UNIVERSITY OF CALIFORNIA, SAN DIEGO

# High-Performance Low-Power VLSI Design

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Computer Science (Computer Engineering)

by

Haikun Zhu

Committee in charge:

Professor Chung-Kuan Cheng, Chair
Professor Peter Asbeck
Professor Fan Chung Graham
Professor Ronald Graham
Professor Michael B. Taylor

2007

.

The dissertation of Haikun Zhu is approved, and it is acceptable in quality and form for publication on microfilm:

Chair

University of California, San Diego

2007

To my forthcoming baby.

## TABLE OF CONTENTS

## LIST OF FIGURES

# LIST OF TABLES

# ACKNOWLEDGEMENTS

First and foremost, I am immensely grateful to my advisor, Professor Chung-Kuan Cheng for his guidance and financial support throughout my Ph.D. journey. I feel fortunate and privileged to have studied with him, from whom I learned not only the science and art of doing research, but also the wisdom of life. The work presented in this dissertation benefits the most from his insightful intuitions, inspiration and encouragement. I could not have forgotten the invaluable advices I received from him during the countless research discussions: always start from the simplest non-trivial case, never lose the big picture, be proactive and do not drop the ball. I thank him for his patience and tolerance during the entire process when I made progress toward the completion of this work. His tremendous influence has shaped my life in the past four and half years, and will continue to impact my future development for many years to come, both intellectually and personally.

I feel obliged to Professor Ronald Graham for his kindly supporting many of my research activities. Professor Graham taught me the mathematical skills for deriving the formula of the width of zero-deficiency adders in Chapter IV and helped me in polishing the early manuscript of that chapter. I thank Professor David Harris for sharing his expertise on digital CMOS circuit design and offering constructive suggestions on the shifter work, without which the content of Chapter III would have been much immature. I would like to thank Professor Hashimoto Masanori for his continual help in the distortionless interconnect project and answering many of my questions regarding transmission line design, which have greatly deepened my understanding of the subject. Dr. Barry Rubin and Dr. Alina Deustch from IBM T. J. Watson Research Center helped me in setting up and learning the EIP tool suite; and for this I thank them. Special thanks go to Professor T. C. Hu, for teaching me combinatorics, sharing his research philosophy and for the pleasure of discussing classical Chinese literature

with him.

I wish to thank Professor Fan Chung Graham, Professor Michael Taylor and Professor Peter Asbeck for taking the time and patience to review my dissertation and serve on my final defense committee. I also would like to thank Professor Paul Chau for sitting on my qualifying exam committee and being supportive on my shifter work.

My fellow graduate students in CK's group and CSE department provide a dynamic, interdisciplinary and fun learning environment, which has made my Ph.D. study a very precious and unique experience; and I wish to take this opportunity to thank them all. Among many others, I would like to thank Dr. Jianhua Liu and Dr. Shuo Zhou for the time and joy of working together as datapath subgroup, and for providing various convenience in my early days in the group. I especially wish to thank Dr. Zhengyong Zhu for many of his selfless helps that I still feel heartwarming to this day. Thanks also go to He Peng, Rui Shi, Yi Zhu, Ling Zhang, Rensheng Wang and Wanping Zhang, for the discussion hours and fun time that we spent together. I feel light-hearted with you guys.

My old buddies from outside UCSD have always been offering me great friendship that I cherish as priceless. They are Dr. Xiangyu Tang, Jiawei Zhang, Dr. Yaping Zhan, Chenbo Liu and too many to list. I wish them all the best.

Finally, I am deeply indebted to my family for their everlasting support and love that I feel powerless to describe in words. My parents have been unconditionally understanding and supportive of my deciding to pursue dreams far away from home, when I should have been accompanying with them. I am truly thankful to my dearest wife, Lingyun, for her irreplaceable love, care and putting up all my complaints during the down time of my Ph.D. life, without which the completion of this dissertation is impossible.

Chpater II, in part, is a reprint of the paper "Distortion minimization for packaging level interconnects" co-authored with Rui Shi, Hongyu Chen, Chung-Kuan Cheng, Alina Deutsch and George Katopis in the proceedings of 2006 EPEP conference, and the paper "Approaching speed-of-light distortionless communication for on-chip interconnect" co-authored with Rui Shi, Hongyu Chen and Chung-Kuan Cheng in the proceedings of ASPDAC 2007. The dissertation author was the primary investigator and author of these two papers.

Chapter III, in part, is a reprint of the paper "An interconnect-centric approach to cyclic shifter design using fanout splitting and cell order optimization" co-authored with Yi Zhu, Chung-Kuan Cheng and David Harris in the proceedings of ASPDAC 2007. The dissertation author was the primary investigator and author of this paper.

Chapter IV, in part, is a reprint of the paper "On the construction of zero-deficiency parallel prefix circuits with minimum depth" co-authored with Chung-Kuan Cheng and Ronald Graham in ACM Transactions on Design Automation of Electronic Systems, Vol. 11, Issue 2, pp.387-409, April 2006. The dissertation author was the primary investigator and author of this paper.

## VITA

| | |
|---|---|
| Oct. 1978 | Born in Zhu Zhou, Hu Nan Province, P.R. China |
| 1999 | B.Eng. in Communications Engineering<br>Shanghai Jiao Tong University, Shanghai, P.R. China |
| 2002 | M.Sc. in Electrical Engineering<br>Fudan University, Shanghai, P.R. China |
| 2007 | Ph.D. in Computer Science (Computer Engineering)<br>University of California, San Diego |

## PUBLICATIONS

Haikun Zhu, Rui Shi, Hongyu Chen and Chung-Kuan Cheng, "Approaching speed-of-light distortionless communication for on-chip interconnect," in Proceeding of 12th Asia and South Pacific Design Automation Conference (ASP-DAC), pp.684-689, Jan. 2007.

Haikun Zhu, Yi Zhu, Chung-Kuan Cheng and David Harris "An interconnect-centric approach to cyclic shifter design using fanout splitting and cell order optimization," in Proceeding of 12th Asia and South Pacific Design Automation Conference (ASP-DAC), pp.616-621, Jan. 2007.

Jianhua Liu, Yi Zhu, Haikun Zhu and Chung-Kuan Cheng, "Optimal prefix adder in a comprehensive area, timing and power design space," in Proceeding of 12th Asia and South Pacific Design Automation Conference (ASP-DAC), pp.609-615, Jan. 2007.

Haikun Zhu, Rui Shi, Hongyu Chen, Chung-Kuan Cheng, Alina Deutsch and George Katopis, "Distortion minimization for packaging level interconnects," 15th Topical Meeting on Electronic Performance of Electronic Packaging (EPEP), pp.189-192, Oct. 2006.

Haikun Zhu, Chung-Kuan Cheng and Ronald Graham, "On the construction of zero-deficiency parallel prefix circuits with minimum depth," in ACM Trans. on Design Automation of Electronics Systems (TODAES), Vol. 11, Issue 2, pp.387-409, Apr. 2006.

