. In *Proc. of the IEEE Symposium on Computer Arithmetic*, pages 43–50, June 1991.

[6] M. Nummer and M. Sachdev. A Methodology for Testing High-Performance Circuits at Arbitrarily Low Test Frequency. In *Proc. of the IEEE VLSI Test Symposium*, pages 68–74, April 2001.

[7] N. Ohkubo et al. A 4.4 ns CMOS 54×54-b Multiplier Using Pass-Transistor Multiplexer. *IEEE Journal of Solid-State Circuits*, 30(3):251–257, March 1995.

[8] S. Sengupta et al. Defect-Based Test: A Key Enabler for Successful Migration to Structural Test. *Intel Technology Journal*, Q1, 1999.

[9] M. Shashani and M. Sachdev. A DFT Technique for High-Performance Circuit Testing. In *Proc. of the International Test Conference*, pages 267–285, October 1999.

[10] H. Speek, H. G. Kerkhoff, M. Sachdev, and M. Shashaani. Bridging the Test Speed Gap: Design for Delay Testability. In *Proc. of the IEEE European Test Workshop*, pages 3–8, 2000.

Chapter 3

A Digitally-Controlled Delay Element with Monotonic and Linear Delay Characteristics

In chapter 2, different phases of the input clock were generated using a DLL that uses voltage-controlled delay elements in order to adjust the delays of the individual stages. In recent years, there has been greater emphasis on using digital control words to adjust the timing of delay lines and delay elements. The elimination of analog control signals and the associated circuity allows for better integration and facilitates the debugging and diagnosis of digital systems.

Many applications use digitally-controlled delay lines (DCDLs) and/or delay elements (DCDEs). In reference [6], a successive approximation register-controlled delay-locked loop (SARDLL) is used to reduce the skew between the core clocks of a multi-chip system. The design uses a digitally-controlled delay cell in order to adjust the delay of the SARDLL's delay line and lock the input clock to a replica of the remote clock. References [9, 11, 14] demonstrate the application of a DCDL in designing DLLs for high-speed random access memory (RAM) controllers.

A dual delay line architecture is used in [5,16,22] to design a DLL with wide dynamic range and high-resolution. Wide dynamic range is made possible through a DCDL while a DCDE is used to achieve high-precision delay interpolation.

On-chip clock skew management is another example for applications that make use of DCDLs/ DCDEs. Such designs are based on adjusting the phases of two (or more) clocks in two clock domains in order to reduce the clock skew or add/remove intentional skew between clocks of performance critical paths. A fine resolution DCDE is used in [7,10,13,21], while a combination of coarse/fine resolution delay elements are used in [20]. The use of digital control allows the deskew buffer's control word to be scanned in/out for test/diagnosis purposes.

All-digital phased-locked loops (ADPLL) have several advantages over the traditional charge-pump-based PLLs. They are capable of achieving fast locking, better integration in standard CMOS processes, and portability over different process generations. In an ADPLL, a digitally-controlled oscillator (DCO) replaces the traditional voltage-controlled oscillator (VCO). DCO design relies on building a ring oscillator using only DCDEs [8,17] or a combination of DCDEs and DCDLs [2–4,12].

ATE is another application domain for DCDLs/DCDEs. Reference [1] proposed a production ATE based on timing verniers that use a combination of both structures in order to generate precisely timed digital signals.

In this chapter, a new design of a DCDE is proposed. The design is capable of achieving precise delay steps while maintaining the monotonicity and linearity of the digital code vs. delay characteristics. Monotonicity means a consistent increase/decrease of the delay of the DCDE with changing input code. Within the context of analog systems, linearity is characterized in terms of the harmonic distortion of the system. However, within the context of this work, linearity is an indication of the possibility of achieving uniform steps for the DCDE. This definition of linearity allows for evaluating the transfer characteristics of the DCDE from the perspective of

Implementation principle	Ref.	Process	Delay step	Uniform steps?	Monotonic?
Logic threshold	[16]	0.15μm	10ps	Yes	Yes
Bias dependence of gate caps	[2]	0.35μm	1.55ps	Yes	Yes
	[3]	0.18μm	?	Yes	Yes
Shunt gate caps	[7,21]	0.18μm	8.5ps	Yes	Yes
	[10]	0.25μm	12ps	Yes	Yes
Tri-state inverters/buffers	[4,5]	0.35μm	3ps (min.)	No	Yes
	[12]	0.6μm	1ps (min.)	No	Yes
	[17]	0.35μm	10-150ps	No	Yes
	[22]	0.18μm	?	Yes	Yes
MUX'ed inverters	[13]	0.18μm	?[1]	Yes	Yes
Current-starved with binary weights	[8]	0.5μm	?[2]	No	?
	[20]	0.13μm	6ps (avg.)	No	Yes[3]
Current-starved with indirect binary control	[15]	0.18μm	2ps (min.)	No	Yes

[1] Estimated from other reported data not to be less than 2.67ps.

[2] Reported a 74fs resolution through simulations.

[3] Not guaranteed as simulations show non-monotonic behaviour.

Table 3.1: Summary of DCDE implementations (table provides reported experimental data only).

digital systems. It also allows for evaluating the robustness of the design with environment variations (i.e the ability to maintain step uniformity with PVT variations).

3.1 Review of DCDE Implementation Techniques

Several DCDE circuit techniques have been proposed in the literature. In general, designing a DCDE depends on digitally modulating the element's logic threshold, load capacitance, or (dis)charge resistance. This section presents a review of these techniques. A summary of this review is given in Table 3.1

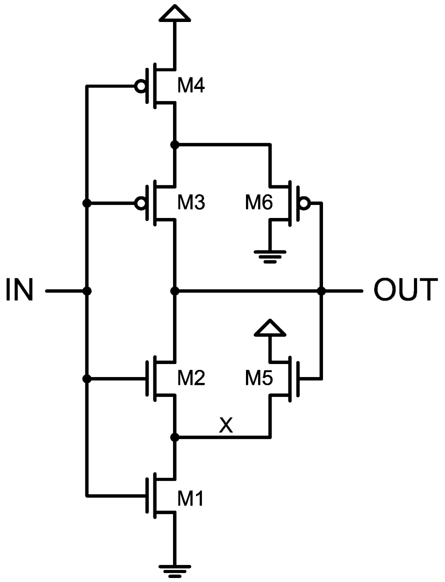


Figure 3.1: Schmitt trigger.

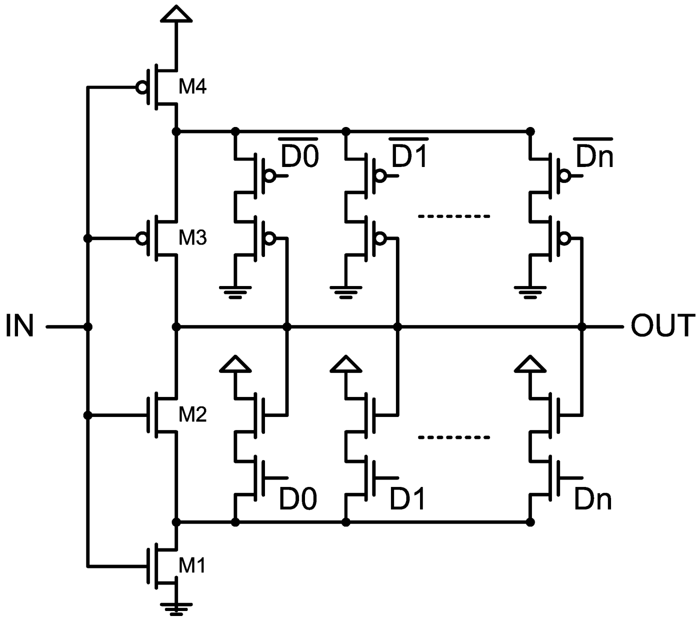


Figure 3.2: Schmitt trigger-based DCDE.

3.1.1 DCDE using adjustable logic threshold

A CMOS inverter has a logic threshold that determines the level that a rising/falling input has to reach before the output node starts to fall/rise. A Schmitt trigger is a variation of the normal inverter that allows the circuit to have non-equal rise/fall logic thresholds. One way to achieve that is to use source followers at the output node as shown in Figure 3.1. For a rising input, the logic threshold depends on the voltage at node X. This voltage is a function of the ratio of the *ON* resistances of transistors M_1 and M_5. For a given W/L for M_1, the larger the *ON* resistance of M_5, the lower the logic threshold and, consequently, the delay would be. Similar analysis applies to the falling input.

By digitally adjusting the equivalent *ON* resistance of transistors M_5 and M_6, it is possible to digitally control the delay of the Schmitt trigger. Such design was proposed in [16] and is illustrated in Figure 3.2. The NMOS block has a number of control branches, each with a transistor acting as a switch and a transistor that controls the logic threshold. When a switch transistor is turned *ON*, the total resistance of the control branches is reduced, resulting in an increase in the threshold voltage for the rising input. Using control branches in the PMOS block allows the delay for both input transitions to be adjusted simultaneously. Using this design, reference [16] reported a 10ps resolution in 0.15μm technology.

The resolution of this design is limited by how small the resistance increments can be. Small resistance increments correspond to larger resistance for the individual control transistors. With this design, it is possible to use long channel transistors to achieve infinitesimally small resolution. The main drawback of this approach is that achieving very small delay steps requires very large channel lengths for the threshold control transistors resulting in a large DCDE area. For example, simulations in 0.18μm CMOS technology show that a 1ps resolution requires the threshold control transistors to have a 10μm channel length. As the threshold control transistors are connected to the output node of the DCDE, the increased area results in increase in DCDE power consumption due to the larger capacitance at that node.

3.1.2 DCDEs using adjustable load capacitance

The delay of a CMOS gate is directly proportional to the capacitance at the output node. This capacitance includes the intrinsic capacitance of the gate, the wiring capacitance, and the input capacitance of the subsequent stage(s). The delay of the gate can be manipulated by adding/removing capacitance to/from this node. Two techniques for achieving that have been proposed in the literature. Both techniques use gate capacitances to achieve delay programmability; one technique uses bias dependence of gate capacitance and the other uses selectable shunt capacitors . Given the parallel nature of the capacitance increments, these techniques achieve good step uniformity when using equal capacitance increments. Because the gate capacitance is proportional to the width and length of a transistor, the resolution of these techniques is limited by the minimum

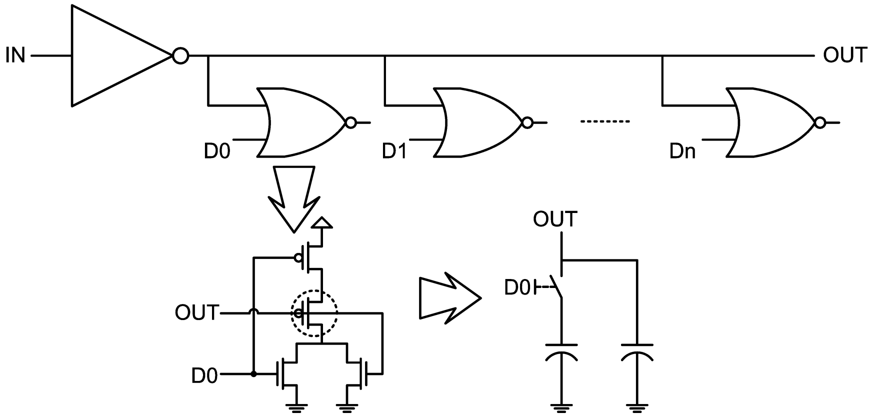


Figure 3.3: DCDE using bias dependence of gate capacitance.

size of a transistor in a particular technology. It is possible, however, to use larger drivers in order to achieve higher resolution. One of the main drawbacks of these techniques is that the gate capacitance is non-linear which requires care in designing these DCDEs.

1. **DCDE using bias dependence of gate capacitance.** The total capacitance at the gate of an MOS transistor depends on the transistor's mode of operation. In the linear mode, the average gate capacitance is approximately one and half times that of the gate capacitance in saturation mode [19]. References [2, 3] proposed a DCDE based on this phenomena. This is illustrated in Figure 3.3, showing an inverter driving NOR gates used as a programmable capacitors. When one of the digital inputs, Dx, is LOW, the average capacitance seen at node OUT becomes higher than when Dx is HIGH. As a result, the delay of the inverter can be adjusted based on the value of Dx. The desired number of delay steps can be achieved by using an equal number of NOR gates at the output of the inverter.

   Using the arrangement shown in Figure 3.3, reference [2] reported a 1.55ps resolution in 0.35μm technology. It should be noted, however, that the capacitance is non-linear and its variation with the digital input is different for rising and falling signals. As a result, this delay element has different delay steps for different signal polarities. In 0.18μm technology, simulations show that the delay step for a rising signal is three times that of a falling signal. This could pose a problem for applications requiring equal rise and fall delays. It is possible to use another type of programmable capacitor or a second stage to alleviate this problem. This would, however, be at the expense of increasing the minimum delay step that can be achieved by this DCDE.

This technique gives monotonic delays and achieves good delay step uniformity. Its main drawback is the large number of transistors needed for each delay step and the associated area overhead. This is even worse when using three input gates as proposed in [3]. Another drawback is the power consumption associated with the switching of the gates used as programmable capacitors. For these reasons, this technique is suitable mainly for cell-based designs.

2. **DCDE using selectable shunt capacitors.** In this technique, the capacitance at the output node is adjusted by multiplexing transistor gates used as load capacitances [7, 10, 21]. This is illustrated in Fig. 3.4 showing selectable shunt capacitors using a transmission gate multiplexer. The load capacitors use both PMOS and NMOS transistors in order to achieve better capacitance linearity and to equally adjust the delay steps for both rising and falling signals [10]. As shown in Table 3.1, using this technique, references [7, 21] reported uniform and monotonic delay steps of 8.5ps in 0.18μm technology. Similar to the previous technique, this design requires a large number of transistors (six) per delay step which makes the associated area large. It also suffers from the non-linear nature of the capacitance at the output node (due to both the transmission gates and the shunt capacitors).

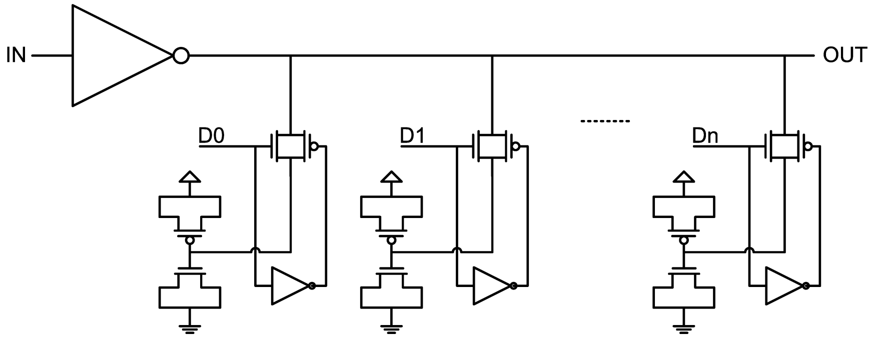


Figure 3.4: DCDE using shunt capacitors.

3.1.3 DCDEs using adjustable (dis)charge resistance

Instead of adjusting the capacitance at the output node, it is possible to achieve delay programmability by adjusting the resistance in the (dis)charge path. Adjusting the resistance can be achieved directly by digitally controlling the effective size of the control transistors in the (dis)charge paths, or indirectly by digitally adjusting the bias of such transistors. Contrary to capacitances, adding resistance increments in parallel results in a non-linear change in the total resistance. As a result, adding/removing equal resistance increments in the (dis)charge paths produces non-uniform delay steps. Careful sizing of the control transistors is needed in order to achieve delay step uniformity. Moreover, using these techniques, the larger the parallel resistance increment, the smaller the delay step would be. Hence, DCDEs using adjustable resistances are, in general, capable of achieving infinitesimally small delay steps either by using smaller transistors widths or larger lengths. Using larger transistor lengths, however, adds to the parasitic capacitances and can result in non-monotonic behavior.

1. **DCDE using parallel tri-state inverters.** A tri-state inverter uses a PMOS and an NMOS control transistors in order to connect/disconnect the inverter to/from the output node. This is illustrated in Figure 3.5. When the enable input is high, the control transistors are ON resulting in normal inverter operation. On the other hand, when the enable is low, both

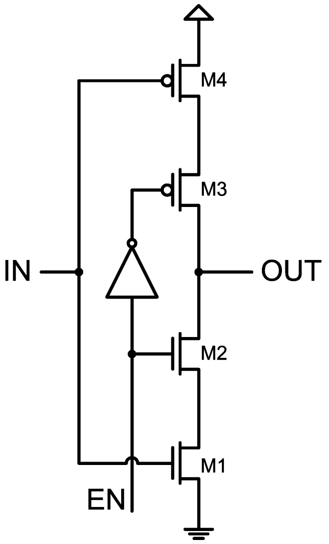


Figure 3.5: Tri-state inverter.

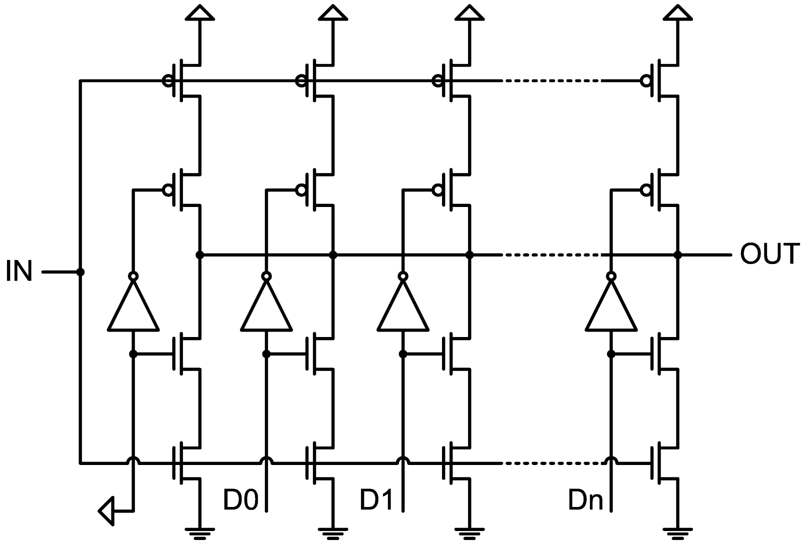


Figure 3.6: DCDE using tri-state inverters.

control transistors are OFF resulting in a high impedance at the output node. By connecting multiple of these inverters in parallel, the ON resistance of the (dis)charge path can be modulated using digital control inputs. This is illustrated in Figure 3.6. By progressively turning the tri-state inverters ON/OFF, the delay of this DCDE can be modulated monotonically and in finite steps. When using equally sized tri-state inverters, the sizes of these steps decrease with the number of inverters that are ON, resulting in non-uniform steps. Reference [17] reported that using this design the delay step ranges from 150ps to 10ps in 0.35μm technology. Similar designs were also used in [4, 5, 12].

In order to improve step uniformity, the tri-state inverters need to be sized such that the design experiences a fixed resistance reduction when an additional tri-state inverter is turned ON. This is a first order analysis that ignores the impact of parasitic capacitances on the delay steps. For example, to achieve three uniform steps, the DCDE requires four tri-state inverters with relative sizes of 3:1:2:6. This sizing ensures a constant resistance step of 1/12 of the resistance of the smallest cell in the design (with relative size “1”). Similarly, for four uniform steps, the relative sizes should be 12:3:5:10:30 giving a resistance step of 1/60 of that of a transistor with a relative size of "1". Reference [22] reported achieving good step uniformity using multi-fingered tri-state inverters. It is obvious that achieving uniform steps using this design can require very large transistors. With the switching of all of these transistors, the associated power consumption can be very large as well. References [12] and [17] reported that the DCOs (built using tri-state inverter DCDEs) consume 85% and 50% of the total design power consumption, respectively.

Instead of using tri-state inverters, reference [13] proposed using a multiplexed inverter DCDE. This design is shown in Figure 3.7. The outputs of a number of inverters (equal to the number of steps+1) are connected through an NMOS-only multiplexer. This design allows the control of both rising and falling edges using three transistors per delay step (versus six for the tri-state inverter DCDE). An additional PMOS transistor is used to ensure proper HIGH level at the output node. Similar to the tri-sate inverter DCDE, this design requires careful sizing of the inverters in order to achieve uniform steps. Its power dissipation is large due to the large number of inverters. The main advantage of this design over the tri-sate inverter DCDE is the reduced number of transistors per delay step.

2. **DCDE using current-starved inverters with binary-weighted or geometrically-sized control transistors.**

   The use of current-starved inverters is one of the popular techniques for designing voltage-controlled delay elements. The basic structure of a current-starved inverter that independently controls the delays of rising and falling inputs is shown in Figure 3.8. Two control voltages, V_n and V_p, modulate the ON resistance of the control NMOS and PMOS transistors. This in turn results in modulating the delays of the inverter. Reference [8] proposed using binary-weighted control transistors in order to design a DCDE based on the current-

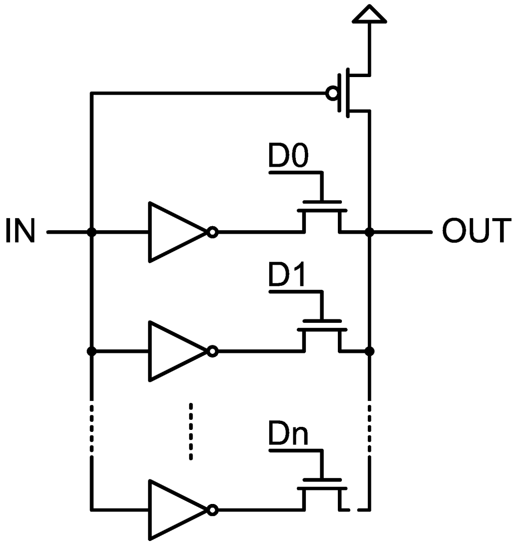


Figure 3.7: DCDE using multiplexed inverters.

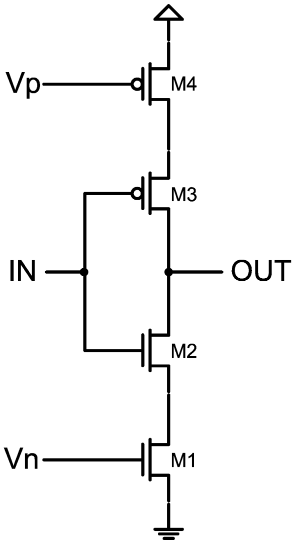


Figure 3.8: Voltage-controlled delay element using current-starved inverter.

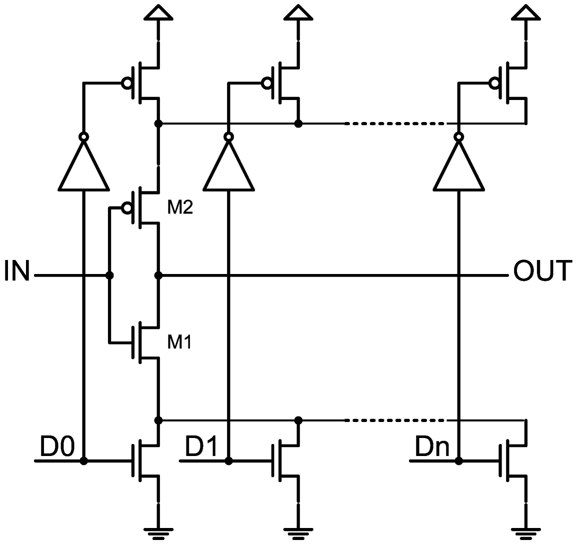


Figure 3.9: DCDE using current-starved inverter.

starved inverter. This design is illustrated in Figure 3.9. Depending on the binary control code, the equivalent (dis)charge ON resistance is modulated resulting in delay adjustment. Using this design, the delay steps are non-uniform and decrease with increasing input code resulting in non-linear delay characteristics.

Another drawback of this design is that it does not guarantee monotonic delay behavior. This was discussed in detail in [15]. For a rising input, as soon as transistor M_1 in Figure 3.9 turns ON, the charge at the output node is shared with the parasitic capacitances of the control transistors resulting in a fast drop in the output voltage. The average drain capacitance is higher when a control transistor is ON. As a result, this charge sharing increases with the input vector resulting in larger than expected delay steps. It is common in this DCDE design to use large L control transistors in order to achieve small delay steps. If a control transistor with a large L is turned ON, it produces a larger than expected delay reduction due to the large parasitic capacitance and the associated charge sharing. Alternatively, if a control transistor with smaller L is turned ON, it might so happen that the delay reduction would be smaller (opposite to what is expected for a transistor with a smaller resistance) due to the smaller parasitic capacitance and the reduced amount of charge sharing. When this situation happens, this DCDE experiences non-monotonic delay behavior.

A similar design that uses binary codes and geometric (non-binary) sizing of the control transistors was proposed in [20]. This design achieves an average delay step of 6ps in 0.13μm technology. Using geometric, non-binary, sizing of the transistors improves the overall delay linearity but produces non-monotonic changes in the delay steps and, potentially, non-monotonic overall delay behavior. Figure 3.10 shows simulated and measured delays of this design. Although not clear in the measurement data, simulation results show the non-monotonic behavior of this design. Moreover, this DCDE design can be affected

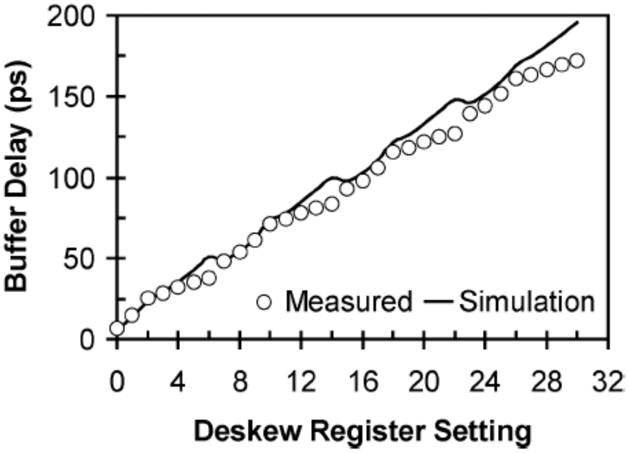


Figure 3.10: Transfer characteristics of DCDE using current-starved inverters with geometric sizing of control transistors [20].

with the charge sharing problem when using large *L* control transistors.

One of the important advantages of the DCDE proposed in [20] is the sharing of the control transistors between two DCDEs. Because the control transistors are only active during the switching of the DCDE input, it is possible to share these transistors between multiple DCDEs that have mutually exclusive transition time slots. This feature is applicable to all current-starved inverters and can significantly reduce the area associated with control transistors.

3. **DCDE using current-starved inverters with indirect delay control using binary-weighted transistors.** One way to overcome the charge sharing problem of the current-starved DCDEs is to control the ON resistance of the (dis)charge path without affecting the parasitic capacitance. This design was proposed in [15] and is shown in Figure 3.11. A control block is constructed using binary-weighted PMOS transistors and a diode-connected NMOS transistor. This control block produces a control voltage, V_n, whose value increases monotonically with the input code. A larger value of V_n reduces the ON resistance of the

NMOS control transistor in the current-starved inverter resulting in faster switching of the output node.

Although this DCDE design ensures monotonic delay characteristics, it produces non-uniform delay steps that decrease with increasing input code. Two factors contribute to this non-linearity in the code-delay characteristics. The first factor is the negative feedback effect of increasing V_n. Increasing the input code value causes both the current in the diode-connected NMOS transistor as well as the control voltage, V_n, to increase. This increase in V_n causes the drain-source voltage of the PMOS control transistors to decrease, resulting in a diminished return on current. As a result, the amount of current contributed when a particular PMOS transistor is turned ON depends on how many other transistors are ON at the same time. Consequently, the increments in current and control voltage are non-uniform and decrease with increasing input code. The second factor that contributes to the non-uniform steps of this DCDE is the inherent non-linear voltage-delay characteristics of the current-starved inverter.

3.2 DCDE Design Considerations

From this review of the different applications and implementations of DCDEs, the important features that a DCDE should have can be summarized in the following.

1. **Monotonicity and Step Uniformity.** One of the most important features a DCDE should have is monotonic and linear code-delay characteristics. This ensures predictable and stable behavior. It also facilitates the analysis of designs using such delay cells. For example, DCDEs with non-monotonic behavior can cause an ADPLL to become unstable, while non-uniform steps make it harder to calculate its loop gain [17]. Non-monotonic behavior happens in designs that have contradictory factors affecting the delay of the DCDE.

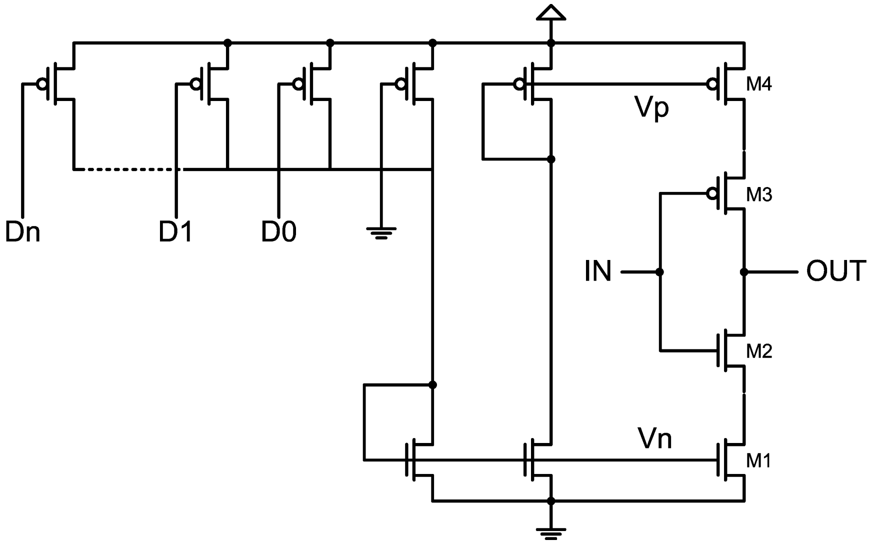


Figure 3.11: DCDE using current-starved inverters with indirect delay control using binary-weighted transistors.

Uniform steps can be easily achieved in designs relying on adjusting the load capacitance while designs using current-starvation are inherently non-linear. Achieving uniform steps using tri-state or multiplexed inverters requires careful transistor sizing and can have a large impact on area and power consumption.

2. **Resolution.** Generally, applications using DCDEs require achieving fine resolutions. Most of the designs proposed in the literature are capable of achieving infinitesimally small steps. This could, however, be at the expense of increased area, power consumption or loss of monotonicity. It should also be noted that such fine resolutions will be prone to variations due to process mismatch and other non-idealities. Additionally, system jitter due to device and supply noise can limit the usability of DCDEs with such fine resolutions.

3. **Control of Different Edges.** Some applications require adjusting both rising and falling edges of a signal in order to maintain its duty cycle. This requires equal steps for rising

and falling edges. Others require independent control of rising and falling edges. For example, references [7, 13, 21] suggest that timing characterization of performance critical paths can be done using duty cycle control of the local clocks. Achieving such requirement is not possible using some of the implementations proposed in the literature (e.g. DCDE implementations relying on adjusting load capacitance).

4. **Area.** When a DCDE is required to have a large number of delay steps, special attention should be given to the area needed for its layout. In addition to a fixed number of transistors, DCDEs proposed in the literature use anywhere between 1 and 6 transistors for every delay step. This number can be doubled if independent control of rising and falling edges is necessary. It can also be reduced by half, or more, if the control transistors can be shared between two or more DCDEs whose inputs have mutually exclusive transition time slots.

5. **Power.** Some DCDE implementations consume large amounts of power. Others are very power efficient. Although other design considerations are of higher importance to DCDE designs, it is always attractive to use more power efficient implementations. This would be particularly attractive for applications whose power consumption comes primarily from the DCDEs or those using a large number of these cells.