Haikun Zhu, Chung-Kuan Cheng and Ronald Graham, "Constructing zero-deficiency parallel prefix adder of minimum depth," in Proceeding of 10th Asia and South Pacific Design Automation Conference (ASP-DAC), pp.883-888, Jan. 2005.

Jianhua Liu, Shuo Zhou, Haikun Zhu and Chung-Kuan Cheng, "An algorithmic approach for generic parallel adders," in Proceeding of International Conference on Computer Aided Design (ICCAD), pp.734-740, Nov. 2003.

# FIELDS OF STUDY

Major Field: Computer Science (Computer Engineering)
Studies in VLSI CAD
Professor Chung-Kuan Cheng

# ABSTRACT OF THE DISSERTATION

High-Performance Low-Power VLSI Design

by

Haikun Zhu
Doctor of Philosophy in Computer Science (Computer Engineering)
University of California, San Diego, 2007
Professor Chung-Kuan Cheng, Chair

It is widely accepted that, as semiconductor technology continues to evolve, interconnects have dominated over transistors in terms of both performance and power consumption in VLSI. Common belief is that interconnects at the on-chip global level or above in a large extent limit the system performance. While we certainly agree with that, we also believe that local interconnects manifest themselves as critical as the devices in datapath design, if not more so. We intend to show that the interconnect dominance is ubiquitous, and deserves a vertical treatment rather than focusing on a certain level only.

In this dissertation, we address several open issues and problems faced by today's designers to reflect the above philosophy:

(1) We propose a passive compensation scheme for improving the signal quality over long metal interconnects. The proposed method tries to mimic the behavior of distortionless transmission line by evenly adding shunt resistors between the signal line and ground. Design parameters such as shunt value, spacing and termination are leveraged to achieve the best eye-diagram jitter, and the tradeoff between jitter and eye-opening is also studied. The evaluation of the worst-case jitter and eye-opening is enabled by a fast analytical prediction method based on any given bitonic step response, which is new to the best knowledge of the author.

(2) We revisit the design of cyclic shifter by introducing two orthogonal new techniques: fanout splitting and cell order optimization using Integer Linear

Programming. Both methods emphasize on reducing the interconnect burden in the shifter network, and significant savings on delay and power are reported.

(3) We solve the open problem of constructing zero-deficiency prefix adder of minimum depth. This work complements previous studies of the asymptotic behavior of depth-size tradeoff in prefix adders. To designers, it answers the question of how far should we push the timing so as not to incur exponential cost of area and power in prefix adders.

Together these contributions made efforts toward high-performance lower-power VLSI design in the nanometer era.

# I Introduction

## I.A Technology Trends of VLSI Systems

### I.A.1 Challenges on interconnect design

While the debate of when semiconductor technology will scale to its limit still goes on, Moore's law amazingly continues to prove its validity. With over one billion transistors integrated in a single chip, it becomes of utmost interest to interconnect the devices efficiently so as not to compromise much of the system performance and maintain a reasonable power consumption.

Concerning interconnect performance there are two important metrics: latency and bandwidth. Latency quantifies the point-to-point delay of electrical signals. It has been pointed out by many researchers that typical on-chip global RC wires are not sufficient to support future high-speed microprocessors and introducing on-chip transmission line is necessary [55] [22] [10] [30]. However, transmission line does not automatically solve the bandwidth problem because when frequency goes even higher it may suffer from the inter-symbol interference, a phenomenon usually seen at the board level. This ISI-induced performance limitation is also important for multi-chip modules as packaging level integration becomes popular.

It is reported in [43] that the interconnects account for over 50% of the dynamic power consumption in a microprocessor, and this number is about 80% if only global interconnects are considered. Thus power consumption shall never be left out in designing high-performance interconnects.

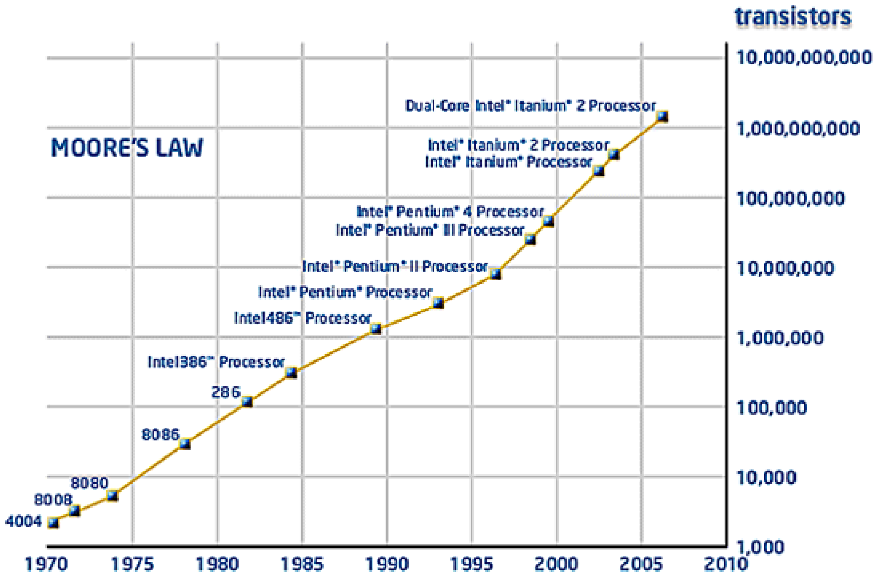


Figure I.1 Moore's law [3]

### I.A.2 Challenges on datapath optimization

Datapath design sees a similar challenge of interconnect playing a role of increasing importance in determining the overall performance. Although local interconnects also scale to be smaller and faster with succeeding technology generations, they scale at a slower rate than that of the transistors. Thus, the relative delay contributed by wires in a datapath system increases steadily, as is clearly illustrated in Figure I.2. Since the distributed RC delay for local interconnects is largely negligible [27], one can conclude that the raise of wire delay component is primarily due to the wire load effect, which indicates that a similar trend shall be observed for gate/wire power dissipation in datapath.

For ASIC designs, what this means is that proper physical information shall be taken into account during the synthesis step. Commercial tools such as Synopsys DC-Topographical adopt such an idea and do a global placement behind the scene to guide the logic synthesis. On the other hand, datapath in microprocessor design typically assumes bit-slice structure and involves a large amount of custom design. In this case, designers shall bear in mind interconnect/layout

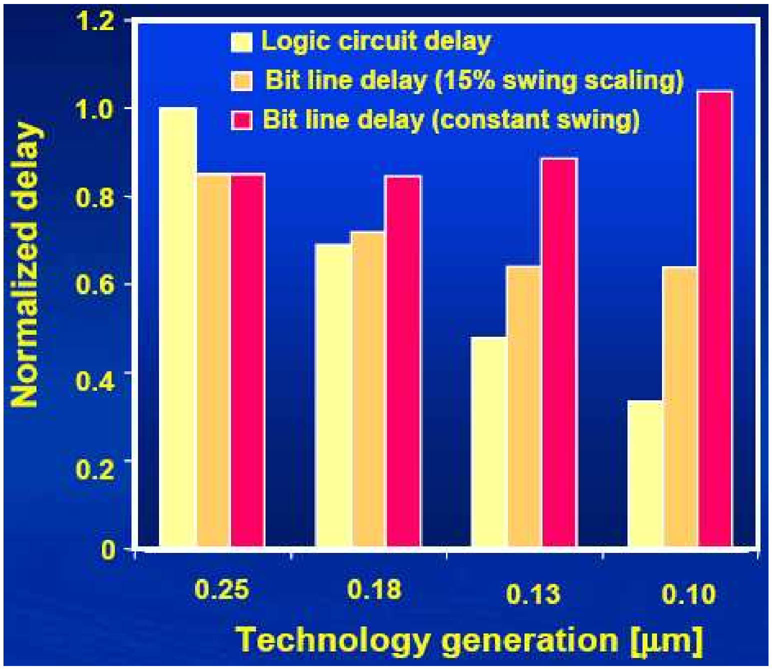


Figure I.2 Scaling effect on gate and wire delay [47]

planning early when determining which logic function blocks to use.

Another important aspect in datapath design is to understand the tradeoffs between different metrics including timing, area and power. An interesting problem of this category is the depth-size tradeoff in prefix adders. It is shown in [51] that a reduction of computation time from $2\log n$ to $\log n$ will asymptotically increase the area from $O(n\log n)$ to $O(n^2)$ for a prefix adder. However, it is not clear when exactly this exponential increase of area will occur, knowing which is instrumental for the designers to make proper tradeoff.

## I.B Dissertation Outline

We dedicate this dissertation to address the aforementioned challenges on interconnect and datapath design, and the remaining chapters are organized as follows.

Chapter II describes a novel passive compensation scheme for high-speed electrical serial link design. The proposed method is shown to be cost-effective, robust and superior to the best alternative technique known to the author, and is applicable to various level of interconnects ranging from on-chip, packaging to board.

Chapter III presents two independent techniques for optimizing cyclic shifters: fanout splitting and cell order optimization using Integer Linear Programming. Both methods emphasize on reducing the interconnect burden in the shifter network, and significant savings on delay and power are reported.

Chapter IV addresses a twenty-year open question regarding the depth-size tradeoff in prefix adders, namely, constructing zero-deficiency prefix adder of minimum depth.

Chapter V concludes the whole dissertation with a brief summary of contribution and discusses possible directions for future work.

# II Distortionless Electrical Signaling via Passive Compensation Scheme

## II.A Introduction

In this chapter, a passive compensation scheme for approaching distortionless electrical signaling is proposed. The method only utilizes passive shunt resistors, thus is applicable at different levels of integration including on-chip, packaging and board. We then propose an analytical method for predicting eye-opening and jitter from any given bitonic step response. The analytical model enables us to quickly evaluate the performance of a large class of interconnect systems, laying the foundation for further optimization.

It has been widely accepted that interconnect has arisen as the prime bottleneck for current and future computing systems. Not only such, the interconnect dominance is omnipresent, spreading from on-chip, multi-chip modules to board level systems, and it imposes challenges on communication latency, bandwidth as well as power consumption. According to the ITRS roadmap [6], on-chip global clock frequency will reach 12GHz (or equivalently 83.3ps cycle time) at 45nm technology node by year 2010. At the same time, signal propagation on 1mm global wire alone will take up to 523ps. Note that 1mm lines are absolutely not rare; based on Rent's rule the number of wires over 1cm at 45nm technology will reach

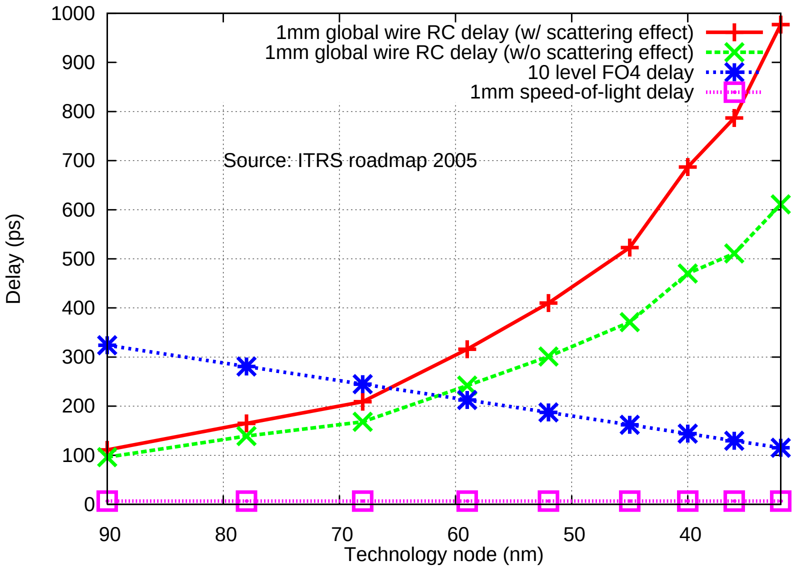


Figure II.1 Delay trend of on-chip global interconnects

Table II.1 Silicon Technology Trend

| Year | 2005 | 2010 | 2015 |
|---|---|---|---|
| D 1/2 pitch (nm) | 80 | 45 | 25 |
| Chip Size (mm$^2$) | 310 | 310 | 310 |
| Pin Count | 3,400 | 4,009 | 6,402 |
| Cents/Pin | 1.78 | 1.37 | 1.05 |
| On-chip (MHz) | 5,170 | 12,000 | - |
| Off-chip (MHz) | 3,125 | - | 29,103 |
| Power Density (Watt/mm$^2$) | 0.54 | 0.64 | - |

1 million. On the other hand, off-chip frequency is also expected to grow, and the ITRS predicts an astonishing 29.1GHz in 2015, while the pin count per chip will only moderately double from 3,400 in 2005 to 6,402 in 2015. As a consequence, we will need more bandwidth per serial link.

To tackle this “interconnect wall”, synergic efforts across academia and industry are being made. Since interconnects at different scales show distinct physical characteristics, their causes of performance limitation are also different.

### On-chip Global Interconnects

For on-chip interconnects, since the series resistance is large, we usually see large attenuation and small frequency dependency. As a result, on-chip global interconnects mostly suffer from large distributive RC delay, to which the conventional solution is buffer insertion [7] [12]. By dividing a long interconnect into small segments, the quadratic distributive RC delay becomes linear w.r.t. the wire length. The common objectives for on-chip buffer planning are delay, power, and more recently, bandwidth [11]. However, buffer insertion is more a mitigation than a solution to the interconnect nightmare. For example, at 70 nm the optimal buffer spacing and sizing for minimum delay are 200 $\mu$m and 55x, respectively [59]. This indicates that we need a multitude of large buffers going everywhere on the chip. These large buffers not only take up a lot of active area on the substrate, but also consume quite a fraction of power. Even worse, large number of buffers creates new problems for routing, placement and power/ground noise. Even at the expense of significant on-chip real estate, the global wire delay can only be improved to 158 ps/mm with optimally planned buffers [59].