In the following section, a new DCDE is proposed. The design achieves monotonic and linear code-delay characteristics. It requires one control transistor per delay step. These control transistors can be sized in order to achieve infinitesimally small delay steps. The design can be easily modified in order to allow simultaneous or independent control of rising and falling edges or sharing of control transistors between multiple DCDEs. In terms of its power consumption, the proposed design falls half way between very power efficient designs (like the shunt capacitor DCDE) and power inefficient designs that have many transistors (dis)charing simultaneously.

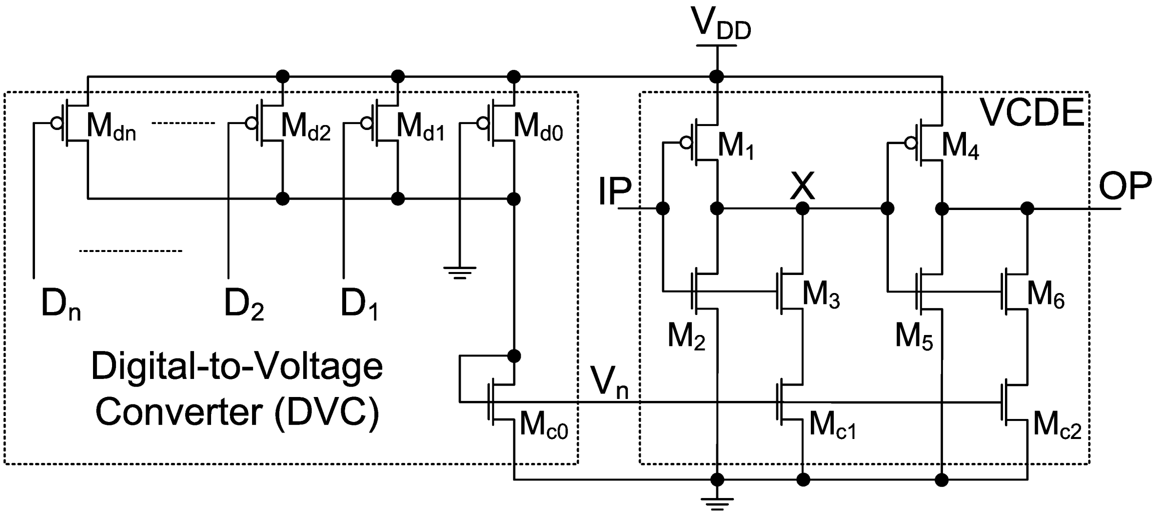


Figure 3.12: Schematic diagram of the proposed DCDE.

3.3 Proposed DCDE

The schematic diagram of the proposed DCDE is shown in Fig. 3.12. The DCDE consists of two basic building blocks: a voltage-controlled delay-element (VCDE) and a digital-to-voltage converter (DVC). The VCDE is composed of two identical stages each consisting of a static inverter connected in parallel with a current-starved NMOS block. Using two identical stages ensures balanced rising and falling edge delays.

The voltage-delay characteristics of the VCDE are shown in Fig. 3.13. For small values of the control voltage, V_n, the current-starved branches are OFF and the delay of the VCDE is determined by the static inverter. This gives the maximum delay that can be achieved using the VCDE. As V_n increases, currents through the current-starved blocks start to increase resulting in reduced delays. As V_n approaches the supply voltage, V_{DD}, the delay of the VCDE starts to saturate reaching its minimum value determined by the combined sizes of the static inverters and the current-starved branches.

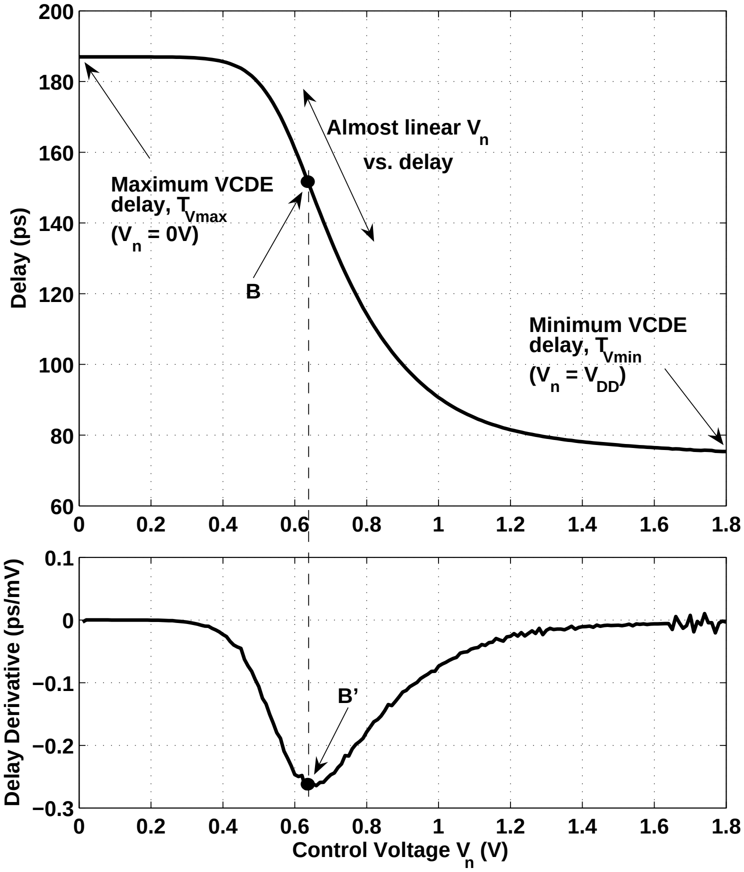


Figure 3.13: Voltage-delay characteristics of the VCDE.

The characteristics in Fig. 3.13 show that there is a region where the VCDE exhibits an almost-linear V_n versus delay behavior. This is further illustrated by the delay derivative (slope) characteristics shown in Fig. 3.13. For “perfectly” linear voltage-delay characteristics, the slope needs to be constant. At point B’ in the derivative characteristics, the slope of the delay derivative changes it polarity. This means that around this point there is a minimal change in the delay slope with changing V_n. As a result, if the values of the control voltage, V_n, are confined to the region around this inflection point, the operation of the VCDE will be limited to this almost-linear region of the characteristics. This is achieved using the DVC of the proposed DCDE. The existing non-linearity can be corrected by proper sizing of the control transistors in the DVC as will be illustrated in the next section.

The DVC converts a digital code to the value of V_n that produces the desired delay. As shown in Fig. 3.12, the DVC block consists of a diode-connected NMOS transistor, M_{c0}, and $n + 1$ load PMOS transistors, M_{d0}-M_{dn}, giving n delay steps for the DCDE. The load transistors control the current flowing in M_{c0} which is then converted to the control voltage V_n. This arrangement is similar to that used in [15]. The main difference is that the DVC of the proposed DCDE uses thermometer rather than binary coding. Consequently, the PMOS load transistors are not sized in a binary-weighted fashion. Instead, the sizing is done in order to prevent the negative feedback effect of increasing V_n and to ensure delay step uniformity.

The following section discusses design tradeoffs that need to be taken into account when designing the proposed DCDE. This is done through a design example illustrating the design of both the VCDE and the DVC.

3.4 Designing the Proposed DCDE

The example VCDE characteristics shown in Figure 3.14 are used to illustrate some of the design parameters related to the design of the proposed DCDE. In this discussion, T_V refers to the delays of the VCDE and T_D refers to the delays of the DCDE. Additionally, the “size” of a transistor refers to its width/length (W/L) ratio. Hence, making a transistor “smaller” means either reducing W or increasing L.

Because PVT variations cause the delays achieved by the DCDE to change with changes in the delay characteristics of the VCDE, it is important that the DCDE is designed such that the VCDE has a high degree of linearity and that the DVC ensures operation in the almost-linear region of the VCDE delay characteristics. The operating region is determined by the

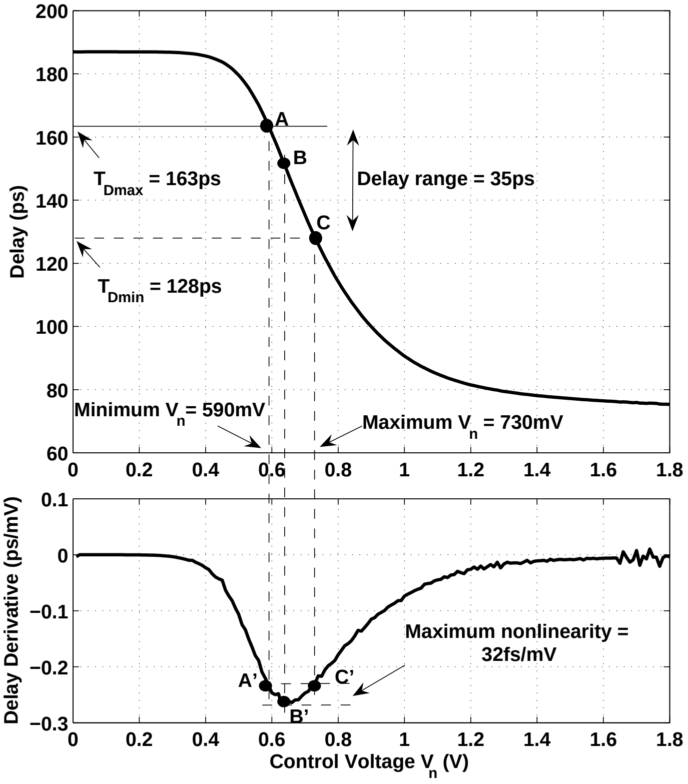


Figure 3.14: VCDE design example.

minimum and maximum V_n generated from the DVC which correspond to the maximum and minimum DCDE delays, T_{Dmax} and T_{Dmin}. These voltage values (given by points A and B in Figure 3.14) are chosen such that they give minimum slope deviation with respect to the slope of the voltage-delay characteristics at the inflection point B. Some important observations regarding the example shown in Fig. 3.14 and the design of the VCDE are given below.

1. Points A and C give the voltage and delay range of the DCDE. These two points should be chosen such that the corresponding delays give the required delay range and that the delay slopes at these two points are equal. This ensures minimum slope deviation from the slope

at the inflection point and gives best linearity for a given VCDE characteristics and DCDE delay range. In this example, A and C are chosen such that the DCDE would have a delay range of 35ps.

2. The maximum non-linearity that the DCDE would exhibit is determined by the difference in the delay characteristics' slopes between point B and either of points A or C. For the example shown in Figure 3.14, this is equal to 32fs/mV. This difference can be used to compare the non-linearity of different VCDE designs for a given delay range. Another way to perform this comparison is to find the delay range corresponding to a fixed degree of non-linearity. The larger that range, the better the linearity of the VCDE would be.

3. As will be explained in section 3.4.2, the DVC is used to linearize the DCDE delay characteristics. As a result, the actual DCDE characteristics would be a straight line between points A and C with a slope that can be estimated to be roughly equal to the slope of the delay characteristics at point B. This can be used to estimate the average V_n step corresponding to the required delay step. For the example shown in Figure 3.14, the slope at point B is equal to 0.268ps/mV. For a delay step of 5ps, this slope corresponds to an average voltage step of 18.66mV. Points A and C are chosen such that the DCDE would have a delay range of 35ps. The corresponding voltage range is 140mV, giving a DCDE delay characteristics slope of 0.25ps/mV, a value close to the estimate given by the slope at B.

4. For a given delay range, the larger the slope at B, the smaller the average voltage step would be. This results in larger sensitivity of the DCDE to variations in V_n. On the other hand, the smaller the slope at B, the larger the voltage step and consequently the V_n range would be. This requires the DVC to produce larger values of V_n and increases the static current flowing in the DVC resulting in larger overall power consumption of the DCDE.

3.4.1 VCDE Design

There are many trade-offs that need to be taken into account when designing the VCDE used in the proposed DCDE. These trade-offs affect the delay range of the DCDE as well as the linearity of the delay characteristics within that range. They also affect area and power consumption of the DCDE. These design trade-offs and their impact on designing the VCDE are summarized in the following.

1. The size of the NMOS transistor in the static inverter, M_2, determines the maximum delay of the VCDE, $T_{V_{max}}$. The minimum VCDE delay, $T_{V_{min}}$, is determined by the sizes of this transistor as well as the transistors in the current-starved NMOS block. The difference between these two delays should be chosen in order to achieve a reasonable delay range for the DCDE with high linearity.

2. The smaller the size of transistor M_{c1}, the better the linearity of the VCDE would be. This is illustrated in Fig. 3.15 showing the voltage-delay and delay-derivative characteristics of the VCDE for four different sizes of M_{c1} and an M_3 size of 15 times the minimum size. The figure also shows the delay ranges (given by points A and C) corresponding to a non-linearity of 20fs/mV in the delay characteristics for all four sizes of M_{c1}. As shown in the figure, smaller sizes of M_{c1} result in larger delay ranges for the same amount of non-linearity, i.e better linearity of the delay characteristics. The figure also shows that with smaller M_{c1} the inflection point happens at larger values of V_n. This means that the improved linearity due to smaller M_{c1} comes at the expense of increased power consumption of the DCDE as the DVC needs to create larger values of V_n. It should be noted that the sizing of M_{c1} has a minimal impact on the power consumption of the VCDE. This is because, although larger M_{c1} causes larger currents to flow in the VCDE, these currents flow for a smaller period of time due to smaller delays. It should also be noted

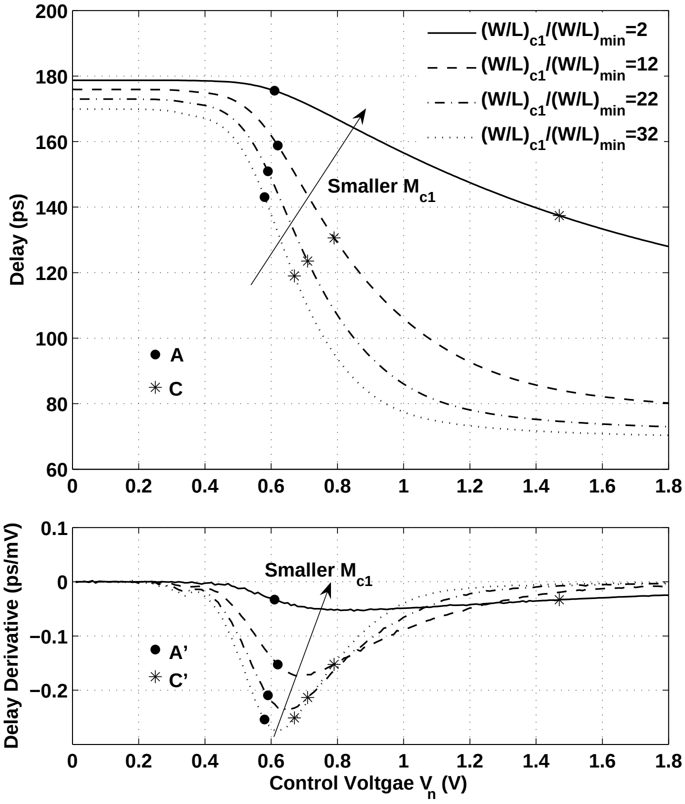


Figure 3.15: Impact of the size of M_{c1} on the voltage-delay characteristics of the VCDE for $(W/L)_3/(W/L)_{min} = 15$.

that smaller sizes of M_{c1} result in larger absolute delays of the DCDE, and consequently larger delay uncertainty due to process variations.

3. For a fixed size of transistor M_{c1}, making transistor M_3 larger improves the linearity of the voltage-delay characteristics of the VCDE. This is illustrated in Figure 3.16 showing the VCDE characteristics as well as the delay ranges corresponding to a non-linearity of 20fs/mV for four different sizes of M_3 and an M_{c1} size of 15 times the minimum (W/L). As shown in the figure, these delay ranges increase with increasing the size of M_3. The improvement in linearity becomes minimal when the size of M_3 exceeds a critical size,

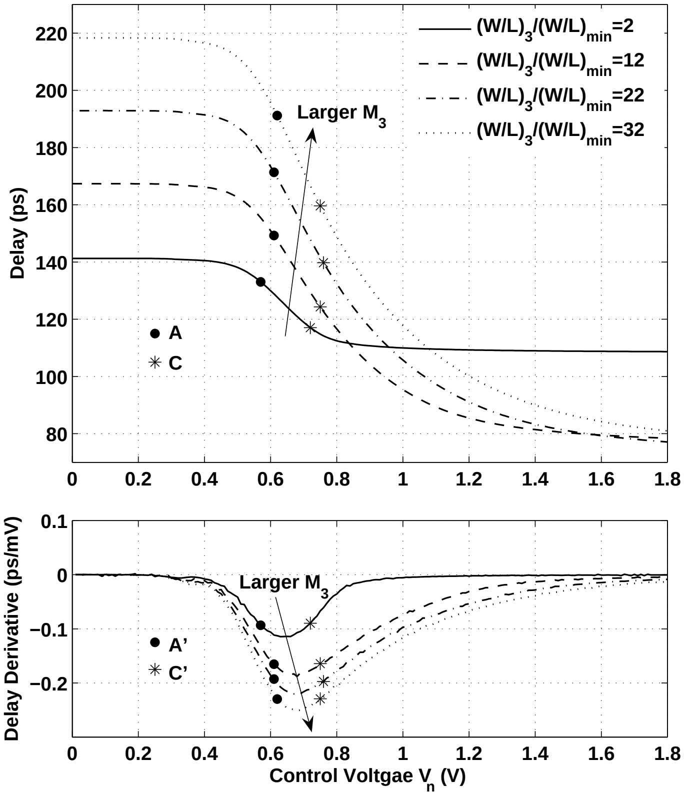


Figure 3.16: Impact of the size of M_3 on the voltage-delay characteristics of the VCDE for $(W/L)_{c1}/(W/L)_{min} = 15$.

$(W/L)_{3_{critical}}$. For this critical size, M_3 becomes large enough so that the current of the current-starved branch is controlled by M_{c1} for all values of the control voltage V_n. It should be noted that the improvement in linearity with larger M_3 comes at the expense of smaller average V_n step. This is because of the larger slope of the VCDE characteristics at the inflection point. More importantly, larger sizes of M_3 cause a proportional increase in the power consumption of the VCDE. Simulations show that sizing M_3 at the critical (W/L) provides a good trade-off between linearity, V_n step, and power consumption.

4. If the size of transistor M_3 is much smaller than that of M_{c1}, M_3 controls the discharge

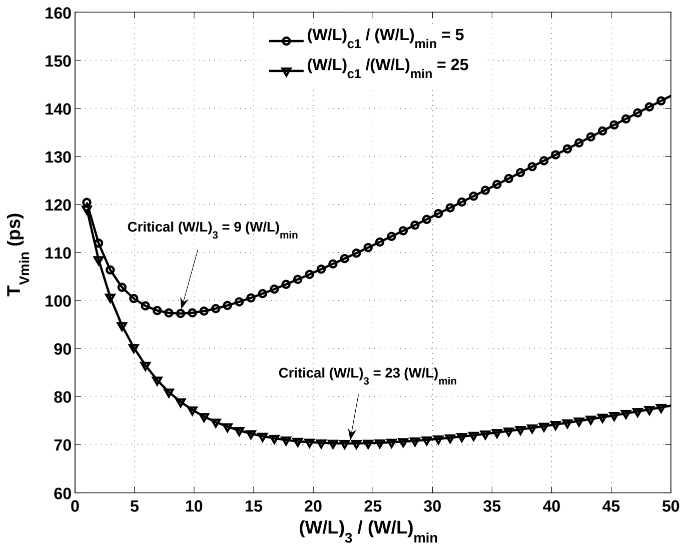


Figure 3.17: Impact of the relative size of M_{c1} and M_3 on the minimum delay of the VCDE, $T_{V_{min}}$ (V_n=V_{DD}).

current in the current-starved block resulting in larger $T_{V_{min}}$. This is illustrated in Fig. 3.17 showing the impact of the size of M_3 on $T_{V_{min}}$ for two different sizes of M_{c1}. As M_3 becomes larger, $T_{V_{min}}$ decreases until it reaches a minimum when the size of M_3 reaches $(W/L)_{3_{critical}}$. At this critical value, M_{c1} starts to control the discharge current of the current-starved NMOS block. When the size of M_3 is increased beyond this critical value, $T_{V_{min}}$ starts to increase due to the increased capacitive loading at node X.

5. Due to the body effect on transistor M_3, it is expected that $(W/L)_{3_{critical}}$ would always be larger than $(W/L)_{c1}$. However, this is only true a long as the increase in delay due to the capacitive loading of larger M_3 does not out-weigh the reduction in its resistance. If that is not the case, $(W/L)_{3_{critical}}$ can be equal or even smaller than $(W/L)_{c1}$. Fig. 3.17 shows that when $(W/L)_{c1}$ is only 5 times the minimum W/L, the critical size of M_3 is 9 times as much. On the other hand, when $(W/L)_{c1}$ is 25 times the minimum W/L, the critical size for M_3 is

actually smaller than the size of M_{c1}.

Taking these trade-offs into account, the VCDE can be designed in order to achieve the desired delay range with the desired linearity, while achieving acceptable area and power consumption. Following the design of the VCDE, the DVC is designed in order convert the digital input codes to control voltage levels that produce the desired DCDE characteristics.

3.4.2 DVC Design

Designing the DVC requires knowledge of the minimum and maximum control voltage values that produce the maximum and minimum delays of the DCDE. The DVC is then designed so that the DCDE would have the required number of linear delay steps between these two delay limits.

Ideally, the diode-connected transistor, M_{c0}, should be identical in size to M_{c1} in order to provide good matching between these two transistors. However, if the power consumption of the DVC (due to static current flowing in M_{c0}) is a concern, M_{c0} can be made smaller in order to reduce this current. This would also require scaling the load transistors M_{d0}-M_{dn} accordingly, which might cause an increase in the area of these transistors (due to larger L) if very small delay steps are needed.

The gate of transistor M_{d0} is connected to ground ensuring a minimum amount of current which also corresponds to the required minimum V_n and maximum DCDE delay, T_{Dmax}. All the other load transistors are controlled by a thermometer code digital word. Using thermometer code ensures a monotonic and linear code versus delay characteristics of the DCDE. If binary code is used and assuming that M_{d1} is sized so that, when ON, it would add a certain amount of current, I, and that M_{d2} would add 2I, the current added when both transistors are ON would be less than 3I. This is attributed to the fact that the PMOS load transistors are operating in the linear region. Hence, larger current results in larger V_n (i.e. smaller V_{DS} for these transistors)

M_1, M_4	M_2, M_5	M_3, M_6	M_{c1}, M_{c2}
2/0.18	0.5/0.18	4.75/0.18	5/0.18

(a)

M_{c0}	M_{d0}	M_{d1}	M_{d2}	M_{d3}	M_{d4}	M_{d5}	M_{d6}	M_{d7}
5/0.18	5.455/2.2	1.455/2.2	1.575/2.2	1.73/2.2	1.895/2.2	2.085/2.2	2.33/2.2	2.605/2.2

(b)

Table 3.2: Sizes of the transistors in the (a) VCDE and (b) DVC used to design a DCDE with seven delay steps of 5ps each.

which in turn reduces the actual current added by these devices. Using thermometer code, on the other hand, ensures that when an additional transistor in turned ON, it can be sized in order to compensate for the reduction in the current produced by the already-ON transistors and to add the required additional current. As a result, transistors M_{d1}-M_{dn} are sized such that each transistor is slightly larger than the one to its right. The sizing is done such that when an additional transistor is turned ON, it provides the required reduction in the delay of the VCDE. This allows the sizing of these transistors be used to linearize the code-delay characteristics in the operating region of the VCDE. The smaller the W/L of the PMOS load, the smaller the delay step would be. As there is no limit on how small the W/L of a transistor can be, the proposed design of the DCDE has no limit on how small the delay step could be.

3.5 Simulation Results

The proposed DCDE has been designed in 0.18μm CMOS technology such that it would have seven equal delay steps of 5ps for the typical process and supply voltage and a temperature of 27^oC. This gives a total delay range of 35ps. The characteristics of the VCDE used in this example design are the ones given in Figure 3.14 with T_{Dmax} and T_{Dmin} chosen to be 163 and 128ps. The sizes of the transistors used in this design are given in Table 3.2

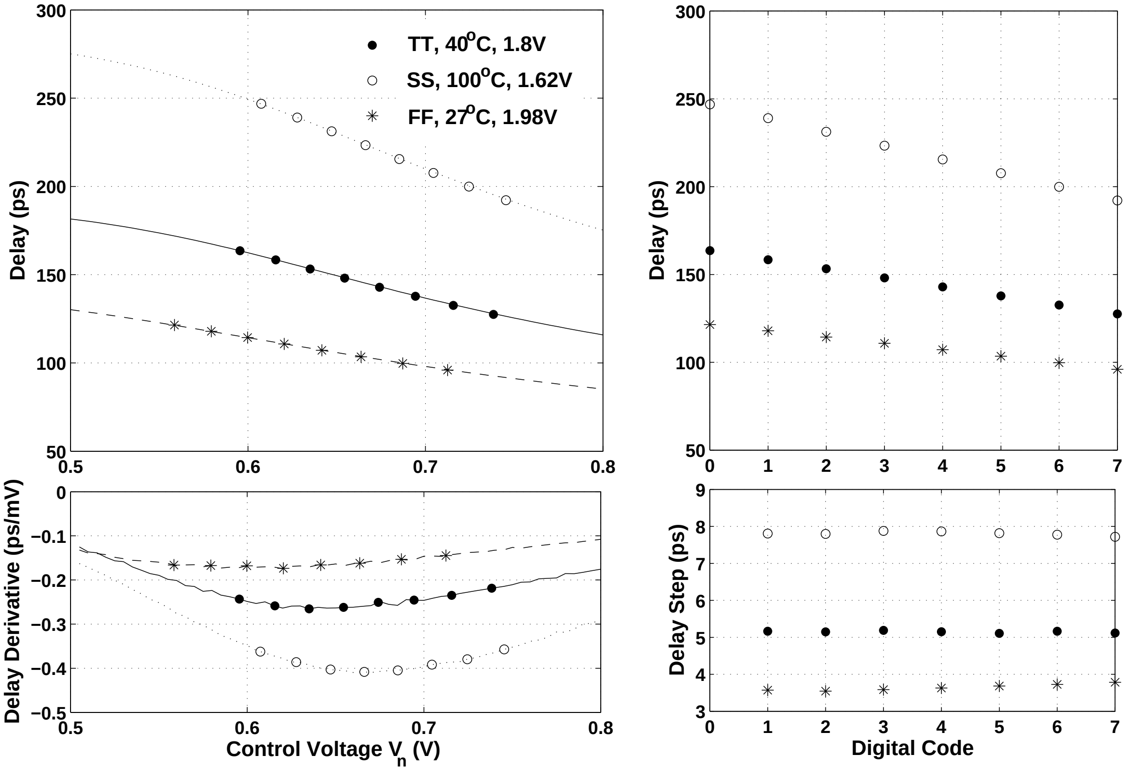


Figure 3.18: Delay characteristics of the proposed DCDE.

Simulation results for this DCDE are presented in Figure 3.18. The leftmost half of the figure shows the eight DCDE operating points superimposed on the VCDE delay and delay-derivative characteristics. The curves show simulation results for three operating conditions; *typical* (TT process, 40^oC, and 1.8V VDD), *slow* (SS process, 100^oC, and 1.62V VDD), and *fast* (FF process, 27^oC, and 1.98V VDD). As shown by the delay derivative characteristics, even with extreme PVT conditions, the DCDE operating points fall in the area around the inflection point. This means that the DCDE would maintain the linearity of its delay characteristics with these variations. This is illustrated in the rightmost half of Figure 3.18, showing the code-delay and delay step characteristics of the DCDE.

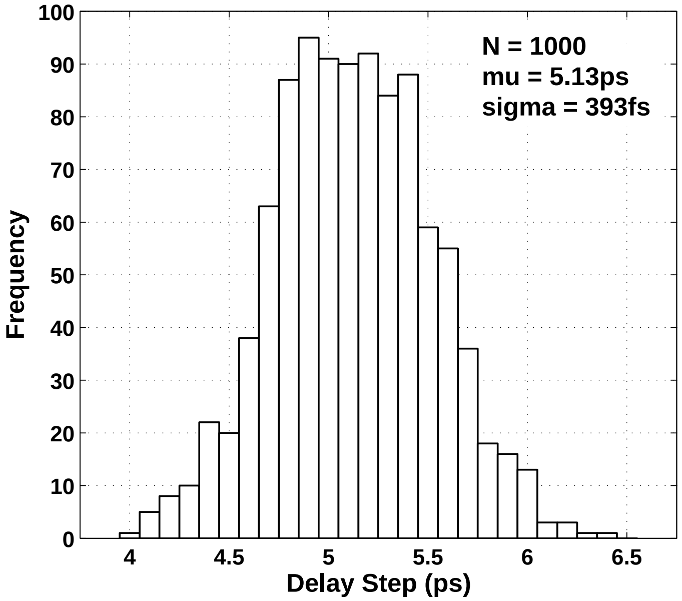


Figure 3.19: Monte Carlo analysis of the impact of process variation on the DCDE delay step.

For the *typical* condition, the DCDE has delay steps ranging from 5.1 to 5.19ps with an average of 5.15ps. With PVT variations, the average delay step can be as high as 7.8ps for the *slow* condition and as low as 3.65ps for the *fast* condition. These variations are consistent with the variations in the absolute delays of the DCDE. As shown in Figure 3.18, for the *slow* condition, the absolute delays range from a T_{Dmax} of 246.8ps to a T_{Dmin} of 192.1ps while for the *fast* condition, these delay values are 121.5 and 96ps, respectively. Monte Carlo analysis of the DCDE shows that with process variations the delay step has a standard deviation of 393fs. These results are shown in Figure 3.19 for 1000 runs of Monte Carlo analysis.

In order to estimate the DCDE jitter due to device noise, jitter simulations have been performed using the methodology described in [18]. In order to obtain the worst case estimate of jitter, these simulations were performed with the DCDE input code set to achieve the largest DCDE delay. Based on these simulations, the rms value of the device contribution to jitter was

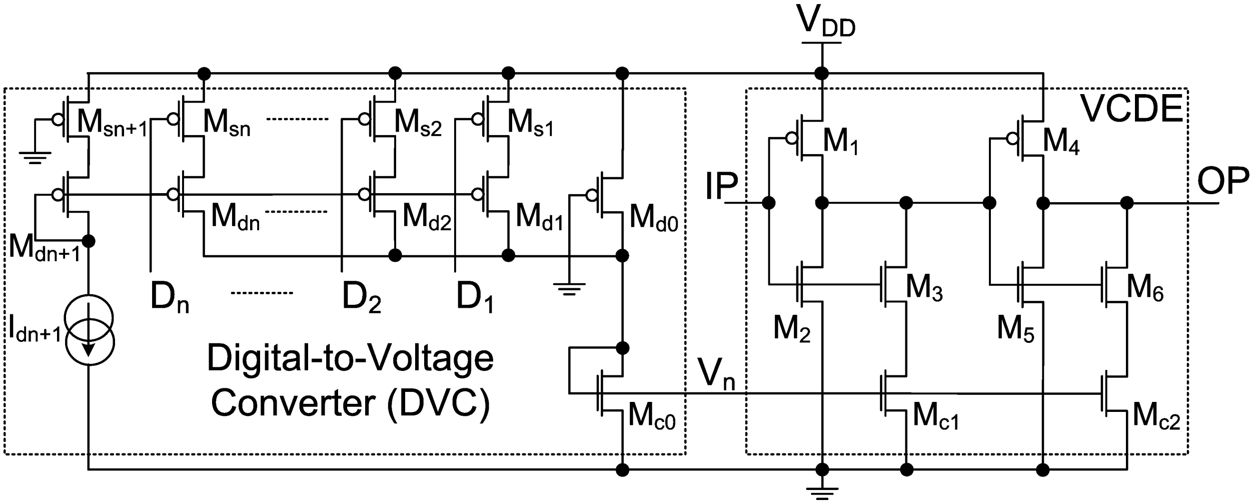


Figure 3.20: Schematic diagram of the controlled-current DCDE.

found to be 0.138ps. In reality, it is expected that the DCDE jitter would be higher than this value due to supply noise and other sources of timing uncertainties.