As an attractive alternative, on-chip signaling using transmission line (T-Line) has received intensive research focus in recent years. The fundamental concept behind T-Line is that signals propagate as electromagnetic waves[1] rather than holistic diffusion of the electrons in the conductor. This wave behavior has a two-fold implication. First, waves are much faster than electron diffusion; they move at the speed-of-light in the dielectric. Second, wave propagation consumes much less power, because there is no need for the driver to drive the whole wire during the transmission of the symbol. The signal, once injected, leaves the driver and propagates on its own as supported by the T-Line.

### Packaging Level Interconnects

Packaging level interconnects such as those used for multi-chip modules exhibit large frequency dependency because of skin effect and proximity effect, although the attenuation becomes smaller. The packaging substrate also has a larger

[1] More specifically, signals travel in the form of TEM (Traverse Electromagnetic) for most on-chip structures with uniform cross-section.

Table II.2 Representative On-chip, Packaging and Board Level Interconnects

| | On-chip | Packaging | Board |
|---|---|---|---|
| Length | 10 mm | 10 cm | 100 cm |
| Series Resistance | 10 $\Omega$/mm | 1 $\Omega$/mm | |
| Cross Section Dimension | 1 $\mu$m$\times$1$\mu$m | 8 $\mu$m$\times$4.5 $\mu$m | 3 mil$\times$ 1.4 mil |
| Dielectric Materal | $SiO_2$ | Ceramic | FR4 |
| Dielectric Constant | 3.9 | 3.4 | 3.4 |
| Loss Tangent | 0.00068 | 0.0018 @10GHz | 0.010 |
| Frequency dependency | small | large | significant |
| Operation Region | RC | RLC | RLC |
| Skin depth of barrier copper | 0.66$\mu$m @ 10GHz | | |

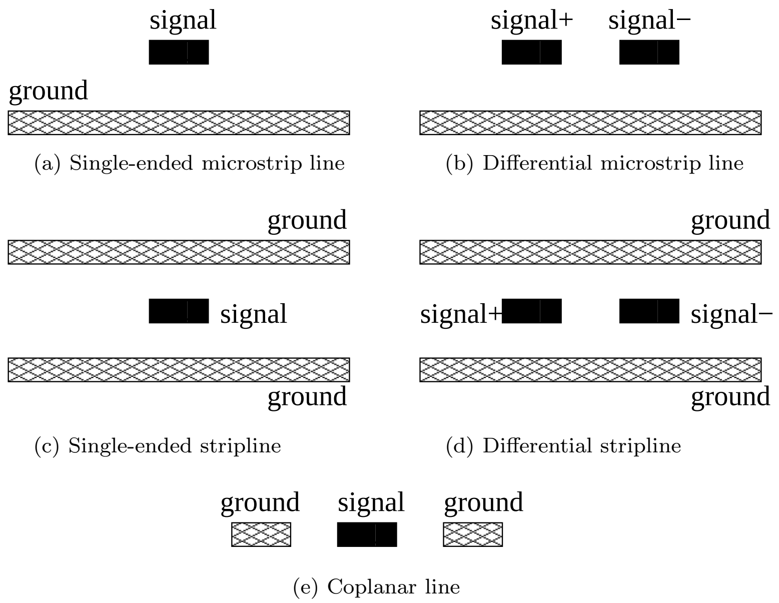
loss tangent, resulting in noticeable substrate loss at high frequencies. Overall, the frequency dependency leads to ISI (Intersymbol Interference) effect, which is the main limiting factor of bandwidth at the packaging level.

### Board Level Interconnects

Board level interconnects display the most severe frequency dependency because the wire width/thickness are much larger than the skin depth at the frequency of interest. Therefore, ISI control is of utmost interest to board level designers.

Physically, T-line usually requires a larger width/spacing than RC wire with standard pitch. More importantly, to form a T-Line structure well-defined return path must be provided so as to control the inductance. For example, Fig. II.2 shows some classical T-Line structures that are suitable for on-chip implementation.

In this chapter, we propose a passive compensation scheme to minimize distortion induced by ISI effect. Our method is based on the theory of distortionless transmission line which has the unique property of enabling speed-of-light transmission with perfect signal fidelity. Evenly distributed shunt resistors are added between the signal line and ground to mimic the behavior of distortionless transmission line.



(a) Single-ended microstrip line

(b) Differential microstrip line

(c) Single-ended stripline

(d) Differential stripline

(e) Coplanar line

Figure II.2 Classical transmission line structures

The rest of the chapter is organized as follows. In Section II.B we review the fundamentals of signal integrity and summarize some of the previous work on high-speed electrical signaling. The theory of distortionless transmission line is then covered in Section II.C. Section II.D provides real design cases to demonstrate the superiority of the proposed technique, and practical concerns such as optimal shunt value and spacing are discussed. In Section II.E an analytical method for predicting worst case eye-opening and jitter from arbitrary bitonic step response is described. This technique enables fast sensitivity analysis at the best shunt value. The chapter ends with a summary in Section II.F.

## II.B Preliminaries and Previous Work

### II.B.1 Fundamentals of Signal Integrity

Digital signals are broadband signals which comprise a wide spectrum of frequency content up to the knee frequency [12],

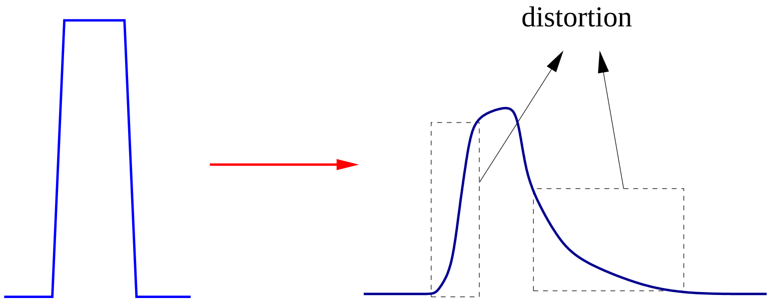


Figure II.3 Distortion due to the frequency dependency of the channel

$$f_{\text{knee}} = \frac{0.35}{t_r} \tag{II.1}$$

where $t_r$ is the rise time of the digital pulse. Thus the faster the digital signal switches, the more high-frequency components is contains. On the other hand, wireline communication channels are by nature low-pass systems. High-frequency components of a digital signal tend to travel faster but attenuate more compared to its low-frequency counterpart. The effect of this frequency dependency in time domain is that a transmitted digital pulse will be distorted at the receiver end, resulting in reduced amplitude and broadened pulse width (long tail). In the case of a digital pulse train, the tails of the proceeding bits will interfere with the succeeding bits, making the signals hard to discern. This is typically referred to as the inter-symbol interference (ISI) effect, and is understood to be the major factor that limits the bandwidth of transmission line.

To assess the quality of the received signal, an eye-diagram is usually created by folding the signal over a period of multiple cycle time. The eye-diagram is characterized by two metrics: jitter and eye-opening. The former quantifies the timing error of the signal while the latter marks the peak-to-peak voltage of the eye.

Various factors contribute to the total jitter, and a complete treatment of them is out of the scope of this dissertation. Among many others, we are interested in the ISI-induced jitter. Such kind of jitter is data-dependent; different

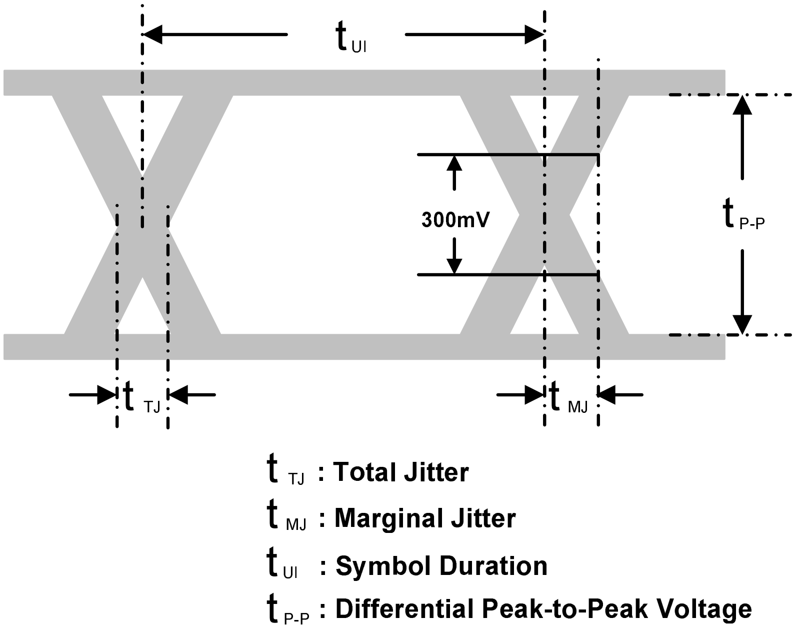


Figure II.4 Eye diagram

bit sequence will likely to produce different amount of jitter, and it is of interest to bound the worst-case jitter for all possible input bit sequences.

### II.B.2 Previous Work

In [30] and [18], Ito et al. proposed to use differential transmission line for global layer signaling, and reported 8Gbps of measured data rate with 180nm technology. Hashimoto et al. further investigated the performance limitation of transmission lines under present and future technologies. In [22], they compared single-ended and differential transmission line against conventional buffered RC wire in terms of bit rate capacity and energy, and suggested that differential transmission line is the most efficient. In [55], Akira et al. studied the tradeoffs between bit rate, interconnect length and eye-opening for both single-ended and differential transmission lines. More realistic situation of including power/ground noise in designing transmission line is considered in [23].

However, conventional RLC transmission line is not once for all. As we have discussed in Section II.B.1, when the frequency gets even higher the ISI effect starts to kick in and limits the bandwidth.

Various techniques were proposed to suppress the ISI effect. In [13], a pre-emphasis filter at the transmitter side was adopted to compensate for the high-frequency loss. Ron Ho et al. proposed to use a clocked discharging scheme to realize time-domain equalization directly on the wire [26]. In [5] and [32], both groups of researchers proposed to implement nonlinear transmission line by adding variable capacitors. The nonlinear transmission line supports solitons which can propagate with little dispersion and loss. Frequency modulation borrowed from RF communication is also attempted. In [10], a 1Gbps data sequence is modulated onto a 7.5Gbps carrier to ensure transmission in the LC region. Jose et al. suggested that by reducing the duty cycle of a RZ (return zero) bit piece, the frequency content of the data can be pushed more to the higher spectrum. More recently, resistive termination for transmission line has been demonstrated to be an effective way for ISI control in [17] and [56]. Both work derived analytical formula for optimal termination resistance.

In this chapter, we apply the theory of distortionless transmission line which states that by making $RC = GL$ we can achieve frequency independent phase velocity and attenuation over the whole spectrum. In the ideal case, all the frequency components travel at the same speed (the speed of light) and all the frequency contents undergo the same amount of attenuation. The overall effect is that the transmitted pulse arrives at the receiver side with perfectly preserved shape and a reduced amplitude. The proposed passive compensation scheme explicitly adds leakage resistors between the signal line and ground (or between the two signal lines in the case of differential signaling) to meet the condition.

The contributions of this chapter can be summarized as follows:

- We experimentally show that, thanks to distortionless transmission, not only the ISI is highly suppressed, but also the jitter and edge rate are well-controlled. Small jitter and high edge rate at the receiver side are essential to avoid timing errors.

- By real design cases we show that the shunt resistors can be easily implemented using conventional high-resistive poly for on-chip application and carbon paste for packaging application, respectively.
- By properly designing the wire geometry and length, we argue that termination for the distortionless transmission line is not necessary. Thus the proposed distortionless transmission line is compatible with conventional static CMOS buffers. No special transceiver circuitry is needed.
- Because the distortionless transmission line does not take full swing, and because there is no active components on the wire, the power consumption is very low. We show that, contrary to our intuition, power per bit for the distortionless transmission line actually decreases with the data rate.
- We develop an analytical method for predicting the worst-case jitter and eye-opening based on any given bitonic response. The analytical prediction technique allows us to quickly explore the design space and maximize the benefit of the proposed passive compensation scheme.

## II.C Theory of Distortionless Transmission Line

### II.C.1 Single-ended Transmission Line

We include the theory of transmission line [45] for self-completeness of this chapter. The telegrapher's equations of the transmission line is the fundamental theory behind almost all kinds of electrical interconnects, being it on-chip, packaging level or board level interconnect. Rather than lumped circuit theory, transmission line theory treats the wire as the conglomerate of numerous infinitesimal RLGC segments, one of which is shown in Fig. II.5, where R, L, G, C are per unit length electrical properties defined as follows:

- R = series resistance per unit length, in $\Omega$/m.
- L = series self loop inductance per unit length, in H/m.

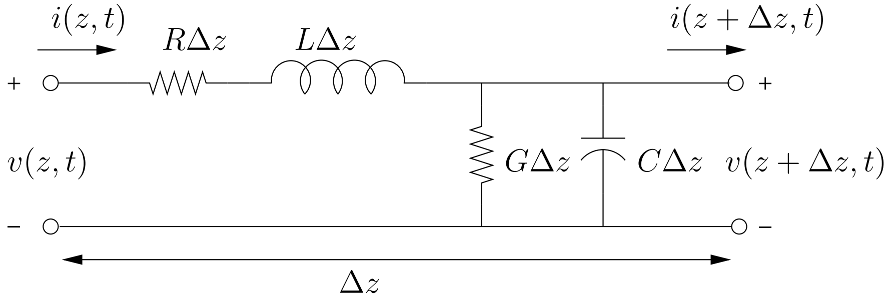


Figure II.5 RLGC model of a transmission line segment.