Although the DCDE maintains the linearity of its delay characteristics with PVT variations, the delay step changes in the same fashion in which the absolute delays change with such variations. These changes in the step size can be undesirable for some applications that require the DCDE to have a fixed step regardless of PVT variations. The DCDE design can be modified in order to provide the ability to adjust the delay step based on the PVT condition. A possible implementation of the modified DCDE is illustrated in Figure 3.20. This controlled-current DCDE (CC-DCDE) is based on adjusting the currents through the PMOS transistors in the DVC in order to achieve the desired step size. The design uses the same VCDE as the original DCDE. It also uses the same M_{d0} and M_{c0} transistors in the DVC. This ensures that $T_{D_{max}}$ (given by point A in Figure 3.14) stays the same as the original DCDE design. Each of the remaining PMOS transistors in the DVC is replaced by two PMOS transistors, one transistor acting as a current mirror (M_{di}) and the other acting as a switch (M_{si}), with i ranging from 1 to n. The current mirrors

M_{d1}-M_{dn} are controlled by a diode-connected PMOS transistor, M_{dn+1}. The relative sizes of these transistors are determined in the same fashion used for sizing the PMOS load transistors in the original DCDE design. The size of M_{dn+1} is chosen to be equal to that of M_{dn}. The sizes of the PMOS switches, M_{s1}-M_{sn+1}, are chosen such that they are larger than the current mirrors. This is important so that these transistors would not limit the current through the different branches. These transistors are controlled by the digital code of the DCDE. When the input to any of these switches is LOW, the current flowing in that branch would be equal to the current in M_{dn+1} (I_{dn+1}) scaled by the ratio of the sizes of the current mirror of that branch and the diode-connected transistor, M_{dn+1}. By adjusting the value of I_{dn+1}, the delay step of the CC-DCDE can be adjusted accordingly.

These changes have been implemented in the DCDE described earlier resulting in a CC-DCDE with seven delay steps that can be adjusted to achieve an average delay step of 5ps under different PVT conditions. Figure 3.21 shows the code-delay and delay step characteristics of the CC-DCDE under typical and extreme PVT conditions. For these simulations, the current I_8 is adjusted to achieve the required average delay step. Comparing Figures 3.18 and 3.21, it is clear that the delays for code “0” are the same for both designs. All other delays of the CC-DCDE are adjusted to achieve the desired step. From the comparison, it is also clear that the adjustment in the delay step comes at the expense of the linearity of the code-delay characteristics. The loss of linearity is attributed to the shifting of the delay operating region with respect to the inflection point. For the CC-DCDE, the delay step ranges from 4.4ps to 5.7ps for the *slow* condition and from 4.5ps to 5.55ps for the *fast* condition. This means that using this CC-DCDE, the individual delay steps can be adjusted to within 14% of the desired average step of 5ps.

It is possible to use the CC-DCDE to achieve programmable delay steps. However, code-delay characteristics would exhibit high linearity only for the intended delay step of the design. This is illustrated by the simulation results in Figure 3.22. The figure shows the code-delay

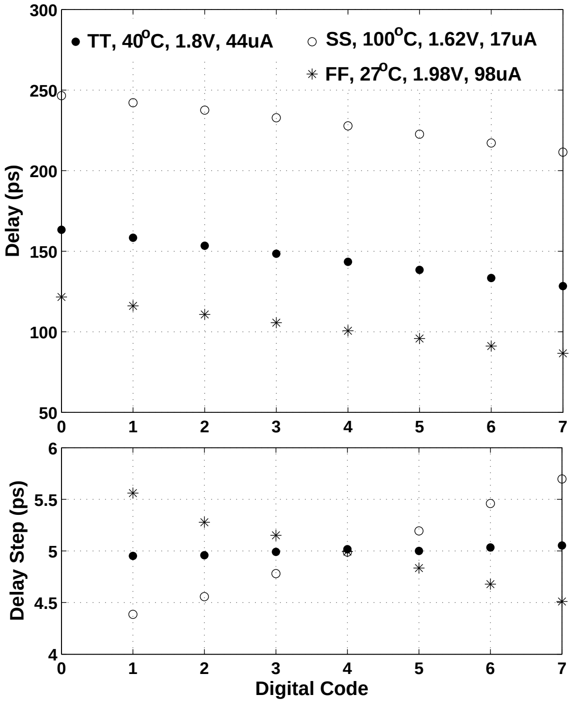


Figure 3.21: Simulation results of the CC-DCDE under typical and extreme PVT conditions with I_8 adjusted for an average delay step of 5ps.

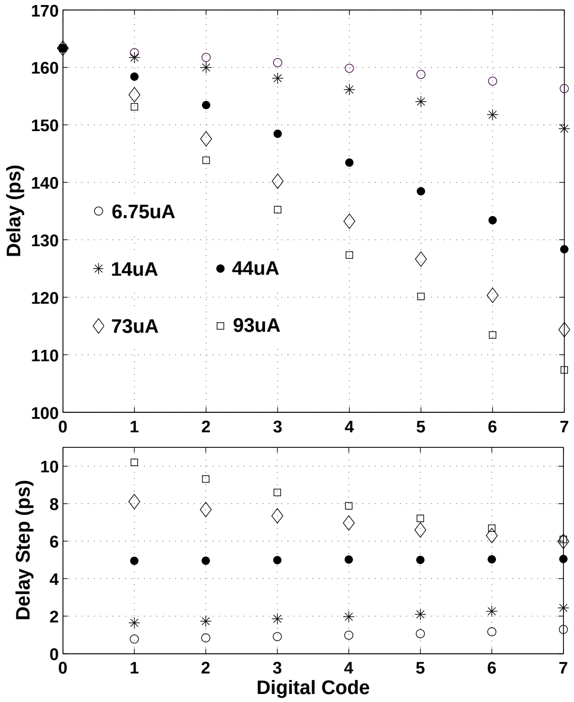


Figure 3.22: Simulation results of the CC-DCDE under typical PVT conditions with five values of I_8 giving average delay steps of 1, 2, 5, 7, and 8ps.

characteristics of the CC-DCDE under *typical* conditions for five different values of I_8. These values are chosen to achieve average delay steps of 1, 2, 5, 7, and 8ps. As shown by the delay step characteristics, the further away from the target (5ps) the delay step is, the more non-linear the code-delay characteristics would be. For an average step of 2ps, the maximum deviation is 22%. This value increases to 30% for an average step of 1ps. Similarly, for an average delay step of 7ps, the maximum deviation is 16% which increases to 27.5% when trying to achieve an average step of 8ps. It should be noted that it is possible to correct this non-linearity by shifting the delay characteristics back around the inflection point. This can be achieved by separately adjusting the current through M_{d0}.

3.6 Experimental Results

The proposed DCDE and CC-DCDE have been implemented on a test chip in 0.18μm CMOS technology. The chip micrograph is shown in Figure 3.23. The chip includes 6 delay elements, DE0-DE5, connected in a delay chain whose input is supplied through an input buffer (IP). The input to IP is supplied from an off-chip digital signal source. DE0 is a VCDE used to condition the input to the subsequent DCDEs. DE1 and DE2 are two DCDEs each designed with 7 delay steps of 5ps. DE3 and DE4 are the corresponding CC-DCDEs. DE5 is a VCDE used to obtain the voltage delay characteristics of the VCDE used in the two versions of the DCDE. The outputs from the six delay elements are buffered and fed to a 6-1 multiplexer (MUX). One of the six outputs is selected by the MUX to feed the output buffer (OP). The select inputs of the MUX as well as the digital codes of the four DCDEs are stored in a control shift register (SR) which accepts input data serially. The VCDE occupies an area of 130μm^2 divided between its two stages. The DVC's area is 250μm^2 for the DCDE and 440μm^2 for the CC-DCDE.

The inputs of the control SR are supplied from an Agilent VXI test system which also supplies the input signal of the delay chain. The output of the output buffer is monitored using an Infiniium 54853A digital storage oscilloscope. In order to obtain accurate delay measurements, it is important to eliminate the jitter of the output waveform. This is done by averaging the output signal over a larger number of measurements. The testing is done by storing the required test vector using the control SR and observing the output signal on the scope. In order to compare the delays of one of the DCDEs for different codes, the output for a given code is stored using the waveform memory function of the scope. The code is then modified and the new output is compared to the stored waveform.

In order to obtain the voltage-delay characteristics of the VCDE, the control voltage V_n of DE5 is adjusted using an off-chip voltage source and the corresponding delays are recorded. These results are shown in Figure 3.24 along with the simulated voltage-delay characteristics of

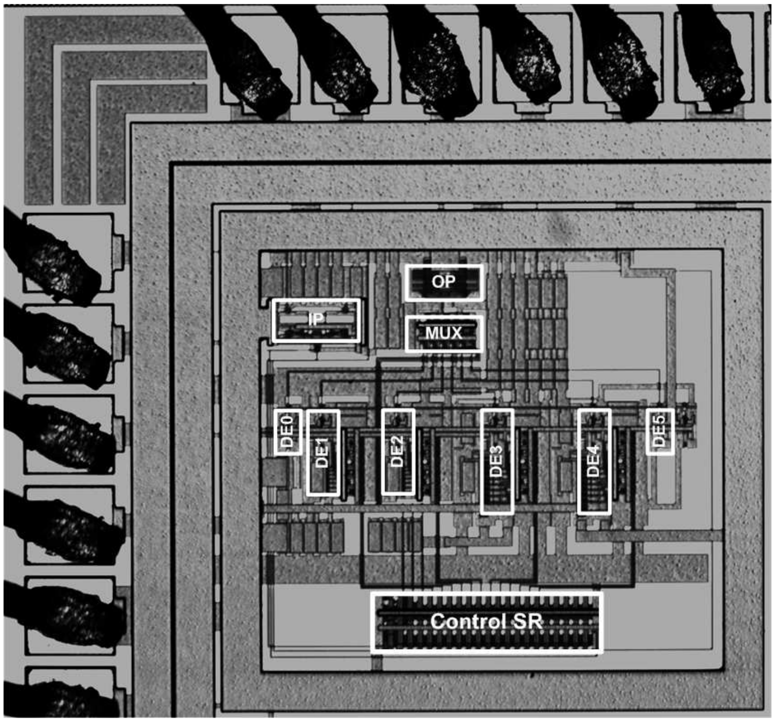


Figure 3.23: DCDE and CC-DCDE chip micrograph.

the VCDE for *typical* PVT conditions. As shown in the figure, the measured delays are larger than those obtained from simulations. This indicates that the PVT conditions at which the chip is operating is on the slower side of the *typical* PVT conditions. As a result, it is expected that the average delay step of the DCDE would be larger than the intended 5ps. This is illustrated by the DCDE (DE2 on the chip) measurement results presented in Figure 3.25. Because the scope can store a maximum of four waveforms, these results are taken over two sets of measurements. Figure 3.25(a) presents the delay steps for codes 0 to 3 and Figure 3.25(b) presents the steps for codes 3 to 7. The relative delays of these waveforms were measured using the delay measurement function of the scope. As shown in the figure, the delay steps range from 5.15ps to 5.64ps with an average of 5.39ps. The maximum deviation from this average is 4.6% which can be attributed to mismatch between the PMOS control transistors as well as measurement uncertainty.

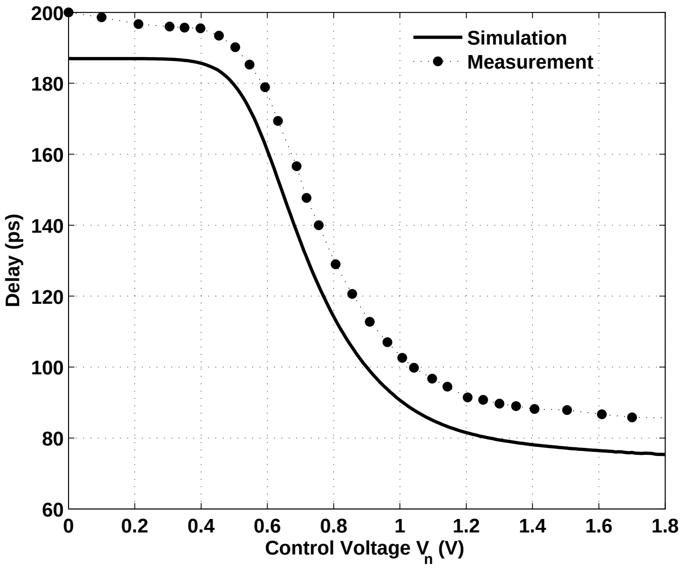


Figure 3.24: Simulations vs. measurement results of the voltage-delay characteristics of the VCDE (DE5 on the test chip).

Jitter measurement for the DCDE was performed by measuring jitter at the input and output of DE2 when the input codes are set to give maximum DCDE delays. The rms jitter at these two nodes was measured to be 5.33ps and 5.42ps respectively. From these results, the rms jitter contribution of DE2 is estimated to be 0.98ps. This number is much larger than that estimated from device noise simulations (0.138ps). The larger jitter contribution can be attributed to supply noise and coupling between signal lines. The output driver used in the test chip shares the same supply and ground pins with the rest of the chip. This leads to larger supply noise and larger jitter. Additionally, the six inputs to the MUX are routed using parallel lines resulting in crosstalk between these signals. Jitter on one signal results in variation in coupling with adjacent signals and results in increased timing uncertainty.

Experimental results for the CC-DCDE were obtained using DE3 on the test chip. The first set of results were obtained by finding the value of I_8 that produces an average delay step of 5ps for the CC-DCDE. As shown in Figure 3.21, simulations predict this value to be 44μA for the *typical* PVT conditions. Because the chip is operating at slower PVT conditions, it is expected that the corresponding value of I_8 would be smaller than what is predicted by simulations. Through measurements, this value was found to be 41μA. Figure 3.26 shows the output waveforms of DE3 for all values of the input code as well as the relative delays of these waveforms as measured by the scope. Figure 3.27 shows experimental results for the CC-DCDE for two additional values of I_8. For $I_8 = 25\mu$A, the average delay step of the CC-DCDE is 3ps, while $I_8 = 5\mu$A gives an average delay step of less than 1ps.

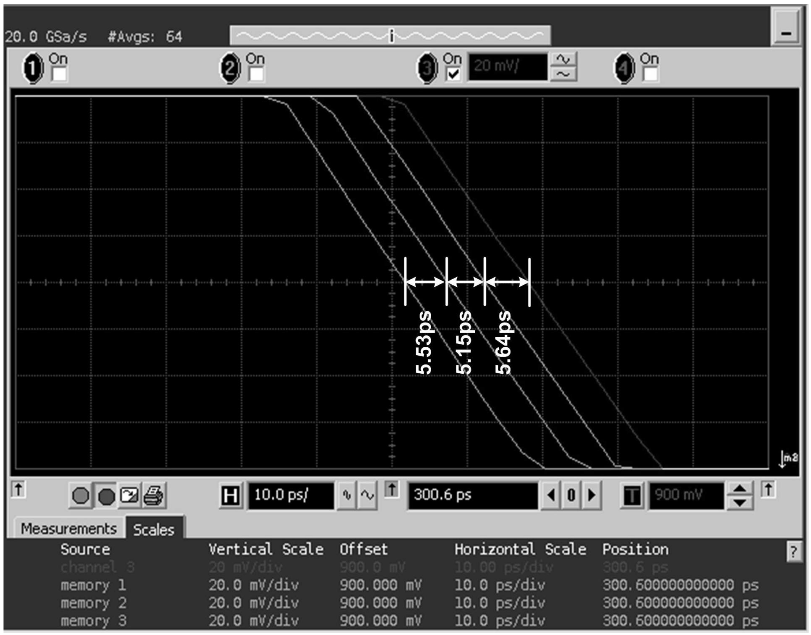
These results shows that the proposed DCDE can be used to achieve small linear delay steps which makes it attractive for many applications. The following chapter demonstrates using this DCDE in designing a high-precision phase aligner for application in clock deskewing.

3.7 Conclusions

In this chapter, a DCDE based on current-starved inverters has been proposed. The design uses a VCDE that has an almost-linear region in its voltage-delay characteristics. By confining the operation of the VCDE to this region and by proper design of the DVC, excellent step uniformity can be achieved even with PVT variations. In addition to the transistors used in the VCDE and two transistors in the DVC, the design requires one transistor per delay step. Using current-starved inverters for delay control allows the design to be easily modified in order to have independent control of rising and falling edges. It also allows the control transistors to be shared between different DCDEs which can help reduce area and power consumption. Using this DCDE, an average delay step of 5.39ps has been demonstrated experimentally in 0.18μm technology.

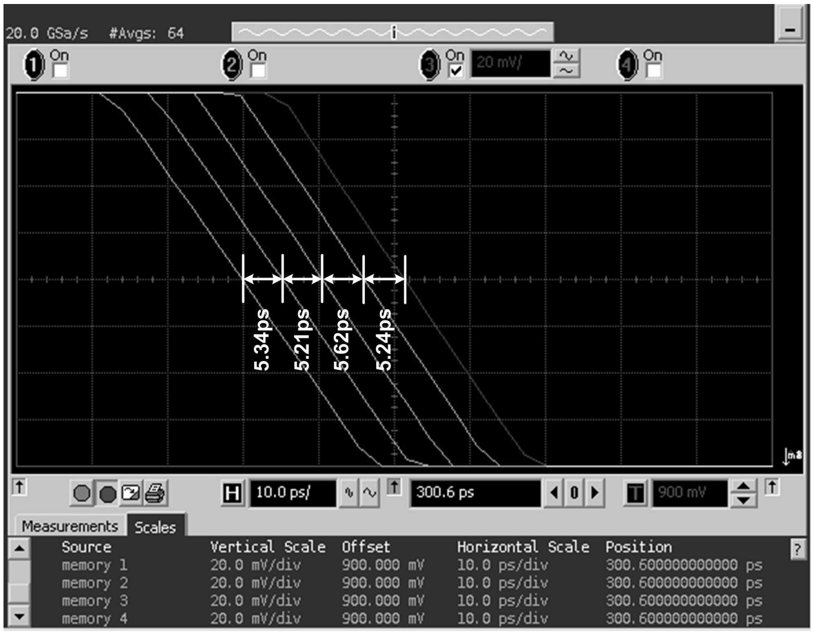
By adjusting the currents in the DVC, the CC-DCDE can be used in order to adjust the average delay step of the proposed DCDE. This comes at the expense of step uniformity. It should be noted, however, that the deviation from the average step is much smaller than the non-uniformity of earlier designs. Using the CC-DCDE, sub-1ps step has been demonstrated experimentally.

Simulations and experimental results demonstrate that the proposed DCDE is capable of achieving precise, monotonic, and linear delay characteristics which makes it attractive in many applications. Some of these applications include all-digital DLLs, DCOs used in all-digital PLLs, timing generators for ATEs, and phase aligners used for deskewing applications.



```
Source        Vertical Scale  Offset        Horizontal Scale  Position
channel 3     20 mV/div       900.0 mV      10.00 ps/div      300.6 ps
memory 1      20.0 mV/div     900.000 mV    10.0 ps/div       300.600000000000 ps
memory 2      20.0 mV/div     900.000 mV    10.0 ps/div       300.600000000000 ps
memory 3      20.0 mV/div     900.000 mV    10.0 ps/div       300.600000000000 ps
```

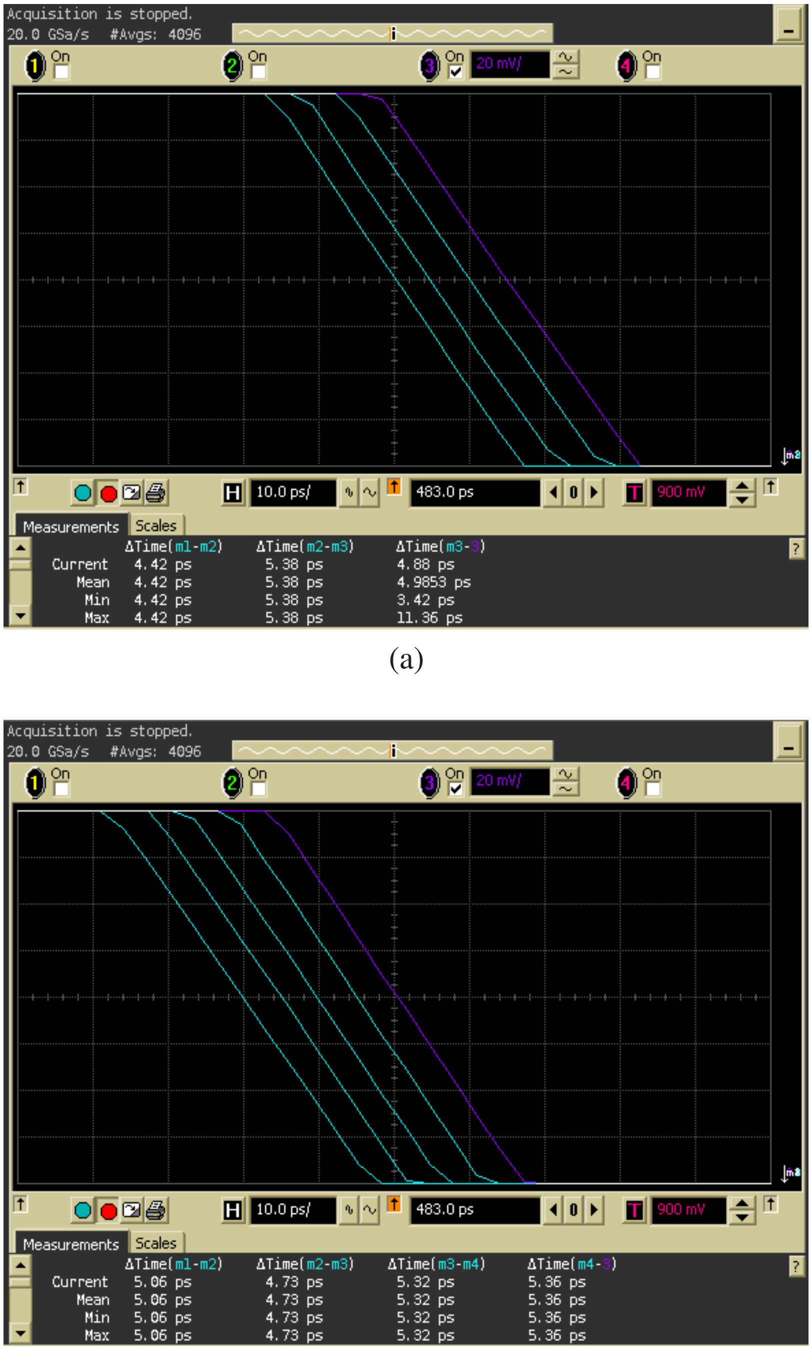
(a)



```
Source        Vertical Scale  Offset        Horizontal Scale  Position
memory 1      20.0 mV/div     900.000 mV    10.0 ps/div       300.600000000000 ps
memory 2      20.0 mV/div     900.000 mV    10.0 ps/div       300.600000000000 ps
memory 3      20.0 mV/div     900.000 mV    10.0 ps/div       300.600000000000 ps
memory 4      20.0 mV/div     900.000 mV    10.0 ps/div       300.600000000000 ps
```

(b)

Figure 3.25: Measurement results of the DCDE (DE2 on the test chip) for (a) codes 0 to 3 and (b) codes 3 to 7.



```
Acquisition is stopped.
20.0 GSa/s   #Avgs: 4096
```

```
           ΔTime(m1-m2)   ΔTime(m2-m3)   ΔTime(m3-3)
 Current    4.42 ps        5.38 ps        4.88 ps
    Mean    4.42 ps        5.38 ps        4.9853 ps
     Min    4.42 ps        5.38 ps        3.42 ps
     Max    4.42 ps        5.38 ps        11.36 ps
```

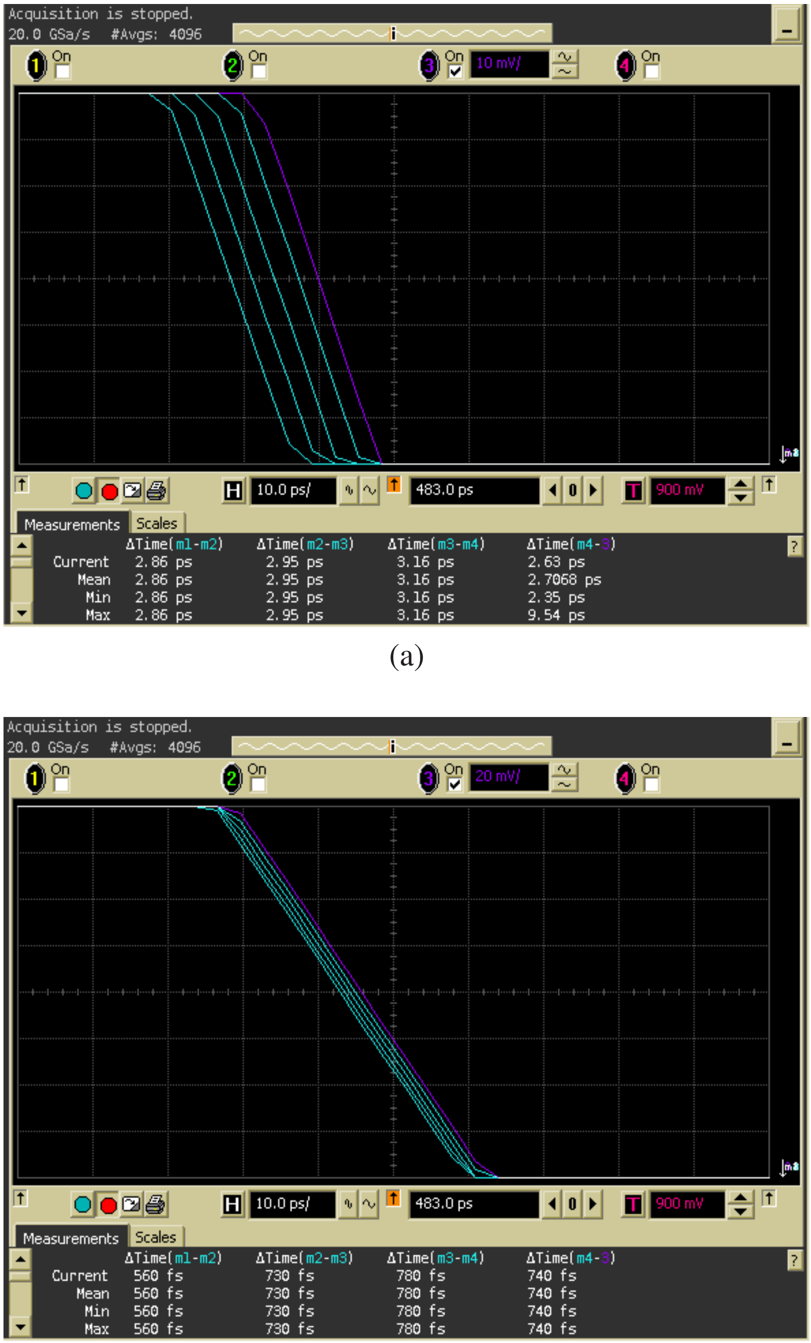
(a)

```
Acquisition is stopped.
20.0 GSa/s   #Avgs: 4096
```

```
           ΔTime(m1-m2)   ΔTime(m2-m3)   ΔTime(m3-m4)   ΔTime(m4-3)
 Current    5.06 ps        4.73 ps        5.32 ps        5.36 ps
    Mean    5.06 ps        4.73 ps        5.32 ps        5.36 ps
     Min    5.06 ps        4.73 ps        5.32 ps        5.36 ps
     Max    5.06 ps        4.73 ps        5.32 ps        5.36 ps
```

(b)

Figure 3.26: Measurement results of the CC-DCDE (DE3 on the test chip) for $I_8 = 41\mu A$ for (a) codes 0 to 3 and (b) codes 3 to 7.



```
            ΔTime(m1-m2)   ΔTime(m2-m3)   ΔTime(m3-m4)   ΔTime(m4-3)
  Current   2.86 ps        2.95 ps        3.16 ps        2.63 ps
     Mean   2.86 ps        2.95 ps        3.16 ps        2.7068 ps
      Min   2.86 ps        2.95 ps        3.16 ps        2.35 ps
      Max   2.86 ps        2.95 ps        3.16 ps        9.54 ps
```

(a)

```
            ΔTime(m1-m2)   ΔTime(m2-m3)   ΔTime(m3-m4)   ΔTime(m4-3)
  Current   560 fs         730 fs         780 fs         740 fs
     Mean   560 fs         730 fs         780 fs         740 fs
      Min   560 fs         730 fs         780 fs         740 fs
      Max   560 fs         730 fs         780 fs         740 fs
```

(b)

Figure 3.27: Measurement results of the CC-DCDE (DE3 on the test chip) for codes 3-7 for (a) $I_8 = 25\mu A$ and (b) $I_8 = 5\mu A$.

References

[1] B. Arkin. Realizing a Production ATE Custom Processor and Timing IC Containing 400 Independent Low-Power and High-Linearity Timing Verniers. In *International Solid-State Circuits Conference Digest of Technical Papers*, pages 348–349, February 2004.

[2] P.L. Chen, C.C. Chung, and C.Y. Lee. A Portable Digitally Controlled Oscillator Using Novel Varactors. *IEEE Transactions on Circuits and System–II: Express Briefs*, 52(5):233–237, May 2005.

[3] P.L. Chen, C.C. Chung, J.N. Yang, and C.Y. Lee. A Clock Generator with Cascaded Dynamic Frequency Counting Loops for Wide Multiplication Range Applications. *IEEE Journal of Solid-State Circuits*, 41(6):1275–1285, June 2006.

[4] C.C. Chung and C.Y. Lee. An All-Digital Phase-Locked Loop for High-Speed Clock Generation. *IEEE Journal of Solid-State Circuits*, 38(2):347–351, February 2003.

[5] C.C. Chung and C.Y. Lee. A New DLL-Based Approach for All-Digital Multiphase Clock Generation. *IEEE Journal of Solid-State Circuits*, 39(3):469–475, March 2004.

[6] G.K. Dehng, J.M. Hsu, C.Y. Yang, and S.I. Liu. Clock-Deskew Buffer Using a SAR-Controlled Delay-Locked Loop. *IEEE Journal of Solid-State Circuits*, 35(8):1128–1136, August 2000.

[7] U. Desai, S. Tam, R. Kim, J. Zhang, and S. Rusu. Itanium Processor Clock Design. In *Proc. of International Symposium on Physical Design*, pages 94–98, April 2000.

[8] J. Dunning, G. Garcia, J. Lundberg, and E. Nuckolls. An All-Digital Phase-Locked Loop with 50-Cycle Lock Time Suitable for High-Performance Microprocessors. *IEEE Journal of Solid-State Circuits*, 30(4):412–422, April 1995.

[9] B.W. Garlepp et al. A Portable Digital DLL for High-Speed CMOS Interface Circuits. *IEEE Journal of Solid-State Circuits*, 34(5):632–644, May 1999.

[10] G. Geannopoulos and X. Dai. An Adaptive Digital Deskewing Circuit For Clock Distribution Networks. In *International Solid-State Circuits Conference Digest of Technical Papers*, pages 400–401, February 1998.

[11] A. Hatakeyama et al. A 256-Mb SDRAM Using a Register-Controlled Digital DLL. *IEEE Journal of Solid-State Circuits*, 32(11):1728–1734, November 1997.

[12] T.Y. Hsu, C.C. Wang, and C.Y. Lee. Design and Analysis of a Portable High-Speed Clock Generator. *IEEE Transactions on Circuits and System–II: Analog and Digital Signal Processing*, 48(4):367–375, April 2001.

[13] N.A. Kurd, J.S. Barkarullah, R.O. Dizon, T.D. Fletcher, and P.D. Madland. A Multigigahertz Clocking Scheme for the Pentium Microprocessor. *IEEE Journal of Solid-State Circuits*, 36(11):1647–1653, November 2001.