- G = shunt conductance per unit length, in S/m.

- C = shunt capacitance per unit length, in F/m.

The voltage and current on the transmission line appear in the form of wave propagation; they are both functions of propagation distance $z$ and time $t$, and are governed by the telegrapher's equations:

$$\frac{\partial V(z,t)}{\partial z} = -RI(z,t) - L\frac{\partial I(z,t)}{\partial t} \tag{II.2}$$

$$\frac{\partial I(z,t)}{\partial z} = -GV(z,t) - C\frac{\partial V(z,t)}{\partial t} \tag{II.3}$$

Assuming sinusoidal steady-state condition, by solving the above telegrapher's equations we can get the expression of the incident wave (which travels in the $z+$ direction):

$$V^{+}(z) = V_0^{+}e^{-\gamma z} = V_0^{+}e^{-\alpha z - j\beta z} \tag{II.4}$$

where

$$\gamma = \alpha + j\beta = \sqrt{(R + j\omega L)(G + j\omega C)} \tag{II.5}$$

is the complex propagation constant. From Equation II.4 we see that the amplitude of the traveling wave is $A(z) = V_0^{+}e^{-\alpha z}$. Thus $\alpha$ is usually referred to as the *attenuation constant*, since 1 volt will attenuate to $e^{-\alpha}$ volt after traveling one unit distance. Similarly $\beta$ is called the *phase constant*, because $\beta z$ gives the phase of the voltage wave at location $z$. The velocity of the traveling wave is

$$v = \frac{\omega}{\beta} \tag{II.6}$$

The characteristic impedance of the line is defined as the ratio of voltage to current on any point of the line:

$$Z_0 = \frac{V^+(z)}{I^+(z)} = \sqrt{\frac{R + j\omega L}{G + j\omega C}} \tag{II.7}$$

Note that the transmission line supports waves in both $z+$ and $z-$ directions. So the general solution to Equation II.2 and II.3 is

$$V(z) = V^+(z) + V^-(z) = V_0^+ e^{-\gamma z} + V_0^- e^{\gamma z} \tag{II.8}$$

Typical on-chip global interconnects are very lossy; The series resistance of the global interconnects is usually at the order of $10\Omega$/mm. On the other hand, the silicon dioxide is a very good insulator, whose loss tangent[2] is only 0.00068. Thus for on-chip transmission line, the shunt conductance $G \approx 0$. Under these conditions, on-chip transmission line could operate in either RC region or LC region, depending on the frequency of interest.

**RC Region:**

When the frequency $\omega$ is low, we have $\omega L \ll R$ and $G \approx 0$. Equation II.5 simplifies to

$$\begin{aligned} \gamma &= \alpha + j\beta = \sqrt{j\omega RC} \\ &= \sqrt{\frac{\omega RC}{2}} + j\sqrt{\frac{\omega RC}{2}} \end{aligned} \tag{II.9}$$

Hence

$$\alpha = \sqrt{\frac{\omega RC}{2}} \tag{II.10}$$

$$v = \frac{\omega}{\beta} = \sqrt{\frac{2\omega}{RC}} \tag{II.11}$$

We see that in RC region, both the attenuation constant and phase velocity are functions of the frequency. More specifically, the lower the frequency, the less the

[2] At high frequencies, leakage currents appear in the dielectric due to the ionization of the atoms. Meanwhile, the periodic oscillation of the magnetic dipoles of the atoms also dissipates energy. These two factors contribute to the signal loss in the dielectric, and are usually indistinguishably quantified by the loss tangent (or dissipation factor) of the material.

attenuation, and the lower the speed. For on-chip interconnect, this $\omega L \ll R$ condition is usually satisfied in up to 10GHz.

**LC Region:**

If the frequency keeps increasing such that $\omega L \gg R$, and $G \approx 0$, then Equation II.5 reduces to

$$\begin{aligned} \gamma = \alpha + j\beta &= \sqrt{(R + j\omega L)j\omega C} \\ &= \frac{R}{2\sqrt{L/C}} + j\omega\sqrt{LC} \end{aligned} \tag{II.12}$$

Thus

$$\alpha = \frac{R}{2\sqrt{L/C}} = \frac{R}{2Z_0} \tag{II.13}$$

$$v = \frac{\omega}{\beta} = \frac{1}{\sqrt{LC}} = \frac{c_0}{\sqrt{\epsilon_r}} \tag{II.14}$$

Where $c_0$ is the speed of light in free space and $\epsilon_r$ is the dielectric constant. We see that in the LC region, if neglecting the variation of R and L, both attenuation and phase velocity are independent of frequency. This result provides the theoretic foundation for the work of [10] and [31], which seek to modulate the low-frequency content to the LC region.

**Distortionless Transmission Line:**

Interestingly if we set

$$\frac{R}{G} = \frac{L}{C} \tag{II.15}$$

and substitute the relation in to Eqn. (II.5), we have

$$\gamma = \alpha + j\beta = \sqrt{(R + j\omega L)(RC/L + j\omega C)} \tag{II.16}$$

$$= \frac{R}{\sqrt{L/C}} + j\omega\sqrt{LC} \tag{II.17}$$

Therefore

$$\alpha = \frac{R}{\sqrt{L/C}} = \frac{R}{Z_0} \tag{II.18}$$

$$v = \frac{\omega}{\beta} = \frac{1}{\sqrt{LC}} = \frac{c_0}{\sqrt{\epsilon_r}} \tag{II.19}$$

Likewise, plug Eqn. (II.15) into Eqn. (II.7), we have

$$Z_0 = \sqrt{\frac{L}{C}} \tag{II.20}$$

Note in Equation II.15 there is no assumption about $\omega$, thus in this case we obtain frequency-independent attenuation and phase velocity across the whole spectrum. And the phase velocity is actually the speed of light in the dielectric. The characteristic impedance (Equation II.20) becomes pure resistive.

Equation II.15 is usually referred to as Heaviside condition to credit Oliver Heaviside who first discovered this elegant result [24]. Based on this result Heaviside proposed to deliberately add inductance for transatlantic cable to achieve distortionless communication, and that was one hundred years ago! For on-chip and packaging level applications, inductance is hard to control, instead we could evenly insert leakage conductance to meet the Heaviside condition.