[14] F. Lin, J. Miller, A. Schoenfeld, M. Ma, and R.J. Baker. A Register-Controlled Symmetrical DLL for Double-Data-Rate DRAM. *IEEE Journal of Solid-State Circuits*, 34(4):565–568, April 1999.

[15] M. Maymandi-Nejad and M. Sachdev. A Monotonic Digitally Controlled Delay Element. *IEEE Journal of Solid-State Circuits*, 40(11):2212– 2219, November 2005.

[16] K. Minami et al. A 1 GHz Portable Digital Delay-Locked Loop with Infinite Phase Capture Ranges. In *International Solid-State Circuits Conference Digest of Technical Papers*, pages 350–351, 469, February 2000.

[17] T. Olsson and P. Nilsson. A Digitally Controlled PLL for SoC Applications. *IEEE Journal of Solid-State Circuits*, 39(5):751–760, May 2004.

[18] J. Phillips and K. Kundert. Noise in Mixers, Oscillators, Samplers, and Logic: An Introduction to Cyclostationary Noise. In *Proc. of the IEEE Custom Integrated Circuits Conference*, pages 431–438, May 2000.

[19] J.M. Rabaey, A. Chandrakasan, and B. Nikolic. *Digital Integrated Circuits: A Design Perspective*. Pearson Education, Inc., second edition, 2003.

[20] S. Tam, R.D. Limaye, and U.N. Desai. Clock Generation and Distribution for the 130-nm Itanium2 Processor with 6-MB On-Die L3 Cache. *IEEE Journal of Solid-State Circuits*, 39(4):636– 642, April 2004.

[21] S. Tam, S. Rusu, U. Desai, R. Kim, Ji Zhang, and I. Young. Clock Generation and Distribution for the First IA-64 Microprocessor. *IEEE Journal of Solid-State Circuits*, 35(11):1545–1552, November 2000.

[22] T. Xanthopoulos, D.W. Bailey, A.K. Gangwar, M.K. Gowan, A.K. Jain, and B.K. Prewitt. The Design and Analysis of the Clock Distribution Network for a 1.2GHz Alpha Microprocessor. In *International Solid-State Circuits Conference Digest of Technical Papers*, pages 402–403, February 2001.

Chapter 4

An All-Digital Phase Aligner with sub-2ps Resolution for Clock Skew Management

This chapter presents an all-digital phase aligner (ADPA) based on the DCDE design presented in Chapter 3 and a PVT tolerant phase detector. Phase aligners have applications in different domains in which synchronization between two signals is essential for correct system operation and for achieving performance goals. This could be needed for signals propagating between two chips or for signals on the same chip. Examples of the former include high speed interfaces for microprocessors and memory busses [25, 28, 30]. One of the most important examples of the later is on-chip clock deskewing.

When synchronizing signals between two chips, the phase adjustment that would be needed tend to be a large percentage of the clock cycle, or even multiple of clock cycles. This is because of the large signal delays through output drives, bond wires, packages, and board traces. For that reason, phase aligners used in such cases need to have a large adjustment range and DLLs are normally used for synchronizing signals in these situations [30]. Several all-digital DLL-based approaches for chip-to-chip synchronization have been proposed in the literature [3,11,15,25,30]

For on-chip phase alignment and when the phase difference range is small, area and power consumption can be saved by using a phase aligner with a limited range. On the other hand, phase alignment for high-speed applications requires the resolution for the phase aligner to be high in order to achieve maximum performance gains. The goal of the work presented in this chapter is to develop a high-precision all-digital phase aligner for on-chip clock deskewing applications.

4.1 Clock Skew Management Techniques

Clock skew can be defined as the difference between the arrival times of the clock signal at different points at the same level of the clock distribution network. As described in [20], a positive skew between two interacting storage devices (i.e. the clock of the destination arrives later than that of the source) has the potential of allowing the circuit to operate at a higher clock frequency but makes it more prone to hold time violations and race conditions. On the other hand, negative skew reduces the probability of race conditions but allows smaller time for logic evaluation and hence reduces performance. Because logic can have data flowing in either direction of clock distribution, it is not always possible to take advantage of either types of skew. In modern high-speed circuit, designing clock distribution networks with minimal clock skew is the preferred design practice [20].

Many different sources contribute to clock skew. They were categorized in [10] as systematic, random, low-frequency, and high-frequency effects. References [5, 10, 18, 23] present details of these different effects.

1. Systematic sources of skew include differences in nominal wire lengths between the clock source and destination registers and differences in nominal clock delays due to load differences. Temperature gradients can also lead to systematic skew. The skew contribution of these sources can be estimated using circuit simulators.

2. Random sources of skew are related to on-die process variations. These variations affect both active buffers and passive clock distribution interconnects. Buffers experience variation in different parameters such as channel lengths and threshold voltages. Interconnects can have variations in parameters such as wire resistivity, dimensions, and dielectric thickness.

3. Low-frequency sources cause slow drifts in clock skew. Temperature changes are one of the most important sources of this skew component.

4. High-frequency sources are normally related to power supply noise causing clock jitter which can be viewed as a special form of clock skew [10].

Modern high-speed custom designs are now operating in the multi-gigahertz regime. Additionally, today's high-speed ASIC designs are becoming more like yesterday's custom designs. These trends towards higher performance are accompanied with trends towards larger uncertainties in the fabrication of nanometer devices [2]. Moreover, with higher levels of integration, die sizes are growing [29] which means that the clock network has to cover increasingly larger areas. All these trends make managing clock skew an essential requirement for current and future high speed designs.

One methodology for clock skew management in high-speed circuits relies on using global clock grids that connect the outputs of a large number of clock drivers. One example for such technique is the Alpha 21264 microprocessor implemented in 0.35μm technology [1, 7]. The global clock distribution network of this microprocessor is depicted in Figure 4.1. The 600MHz clock is generated from a lower frequency input clock using the PLL. This clock is then driven to the center of the chip and then forwarded to a set of distributed clock drivers arranged in the form of four "windowpanes". The outputs from these drivers are then shorted using a grid covering the entire chip area. This arrangement results in clock skew for the global clock of 72ps

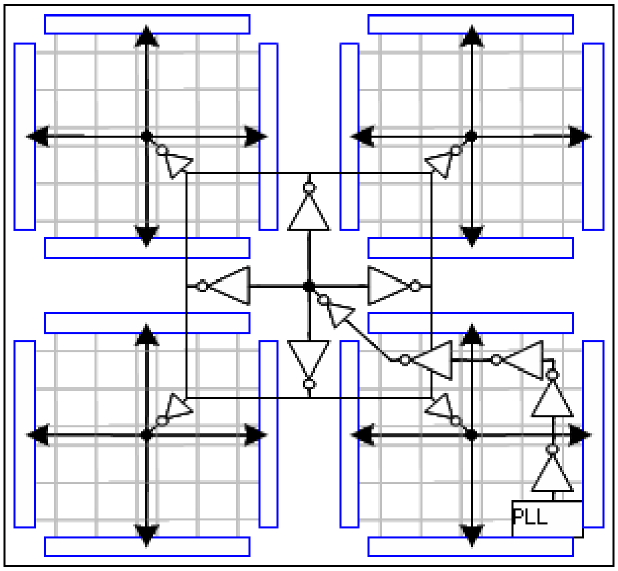


Figure 4.1: Clock distribution of the Alpha 21264 microprocessor [1].

(obtained from simulations). Another example for using the same methodology is the Power4 microprocessor [22]. In this design a balanced and carefully tuned clock tree drives a global clock grid that covers the entire chip area. Reference [22] reports that using this methodology the worst case clock skew due to across-chip variations is 70ps.

Although global clock grids are simple to implement and can effectively minimize clock skew, they have some important drawbacks. They do not account for process variations and require detailed timing analysis. The use of grids requires large use of metal resources and a significant amount of power to drive the clock signals across the chip-wide grid. This becomes even more problematic with power consumption of clock distribution networks becoming a significant percentage of the total power consumption of high-speed circuits [5]. Additionally, global clock grids allow only for localized clock gating [22] which can complicate the clock gating hardware and limit the potential for power reduction through clock gating.

An alternative methodology for skew management is used in [6, 12, 16, 26, 27, 29] and is

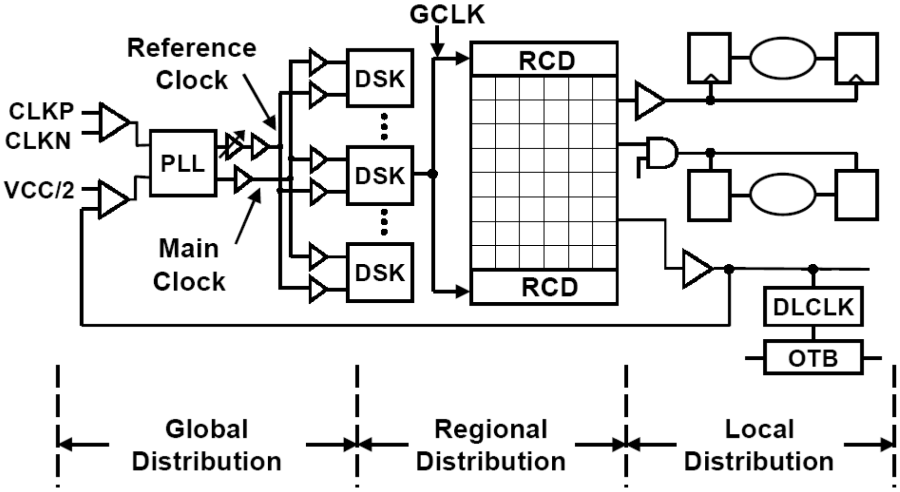


Figure 4.2: Clock distribution of the Itanium microprocessor [27].

illustrated in Figure 4.2. This methodology relies on dividing the chip into different clock domains (or regions) each having a domain clock that controls one functional block or a number of functionally related blocks. Within each clock domain, a low-skew methodology (balanced tree, grid, etc.) is used for the intra-domain clock distribution. Because the size of the clock domain is much smaller than that of the whole chip, it is much more feasible to achieve low levels of skew within the domain. A global clock tree delivers the clock signal to the various domains with active deskew buffers adjusting the skew between all the domain clocks. Clearly, this methodology is more complicated to implement but it offers many benefits. It does not require chip-level detailed timing analysis of the clock distribution as the clock deskew buffers will automatically adjust the skew between the domains. Such analysis would still be needed at the domain level which is much easier to do. The methodology allows for domain-level clock gating. When the deskew buffers are implemented using all-digital phase aligners (ADPAs), they can be used for skew debugging and timing characterization. They can also be used to improve performance by inserting intentional skew at strategic locations [13].

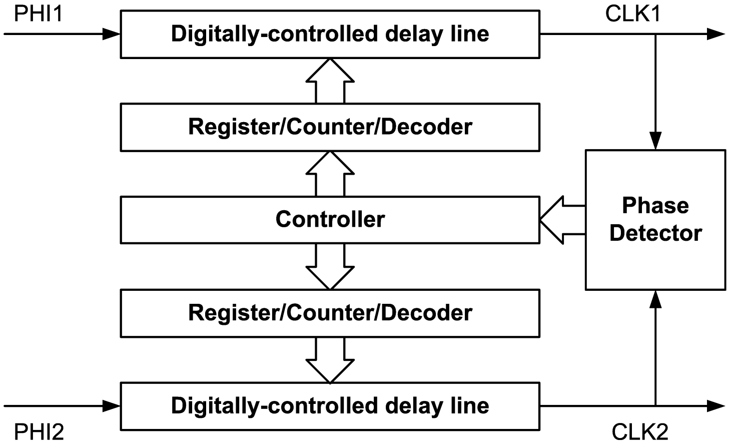


Figure 4.3: Basic architecture of an all-digital phase aligner.

The basic architecture of an ADPA is illustrated in Figure 4.3. Two DCDLs are used to adjust the relative delays of the two input clocks, PHI1 and PHI2. The two delay lines form a differential (or vernier) delay line allowing for fine adjustment of the relative delays of the two inputs. Depending on the implementation of the DCDLs, a combination of a register/counter and/or a decoder could be used to provide the digital control inputs for the DCDLs. For example, if the DCDLs use thermometer coding, only a shift register would be needed [6]. Alternatively, an UP/DOWN counter followed by a binary-to-thermometer decoder could be used. The delayed clocks, CLK1 and CLK2, are sampled using the phase detector (PD) which provides correction signals to the controller. In its basic form, the controller generates the control signals for the register/counter. It can also be used to implement additional functions. For example, the controller can be designed such that the input clocks are sampled at a lower rate. This can be useful in order to allow the PD more time to produce valid outputs if it enters a metastable state [27]. The controller can also be used to implement a digital low pass filter that adjusts the DCDLs delays based on a number of consecutive samples instead of an individual sample. This can be useful to prevent the ADPA from adjusting the delays based on the variations due to the jitter of the input

Ref.	Process	Frequency (GHz)	Deskew resolution (ps)	Total skew(ps)
[6]	0.25μm	–	12	15
[27]	0.18μm	0.8	8.5	28
[29]	0.18μm	1.2	10	<30
[12, 13]	0.18μm	0.1-4GHz	?	<20
[26]	130nm	1.5	7	7
[16]	90nm	0.1-2.5	1	<10

Table 4.1: Performance data for designs using deskew-based methodology.

clocks. It can be seen from Figure 4.3 that the accuracy of delay adjustment depends mainly on the resolution of the DCDLs and the accuracy of the PD, both of which are strong functions of PVT variations.

Table 4.1 summarizes performance data for a number of designs using the deskew-based methodology. Deskew resolution represents the delay step of the DCDLs used in implementing the phase aligner (DLL in reference [29]). As shown in Table 4.1, the total skew is often larger than the DCDL delay step. One source of the additional skew is the offset of the PD due to random process variations. The other source is skew accumulation between different deskew buffers. When the chip area is (relatively) small, a small number of deskew buffers would be needed and the domain clocks are aligned with one reference clock propagated to all domain buffers. This arrangement was used in references [26, 27, 29]. Under these circumstances, the total skew would be limited to the accuracy of the DCDLs and the PD. On the other hand, when a large number of deskew buffers is needed, clock deskewing need to be done in a hierarchical manner which results in skew accumulation across the hierarchy of deskew buffers [12, 16].

From the previous discussion, it is clear that achieving precise phase alignment depends on designing high-resolution DCDEs and PDs that will maintain their precision under PVT variations. The following section presents an ADPA that uses the DCDE presented in Chapter 3 and a PVT tolerant PD.

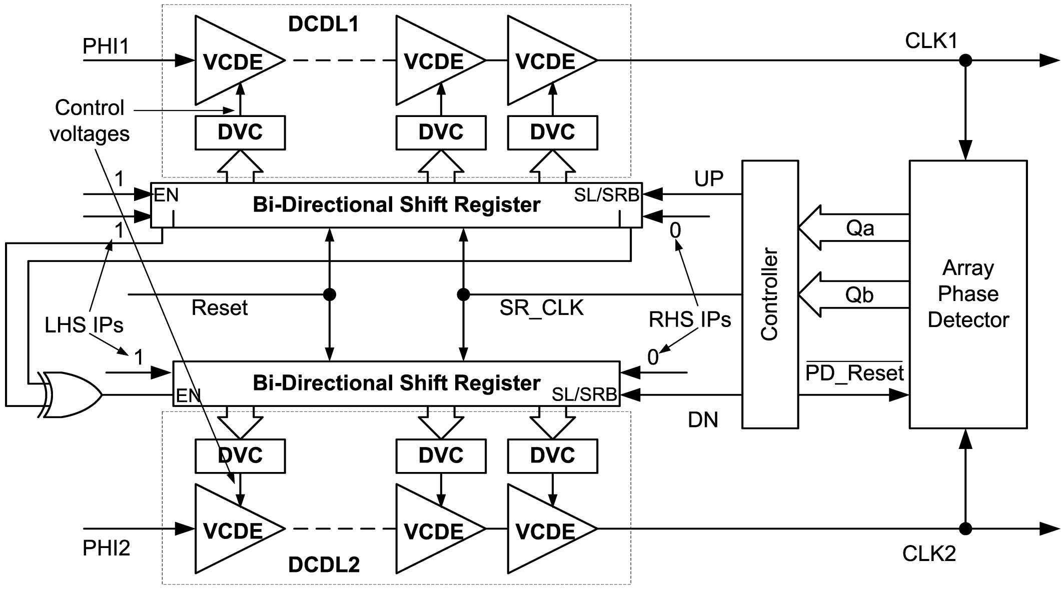


Figure 4.4: Architecture of proposed all-digital phase aligner.

4.2 Proposed All-Digital Phase Aligner Design

Figure 4.4 shows the architecture of the ADPA presented in this work. It consists of two DCDLs, two bi-directional shift registers, a phase detector, and a controller. Details of the design of the main components are presented in the following sections.

4.2.1 Digitally-Controlled Delay Lines and Related Control Circuitry

The DCDLs are designed using the DCDE proposed in Chapter 3 after reducing its delay step from 5ps to 2ps and increasing the number of step per DCDE from 7 to 15. This gives a total delay range of 30ps under *typical* conditions. As can be predicted from the data in Figure 3.18, this number would increase to 3.12ps under *slow* conditions. If this value is not acceptable for some target application, the DCDE design can be easily modified in order to achieve the desired (smaller) delay step. The number of DCDEs used in each of the DCDLs determines the phase alignment range. In this work, seven DCDEs were used in each of the delay lines, giving a nominal phase alignment range of ±210ps. This is also a design variable that depends on the target application.

Mismatch between the DCDEs can cause the delay step to be different from one code to the other within the same DCDE. Such variations would be very limited as they are related only to the mismatch between the currents supplied by the PMOS load transistors in Figure 3.12. On the other hand, mismatch between the different DCDEs, can cause much larger variations in the delay steps. Such uncertainties can cause the ADPA to have non-uniform phase adjustment steps. However, such non-uniformity does not impact the resolution of the ADPA as the delay step would always be limited to the worst case delay step achieved under *slow* conditions. The impact of such variation would mainly be on the phase adjustment range achieved by the ADPA.

The DCDLs are controlled using bi-directional shift registers. Each of the two registers has 105 flip-flops whose outputs are connected to the control inputs of the DCDEs. Multiplexers at the inputs of the flip-flops allow the control word to be shifted to the left or the right depending on the values of the direction control inputs (SL/SRB). When shifting left, zeros are appended to right of the control word, while if shifting right, ones are appended to the left. If the direction input is HIGH, the register performs a shift-left operation resulting in a larger number of zeros in the input code of the DCDL. This results in smaller delays for that DCDL. On the other hand, shifting to the right results in a smaller number of zeros and larger delays.

The direction inputs are supplied from the controller block. These two signals (UP and DN) are mutually exclusive in the sense that, when the shift registers are clocked, only one of these two signals would be HIGH. This ensures that the two registers would always be shifting in opposite directions. The clock for the shift registers, SR_CLK, is generated by ORing the UP and DN signals, ensuring delay adjustment only when the difference between CLK1 and CLK2 is resolvable by the PD (i.e. either UP or DN is HIGH). This approach ensures correct timing of the SR_CLK with respect to the UP and DN signals. In order to ensure delay adjustment by only one DCDE delay step at a time, only one of the two shift registers need to be shifting at any time. This is achieved using the enable inputs of the shift registers, EN. The top shift register is always enabled. However, when all values stores in this shift register are equal (i.e. all ones or all zeros), shifting in the same direction that caused this situation does not change the contents of this register. On the other hand, the EXOR feeding the EN input of the bottom shift register detects this situation and allows that shift register to start shifting in the opposite direction.

4.2.2 Phase Detector with Redundancy for PVT Tolerance

A phase detector samples two input signals and produces outputs that reflect the phase relationship between the two inputs. Many different types of phase detectors have been used in the literature [3, 6, 9, 17, 19, 24, 28]. In some phase detector designs, the two inputs see identical delays in all sections of the phase detector resulting in the ability of the design to resolve very small phase differences, limited by the potential of metastability and sampling offsets. On the other hand, some phase detector designs have two sections with imbalanced delays seen by the two inputs in each section. This creates a guard band (also called dead zone) around the inputs, such that when the phase difference is within this guard band, it will not be "seen" by the phase detector. A guard band is not desirable for charge-pump based phase correction circuits as it opens the feedback loop and stops phase correction when the phase error is within the guard band [21]. The opposite is true for all-digital phase correction circuits. Because of the finite phase adjustment step, a guard band that is slightly larger than the adjustment step is desirable [4]. It ensures system stability by preventing further phase correction when the phase error is less than the delay step. If that is not the case (i.e. if the guard band is smaller than the adjustment step), phase correction for a difference that is slightly larger than the guard band can cause an adjustment that would trigger a correction in the opposite direction. This situation causes deterministic jitter to be added to the output signals.

Process variations introduce offset in the phase detector's characteristics and can seriously degrade the accuracy of phase detection. For example, reference [8] reported that the offset of 64 arbiters implemented in 0.35μm CMOS was in the range of -30 to +40ps. For a similar design implemented in 0.18μm technology, the offset was reported to be in the range of +3 to +16ps (for 32 arbiters). Additionally, temperature and supply variations can cause changes in the size of the guard band resulting in variations in the accuracy. Such variations can result in reduced accuracy with improved stability (for larger guard bands) or improved accuracy and reduced stability (for smaller guard bands).

Reference [14] suggested using redundancy to improve the accuracy of phase detection by implementing multiple copies of the PD and selecting the ones with lower offsets. The selection is done using code density-based calibration. Such calibration would need to be repeated when temperature or supply conditions change, which may not be attractive for clock deskewing applications.

This work proposes implementing redundancy in the phase detector in order to alleviate the loss of accuracy that can result from PVT variations. The technique is fully autonomous and does not require calibration. The basic (i.e without redundancy) phase detector used in this work is chosen to be of a type that has a guard band and is similar to those used in [4, 6]. The focus of this work is not on the particular phase detector, rather it is on using redundancy to improve accuracy. These ideas can be extended to other types and structures of phase detectors.

The basic phase detector used in this work is shown in Figure 4.5. It consists of two D flip-flops (DFFs) with each input clock (CLK1/CLK2) connected to the clock input of one DFF and the D input of the other. The DFF is taken from a standard cells library and has a nominal rising-input setup time (t_{su}) of -15ps. As the setup time of the flip-flop is what determines the size of the guard band, buffers are added to the inputs in order to provide the ability to adjust the setup

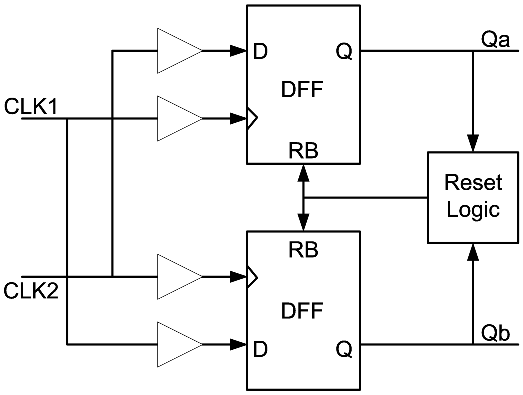


Figure 4.5: Basic phase detector.

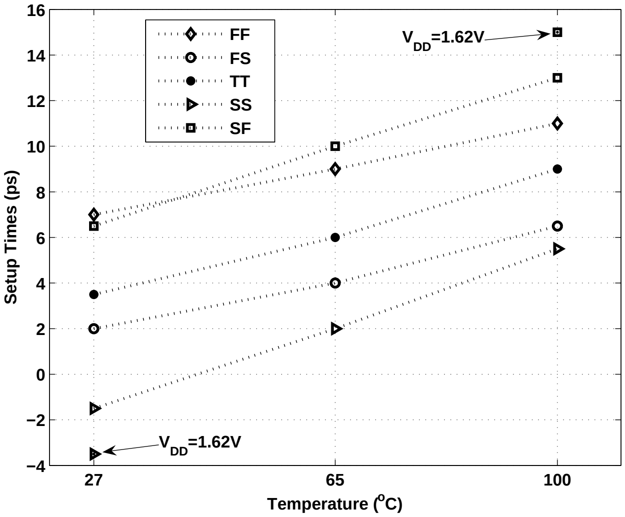


Figure 4.6: DFF setup time vs. PVT variations

time (and hence the guard band) by skewing the delays of the clock and D inputs of the DFF. In the following discussions, setup time refers to the modified, rather than the original, setup time of the DFF.

Nominally, the setup times of both DFFs would be identical and can be either positive, negative, or zero. When both setup times are zero, there is no guard band and the phase detector in Figure 4.5 behaves in the same way as an arbiter or an SR latch. When both setup times are positive, the PD would have a guard band whose width is equal to the setup time. This means that when the phase difference between CLK1 and CLK2 is less than the setup time, neither DFFs would switch to HIGH (after being initially reset to LOW) and the PD will not resolve the phase difference. Alternatively, if both DFFs have negative setup times, a phase difference that is less than the absolute value of the setup time causes both Qa and Qb to switch to HIGH, and, again, the PD will not resolve the phase difference. In both of these situations (positive and negative setup times), a phase difference that is larger than the absolute value of the setup time causes only one of the two outputs to switch to HIGH, depending on which of the two clocks is lagging.

If CLK1 lags CLK2 (by more than the setup time), Qa switches to HIGH, and vice versa. From this discussion, it is clear that the setup times of the DFFs determine the accuracy of the phase detector. Maintaining small DFFs' setup times is essential for maintaining high resolution for phase alignment.

In order to understand the impact of PVT variations on the DFFs' setup time, the DFFs have been designed to have a setup time of 3.5ps (slightly larger than the worst case delay step from the DCDLs) for the TT process, 27^{o}C, and nominal V$_{DD}$. The DFF has then been simulated at different process corners[1], temperatures, and supply voltages. Figure 4.6 shows the variations in the setup time under these conditions. All simulations were done at nominal supply, except the two points with the smallest and largest setup times which were obtained using a supply voltage that is 10% less than nominal. As shown in the figure, going from the SS corner at 27^{o}C and low V$_{DD}$ to the SF corner at 100^{o}C and low V$_{DD}$, the setup time of the DFF changes from -3.5ps to 15ps. These results indicate that in order to maintain system stability, the design has to guarantee that the setup time of the DFF is larger that the worst case delay step under all possible PVT conditions. This means, for example, that the characteristics in Figure 4.6 need to be shifted upwards so that the lowest point (SS, 27^{o}C, low V$_{DD}$) happens at 3.2ps or more. Clearly, achieving that would be at the expense of loss of resolution, which would then take any value from 3.2ps to 21.7ps, depending on the PVT condition.

It is arguable that, when the phase aligner is capable of guaranteeing a small phase adjustment step (e.g. 3.2ps), if the guard band of the phase detector becomes smaller than the delay step, the added deterministic jitter would be much smaller than the system jitter and would have minimal impact on system performance (compared to the benefit of adjusting the phase with high accuracy). Additionally, some techniques can be used in order to alleviate the possible instability due to phase corrections. Examples include turning phase correction off after lock condition is

[1] Only corner simulations were available at the time of designing the test chip.

detected or making phase correction based on the history of a number of samples rather than an individual sample [27].

Based on this discussion, the target accuracy of the phase detector has been changed from 3.2ps to 2ps. In order to guarantee a high probability of achieving the desired target accuracy, an array phase detector (APD) is used. The APD consists of a number of phase detectors with skewed setup times. This redundancy ensures that a subset of the PDs would be active (i.e. have a small guard band) under a possible range of PVT conditions, such that all ranges of PVT conditions would be covered with high accuracy by at least one PD. The number of PDs to be used in the APD poses a trade-off between accuracy and design complexity. More PDs increase the probability that a given target accuracy will be achieved, but it comes at the expense of area and added complexity of the associated control hardware.

From the data of Figure 4.6, it is observed that the variations in the setup time has a total range of 18.5ps. It is also observed that under nominal conditions (TT, 27^{o}C, nominal V$_{DD}$), the setup time has a total spread of approximately 8ps. Based on this data, an APD has been designed to have five PDs with nominal setup times spread at 4ps intervals. One of the PDs is designed to have a nominal setup time of 2ps. All other PDs are placed around this PD in an arrangement that will result in proper shifting of the APD characteristics under PVT variations. This arrangement is depicted graphically in Figure 4.7 showing the setup time characteristics of one set of (five) DFFs used in the APD. The other set of DFFs have the same phase characteristics. The second set of curves in Figure 4.7 represent the characteristics under nominal temperature and supply voltage. Under such conditions, PD2 and PD3 would be the phase detectors providing phase information. When the supply voltage drops to 1.62V, the setup time characteristics shifts to the left (smaller setup times) and PD4 contributes to providing phase information. Similarly, when the temperature increases to 100^{o}C, PD0, PD1, and PD2 would be the phase detectors contributing to phase information. If at 100^{o}C the supply voltage drops to 1.62V, PD0 and PD1 will provide the phase information.

Ideally, all PDs should give consistent phase information. This means that, if one of the five PDs indicates that, for example, CLK1 leads CLK2, each of the other four will either indicate the same phase relationship or indicate that it is not able to resolve the phase difference. Because of mismatch between the DFFs and the overlap in the setup time characteristics, it is possible that two PDs would give opposite phase difference indications. For this reason, the ADPA's controller uses majority voters in order to produce phase correction signals.

4.2.3 ADPA Controller

The ADPA controller accept the outputs from the APD and produces the phase correction signals UP, DN, and SR_CLK. It also produces the $\overline{\text{PD\_Reset}}$ input of the APD. The schematic diagram of the controller is shown in Figure 4.8. The controller inputs, Qa and Qb are the two 5-bit words produced by the APD. As can be seen from Figure 4.5, Qa and Qb are the outputs of the DFFs that take CLK1 and CLK2 as their input clocks, respectively. Two majority voters decide whether a phase correction is to take place and if so they decide on the direction. If the number of ones in Qa is larger than the number of ones in Qb, the output of the top majority voter would be HIGH driving the UP signal HIGH, while the DN signal would be LOW and vice versa.

The two majority voters guarantee that only one of the UP and DN signals can switch HIGH after a sample is taken. Drivers are used to supply the UP and DN signals to the 105 flip-flops in the respective shift registers (Figure 4.4). The SR_CLK is generated from the UP and DN signals using the NOR gate. Because the UP and DN signals drive the direction selection multiplexers in the shift registers, a MUX is used to delay the SR_CLK signal in order to ensure that the UP and DN signals arrive at the shift registers before the SR_CLK signal. The SR_CLK signal is then inverted and driven to the 210 flip-flops in the two shift registers allowing phase correction to take place. The SR_CLK is then used to generate the reset input to the APD allowing a new

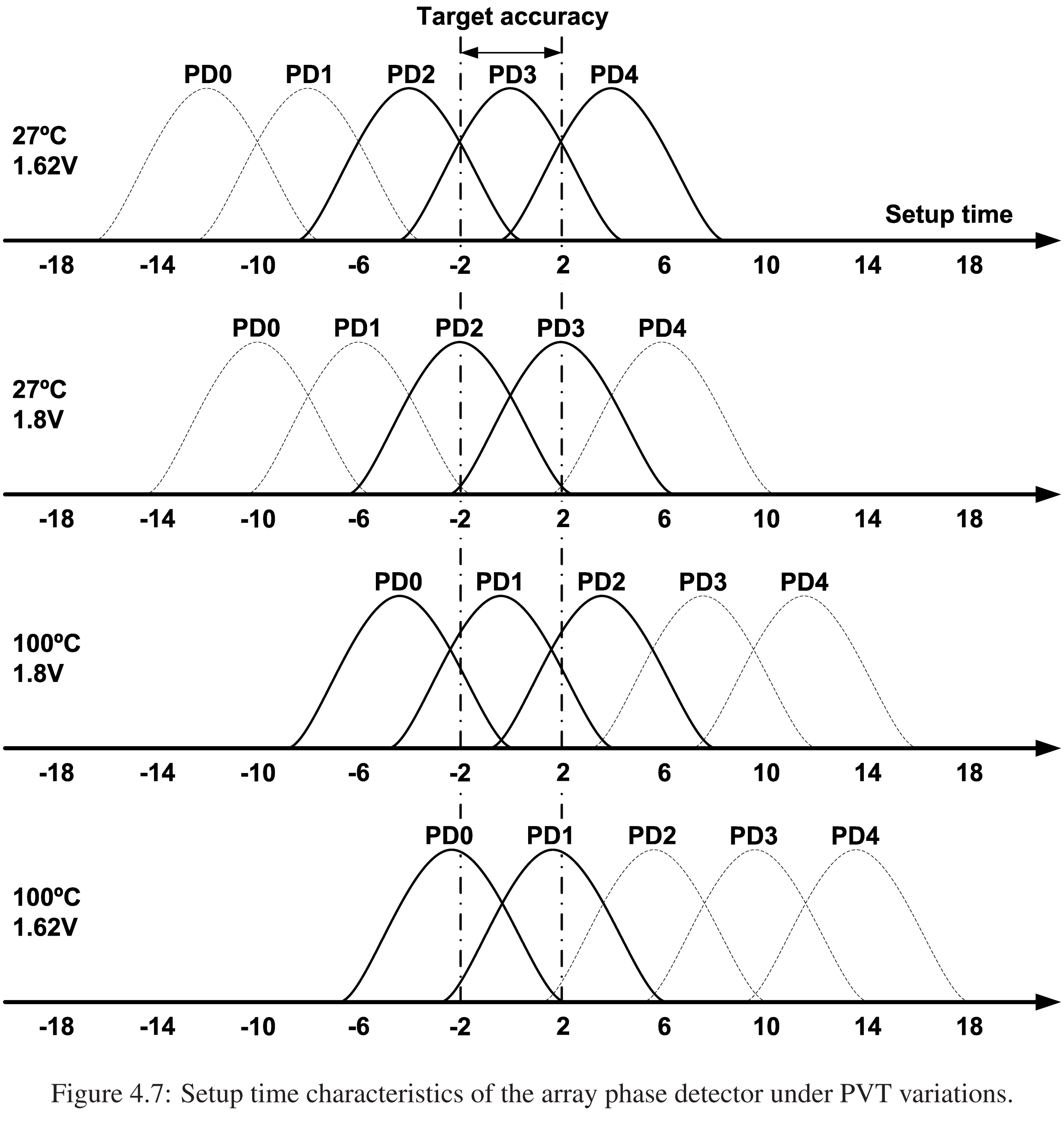


Figure 4.7: Setup time characteristics of the array phase detector under PVT variations.

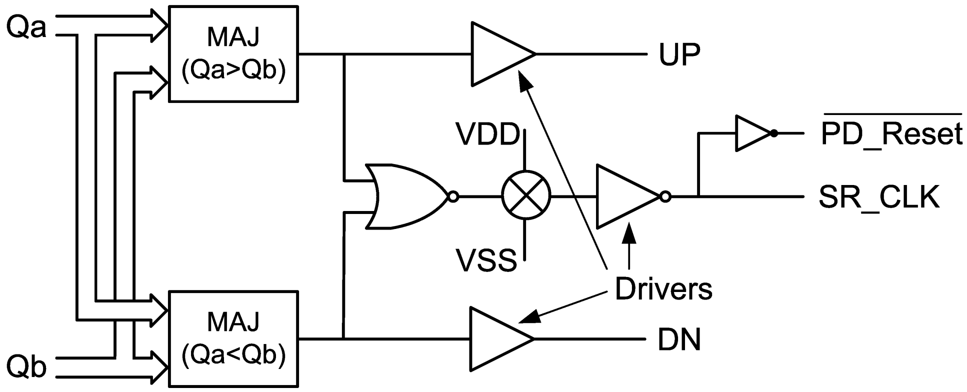


Figure 4.8: Schematic of the ADPA controller.

sample to be taken. Figure 4.9 shows the timing relationship between the various signals in the ADPA.

4.3 ADPA Simulation Results

The proposed ADPA has been designed in 0.18μm CMOS technology. With ASIC design being a target application, the design makes extensive use of standard cells. The only exceptions to this are the DCDEs and the buffers used to adjust the setup times of the APD DFFs. Extensive simulations were performed in order to verify correct timing and functionality under different PVT conditions. Simulations show that the ADPA's nominal feedback loop delay is 2.5ns. Referring to Figure 4.9, this is the delay from the transition of the earliest of CLK1 and CLK2 until the $\overline{\text{PD\_Reset}}$ signal is de-asserted (to HIGH). Allowing a safety margin of another 0.5ns, this means that the ADPA is able to take a sample and make a phase adjustment (if needed) once every 3ns. This means that for frequencies higher than 333MHz, the ADPA will take a sample once every multiple clock cycles. From the point of view of metastability, such large delay is considered to be advantageous as it allows more time for the system to resolve its output in case the APD

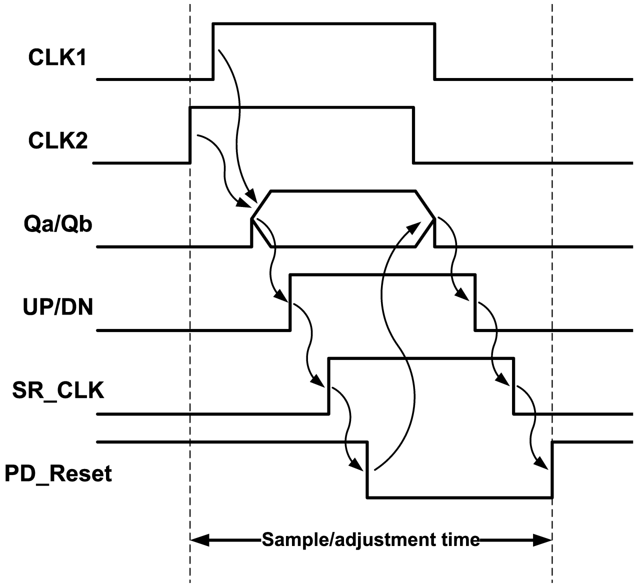


Figure 4.9: Timing relationship between ADPA signals.

enters a metastable state [6, 27]. This comes at the expense of the total number of cycles needed to align the two clocks.

A simulation example that shows the operation of the ADPA with input frequency of 1GHz is presented in Figure 4.10. This example is set up in order to show how the ADPA would respond to a sudden change in the phase relationship that causes the polarity of the phase difference between the two clocks to be reversed. Before running this simulation, CLK1 was set to lead CLK2 by approximately 50ps and the ADPA was simulated until the two clocks were aligned. This resulted in the initial values of the control voltages of the two DCDLs shown at 0ns in Figure 4.10. Because CLK1 was lagging, the ADPA caused the control voltage of two of the DCDEs in the CLK1 DCDL to increase (one after the other) until the two clock were aligned. Because the CLK2's shift register was not enabled, the control voltages of the corresponding DCDL all remained at their minimum values. At the beginning of the simulation shown in Figure 4.10, the relative delays between CLK1 and CLK2 are adjusted so that CLK1 leads CLK2 by 50ps. This is shown in the close-up view (before alignment) in the top left corner of the figure. Once the ADPA detects this situation, it start sampling the two clocks and asserting the DN signal once every three clock cycles. The UP signal remains LOW throughout this simulation. Because only the CLK1 shift register is enabled at this point, every sample taken by the ADPA causes an additional “1” to be shifted from the left side of DCDL1 (Figure 4.3) reducing the control voltages of DCDL1 and increasing the delay of CLK1, bringing it closer to CLK2. This operation continues until the CLK1 shift register stores all ones, at which point the CLK2 shift registers is enabled and zeros start to get shifted from the right side of this shift register. This results in the control voltages of DCDL2 to increase (one after the other) reducing the delay of CLK2 and bringing it closer to CLK1. This operation continues until the two clocks are aligned and no additional correction signals are generated. This is shown in the close-up view at the top right corner of Figure 4.10.

4.4 Chip Implementation and Experimental Results

The proposed ADPA has been implemented in 0.18μm CMOS technology. The chip micrograph is shown in Figure 4.11. The chip is pad-limited with a total area of 2.25mm^2. The area used for the various components of the ADPA is 0.1154mm^2. An input buffer is used to drive the ADPA inputs, PHI1 and PHI2. In order to observe the inputs and outputs of the ADPA, four output drivers have been implemented on the test chip. The topmost drivers are used to drive PHI1 and PHI2, with the possibility of multiplexing either of the two signals to either of the two drivers. This arrangement is used in the testing in order to extract away differences in the delays of the output drivers, bond-wires, and PCB traces. Similarly, the bottom drivers are used for either CLK1 and CLK2. A control shift register is used to store the control signals of the four multiplexers.

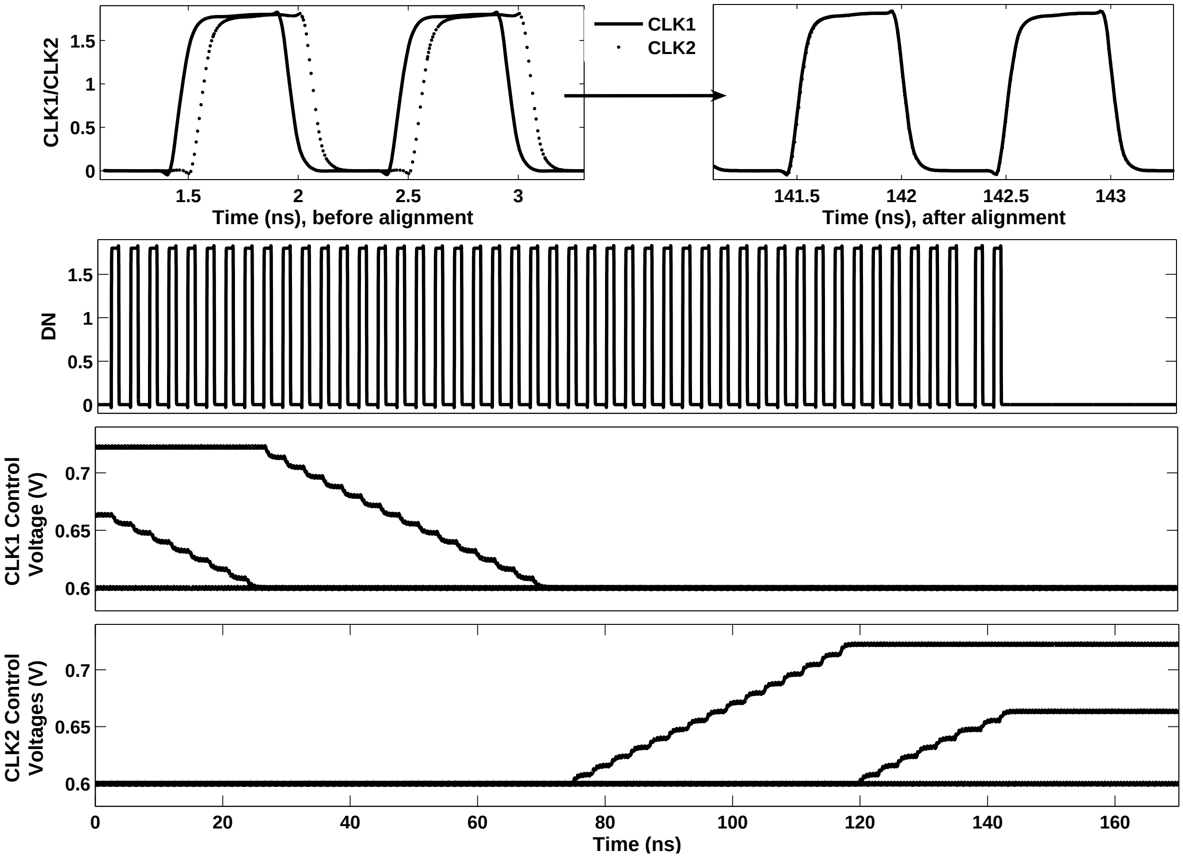


Figure 4.10: Simulation example for the ADPA at 1GHz.

The chip is tested by providing input clocks from an Agilent VXI test system which allows the relative delays of the inputs to be adjusted in steps of 1ps. An Infiniium 54853A digital storage oscilloscope is used to observe and measure the delays of the inputs and outputs of the ADPA. As was done in the testing of the DCDE, averaging and waveform memory were used to obtain accurate delay measurements.

The PCB used for testing the design was implemented as a two layer board which is not suitable for high speed operation. Because of this limitation and the limitations of the output drivers, the ADPA was tested at an input frequency of 150MHz. Example measurement results

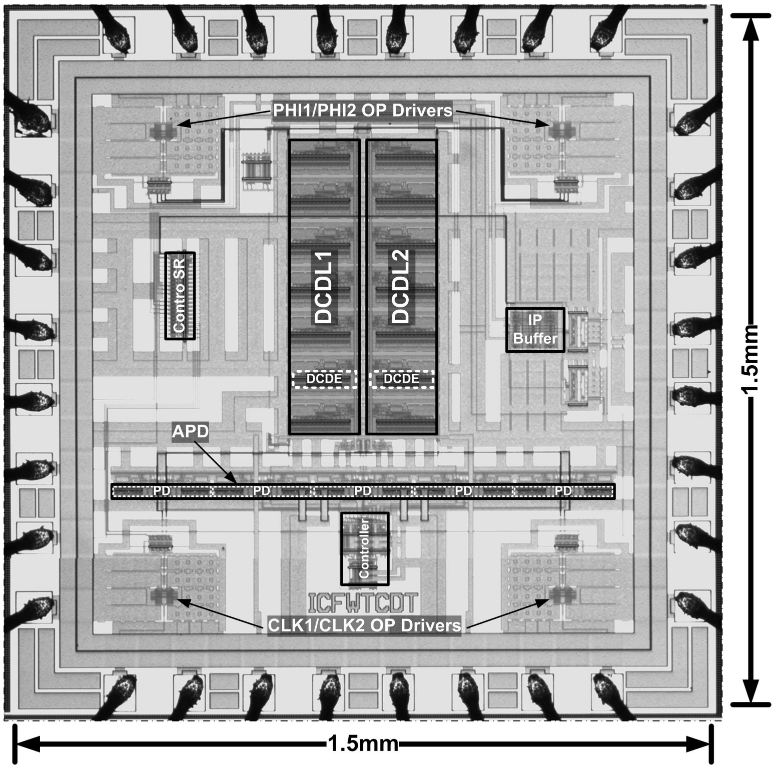
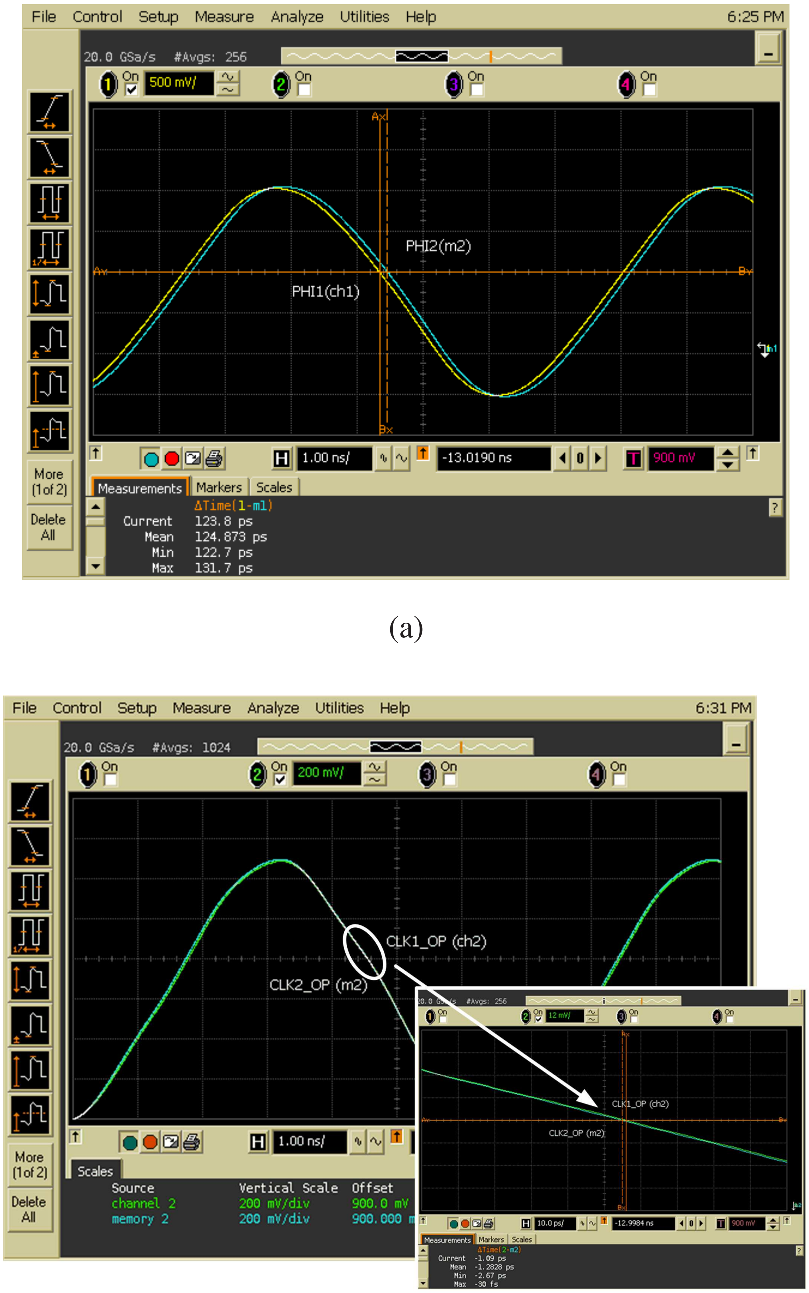


Figure 4.11: ADPA test chip micrograph.

are shown in Figures 4.12 and 4.13 showing ADPA inputs and outputs for phase differences of +123.8ps and -158.4ps (relative to PHI1) at the nominal supply voltage of 1.8V. In both cases, the ADPA adjusts the phase difference between the output to less than 2ps. Similar results were obtained under various values of the supply voltage ranging from 1.6 to 2V. For the nominal supply voltage, measurements show that the tested ADPA has a phase alignment range of -264ps to +247ps (compared to the nominal ±210ps) which corresponds to an average DCDE delay step of 2.43ps. At a supply voltage of 1.8V and an input frequency of 150MHz, the ADPA consumes 12mW of power.

Jitter contributions of the DCDLs were estimated by measuring DCDL1's input and output jitter when it has maximum delay (which corresponds to maximum jitter contribution). The rms



(a)

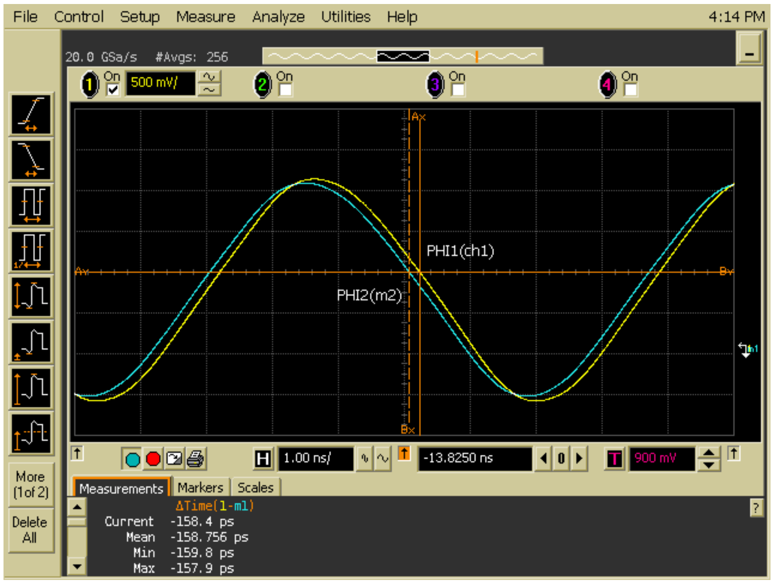
```
ΔTime(1-m1)
Current   123.8 ps
   Mean   124.873 ps
    Min   122.7 ps
    Max   131.7 ps
```

(b)

```
Source      Vertical Scale  Offset
channel 2   200 mV/div      900.0 mV
memory 2    200 mV/div      900.000 m
```

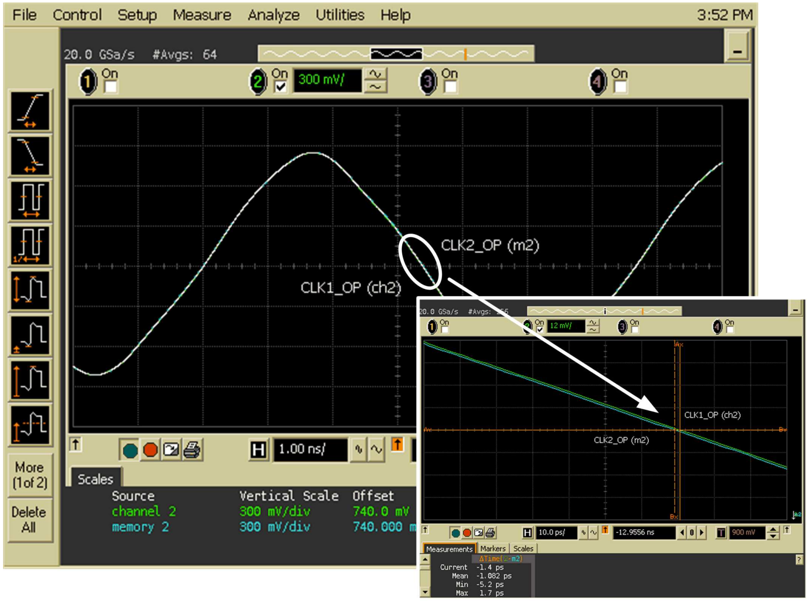
```
ΔTime(2-m2)
Current  -1.09 ps
   Mean  -1.2828 ps
    Min  -2.67 ps
    Max  -30 fs
```

Figure 4.12: Measurement results for PHI1 leading PHI2.



```
ΔTime(1-m1)
Current  -158.4 ps
   Mean  -158.756 ps
    Min  -159.8 ps
    Max  -157.9 ps
```

(a)



```
Source     Vertical Scale  Offset
channel 2  300 mV/div      740.0 mV
memory 2   300 mV/div      740.000 m
```

```
ΔTime(2-m2)
Current  -1.4 ps
   Mean  -1.082 ps
    Min  -5.2 ps
    Max   1.7 ps
```

(b)

Figure 4.13: Measurement results for PHI1 lagging PHI2.

input and output jitter values were measured to be 4.866 and 4.994ps resulting in a maximum DCDL rms jitter of 1.123ps. This value translates to a maximum ADPA rms jitter of 1.59ps. It also indicates that the rms jitter contribution of the individual DCDE (each DCDL has seven DCDEs) is 0.424ps. This value is smaller than that obtained from the DCDE chip in Chapter 3. This can be attributed to using separate supply and ground pins for the chip core and each of the output drivers which reduces the core supply noise considerably.

4.5 Conclusions

In this chapter, an ADPA for clock deskew applications has been proposed. The design uses the DCDE proposed in Chapter 3 with a delay step of 2ps. Redundancy is used in the phase detector in order to provide PVT tolerance and increase the probability of achieving high accuracy under such variations. Simulation and experimental results demonstrate that the design is capable of achieving sub-2ps resolution in 0.18μm technology. The design makes extensive use of standard cells which makes it attractive for ASIC applications.

References

[1] D.W. Bailey and B.J. Benschneider. Clocking Design and Analysis for a 600-MHz Alpha Microprocessor. *IEEE Journal of Solid-State Circuits*, 33(11):1627–1633, November 1998.

[2] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De. Parameter Variations and Impact on Circuits and Microarchitecture. In *Proc. of the Design Automation Conference*, pages 338–342, June 2003.

[3] G.K. Dehng, J.M. Hsu, C.Y. Yang, and S.I. Liu. Clock-Deskew Buffer Using a SAR-Controlled Delay-Locked Loop. *IEEE Journal of Solid-State Circuits*, 35(8):1128–1136, August 2000.

[4] C.E. Dike, N.A. Kurd, P. Patra, and J. Barkatullah. A Design for Digital, Dynamic Clock Deskew. In *2003 Symposium on VLSI Circuits Digest of Technical Papers*, pages 21–24, June 2003.

[5] E.G. Friedman. Clock Distribution Networks in Synchronous Digital Integrated Circuits. *Proc. of the IEEE,*, 89(5):665–692, May 2001.

[6] G. Geannopoulos and X. Dai. An Adaptive Digital Deskewing Circuit For Clock Distribution Networks. In *International Solid-State Circuits Conference Digest of Technical Papers*, pages 400–401, February 1998.

[7] P.E. Gronowski, W.J. Bowhill, R.P. Preston, M.K. Gowan, and R.L. Allmon. High-Performance Microprocessor Design. *IEEE Journal of Solid-State Circuits*, 33(5):676–686, May 1998.

[8] V. Gutnik and A. Chandrakasan. On-Chip Picosecond Time Measurement. In *2000 Symposium on VLSI Circuits Digest of Technical Papers*, pages 52–53, June 2000.

[9] T. Hamamoto, S. Kawasaki, K. Furutani, K. Yasuda, and Y. Konishi. A Skew and Jitter Suppressed DLL Architecture for High Frequency DDR SDRAMs. In *2000 Symposium on VLSI Circuits Digest of Technical Papers*, pages 76–81, June 2000.

[10] D. Harris and S. Naffziger. Statistical Clock Skew Modeling with Data Delay Variations. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 9(6):888–898, December 2001.

[11] A. Hatakeyama et al. A 256-Mb SDRAM Using a Register-Controlled Digital DLL. *IEEE Journal of Solid-State Circuits*, 32(11):1728–1734, November 1997.

[12] N.A. Kurd, J.S. Barkarullah, R.O. Dizon, T.D. Fletcher, and P.D. Madland. A Multigigahertz Clocking Scheme for the Pentium Microprocessor. *IEEE Journal of Solid-State Circuits*, 36(11):1647–1653, November 2001.

[13] N.A. Kurd, J.S. Barkarullah, R.O. Dizon, T.D. Fletcher, and P.D. Madland. Multi-GHz Clocking Scheme for Intel Pentium 4 Microprocessor. In *International Solid-State Circuits Conference Digest of Technical Papers*, pages 404–405, February 2001.

[14] L.-M. Lee, D. Weinlader, and C.-K. K. Yang. A Sub-10-ps Multiphase Sampling System Using Redundancy. *IEEE Journal of Solid-State Circuits*, 41(1):265–273, January 2006.

[15] F. Lin, J. Miller, A. Schoenfeld, M. Ma, and R.J. Baker. A Register-Controlled Symmetrical DLL for Double-Data-Rate DRAM. *IEEE Journal of Solid-State Circuits*, 34(4):565–568, April 1999.

[16] P. Mahoney, E. Fetzer, B. Doyle, and S. Naffziger. Clock distribution on a Dual-Core, Multi-Threaded Itanium-Family Processor. In *International Solid-State Circuits Conference Digest of Technical Papers*, pages 292–293,599, February 2005.

[17] K. Minami et al. A 1 GHz Portable Digital Delay-Locked Loop with Infinite Phase Capture Ranges. In *International Solid-State Circuits Conference Digest of Technical Papers*, pages 350–351, 469, February 2000.

[18] S.R. Nassif. Within-Chip Variability Analysis. In *Technical Digest of Electron Devices Meeting*, pages 283–286, December 1998.

[19] T. Olsson and P. Nilsson. A Digitally Controlled PLL for SoC Applications. *IEEE Journal of Solid-State Circuits*, 39(5):751–760, May 2004.

[20] J.M. Rabaey, A. Chandrakasan, and B. Nikolic. *Digital Integrated Circuits: A Design Perspective*. Pearson Education, Inc., second edition, 2003.

[21] B. Razavi. *Design of Analog CMOS Integrated Circuits*. McGraw-Hill series in electrical and computer engineering. McGraw-Hill, 2001.

[22] P.J. Restle et al. The Clock Distribution of the Power4 Microprocessor. In *International Solid-State Circuits Conference Digest of Technical Papers*, pages 144–145, February 2002.

[23] S. Sauter, D. Schmitt-Landsiedel, R. Thewes, and W. Weber. Effect of Parameter Variations at Chip and Wafer Level on Clock Skews. *IEEE Transactions on Semiconductor Manufacturing*, 13(4):395–400, November 2000.

[24] S. Sidiropoulos and M. Horowitz. A Semidigital Dual Delay-Locked Loop. *IEEE Journal of Solid-State Circuits*, 32(11):1683–1692, November 1997.

[25] H. Song. Programmable De-Skew Clock Generation Based on Dual Digital Delay-Locked Loop Structure. In *Proc. of the IEEE International System-on-Chip Conference*, pages 249–250, September 2003.

[26] S. Tam, R.D. Limaye, and U.N. Desai. Clock Generation and Distribution for the 130-nm Itanium2 Processor with 6-MB On-Die L3 Cache. *IEEE Journal of Solid-State Circuits*, 39(4):636– 642, April 2004.

[27] S. Tam, S. Rusu, U. Desai, R. Kim, Ji Zhang, and I. Young. Clock Generation and Distribution for the First IA-64 Microprocessor. *IEEE Journal of Solid-State Circuits*, 35(11):1545–1552, November 2000.

[28] S. Tanoi, T. Tanabe, K. Takahashi, S. Miyamoto, and M. Uesugi. A 250-622 MHz Deskew and Jitter-Suppressed Clock Buffer using Two-Loop Architecture. *IEEE Journal of Solid-State Circuits*, 31(4):487–493, April 1996.

[29] T. Xanthopoulos, D.W. Bailey, A.K. Gangwar, M.K. Gowan, A.K. Jain, and B.K. Prewitt. The Design and Analysis of the Clock Distribution Network for a 1.2GHz Alpha Microprocessor. In *International Solid-State Circuits Conference Digest of Technical Papers*, pages 402–403, February 2001.

[30] C.Y. Yang and S.I. Liu. A One-Wire Approach for Skew-Compensating Clock Distribution based on Bidirectional Techniques. *IEEE Journal of Solid-State Circuits*, 36(2):266–272, February 2001.

Chapter 5

A 6-bit Flash Time-to-Digital Converter

A time-to-digital converter (TDC) converts the time difference between two input signals or events into digital code. It has a wide range of applications including time-of-flight detection [21], laser range finding [25], temperature sensing [8], medical imaging [40], high-energy physics experiments [5], and optical characterization of CMOS circuits [39]. In recent years, TDCs have been used in all-digital PLLs and DLLs to provide phase and frequency detection [12,38,43]. One of the most common application domains for TDCs is in the area of test and measurement. TDCs are used for performance characterization of systems-on-chip (SoCs) [24], for jitter and skew measurement [7, 23], and for on-chip timing characterization of I/Os [42]. They are also used for timing control in different test equipment including oscilloscopes, logic analyzers, and signal generators [26, 37].

5.1 Review of Time-to-Digital Converter Implementations

Many analog and digital techniques have been used for TDC implementation. Analog techniques rely on converting time into a voltage level using a time-to-amplitude converter (TAC).

The TAC is then followed by an analog-to-digital converter (ADC) [35,40] or an analog content-addressable-memory (CAM) [14]. Although the analog techniques can achieve high measurement precision, they have some important drawbacks that make them less attractive from the point of view of modern high-speed digital systems. These drawbacks include large area and power consumption [41], requirement of good analog processes and difficulty of integration using standard digital processes [11], longer conversion time, and sensitivity to noise [20].

The simplest digital technique for time measurement uses a counter to measure the number of clock cycles that lapse between two events. This technique is simple to implement, has a wide measurement range, but has a low resolution. Many digital techniques that can achieve a much higher resolution have been proposed in the literature. Such techniques can be used independently or can be combined with a counter in order to achieve a large measurement range with high resolution. Following is a summary of the high-resolution digital time measurement techniques. A comprehensive review of such techniques was presented in [20].