### II.C.2 Differential Transmission Line

A small revisit of the distortionless condition is needed for differential transmission line, a circuit model of which is shown in Figure II.6. In the case of differential signaling (i.e., currents in the two signal lines flow in the opposite directions), we can derive that the leakage resistance between the two signal lines shall satisfy

$$R_x = \frac{2Z_{\mathrm{odd}}^2}{R} \tag{II.21}$$

where $Z_{\mathrm{odd}}$ is the odd-mode characteristic impedance given by

$$Z_{\mathrm{odd}} = \sqrt{\frac{(1-M)L}{C+C_x}} \tag{II.22}$$

If the differential pair operates in common mode, that is, currents in the two signal lines flow in the same directions, the distortionless condition becomes

$$R_s = \frac{Z_{\mathrm{even}}^2}{R} \tag{II.23}$$

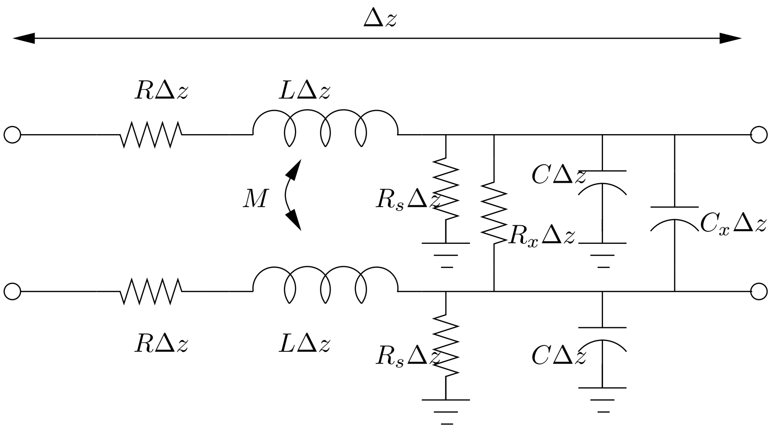


Figure II.6 RLGC segment of a differential transmission line

where $Z_{\text{even}}$ is the even-mode characteristic impedance given by

$$Z_{\text{odd}} = \sqrt{\frac{(1+M)L}{C}} \tag{II.24}$$

Nevertheless, in differential pairs the common mode signal is usually perceived as noise, thus it is of more interest to compensate the differential mode signal using Equation II.21.

## II.D Case Studies

### II.D.1 On-Chip Interconnect

In this section, we use a real on-chip design case to illustrate the superiority of the distortionless transmission line. To get a better demonstration we use a single-ended microstrip line instead of a differential pair, although the latter is shown to be able to achieve higher data rate [22]. We choose 0.10$\mu$m as the target technology, and implement the microstrip line using M7 and M9, as shown in Fig. II.7. We assume the resistivity of barrier copper, $\rho$=2.2e-06 S/cm. The line length is 10mm.

The first step in designing the distortionless transmission line is to determine the shunt conductance. In Equation II.15, a hidden assumption is that RLC values of the transmission line is independent of frequency so that we can find a

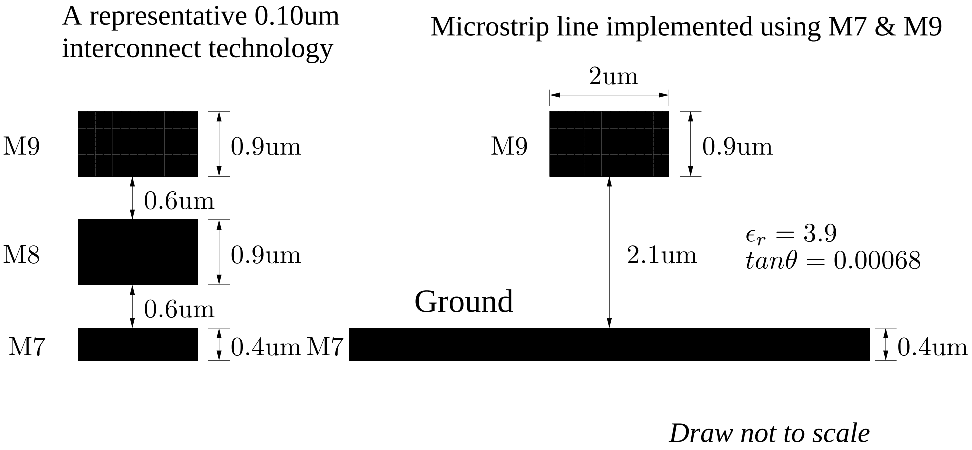


Figure II.7 A microstrip line implemented in 0.10$\mu$m technology.

single G to meet the heaviside condition. However, this is not the real scenario. For on-chip interconnect, although RLC values do not vary much, they do change over frequency. Thus a single conductance value for meeting the Heaviside condition at every frequency point is not possible. This brings up the question of what is the optimal shunt conductance.

We use a 2D EM solver called CZ2D from IBM to extract the RLGC values of the microstrip line up to 50GHz, and the results are shown in Figure II.8. CZ2D has the capability of considering both skin effect and proximity effect yet it is much faster than 3D extraction tools such as Raphael [1]. Clearly, as the frequency passes roughly 1GHz, the line resistance starts to increase due to the skin effect. Meanwhile, the total inductance decreases because the internal inductance of the line vanishes when currents rush to the surface of the conductor. The capacitance, on the other hand, is virtually constant over the whole spectrum. The leakage conductance due to the dielectric, though increases sharply at high frequencies, is still negligible compared to the series line resistance. The high-frequency characteristic impedance of the line is $Z_0$=54.9Ω, and the time of flight is 65.87ps/cm.

As we discussed in Section II.C, at high frequencies the transmission line operates in the good LC region. It is in the RC region that both attenuation and