1. **Delay Line Technique**. The delay line technique is based on sampling the input signal using shifted versions of a reference signal created using a delay line. In order to stabilize and calibrate the delay of the delay elements, a DLL [2] or a PLL [3] can be used. In another variation, the sampling phases are created using a ring oscillator [27]. The delay line technique is simple and achieves good linearity but its resolution is limited to the delay of the basic delay cell. The conversion rate of the delay line technique is limited by the total delay through the delay line. Its INL[1] is limited by the mismatch between the delay elements and the sampling offset of the sampling circuits (usually flip-flops). Additionally, long delay chains allow noise to accumulate in the delay line. In [33], the delay line approach was used for measuring on-chip timing uncertainty. In 130nm technology, the

[1] Integral non-linearity (INL) is the maximum deviation of the TDC transfer characteristics from a straight line that goes from the smallest through the largest codes. INL is a measure of the accuracy of the TDC measurement characteristics.

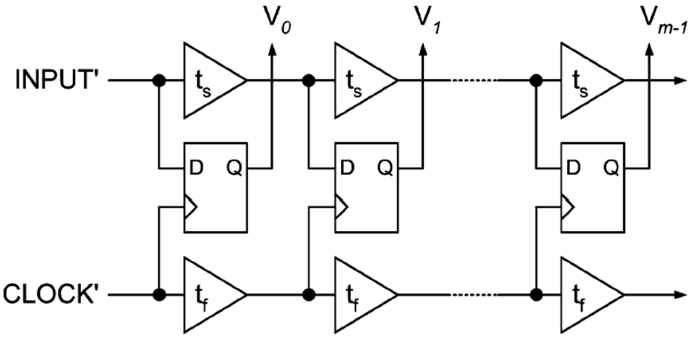


Figure 5.1: TDC based on the Vernier delay line [17].

design achieves an LSB of 8ps and an accuracy of 9ps. In another design, a delay resolution of 20ps with an INL of 0.7 LSB was demonstrated in 90nm CMOS technology using inverters as the delay cell [38]. Because time measurement in the delay line based techniques does not require any processing of the signals (e.g counting), they are sometimes referred to as “flash” TDCs.

2. **The Vernier Delay Line (VDL).** The VDL technique improves the resolution of the basic delay line by using two delay lines in a differential arrangement. One delay line is designed with faster delay elements than those used in the other delay line. This arrangement is illustrated in Figure 5.1. Because of the differential nature of the delays, the impact of mismatch is higher than in the case of a single delay line. The VDL has similar conversion time and noise accumulation characteristics. Using VCDEs, references [15] and [1] demonstrated resolutions as low as 25ps and 18ps in 1.2μm and 0.35μm technologies, respectively. Using two DLLs to stabilize the delays of the VDL was demonstrated in [13, 17].

3. **Sampling Offset TDC (SOTDC).** The SOTDC eliminates the delay lines used for the flash TDCs and relies on exploiting the random variations in the sampling offset of the flip-flops

to provide high-resolution delay measurement [16, 24]. Although the SOTDC is capable of achieving very high resolution, this resolution can not be predicted at design time and has to be estimated through calibration. Additionally, because of the limited offset due to process variations, the dynamic range of the SOTDC is limited and also can not be predicted at design time.

4. **"The Vernier" Technique.** According to reference [20], the Vernier technique for time measurement (which should be differentiated from the VDL, although the two have many similarities) is the first fully digital time measurement technique. It was proposed in [4] which presented an implementation using discrete components. Implementations in bipolar and CMOS technologies were presented in [28] and [7], respectively. As illustrated in Figure 5.2, the technique is based on expanding the time difference between the START and STOP signals. Two triggered ring oscillators with slightly different frequencies (which determines the resolution of the TDC) and two counters are used to measure the time difference between the two inputs. References [7, 28] used only one counter with a phase detector. The Vernier technique effectively eliminates the problem of mismatch between delay elements; only one delay element per ring oscillator is used. It offers attractive area and power consumption characteristics. However, these advantage come at the expense of increased measurement time. In order to reduce conversion time, reference [7] suggested using a combination of the delay line and the Vernier techniques to form an array of converters, which offsets the area and power advantage of the Vernier technique. As indicated in [7], although the mismatch problem is eliminated, noise in the Vernier-based TDC is still additive.

5. **Pulse Shrinking (Stretching) TDC.** In this technique, the width of a pulse that represents the time difference between the two inputs is adjusted by feeding the pulse through a set

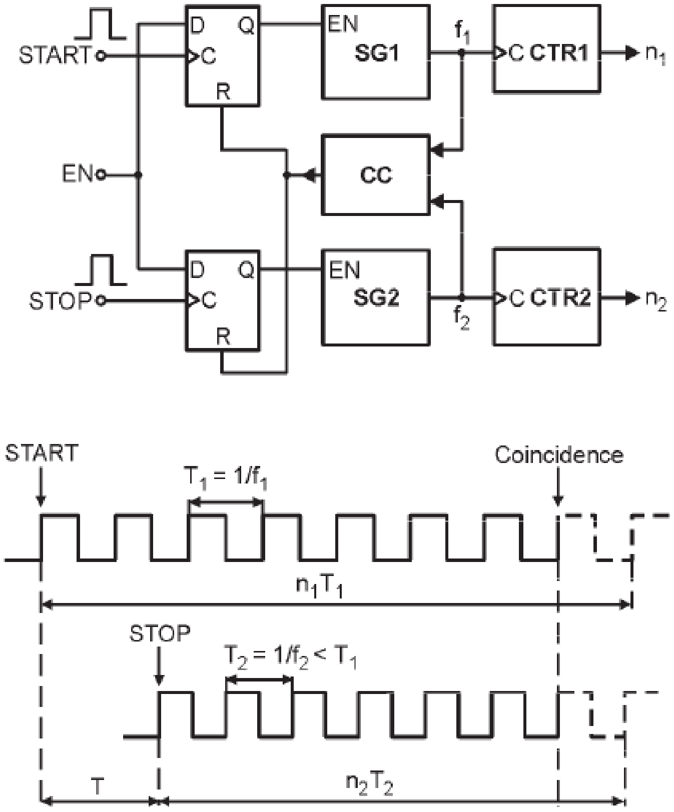


Figure 5.2: Circuit and operation example for time measurement using the Vernier technique [20].

of delay elements with imbalanced rise and fall time [31, 32]. The number of elements needed to cause the pulse of vanish is proportional to the original width of the pulse. In a 0.5μm technology, the pulse shrinking technique was used to achieve a resolution of 50ps [21] using a DLL for calibration. Similar to the VDL, the main drawbacks of the pulse shrinking technique are long conversion times [38], noise accumulation [31], and non-linearity due to mismatch [9]. In order to reduce the non-linearity, a cyclic version of the pulse shrinking TDC has been proposed [9, 41]

6. **TDC with Hierarchical DLLs.** Reference [25] suggested the use of hierarchical DLLs in order to achieve high measurement resolution with a small number of delay elements. In 0.8μm technology and using a two level interpolator, a 92ps LSB was achieved using 32 delay elements and a reference clock frequency of 85MHz. This design is illustrated in Figure 5.3. The second level of interpolation uses parallel delay lines that use shunt capacitors to adjust the delay of the individual delay lines. A replica-based calibration is provided using DLL2. Using multi-level interpolation, resolutions of 30 and 12.2ps were achieved in 0.6 and 0.35μm technologies [18, 26].

7. **Interpolation using weighted averaging of delays.** This technique is an extension to the delay line with DLL technique. Sub-gate delays can be achieved by averaging the delays of two delay cells whose delays differ by one gate delay. In a 25GHz f_T silicon bipolar process, this approach achieved an LSB of 15.625ps [22]. References [23, 44] used digital calibration of the weights in order to achieve high precision and reduce non-linearity. Calibration codes were stored in SRAM cells. For example, the design presented in [23] allows for a delay tuning step of 1.6ps. The design also uses offset compensation for the sampling comparators. Sampling offset is further reduced using redundancy by eliminating the comparators with larger offsets. In 0.18μm CMOS technology, the design operates at 700MHz and achieves an 8ps resolution

From this review, it can be seen that many of the techniques proposed in the literature are capable of achieving high resolution time measurements. However, because of the target applications, most of the proposed architectures focus on achieving large dynamic range rather than high conversion rate. High conversion rates are essential for performing real-time on-chip timing measurements. Measuring on-chip skew requires both a high resolution and a high sample rate [34]. This is important in order to capture intermittent and sporadic timing failures. For measuring jitter using on-chip test techniques (e.g BIST jitter measurement), many techniques

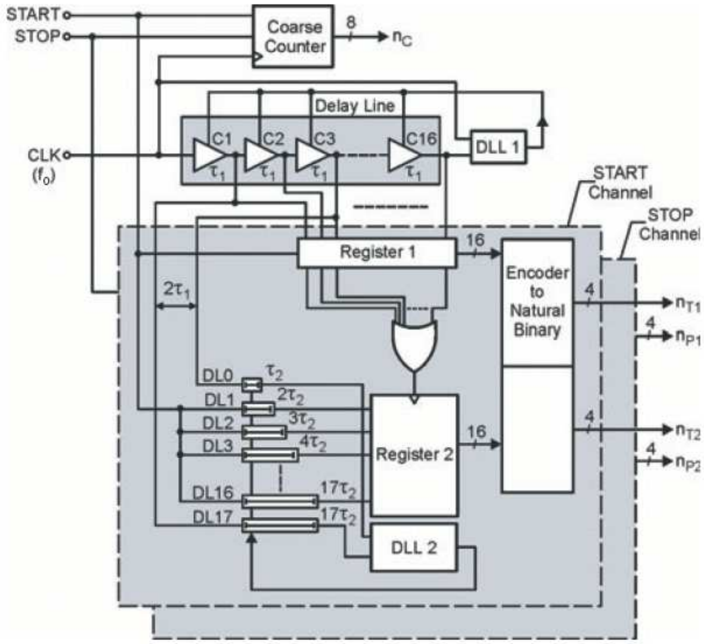


Figure 5.3: Block diagram of a TDC with two level interpolating DLLs [20].

use under-sampling with the assumption that, due to the random nature of jitter, such techniques can still give a correct estimate of jitter in the system. In a statistical analysis supported by experimental results, reference [6] showed that such techniques give an incorrectly optimistic measure of the actual jitter values.

5.2 Proposed TDC Architecture

The goal of this work is to implement a TDC architecture that would be capable of achieving very high conversion rate for real-time on-chip time measurement. When performing such measurements, it is possible for the measured delay to be either positive or negative (skew, e.g) and hence the TDC implementations should support both positive and negative delay measurements.

For implementing high speed analog-to-digital converters (ADCs), it is a common practice to use the flash architecture. For example, references [10] and [36] demonstrate flash ADCs with 1.3 and 1.6Gsample/s in 0.35μm and 0.18μm CMOS technologies, respectively. On the other hand, there is little published work on using the same flash architecture for high conversion rate TDCs. References [16, 24] suggested building a flash TDC that takes advantage of the offset of the sampling flip-flops. As discussed earlier, this design can operate at very high speeds but suffers from small and unpredictable dynamic range.

In a flash ADC, the level of the input signal is compared to a number of reference voltage levels resulting in a thermometer digital code. This code is then converted to a more compact digital representation such as binary or gray codes. The proposed architecture of the flash TDC follows this model of ADC design and is illustrated in Figure 5.4 for a 6-bit TDC. The TDC measures the delay of the input signal, INP, with respect to the reference signal, REF. These two inputs feed 63 delay stages (ST0–ST62) each consisting of two delay lines and an arbiter. The difference in delays applied to the two inputs and the corresponding delay thresholds that the stages can discriminate in multiples of LSB (delay resolution of the TDC) are as shown in the figure. For example, ST3 delays REF by 1 LSB and INP by 29 LSB. As a result, this stage causes the delay of INP seen at i2 of the arbiter A3 to be skewed by 28 LSBs with respect to the delay of REF seen at i1 of A3. If the original delay of INP with respect to REF is -28 LSB, the delayed versions would be seen at the arbiter as happening at the same time. This means that ST3 has a delay threshold of -28 LSB and its output (and the outputs of all arbiters to its left) would be HIGH if $t_{REF} - t_{INP}$ is less than -28 LSB. The outputs of all other arbiters would be LOW resulting in a thermometer code at the input of the latch and encoder block. After the latch stores the outputs of the arbiters, the code is then converter to gray or binary representation using the encoder.

There are some observation regarding this TDC architecture:

1. The delays of the signal paths are limited to the delay lines, the arbiter, and the latch. If

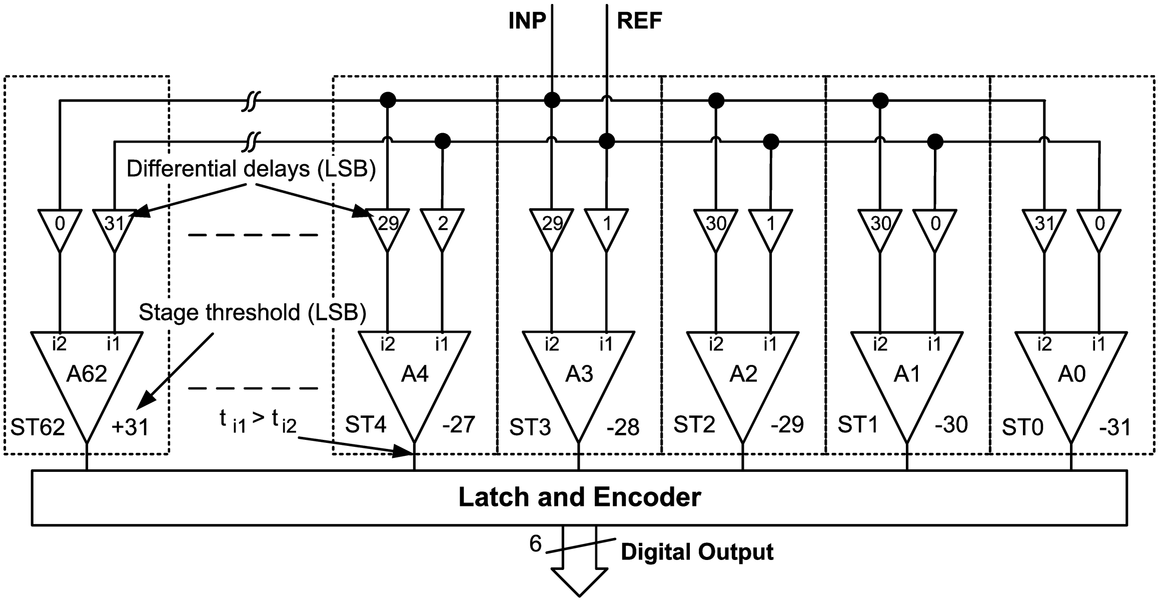


Figure 5.4: Proposed TDC architecture.

these blocks are designed with small delays, this TDC can operate at a high conversion rate.

2. The two input signals drive a number of stages equal to 2^n, where n in the number of bits in the output. Drivers might be needed in order to maintain an adequate rise and fall times of these two inputs.

3. Skew between the two inputs can cause a static error in the measured delay. Additionally, skew between the inputs to the different stages can introduce non-linearity and affect measurement precision.

4. Because the inputs propagate through the stages in parallel, noise does not accumulate. If the stages are designed to have small delays, noise impact on measurement precision can be negligibly small.

5. Due to the open loop nature of the design, PVT variations can cause variation in the delay resolution of the TDC. Consequently, characterization would be essential. Alternatively, a replica-based calibration of the delay step can be used if the delay lines support delay step adjustment.

6. Mismatch can cause the stage delays to deviate from the ideal values and the arbiters to have sampling offsets. This results in nonlinearity in the TDC characteristics which affect its precision. For real-time measurements, it is essential that the precision of the TDC be ≤ LSB [20]. If that is not the case, calibration techniques can be used to adjust the delays and compensate for the nonlinearity.

7. As shown in Figure 5.4, for each of the REF and INP signals, two stages delay them by equal delays. For example, both of stages ST2 and ST3 delay REF by a differential delay of 1 LSB. Similarly, both of ST3 and ST4 delay INP by 29 LSBs. This means that the number of delay lines can be reduced by half if each of them is shared between two stages in the TDC. The only exceptions to this are the 31 LSBs delay lines of both REF and INP (in ST0 and ST62, respectively).

A 6-bit TDC that uses the architecture in Figure 5.4 has been implemented in 0.18μm CMOS technology. The main goal of this experiment is to demonstrate the high conversion rate possible through this implementation. Although, a very small LSB has been targeted, no calibration features were included in this preliminary experiment. However, the design is made such that calibration features can be easily included in future experiments.

5.3 Proposed TDC Implementation

Figure 5.5 illustrates the TDC implementation. It uses 64 delay lines to adjust the delays of the REF and INP signals. In order to reduce the number of delay lines used, each of the delay lines feeds two latched arbiters (LAs). The delay lines are implemented using the DCDE proposed in chapter 3. Although the digital input codes for DCDEs are hard-wired in the design, using this DCDE allows the design to be easily modified in order to allow for calibration in future experiments. Modifying the DCDE to the CC-DCDE version allows the delay step to be adjusted using replica-based calibration. Additionally, adjusting the input codes of the DCDEs allows for linearity correction of the TDC characteristics.

The delay step of the DCDEs is chosen to be 2.5ps (the TDC's LSB). For this 6-bit TDC implementation, the delay lines needs to be able to provide 31 uniform delay steps (0-31 LSBs as shown in Figure 5.5). Uniformity of the delay steps is important in order to maintain the linearity of the TDC transfer characteristics (under ideal conditions). Because this range is much larger than what can be achieved using the VCDE used in chapter 3, every delay line uses two DCDEs, one with 16 delay steps and the other with 15 resulting in a 31-bit thermometer code. When this code is set to all 0s, the delay line gives the smallest delay which corresponds to a differential delay of 0 LSB. This setting is used in DL1 and DL62 as shown in Figure 5.5. On the other hand, when the code is set to all 1s, the delay of the delay line is at its maximum which corresponds to a differential delay of 31 LSBs. This setting is used for DL0 and DL63. Between these two extremes, the codes of the delay lines are set in order to give differential delays at the inputs to the arbiters that range from -31 to +31 LSBs resulting in a TDC measurement range of ±77.5ps. As each of the REF and INP signals drive 32 delay lines, a four inverter buffer is used for each of them. The four inverters are sized to achieve minimum delay with each inverter 3-4 times the one at its input [30]. The use of these drivers has the potential of introducing skew between the two inputs and between the same input as seen at the input of the different delay

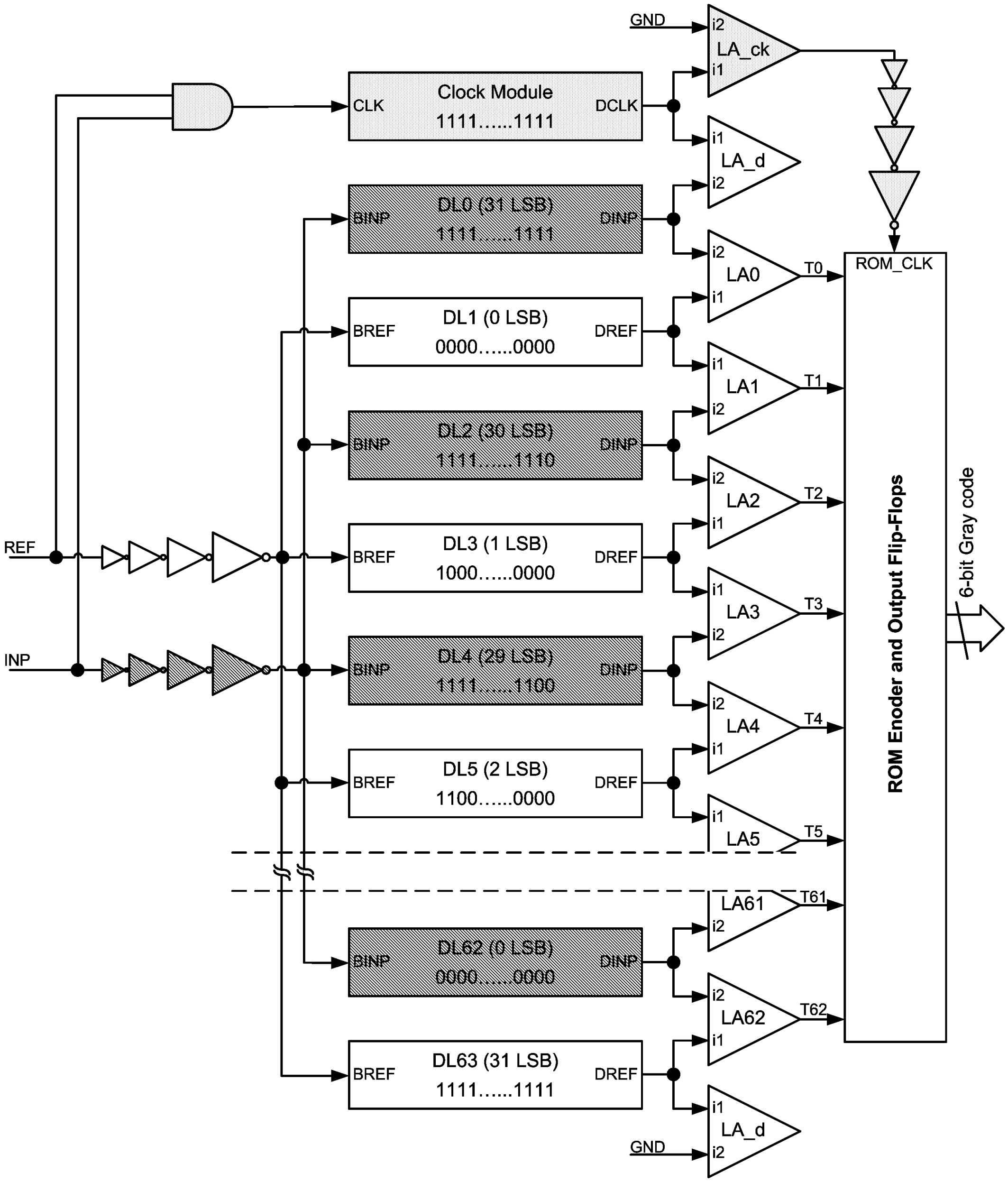


Figure 5.5: TDC implementation.

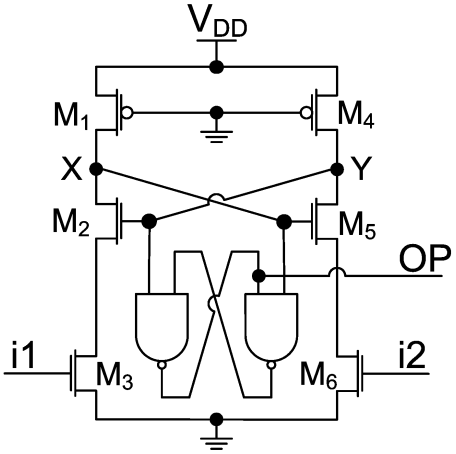


Figure 5.6: Schematic of the latched arbiter.

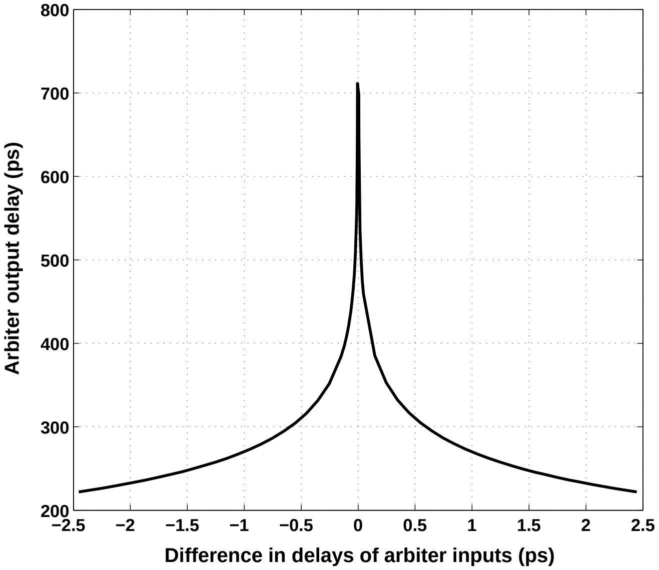


Figure 5.7: Metastability characteristics of the latched arbiter under *typical* conditions.

lines. Careful layout of these drivers is important in order to minimize the skew.

The latched arbiter used in this design is built using a CMOS arbiter [16] followed by an SR latch (two cross-coupled NAND gates). The schematic diagram of the arbiter is shown in Figure 5.6. When both inputs i1 and i2 are LOW, nodes X and Y are pre-charged to HIGH using transistors M_1 and M_4. This allows the SR latch to keep the value it has previously stored. It also causes transistors M_2 and M_5 to be ON and the arbiter to be ready to receive the input transitions. Depending on which of the two inputs switch to HIGH earlier, one of nodes X and Y switches to LOW forcing the NAND gate on the opposite side to have a HIGH output and the other to have a LOW output. For example, if i2 switches from LOW to HIGH earlier than i1, transistor M_6 starts to turn ON before M_3 and node Y starts to discharge before node X. The earlier discharging of node Y causes M_2 to turn OFF which stops the discharging of node X and allows M_1 to restore the voltage level at this node to HIGH while Y continues to discharge until it become LOW. A HIGH on X and a LOW on Y force the SR latch's output, OP (which is also the LA's output), to be HIGH. Due to metastability, the delay of the LA depends on the difference in the delay between i1 and i2. The smaller the delay, the more time it takes for the LA to produce a valid output. This is illustrated in Figure 5.7, showing the delay of the arbiter's output as the input delay changes from -1 LSB to +1 LSB (-2.5ps to +2.5ps). In order to capture the correct output from the arbiter, enough time need to be allowed for its evaluation. This, however, can significantly reduce the conversion rate of the TDC. A good tradeoff is to allow enough time that would result in correct output if the delay is larger than a small fraction of the LSB.

The TDC contains 63 arbiters (LA0-LA63) whose outputs T62.....T0 constitute the thermometer code representation of the measured delay. Two dummy arbiters (LA_d) are used to balance the load at the outputs of DL0 and DL63. A read-only memory (ROM) encoder is used to convert the thermometer code to 6-bit gray code. The encoder uses a three input NAND gate for single bubble error removal [19]. In order to achieve high speed operation, the ROM encoder is designed using dynamic CMOS logic. Flip-flops are used to store the outputs of the encoder and a clock module is used to generate the clock signal, ROM_CLK used for the ROM and the output flip-flops. The clock is generated internally using the inputs signals REF and INP. This self-timing approach allows the TDC to operate at very high conversion rates. To achieve the smallest conversion time, the ROM_CLK transition should happen as soon as all bits in the thermometer code have settled to the values corresponding to the time to be measured. Due to the potential increase in delay due to metastability, the timing of ROM_CLK should be such that some margin is allowed for the additional delay. Because the delay lines have different delays, the timing of the ROM_CLK need to take into account the worst case delay from inputs to thermometer code.

The clock is generated by ANDing the two input signals resulting in a LOW-to-HIGH transition that follows the latest of the two inputs. This transition feeds the clock module which is identical to the delay lines used in the TDC and has all of its digital inputs set to 1s. This causes the clock module to have a delay that is equal to the largest delay line delay (equal to the delay of DL0 and DL63). The output from the clock module feeds an arbiter and a four inverter buffer whose delay is equal to the buffer used for REF and INP. Using this arrangement, the total of the clock module, arbiter, and clock buffer delays would be equal to the largest delay that would be experienced by the latest of the two inputs REF and INP (excluding additional delays due to metastability). The delay of the AND gate is used to set the delay margin allowed for metastable behavior. In this design, the AND gate delay is set to a nominal value of 300ps. As shown in Figure 5.7, this value allows enough time for the LA to resolve a minimum input delay difference of ±0.5ps (1/5 LSB).

Extensive simulations of the proposed TDC have been performed in order to verify its functionality at different operating conditions. Using the proposed architecture, a 6-bit TDC design in 0.18μm CMOS technology is capable of achieving a conversion rate as high as 1.5GSa/s under *typical* conditions. It is expected, however, that this rate would decrease in actual implementation due to supply noise, skew, jitter, and other source of uncertainty. Device noise simulations show that the maximum rms jitter of the TDC delay lines is 0.24ps.

5.4 Chip Implementation and Test Setup

This section describes the implementation of the proposed 6-bit TDC on a test chip in 0.18μm CMOS technology. The block diagram of the test chip is shown in Figure 5.8. In order to allow for high-speed inputs, an input buffer is used to generate the inputs of the TDC from a small signal sinusoidal input, ACINP. This small signal input is terminated using an on-chip 50Ω resistor connected to the input of a CMOS buffer (two back-to-back inverters). If the DC level of ACINP is set to the threshold level of the first inverter, the buffer converts the small signal input

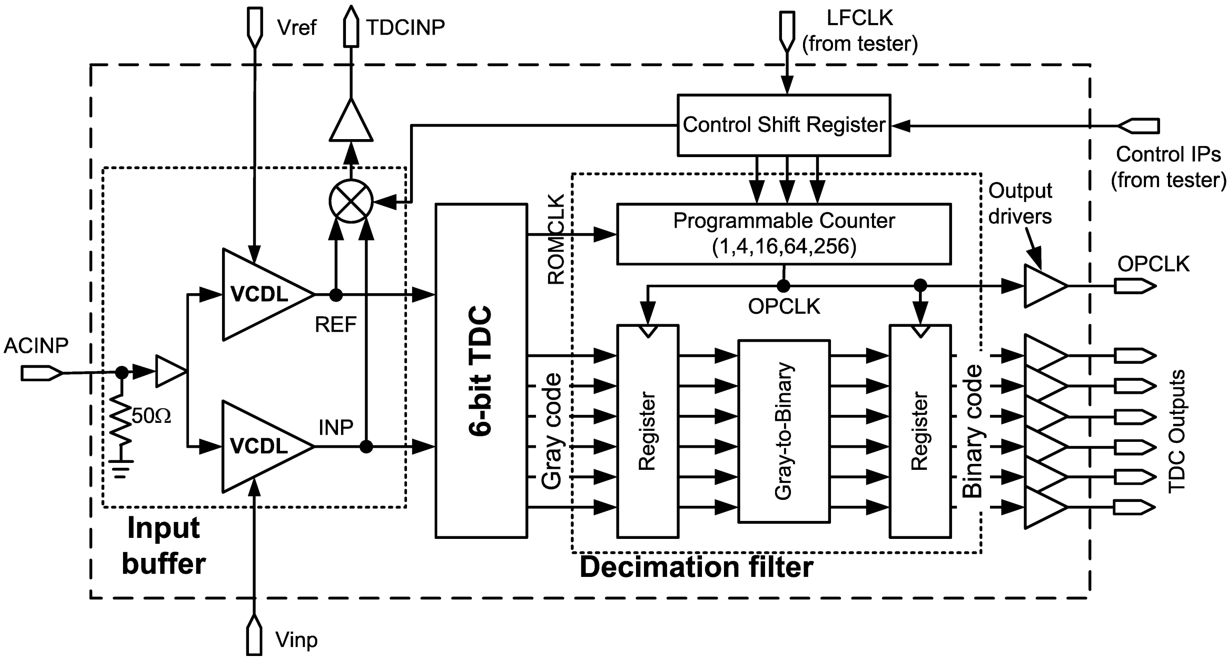


Figure 5.8: Block diagram of the TDC test chip.

into a high-speed digital signal. Two voltage-controlled delay-lines (VCDLs) are used to create the TDC inputs, REF and INP, from this high speed signal. The input delay is adjusted by setting the values of the VCDLs control voltages, V_{ref} and V_{inp}, off-chip. This arrangement effectively eliminates the impact of the jitter on the small signal input.

In order to characterize the TDC performance, a mechanism for measuring the input delay has to be provided. In order to achieve that, the TDC inputs, REF and INP, are multiplexed and driven off-chip using an output driver. Multiplexing the two signals ensures that the path followed by both of them through the output driver, bond-wire, package pin, PCB trace, and scope probe is the same. Such arrangement eliminates many source of measurement uncertainty but requires storing one of the two signals and then observing the other on the scope.

Because of the limitations of the speed of the logic analyzer used for testing the TDC, a decimation filter is used to reduce the rate of the TDC outputs. The filter consists of a programmable

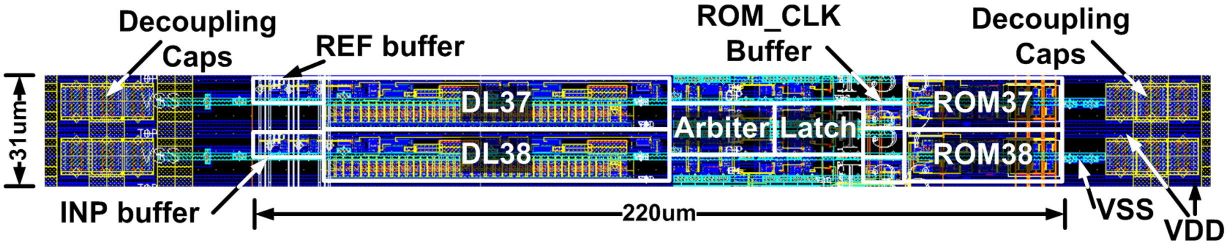


Figure 5.9: TDC row layout for rows 37 and 38.

counter and two registers. The counter is capable of dividing the ROM_CLK (created internally within the TDC and used for the ROM in Figure 5.5) by 4, 16, 64, or 256. It can also provide full-rate outputs if necessary. The first of the two registers is used to sample the TDC outputs at the rate of the counter's output clock, OPCLK. A gray-to-binary conversion logic is used to convert the sampled outputs into binary code which is then stored using the second register. This register feeds the output drivers that drive the TDC outputs off-chip. OPCLK is also driven off-chip and is used to verify the correct operation of the counter and as a trigger input to the logic analyzer used to monitor the TDC outputs.

It is important that care is taken in doing the layout of the TDC in order to minimize the potential impact of skew, supply noise, and crosstalk on its operation. The layout is done in 64 identical rows based on the diagram in Figure 5.5. The area of each row is $220\times15.5\mu\text{m}^2$. The layout of rows 37 and 38 is shown in Figure 5.9. In order to minimize the skew of the REF and INP signals, their buffers are implemented using distributed inverters. The last three inverters in these buffers are distributed over the 32 stages fed by each of the two signals. The outputs of these inverters are shorted together in order to minimize skew. This requires having long metal lines that extend over the height of the TDC (approximately 1mm). In order to minimize coupling with other metal layers, these metal lines are implemented using the next-to-top metal layer. Additionally, the metal lines carrying the buffered REF and INP signals are isolated using GND lines in order to minimize coupling between the two signals. The layout is done such that the two signals see exactly the same lengths of metal lines in all buffer levels.

As shown in Figure 5.9, the latched-arbiter is placed halfway between each two adjacent delay lines. The clock buffer for ROM_CLK is also implemented using distributed inverters spread across the 64 TDC rows. Each row has one VSS line routed halfway through the row and two VDD lines routed at the top and bottom sides of the row. The adjacent VDD lines from neighboring rows are not connected together across the length of the row. The connection is made at the two ends of the rows where the VDD and VSS lines are connected to decoupling capacitors forming a VDD/VSS grid. This arrangement ensures minimal interaction between supply and ground lines of adjacent rows and minimizes the potential of supply noise coupling.

The TDC chip micrograph is shown in Figure 5.10. The total chip area is 3.75mm^2. The chip is pad-limited with the TDC occupying an area of 0.23mm^2. Separate VDD and VSS pins have been used for unrelated parts of the test chip. Each of the TDC, the decimation filter, and the output drivers have their own supply and ground pins. This is important in order to prevent the supply noise in the decimation filter and the output drivers (which are included for test purposes only) from propagating to the TDC. Additionally, it allows the current drawn by the TDC to be measured independently. Due to the inductance of the bond wires and the package, the chip includes a large amount of decoupling capacitors. The decoupling capacitors allow the supply grid to have a faster response to sudden changes in the TDC current and help reduce the supply noise.

The PCB used for mounting the TDC chip has been designed with considerations for high-speed operation. The PCB has four layers with the two internal layers used as supply and ground planes. This provides proper return paths for high-speed currents and also provides additional decoupling for high-speed supply noise. The supply and ground planes were split into a number of sections according to the division of chip supply and ground pins. The different sections of the ground plane were connected using ferrite beads providing isolation for high-frequency noise.

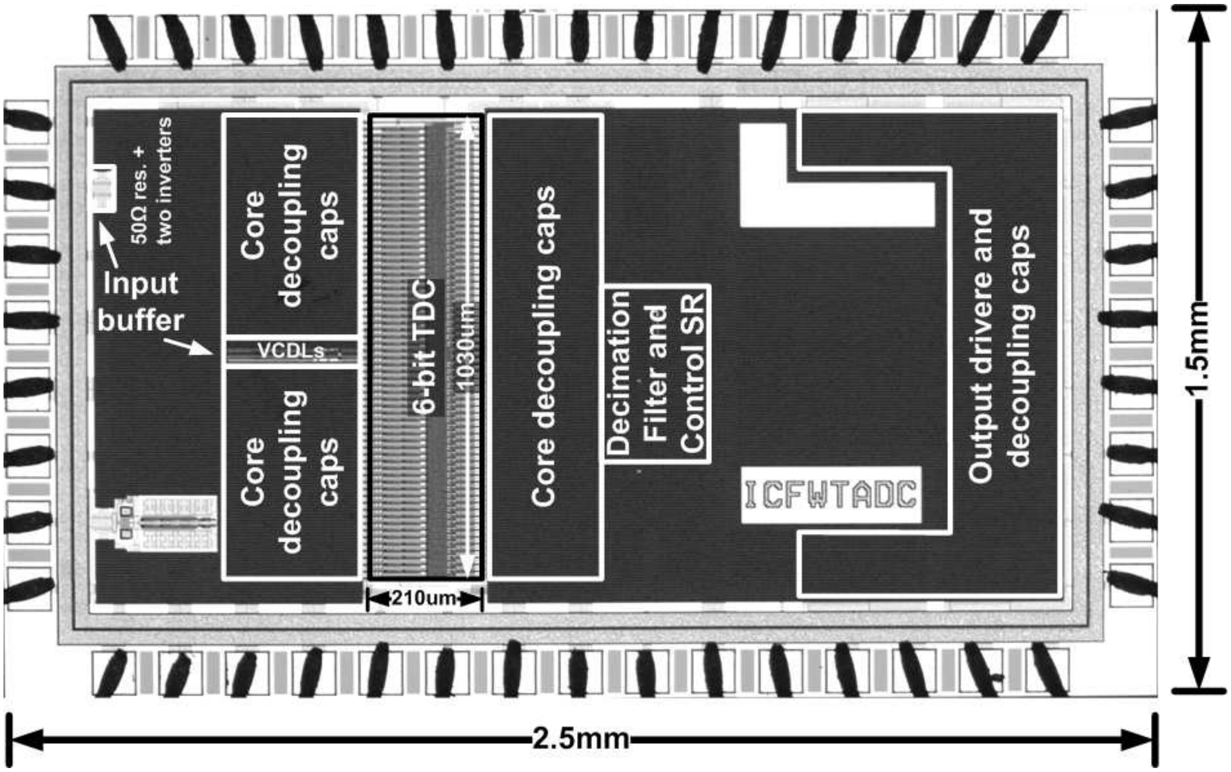


Figure 5.10: TDC chip micrograph.

Separate DC sources were used for the different supply plane sections. In order to minimize reflection, the PCB traces used for the high-speed input, ACINP, were designed to have a 50Ω characteristic impedance.

Figure 5.11 shows the test setup used for the TDC chip. An R&S SMT03 signal generator generates the high frequency input signal. This signal is biased to the logic threshold of the input buffer (approximately VDD/2) using a bias-T. V_{ref} and V_{inp} (Figure 5.8) are generated on-board using a variable resistor operating as a voltage divider. A noise, ramp, or low frequency sinusoid can be added to V_{inp} using the AC/Noise functions of the SMT03 signal generator. A VXI test system (digital tester) is used to supply the low speed control inputs for the chip. An Agilent Infiniium 54853A digital storage scope is used to observe the (multiplexed) inputs of the TDC

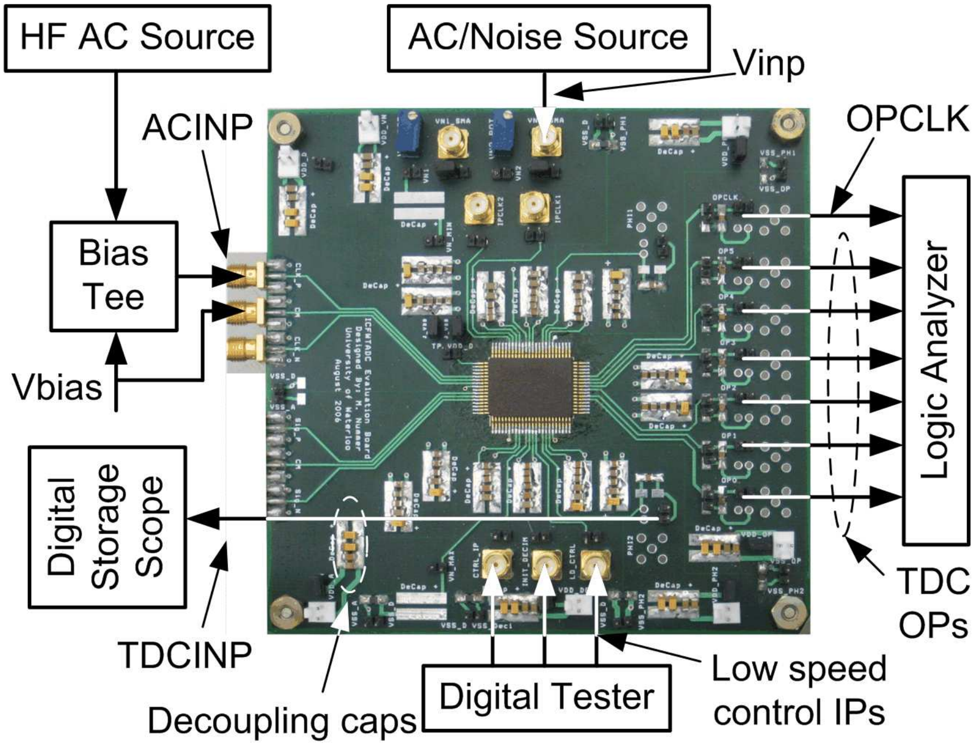


Figure 5.11: TDC board and test setup.

while an Agilent 1680AD logic analyzer monitors OPCLK as well as the digital outputs.

5.5 Experimental Results

The TDC chip has been tested in order to estimate its performance characteristics. The testing includes estimating the actual LSB of the TDC and its INL at different operating frequencies.

Estimating the average LSB is based on finding the delay range that corresponds to full range TDC output codes. This is done by adjusting the delays of the VCDLs at the TDC inputs and finding the largest (negative) INP delay (with respect to REF) that results in all output codes being equal to 0s and the smallest (positive) INP delay that results in all output codes being 63.

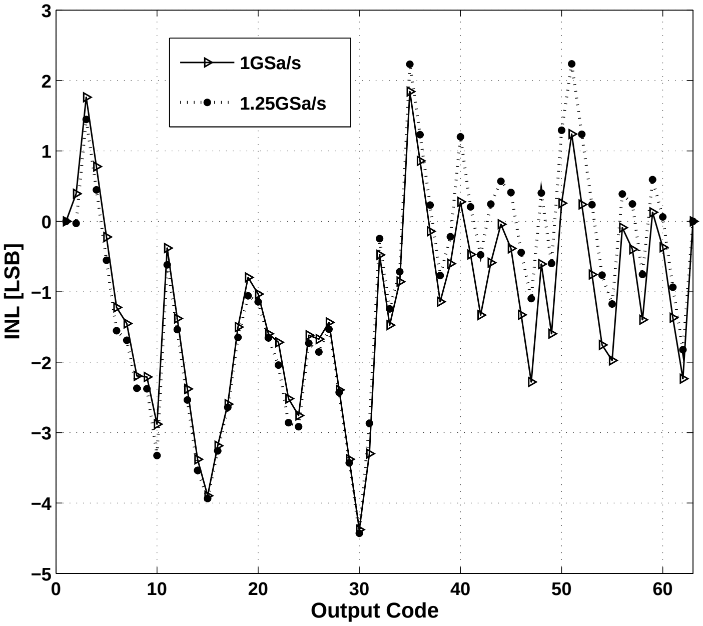


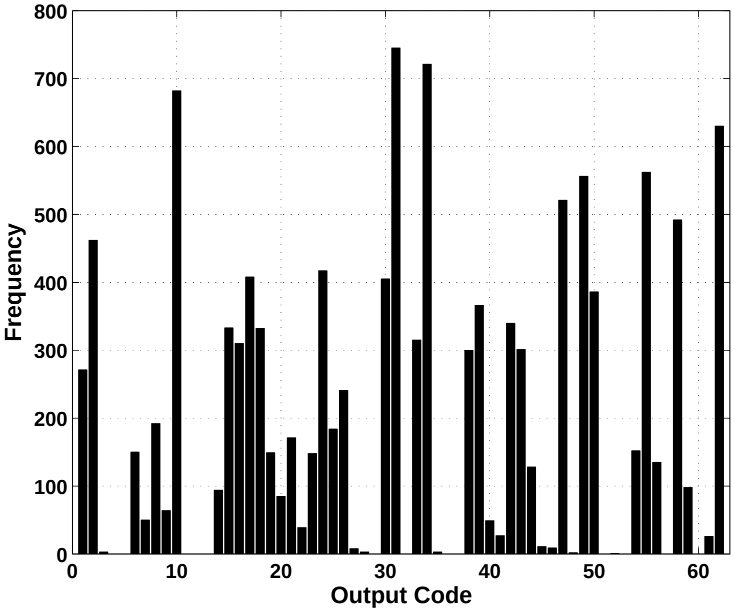
Figure 5.12: Measured INL of the TDC at 1GSa/s and 1.25GSa/s.

The difference between these two delays is the full range delay and corresponds to 62 LSBs (64 - the two over-range bins, 0 and 63). Measurements show that the full range delay is 180ps which corresponds to an LSB of 2.9ps. This result indicates that the chip operates on a slower than typical process split. The average output code for zero input delay is 30.887. This translates to a TDC delay measurement range between -89.6 and +90.4ps.

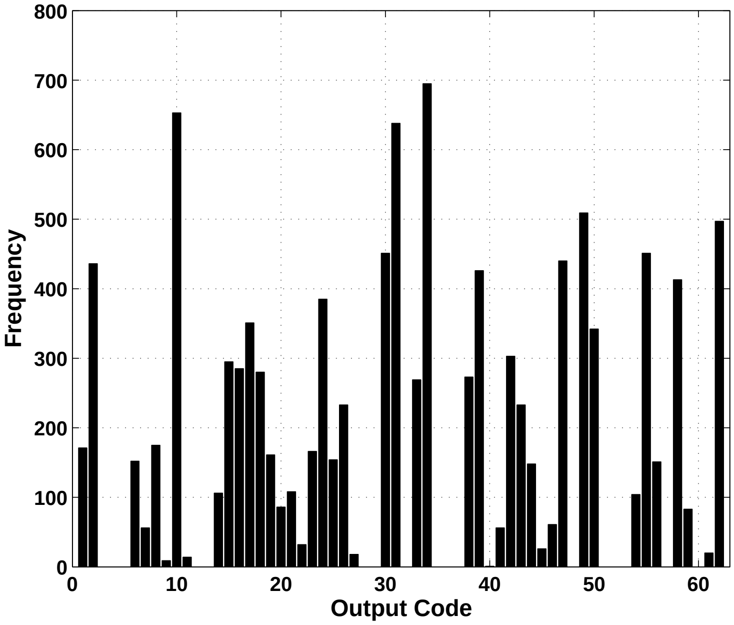
In order to estimate the accuracy of delay measurements, the INL of the TDC is measured using code density test with ramp input. Given that the VCDLs at the inputs of the TDC have a region that exhibits a high degree of linearity, the ramp input is set to operate in that region. The slope of the ramp is set to produce an average of 200 samples per bin with a decimation ratio of 64 (corresponding to 12,800 samples per bin at full rate). Figure 5.12 shows the measured INL of the TDC at sampling frequencies of 1GSa/s and 1.25GSa/s. For both frequencies the maximum INL is -4.43 LSB. The figure also shows that the INL increases slightly when going from 1GSa/s to 1.25GSa/s. When the input frequency increases to 1.3GSa/s, the TDC starts to fail by producing the output code 42 in unexpected locations of the output sequence. In gray coding, code 42 is represented by all 1s, indicating that the clock input to the ROM goes into evaluation too early before the outputs of some of the arbiters are produced. The fact that this happens for some codes and not the others (at 1.3GSa/s) indicates that the arbiters responsible for generating the output code are experiencing increased delays due to metastability. From these tests, it is concluded that the maximum conversion rate of the TDC is 1.25GSa/s. It should be noted that this result is in agreement with the previous conclusion that the tested TDC chip operates on a slower-than-typical process split. At 1.8V supply, the TDC consumes 85mA at 1GSa/s and 110mA at 1.25GSa/s, out of which 30mA are consumed as static current by the DVC blocks of the DCDEs. This leaves much room for power reduction if the sizes of the transistors in the DVC blocks are adjusted for that purpose.

The large negative INL indicates that the TDC has a number of consecutive missing codes in its transfer characteristics. This is illustrated by the code density histograms in Figure 5.13. The 1GSa/s histogram shows that there is a set of 13 codes that never appear in the output of the TDC. Additionally, another set of codes appear in the output but with a very low density. These missing and “weak” codes are replaced by other “dominant” codes. The code density histogram for a conversion rate of the 1.25GSa/s shows a similar pattern, however at this higher rate some of the weak codes never appear at the TDC output, giving a total of 17 missing codes.

In order to understand the source of the high INL, a number of possibilities were explored. The fact that the number of missing codes increased when going from 1GSa/s to 1.25GSa/s suggests that the non-linearity could be related to the operating frequency of the TDC. For that reason, the TDC was tested at a number of different frequencies down to 10MHz. No significant differences in the INL or the number of missing codes were observed. When operating at 1.25GSa/s, the TDC is effectively being pushed towards its performance limit. Under such



(a)



(b)

Figure 5.13: Code density histograms for the TDC at (a) 1GSa/s and (b) 1.25GSa/s.

circumstances, an increase in the TDC's non-linearity is expected and the disappearance of the "weak" codes is one possible manifestation of that. This analysis rules out the possibility of a relationship between the missing codes and the operating frequency. Similar testing indicated that the input supply voltage can not be the source of the high non-linearity. Another possibility is the presences of systematic errors in the design or the layout of the TDC or the presence of localized supply noise or crosstalk in some areas of the design. Testing four different chips showed that all of them experience the same general pattern (some missing codes, some weak codes, and some dominant codes). However, none of the four test chips had the exact same code density characteristics.

This discussion leaves only the possibility of device mismatch which is a common source of non-linearity in ADCs and TDCs. In order to verify this hypothesis, a behavioral model of the TDC was developed. The model is based on the work presented in [24, 34] and incorporates the random variations in the DCDE delays as well as the offsets of the arbiters. These estimates were obtained using Monte Carlo mismatch analysis[2]. The model also incorporates an estimate of the TDC jitter that was obtained using the methodology presented in [29]. The standard deviation of the DCDE delays was estimated to range from 2ps for the DCDE with the smallest delay up to 4ps for the DCDE with the largest delay. The arbiter offset was estimated to have a standard deviation of 1.67ps and the jitter was estimated to have an rms value of 0.24ps. Figure 5.14 shows an example code density histogram created using this model. Clearly, the histogram resembles those obtained from measurements which verifies that mismatch between the delay elements as well as arbiter offsets are the main sources of the high INL for this TDC. This example has 19 missing codes and an INL of -5.1 LSB.

In order to demonstrate using the TDC for delay measurements, it has been used to measure gaussian and sinusoidal jitter and the measurement results were compared to those obtained

[2] The models used for this analysis were not available at the time of designing the TDC.

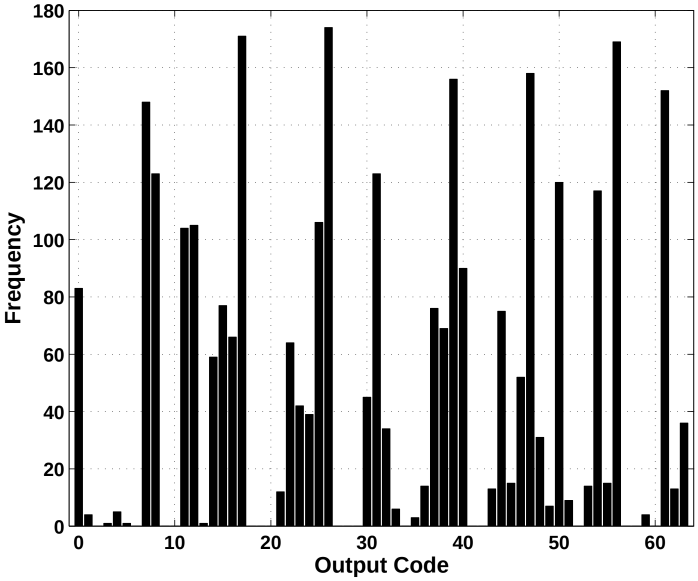


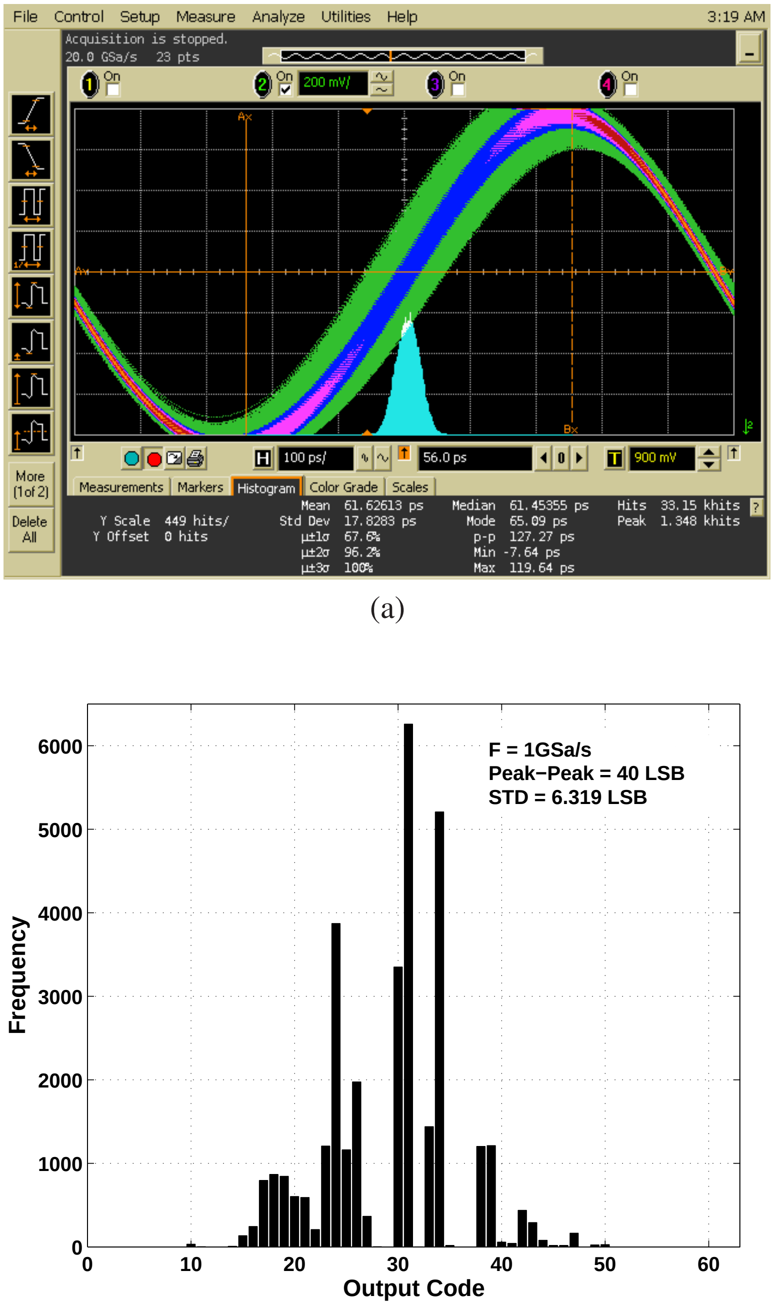
Figure 5.14: Example code density histogram obtained using TDC model.

from the scope. Jitter was induced by connecting a noise or sinusoidal signal to Vinp while keeping Vref at a constant DC voltage. Figure 5.15 shows these results for the noise input and Figure 5.16 shows the results for the sinusoidal input. As shown in Figure 5.15, for the noise input, scope measurements indicate peak-peak and rms jitter values of 127 and 17.83ps. For the TDC, these values are measured at 1GSa/s to be 40 and 6.32LSB which are equivalent to 116 and 18.33ps (LSB is 2.9ps), respectively. For sinusoidal jitter, results shown in Figure 5.16 indicate peak-peak and rms jitter values of 136 and 46ps using the scope compared to 125 (43LSB) and 43.36ps (15.3LSB) using the TDC at 1GSa/s.

5.6 Conclusions

In this chapter, a high-conversion rate TDC for real-time on-chip delay measurement has been proposed. The design is based on the true flash architecture commonly used for ADCs. Experimental results demonstrate a conversion rate as high as 1.25GSa/s in 0.18μm CMOS technology.

Although, the measured LSB of the design is 2.9ps, accuracy is limited by the mismatch between the different signal paths in the design as well comparator offsets. Improving the design through calibration techniques has the potential for improving the accuracy of the design while maintaining the high-conversion rate advantage.



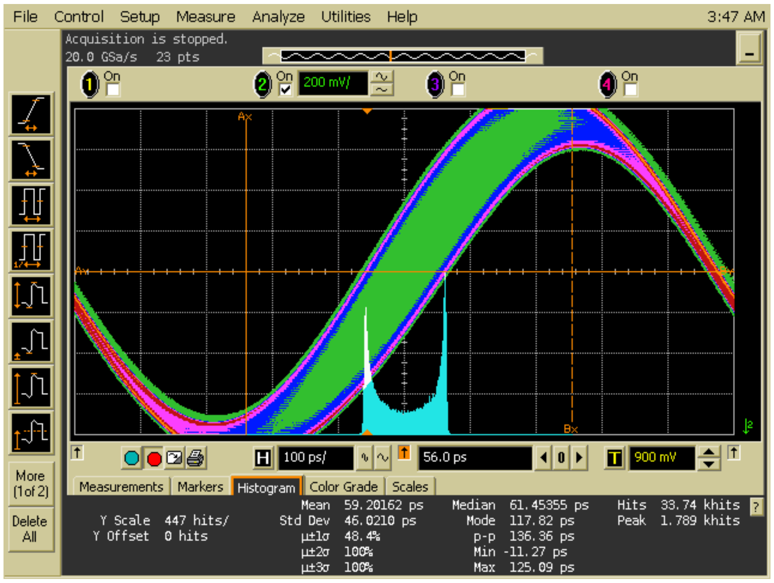
```
Acquisition is stopped.
20.0 GSa/s   23 pts

                         Mean     61.62613 ps    Median  61.45355 ps   Hits  33.15 khits
  Y Scale   449 hits/    Std Dev  17.8283 ps     Mode    65.09 ps      Peak  1.348 khits
  Y Offset  0 hits          µ±1σ  67.6%           p-p    127.27 ps
                            µ±2σ  96.2%           Min    -7.64 ps
                            µ±3σ  100%            Max    119.64 ps
```

(a)

(b)

Figure 5.15: Gaussian jitter measurement using (a) the scope and (b) the TDC running at 1GSa/s.

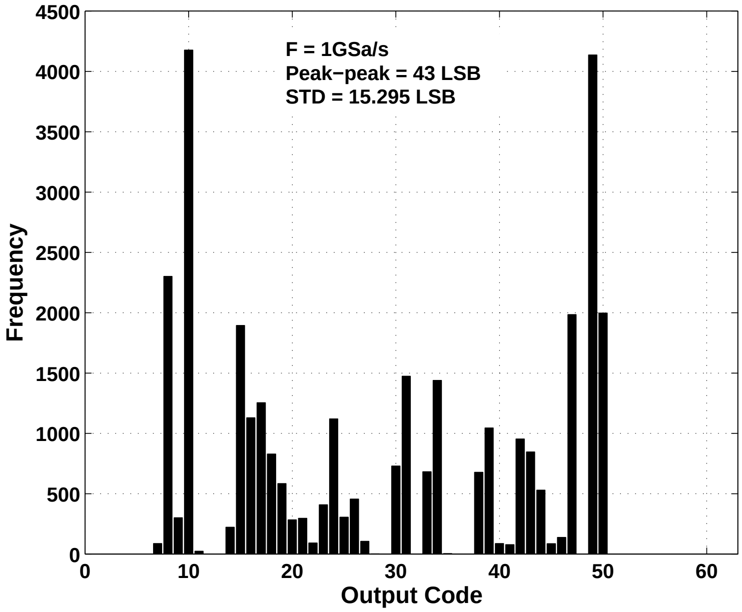


```
Acquisition is stopped.
20.0 GSa/s   23 pts
```

```
100 ps/        56.0 ps        0        900 mV
```

```
                          Mean   59.20162 ps    Median  61.45355 ps     Hits  33.74 khits
  Y Scale   447 hits/  Std Dev   46.0210 ps       Mode  117.82 ps       Peak  1.789 khits
 Y Offset   0 hits        μ±1σ   48.4%             p-p  136.36 ps
                          μ±2σ   100%              Min -11.27 ps
                          μ±3σ   100%              Max  125.09 ps
```

(a)



(b)

Figure 5.16: Sinusoidal jitter measurement using (a) the scope and (b) the TDC running at 1GSa/s.

References

[1] N. Abaskharoun, M. Hafed, and G.W. Roberts. Strategies for On-Chip Sub-Nanosecond Signal Capture and Timing Measurements. In *Proc. of the IEEE International Symposium on Circuits and Systems*, pages 174–177, May 2001.

[2] P. Andreani et al. Multihit Multichannel Time-to-Digital Converter with ±1% Differential Nonlinearity and Near Optimal Time Resolution. *IEEE Journal of Solid-State Circuits*, 33(4):650–656, April 1998.

[3] Y. Arai and M. Ikeno. A Time Digitizer CMOS Gate-Array with a 250 ps Time Resolution. *IEEE Journal of Solid-State Circuits*, 31(2):212–220, February 1996.

[4] J. Aveynier and R. van Zurk. Vernier Chronotron Reflex. *Nuclear Instruments and Methods*, 78:161–170, 1970.

[5] F. Baronti, L. Fanucci, D. Lunardini, R. Roncella, and R. Saletti. On the Differential Nonlinearity of Time-to-Digital Converters Based on Delay-Locked-Loop Delay Lines. *IEEE Transactions on Nuclear Science*, 48(6):2424–2431, December 2001. 9-40.

[6] D. Bordoley, H. Nguyen, and M. Soma. A Statistical Study of the Effectiveness of BIST Jitter Measurement Techniques. In *Proc. of the International Conference on Computer Aided Design*, pages 100–107, November 2005.

[7] A.H. Chan and G.W. Roberts. A Jitter Characterization System Using a Component-Invariant Vernier Delay Line. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 12(1):79–95, January 2004.

[8] P. Chen, C.-C. Chen, C.-C. Tsai, and W.-F. Lu. A Time-to-Digital-Converter-Based CMOS Smart Temperature Sensor. *IEEE Journal of Solid-State Circuits*, 40(8):1642–1648, August 2005.

[9] P. Chen and S.-I. Liu. A Cyclic CMOS Time-to-Digital Converter with Deep Subnanosecond Resolution. In *Proc. of the IEEE Custom Integrated Circuits Conference*, pages 605–608, May 1999.

[10] M. Choi and A.A. Abidi. A 6-b 1.3-Gsample/s A/D Converter in 0.35-μm CMOS. *IEEE Journal of Solid-State Circuits*, 36(12):1847–1858, December 2001.

[11] J. Christiansen. An Integrated CMOS 0.15 ns Digital Timing Generator for TDC's and Clock Distribution Systems. *IEEE Transactions on Nuclear Science*, 42(4):753–757, August 1995.

[12] C.C. Chung and C.Y. Lee. A New DLL-Based Approach for All-Digital Multiphase Clock Generation. *IEEE Journal of Solid-State Circuits*, 39(3):469–475, March 2004.

[13] P. Dudek, S. Szczepanski, and J.V. Hatfield. A High-Resolution CMOS Time-to-Digital Converter Utilizing a Vernier Delay Line. *IEEE Journal of Solid-State Circuits*, 35(2):240–247, February 2000. 9-46.

[14] E.J. Gerds et al. A CMOS Time to Digital Converter IC with 2 Level Analog CAM. *IEEE Journal of Solid-State Circuits*, 29(9):1068–1076, September 1994.

[15] C.T. Gray, W. Liu, W.A.M. Van Noije, Jr. Hughes, T.A., and III Cavin, R.K. A Sampling Technique and its CMOS Implementation with 1 Gb/s Bandwidth and 25 ps Resolution. *IEEE Journal of Solid-State Circuits*, 29(3):340–349, March 1994.

[16] V. Gutnik and A. Chandrakasan. On-Chip Picosecond Time Measurement. In *2000 Symposium on VLSI Circuits Digest of Technical Papers*, pages 52–53, June 2000.

[17] C.S. Hwang, P. Chen, and H.W. Tsao. A High-Precision Time-to-Digital Converter Using a Two-Level Conversion Scheme. *IEEE Transactions on Nuclear Science*, 51(4):1349–1352, August 2004.

[18] J.P. Jansson, A. Mantyniemi, and J Kostamovaara. A CMOS Time-to-Digital Converter with better than 10 ps Single-Shot Precision. *IEEE Journal of Solid-State Circuits*, 41(6):1286– 1296, June 2006. 9-13.

[19] D.A. Johns and K. Martin. *Analog Integrated Circuit Design*. John Wiley & Sons, Inc., 1997.

[20] J. Kalisz. Review of Methods for Time Interval Measurements with Picosecond Resolution. *Metrologia*, 41(1):17–32, February 2004.

[21] K. Karadamoglou, N.P. Paschalidis, E. Sarris, N. Stamatopoulos, G. Kottaras, and V. Paschalidis. An 11-bit High-Resolution and Adjustable-Range CMOS Time-to-Digital Converter for Space Science Instruments. *IEEE Journal of Solid-State Circuits*, 39(1):214–222, January 2004.

[22] T.A. Knotts, D. Chu, and J. Sommer. A 500 MHz Time Digitizer IC with 15.625ps Resolution. In *International Solid-State Circuits Conference Digest of Technical Papers*, pages 58–59, February 1994.

[23] L.-M. Lee, D. Weinlader, and C.-K. K. Yang. A Sub-10-ps Multiphase Sampling System Using Redundancy. *IEEE Journal of Solid-State Circuits*, 41(1):265–273, January 2006.

[24] P.M. Levine and G.W. Roberts. High-Resolution Flash Time-to-Digital Conversion and Calibration for System-on-Chip Testing. *IEE Proceedings - Computers and Digital Techniques*, 152(3):415–426, May 2005.

[25] A. Mántyniemi, T. Rahkonen, and J. Kostamovaara. An Integrated Digital CMOS Time-to-Digital Converter with Sub-Gate-Delay Resolution. *Analog Integrated Circuits and Signal Processing*, 22(1):61–70, January 1999. 9-55.

[26] A. Mántyniemi, T. Rahkonen, and J. Kostamovaara. An Integrated 9-Channel Time Digitizer with 30ps Resolution. In *International Solid-State Circuits Conference Digest of Technical Papers*, pages 266–267,465, February 2002.

[27] I. Nissinen, A. Mantyniemi, and J. Kostamovaara. A CMOS Time-to-Digital Converter based on a Ring Oscillator for a Laser Radar. In *Proc. of the European Solid-State Circuits Conference*, pages 469–472, September 2003.

[28] T.I. Otsuji. A Picosecond-Accuracy, 700-MHz Range, Si-Bipolar Time Interval Counter LSI. *IEEE Journal of Solid-State Circuits*, 28(9):941–947, September 1993.

[29] J. Phillips and K. Kundert. Noise in Mixers, Oscillators, Samplers, and Logic: An Introduction to Cyclostationary Noise. In *Proc. of the IEEE Custom Integrated Circuits Conference*, pages 431–438, May 2000.

[30] J.M. Rabaey, A. Chandrakasan, and B. Nikolic. *Digital Integrated Circuits: A Design Perspective*. Pearson Education, Inc., second edition, 2003.

[31] T.E. Rahkonen and J. Kostamovaara. The Use of Stabilized CMOS Delay Lines for the Digitization of Short Time Intervals. *IEEE Journal of Solid-State Circuits*, 28(8):887–894, August 1993.

[32] E. Raisanen-Ruotsalainen, T. Rahkonen, and J. Kostamovaara. A Low-Power CMOS Time-to-Digital Converter. *IEEE Journal of Solid-State Circuits*, 30(9):984–990, September 1995.

[33] P.J. Restle et al. Timing Uncertainty Measurements on the Power5 Microprocessor. In *International Solid-State Circuits Conference Digest of Technical Papers*, pages 354– 355, February 2004.

[34] J. Rivoir. Fully-Digital Time-to-Digital Converter for ATE with Autonomous Calibration. In *Proc. of the International Test Conference*, pages 1–10, October 2006.

[35] O. Sasaki, T. Taniguchi, T.K. Ohska, and H. Kurashige. A High-Resolution TDC in TKO BOX System. *IEEE Transactions on Nuclear Science*, 35(1):342–345, February 1988.

[36] P.C.S. Scholtens and M. Vertregt. A 6-b 1.6-Gsample/s Flash ADC in 0.18-μm CMOS Using Averaging Termination. *IEEE Journal of Solid-State Circuits*, 37(12):1599–1609, December 2002.

[37] K.L. Shepard and Y. Zheng. On-Chip Oscilloscopes for Noninvasive Time-Domain Measurement of Waveforms. In *Proc. of the International Conference on Computer Design*, pages 221–226, September 2001.

[38] R.B. Staszewski, S. Vemulapalli, P. Vallur, J. Wallberg, and P.T. Balsara. 1.3 V 20 ps Time-to-Digital Converter for Frequency Synthesis in 90-nm CMOS. *IEEE Transactions on Circuits and System–II: Express Briefs*, 53(3):220–224, March 2006.

[39] F. Stellari, F. Zappa, S. Cova, and L. Vendrame. Tools for Non-Invasive Optical Characterization of CMOS Circuits. In *Technical Digest of Electron Devices Meeting*, pages 487–490, December 1999.

[40] B.K. Swann et al. A 100-ps Time-Resolution CMOS Time-to-Digital Converter for Positron Emission Tomography Imaging Applications. *IEEE Journal of Solid-State Circuits*, 39(11):1839–1852, November 2004.

[41] S. Tisa, A. Lotito, A. Giudice, and F. Zappa. Monolithic Time-to-Digital Converter with 20ps Resolution. In *Proc. of the European Solid-State Circuits Conference*, pages 465–468, September 2003.

[42] N. Vijayaraghavan, B. Singh, S. Singh, and V. Srivastava. Novel Architecture for On-Chip AC Characterization of I/Os. In *Proc. of the IEEE International Test Conference*, pages 1–10, October 2006.

[43] T. Watanabe and S. Yamauchi. An All-Digital PLL for Frequency Multiplication by 4 to 1022 with Seven-Cycle Lock Time. *IEEE Journal of Solid-State Circuits*, 38(2):198–204, February 2003.

[44] D. Weinlader, R. Ho, C.-K. K. Yang, and M. Horowitz. An Eight Channel 35 GSample/s CMOS Timing Analyzer. In *International Solid-State Circuits Conference Digest of Technical Papers*, pages 170–171, February 2000.

Chapter 6

Conclusions and Future Work

In this thesis, it has been demonstrated that it is feasible to design and implement circuit and architectural techniques that can be used to address some of the emerging timing issues in modern VLSI technology. With precise and accurate timing of digital signals, such techniques help in the design and characterization of performance critical digital systems.

Testing high-performance pipelined datapaths using slow-speed testers has been illustrated experimentally using a DLL-based approach for on-chip clock timing. The design achieves a timing resolution of 52.9ps in 0.18μm CMOS technology. Using the closed-loop feature of the DLL allows the clock timing circuit itself to be testable.

In order to achieve much higher resolution than can be offered by a basic delay element, a DCDE has been proposed and its accuracy and linearity has been verified experimentally. Infinitesimally small delay steps have also been demonstrated, which can be exploited by many applications.

A possible application that can benefit from such precision is phase alignment for clock deskewing and other application. With the high resolution of the DCDE used, the precision is limited by the accuracy of the phase detector, which is a strong function of PVT conditions. It has been demonstrated that a phase aligner can achieve high accuracy using the proposed DCDE and a PVT tolerant phase detector that uses redundancy. The design is fully autonomous and achieves a sub-2ps resolution in the same technology.

In order to achieve high-precision real-time measurement of skew and jitter, an implementation of the flash architecture for TDCs has been demonstrated. The design achieves a maximum conversion rate of 1.25GSa/s. With a 2.5ps LSB, the design suffers from high non-linearity due to mismatch between the DCDEs and sampling offset of the arbiters. Having demonstrated the feasibility of achieving high conversion rate, additional design and calibration techniques need to be implemented in order to improve the converter's linearity and precision.

There are many aspects of the work presented in this thesis that can prompt future work on the same areas. Some of these include:

1. Implementing the pipeline testing methodology using an all-digital DLL that benefits from the precision offered by the DCDE and APD. This will not only improve the design's precision, but will make the testing of the clock-timing circuit much easier to perform in a digital system.

2. Modifying the design of the DCDE in order to reduce its static power consumption. This can be done by proper sizing of the transistors in the DVC block.

3. Implementing the DCDE by adjusting the delay in both stages of the design. This arrangement will allow wider delay adjustment range with smaller total absolute delays. This can be beneficial in reducing the number of stages needed to achieve a certain number of steps and hence reduce the non-linearity due to mismatch.

4. Exploring techniques for improving the linearity of the TDC. These can include circuit techniques for reducing offset and mismatch (e.g smaller absolute delays, redundancy in

the arbiters) or calibration and correction techniques (e.g. code re-ordering, linearity correction and offset cancelation).

Bibliography

[A+98] P. Andreani et al. Multihit Multichannel Time-to-Digital Converter with ±1% Differential Nonlinearity and Near Optimal Time Resolution. *IEEE Journal of Solid-State Circuits*, 33(4):650–656, April 1998.

[AC95] V.D. Agrawal and T.J. Chakraborty. High-Performance Circuit Testing with Slow-Speed Testers. In *Proc. of the International Test Conference*, pages 302–310, 1995.

[AHR01] N. Abaskharoun, M. Hafed, and G.W. Roberts. Strategies for On-Chip Sub-Nanosecond Signal Capture and Timing Measurements. In *Proc. of the IEEE International Symposium on Circuits and Systems*, pages 174–177, May 2001.

[AI96] Y. Arai and M. Ikeno. A Time Digitizer CMOS Gate-Array with a 250 ps Time Resolution. *IEEE Journal of Solid-State Circuits*, 31(2):212–220, February 1996.

[Ark04] B. Arkin. Realizing a Production ATE Custom Processor and Timing IC Containing 400 Independent Low-Power and High-Linearity Timing Verniers. In *International Solid-State Circuits Conference Digest of Technical Papers*, pages 348–349, February 2004.

[Av70] J. Aveynier and R. van Zurk. Vernier Chronotron Reflex. *Nuclear Instruments and Methods*, 78:161–170, 1970.

[BB98] D.W. Bailey and B.J. Benschneider. Clocking Design and Analysis for a 600-MHz Alpha Microprocessor. *IEEE Journal of Solid-State Circuits*, 33(11):1627–1633, November 1998.

[BFL$^+$01] F. Baronti, L. Fanucci, D. Lunardini, R. Roncella, and R. Saletti. On the Differential Nonlinearity of Time-to-Digital Converters Based on Delay-Locked-Loop Delay Lines. *IEEE Transactions on Nuclear Science*, 48(6):2424–2431, December 2001. 9-40.

[BKN$^+$03] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De. Parameter Variations and Impact on Circuits and Microarchitecture. In *Proc. of the Design Automation Conference*, pages 338–342, June 2003.

[BNS05] D. Bordoley, H. Nguyen, and M. Soma. A Statistical Study of the Effectiveness of BIST Jitter Measurement Techniques. In *Proc. of the International Conference on Computer Aided Design*, pages 100–107, November 2005.

[CA01] M. Choi and A.A. Abidi. A 6-b 1.3-Gsample/s A/D Converter in 0.35-μm CMOS. *IEEE Journal of Solid-State Circuits*, 36(12):1847–1858, December 2001.

[CCL05] P.L. Chen, C.C. Chung, and C.Y. Lee. A Portable Digitally Controlled Oscillator Using Novel Varactors. *IEEE Transactions on Circuits and System–II: Express Briefs*, 52(5):233–237, May 2005.

[CCTL05] P. Chen, C.-C. Chen, C.-C. Tsai, and W.-F. Lu. A Time-to-Digital-Converter-Based CMOS Smart Temperature Sensor. *IEEE Journal of Solid-State Circuits*, 40(8):1642–1648, August 2005.

[CCYL06] P.L. Chen, C.C. Chung, J.N. Yang, and C.Y. Lee. A Clock Generator with Cascaded Dynamic Frequency Counting Loops for Wide Multiplication Range Applications. *IEEE Journal of Solid-State Circuits*, 41(6):1275–1285, June 2006.

[Che06] T.C. Chen. Where CMOS is Going: Trendy Hype vs. Real Technology. In *International Solid-State Circuits Conference Digest of Technical Papers*, pages 1–7, February 2006.

[Chr95] J. Christiansen. An Integrated CMOS 0.15 ns Digital Timing Generator for TDC's and Clock Distribution Systems. *IEEE Transactions on Nuclear Science*, 42(4):753–757, August 1995.

[CL99] P. Chen and S.-I. Liu. A Cyclic CMOS Time-to-Digital Converter with Deep Subnanosecond Resolution. In *Proc. of the IEEE Custom Integrated Circuits Conference*, pages 605–608, May 1999.

[CL03] C.C. Chung and C.Y. Lee. An All-Digital Phase-Locked Loop for High-Speed Clock Generation. *IEEE Journal of Solid-State Circuits*, 38(2):347–351, February 2003.

[CL04] C.C. Chung and C.Y. Lee. A New DLL-Based Approach for All-Digital Multiphase Clock Generation. *IEEE Journal of Solid-State Circuits*, 39(3):469–475, March 2004.

[CM96] J.T.Y. Chang and E. McClusky. Quantitative Analysis of Very-Low-Voltage Testing. In *Proc of the VLSI Test Symposium*, pages 332–337, April 1996.

[CR04] A.H. Chan and G.W. Roberts. A Jitter Characterization System Using a Component-Invariant Vernier Delay Line. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 12(1):79–95, January 2004.

[DGLN95] J. Dunning, G. Garcia, J. Lundberg, and E. Nuckolls. An All-Digital Phase-Locked Loop with 50-Cycle Lock Time Suitable for High-Performance Microprocessors. *IEEE Journal of Solid-State Circuits*, 30(4):412–422, April 1995.

[DHYL00] G.K. Dehng, J.M. Hsu, C.Y. Yang, and S.I. Liu. Clock-Deskew Buffer Using a SAR-Controlled Delay-Locked Loop. *IEEE Journal of Solid-State Circuits*, 35(8):1128–1136, August 2000.

[DKPB03] C.E. Dike, N.A. Kurd, P. Patra, and J. Barkatullah. A Design for Digital, Dynamic Clock Deskew. In *2003 Symposium on VLSI Circuits Digest of Technical Papers*, pages 21–24, June 2003.

[DSH00] P. Dudek, S. Szczepanski, and J.V. Hatfield. A High-Resolution CMOS Time-to-Digital Converter Utilizing a Vernier Delay Line. *IEEE Journal of Solid-State Circuits*, 35(2):240–247, February 2000. 9-46.

[DTK+00] U. Desai, S. Tam, R. Kim, J. Zhang, and S. Rusu. Itanium Processor Clock Design. In *Proc. of International Symposium on Physical Design*, pages 94–98, April 2000.

[Fra94] P. Franco. Testing Digital Circuits for Timing Failures by Output Waveform Analysis. Technical Report 94-9, Center for Reliable Computing, Stanford University, September 1994.

[Fri01] E.G. Friedman. Clock Distribution Networks in Synchronous Digital Integrated Circuits. *Proc. of the IEEE,*, 89(5):665–692, May 2001.

[G+94] E.J. Gerds et al. A CMOS Time to Digital Converter IC with 2 Level Analog CAM. *IEEE Journal of Solid-State Circuits*, 29(9):1068–1076, September 1994.

[G⁺99] B.W. Garlepp et al. A Portable Digital DLL for High-Speed CMOS Interface Circuits. *IEEE Journal of Solid-State Circuits*, 34(5):632–644, May 1999.

[GBP⁺98] P.E. Gronowski, W.J. Bowhill, R.P. Preston, M.K. Gowan, and R.L. Allmon. High-Performance Microprocessor Design. *IEEE Journal of Solid-State Circuits*, 33(5):676–686, May 1998.

[GC00] V. Gutnik and A. Chandrakasan. On-Chip Picosecond Time Measurement. In *2000 Symposium on VLSI Circuits Digest of Technical Papers*, pages 52–53, June 2000.

[GD98] G. Geannopoulos and X. Dai. An Adaptive Digital Deskewing Circuit For Clock Distribution Networks. In *International Solid-State Circuits Conference Digest of Technical Papers*, pages 400–401, February 1998.

[GLVN⁺94] C.T. Gray, W. Liu, W.A.M. Van Noije, Jr. Hughes, T.A., and III Cavin, R.K. A Sampling Technique and its CMOS Implementation with 1 Gb/s Bandwidth and 25 ps Resolution. *IEEE Journal of Solid-State Circuits*, 29(3):340–349, March 1994.

[H⁺97] A. Hatakeyama et al. A 256-Mb SDRAM Using a Register-Controlled Digital DLL. *IEEE Journal of Solid-State Circuits*, 32(11):1728–1734, November 1997.

[HCT04] C.S. Hwang, P. Chen, and H.W. Tsao. A High-Precision Time-to-Digital Converter Using a Two-Level Conversion Scheme. *IEEE Transactions on Nuclear Science*, 51(4):1349–1352, August 2004.

[HKF⁺00] T. Hamamoto, S. Kawasaki, K. Furutani, K. Yasuda, and Y. Konishi. A Skew and Jitter Suppressed DLL Architecture for High Frequency DDR SDRAMs. In *2000 Symposium on VLSI Circuits Digest of Technical Papers*, pages 76–81, June 2000.

[HM93] H. Hao and E.J. McCluskey. Very-Low-Voltage Testing for Weak CMOS Logic ICs. In *Proc. of the International Test Conference*, pages 275–284, October 1993.

[HN01] D. Harris and S. Naffziger. Statistical Clock Skew Modeling with Data Delay Variations. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 9(6):888–898, December 2001.

[HWL01] T.Y. Hsu, C.C. Wang, and C.Y. Lee. Design and Analysis of a Portable High-Speed Clock Generator. *IEEE Transactions on Circuits and System–II: Analog and Digital Signal Processing*, 48(4):367–375, April 2001.

[JM97] D.A. Johns and K. Martin. *Analog Integrated Circuit Design*. John Wiley & Sons, Inc., 1997.

[JMK06] J.P. Jansson, A. Mantyniemi, and J Kostamovaara. A CMOS Time-to-Digital Converter with better than 10 ps Single-Shot Precision. *IEEE Journal of Solid-State Circuits*, 41(6):1286– 1296, June 2006. 9-13.

[Kal04] J. Kalisz. Review of Methods for Time Interval Measurements with Picosecond Resolution. *Metrologia*, 41(1):17–32, February 2004.

[KBD$^+$01a] N.A. Kurd, J.S. Barkarullah, R.O. Dizon, T.D. Fletcher, and P.D. Madland. A Multigigahertz Clocking Scheme for the Pentium Microprocessor. *IEEE Journal of Solid-State Circuits*, 36(11):1647–1653, November 2001.

[KBD$^+$01b] N.A. Kurd, J.S. Barkarullah, R.O. Dizon, T.D. Fletcher, and P.D. Madland. Multi-GHz Clocking Scheme for Intel Pentium 4 Microprocessor. In *International Solid-State Circuits Conference Digest of Technical Papers*, pages 404–405, February 2001.

[KCS94] T.A. Knotts, D. Chu, and J. Sommer. A 500 MHz Time Digitizer IC with 15.625ps Resolution. In *International Solid-State Circuits Conference Digest of Technical Papers*, pages 58–59, February 1994.

[KPS+04] K. Karadamoglou, N.P. Paschalidis, E. Sarris, N. Stamatopoulos, G. Kottaras, and V. Paschalidis. An 11-bit High-Resolution and Adjustable-Range CMOS Time-to-Digital Converter for Space Science Instruments. *IEEE Journal of Solid-State Circuits*, 39(1):214–222, January 2004.

[LMS+99] F. Lin, J. Miller, A. Schoenfeld, M. Ma, and R.J. Baker. A Register-Controlled Symmetrical DLL for Double-Data-Rate DRAM. *IEEE Journal of Solid-State Circuits*, 34(4):565–568, April 1999.

[LR05] P.M. Levine and G.W. Roberts. High-Resolution Flash Time-to-Digital Conversion and Calibration for System-on-Chip Testing. *IEE Proceedings - Computers and Digital Techniques*, 152(3):415–426, May 2005.

[LWY06] L.-M. Lee, D. Weinlader, and C.-K. K. Yang. A Sub-10-ps Multiphase Sampling System Using Redundancy. *IEEE Journal of Solid-State Circuits*, 41(1):265–273, January 2006.

[M+00] K. Minami et al. A 1 GHz Portable Digital Delay-Locked Loop with Infinite Phase Capture Ranges. In *International Solid-State Circuits Conference Digest of Technical Papers*, pages 350–351, 469, February 2000.

[MFDN05] P. Mahoney, E. Fetzer, B. Doyle, and S. Naffziger. Clock distribution on a Dual-Core, Multi-Threaded Itanium-Family Processor. In *International Solid-State Circuits Conference Digest of Technical Papers*, pages 292–293,599, February 2005.

[MNS05] M. Maymandi-Nejad and M. Sachdev. A Monotonic Digitally Controlled Delay Element. *IEEE Journal of Solid-State Circuits*, 40(11):2212– 2219, November 2005.

[MPES91] M. Mehta, V. Parmar, and Jr. E. Swartzlander. High-Speed Multiplier Design Using Multi-Input Counter and Compressor Circuits. In *Proc. of the IEEE Symposium on Computer Arithmetic*, pages 43–50, June 1991.

[MRK99] A. Mántyniemi, T. Rahkonen, and J. Kostamovaara. An Integrated Digital CMOS Time-to-Digital Converter with Sub-Gate-Delay Resolution. *Analog Integrated Circuits and Signal Processing*, 22(1):61–70, January 1999. 9-55.

[MRK02] A. Mántyniemi, T. Rahkonen, and J. Kostamovaara. An Integrated 9-Channel Time Digitizer with 30ps Resolution. In *International Solid-State Circuits Conference Digest of Technical Papers*, pages 266–267,465, February 2002.

[Nas98] S.R. Nassif. Within-Chip Variability Analysis. In *Technical Digest of Electron Devices Meeting*, pages 283–286, December 1998.

[NMK03] I. Nissinen, A. Mantyniemi, and J. Kostamovaara. A CMOS Time-to-Digital Converter based on a Ring Oscillator for a Laser Radar. In *Proc. of the European Solid-State Circuits Conference*, pages 469–472, September 2003.

[NS01] M. Nummer and M. Sachdev. A Methodology for Testing High-Performance Circuits at Arbitrarily Low Test Frequency. In *Proc. of the IEEE VLSI Test Symposium*, pages 68–74, April 2001.

[O$^+$95] N. Ohkubo et al. A 4.4 ns CMOS 54×54-b Multiplier Using Pass-Transistor Multiplexer. *IEEE Journal of Solid-State Circuits*, 30(3):251–257, March 1995.

[ON04] T. Olsson and P. Nilsson. A Digitally Controlled PLL for SoC Applications. *IEEE Journal of Solid-State Circuits*, 39(5):751–760, May 2004.

[Ots93] T.I. Otsuji. A Picosecond-Accuracy, 700-MHz Range, Si-Bipolar Time Interval Counter LSI. *IEEE Journal of Solid-State Circuits*, 28(9):941–947, September 1993.

[PK00] J. Phillips and K. Kundert. Noise in Mixers, Oscillators, Samplers, and Logic: An Introduction to Cyclostationary Noise. In *Proc. of the IEEE Custom Integrated Circuits Conference*, pages 431–438, May 2000.

[R^+02] P.J. Restle et al. The Clock Distribution of the Power4 Microprocessor. In *International Solid-State Circuits Conference Digest of Technical Papers*, pages 144–145, February 2002.

[R^+04] P.J. Restle et al. Timing Uncertainty Measurements on the Power5 Microprocessor. In *International Solid-State Circuits Conference Digest of Technical Papers*, pages 354– 355, February 2004.

[Raz01] B. Razavi. *Design of Analog CMOS Integrated Circuits*. McGraw-Hill series in electrical and computer engineering. McGraw-Hill, 2001.

[RCN03] J.M. Rabaey, A. Chandrakasan, and B. Nikolic. *Digital Integrated Circuits: A Design Perspective*. Pearson Education, Inc., second edition, 2003.

[Riv06] J. Rivoir. Fully-Digital Time-to-Digital Converter for ATE with Autonomous Calibration. In *Proc. of the International Test Conference*, pages 1–10, October 2006.

[RK93] T.E. Rahkonen and J. Kostamovaara. The Use of Stabilized CMOS Delay Lines for the Digitization of Short Time Intervals. *IEEE Journal of Solid-State Circuits*, 28(8):887–894, August 1993.

[RRRK95] E. Raisanen-Ruotsalainen, T. Rahkonen, and J. Kostamovaara. A Low-Power CMOS Time-to-Digital Converter. *IEEE Journal of Solid-State Circuits*, 30(9):984–990, September 1995.

[S+99] S. Sengupta et al. Defect-Based Test: A Key Enabler for Successful Migration to Structural Test. *Intel Technology Journal*, Q1, 1999.

[S+04] B.K. Swann et al. A 100-ps Time-Resolution CMOS Time-to-Digital Converter for Positron Emission Tomography Imaging Applications. *IEEE Journal of Solid-State Circuits*, 39(11):1839–1852, November 2004.

[SH97] S. Sidiropoulos and M. Horowitz. A Semidigital Dual Delay-Locked Loop. *IEEE Journal of Solid-State Circuits*, 32(11):1683–1692, November 1997.

[SKSS00] H. Speek, H. G. Kerkhoff, M. Sachdev, and M. Shashaani. Bridging the Test Speed Gap: Design for Delay Testability. In *Proc. of the IEEE European Test Workshop*, pages 3–8, 2000.

[Son03] H. Song. Programmable De-Skew Clock Generation Based on Dual Digital Delay-Locked Loop Structure. In *Proc. of the IEEE International System-on-Chip Conference*, pages 249–250, September 2003.

[SS99] M. Shashani and M. Sachdev. A DFT Technique for High-Performance Circuit Testing. In *Proc. of the International Test Conference*, pages 267–285, October 1999.

[SSLTW00] S. Sauter, D. Schmitt-Landsiedel, R. Thewes, and W. Weber. Effect of Parameter Variations at Chip and Wafer Level on Clock Skews. *IEEE Transactions on Semiconductor Manufacturing*, 13(4):395–400, November 2000.

[STOK88] O. Sasaki, T. Taniguchi, T.K. Ohska, and H. Kurashige. A High-Resolution TDC in TKO BOX System. *IEEE Transactions on Nuclear Science*, 35(1):342–345, February 1988.

[SV02] P.C.S. Scholtens and M. Vertregt. A 6-b 1.6-Gsample/s Flash ADC in 0.18-μm CMOS Using Averaging Termination. *IEEE Journal of Solid-State Circuits*, 37(12):1599–1609, December 2002.

[SVV$^+$06] R.B. Staszewski, S. Vemulapalli, P. Vallur, J. Wallberg, and P.T. Balsara. 1.3 V 20 ps Time-to-Digital Converter for Frequency Synthesis in 90-nm CMOS. *IEEE Transactions on Circuits and System–II: Express Briefs*, 53(3):220–224, March 2006.

[SZ01] K.L. Shepard and Y. Zheng. On-Chip Oscilloscopes for Noninvasive Time-Domain Measurement of Waveforms. In *Proc. of the International Conference on Computer Design*, pages 221–226, September 2001.

[SZCV99] F. Stellari, F. Zappa, S. Cova, and L. Vendrame. Tools for Non-Invasive Optical Characterization of CMOS Circuits. In *Technical Digest of Electron Devices Meeting*, pages 487–490, December 1999.

[TLD04] S. Tam, R.D. Limaye, and U.N. Desai. Clock Generation and Distribution for the 130-nm Itanium2 Processor with 6-MB On-Die L3 Cache. *IEEE Journal of Solid-State Circuits*, 39(4):636– 642, April 2004.

[TLGZ03] S. Tisa, A. Lotito, A. Giudice, and F. Zappa. Monolithic Time-to-Digital Converter with 20ps Resolution. In *Proc. of the European Solid-State Circuits Conference*, pages 465–468, September 2003.

[TRD+00] S. Tam, S. Rusu, U. Desai, R. Kim, Ji Zhang, and I. Young. Clock Generation and Distribution for the First IA-64 Microprocessor. *IEEE Journal of Solid-State Circuits*, 35(11):1545–1552, November 2000.

[TTT+96] S. Tanoi, T. Tanabe, K. Takahashi, S. Miyamoto, and M. Uesugi. A 250-622 MHz Deskew and Jitter-Suppressed Clock Buffer using Two-Loop Architecture. *IEEE Journal of Solid-State Circuits*, 31(4):487–493, April 1996.

[VSSS06] N. Vijayaraghavan, B. Singh, S. Singh, and V. Srivastava. Novel Architecture for On-Chip AC Characterization of I/Os. In *Proc. of the IEEE International Test Conference*, pages 1–10, October 2006.

[W+07] S.B. Wijeratne et al. A 9-GHz 65-nm Intel Pentium 4 Processor Integer Execution Unit. *IEEE Journal of Solid-State Circuits*, 42(1):26–37, January 2007.

[WHYH00] D. Weinlader, R. Ho, C.-K. K. Yang, and M. Horowitz. An Eight Channel 35 GSample/s CMOS Timing Analyzer. In *International Solid-State Circuits Conference Digest of Technical Papers*, pages 170–171, February 2000.

[WY03] T. Watanabe and S. Yamauchi. An All-Digital PLL for Frequency Multiplication by 4 to 1022 with Seven-Cycle Lock Time. *IEEE Journal of Solid-State Circuits*, 38(2):198–204, February 2003.

[XBG+01] T. Xanthopoulos, D.W. Bailey, A.K. Gangwar, M.K. Gowan, A.K. Jain, and B.K. Prewitt. The Design and Analysis of the Clock Distribution Network for a 1.2GHz

Alpha Microprocessor. In *International Solid-State Circuits Conference Digest of Technical Papers*, pages 402–403, February 2001.

[YL01] C.Y. Yang and S.I. Liu. A One-Wire Approach for Skew-Compensating Clock Distribution based on Bidirectional Techniques. *IEEE Journal of Solid-State Circuits*, 36(2):266–272, February 2001.