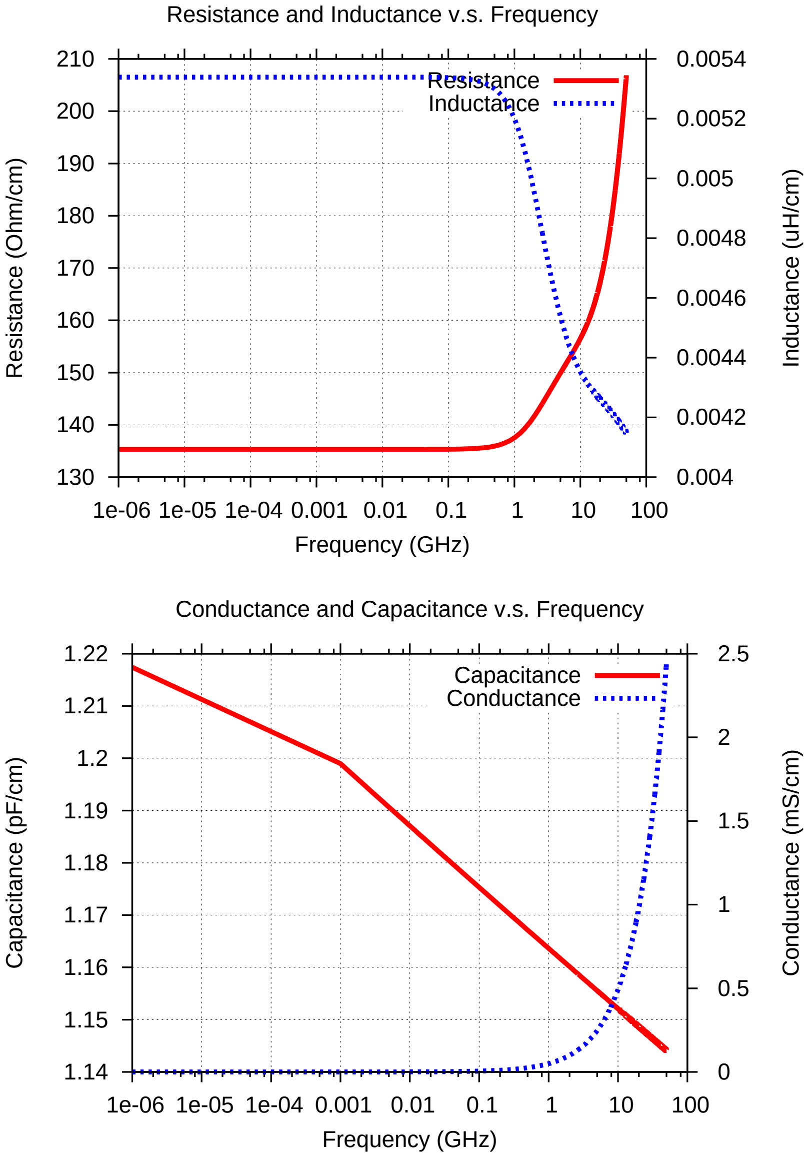


Figure II.8 RLGC values v.s. frequency for the 10 mm microstrip line.

phase velocity see great frequency dependency. Therefore, our rule of thumb for determining shunt conductance is “match-at-DC”. Another way to understand this rule is to realize that the shunt scheme is purely passive. Rather than boosting up high-frequency components (which is not possible without active compensation/equalization), we use shunt resistors to make the low-frequency components attenuate more and travel faster.

At DC mode, we have $R_{\mathrm{DC}}$=135.3Ω/cm, $C_{\mathrm{DC}}$=1.22pF/cm, $L_{\mathrm{DC}}$=5.34e-

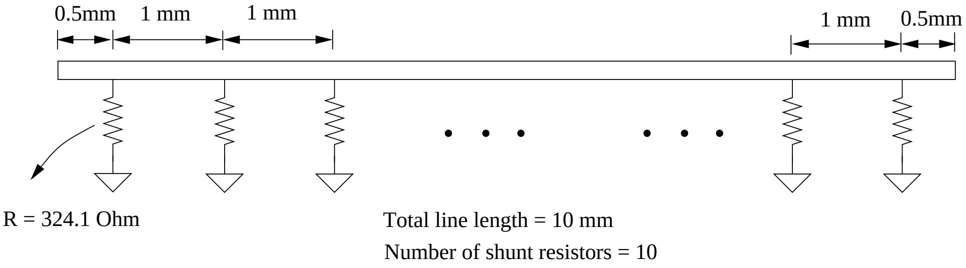


Figure II.9 Implementation of distortionless transmission line

03$\mu$ H/cm. The total shunt resistance needed for meeting the Heaviside condition is

$$R_{\text{shunt,total}} = \frac{L_{\text{DC}}}{R_{\text{DC}} C_{\text{DC}}} = 32.41\Omega/cm. \tag{II.25}$$

Assuming we evenly insert $N$ shunt resistors for the 10mm line, each resistor has the value

$$R_{\text{single}} = N \cdot R_{\text{shunt,total}} \tag{II.26}$$

In theory, we can increase $N$ so that the line approaches a distributive distortionless transmission line. However, inserting too many shunt resistors can be prohibitive in terms of silicon resources, nor is it necessary. According to our simulation, if the spacing of the shunt resistors is less than the critical length $l_{\text{crit}} = \frac{c}{\sqrt{\epsilon_r}} \cdot t_r$, where $t_r$ is the rising time of the signal, the transient response of the system becomes very close to that of a real distributive distortionless transmission line. If the system is targeted for 10Gbps data transmission, the critical length $l_{\text{crit}}$ is roughly 1.5mm. Thus, for our 10mm microstrip line we insert a shunt resistor every 1mm. Each shunt resistor is 324.1$\Omega$. The design is shown in Fig. II.9.

Note that a resistor of 324.1$\Omega$ can be easily implemented on-chip. For example, the sheet resistance of $n^+/p^+$ unsilicided polysilicon is 150$\sim$200$\Omega/\Box$ for a typical 0.25$\mu$m technology [46]. Hence the shunt scheme is completely compatible with the conventional silicon process.

Figure II.10 shows the pulse responses of the 10mm line w/o shunts and with 10 shunt resistors inserted. The input is a trapezoidal pulse with 10ps rising/falling time and 90ps duration. For the typical RLC transmission line without

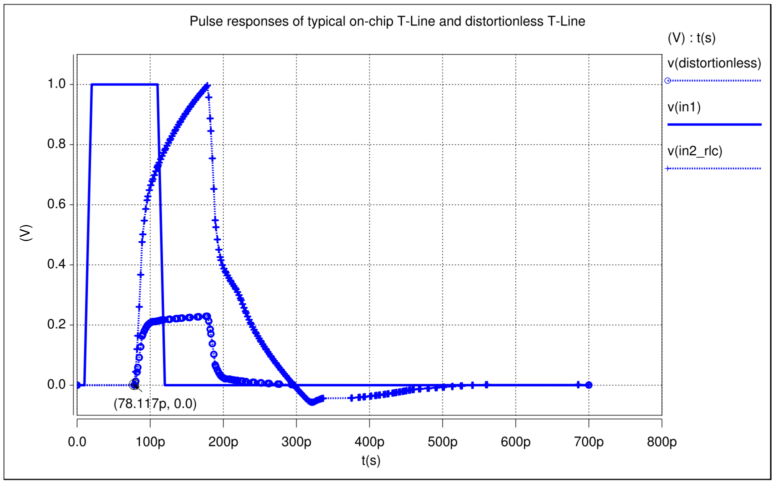


Figure II.10 Pulse responses of the 10mm microstrip line

shunts, we see both slow rising top and long tail which will lead to significant ISI. For the transmission line with 10 shunts, the pulse shape is largely preserved, but the height is reduced to approximately 0.23 volt. The DC saturation voltage, in this case, is determined by the resistance ladder formed by the series line resistance and shunt resistors. At the receiver side, a state-of-the-art sense amplifier can easily detect this amount of voltage. Both responses rise at about 78ps, which corresponds to the time of flight of the 10mm line.

The advantage of the shunted transmission line can also be explained in terms of the attenuation and phase velocity. Figure II.11 shows $e^{-\alpha}$ and phase velocity for both shunted transmission line and typical transmission line. We see that for the shunted transmission line, the variation of attenuation over frequency is less severe than that of typical transmission line, although the overall curve is lower. More importantly, for shunted transmission line the velocity of the low-frequency components is greatly boosted up and the phase velocity curve is flatter. In the case of typical RLC transmission line, the signal speed goes up from almost zero at DC and saturates at the speed of light when the frequency increases.

To evaluate the performance of the shunted transmission line, we extract the 2-port S-parameter of the 0.5mm/1mm segments using CZ2D, and then build

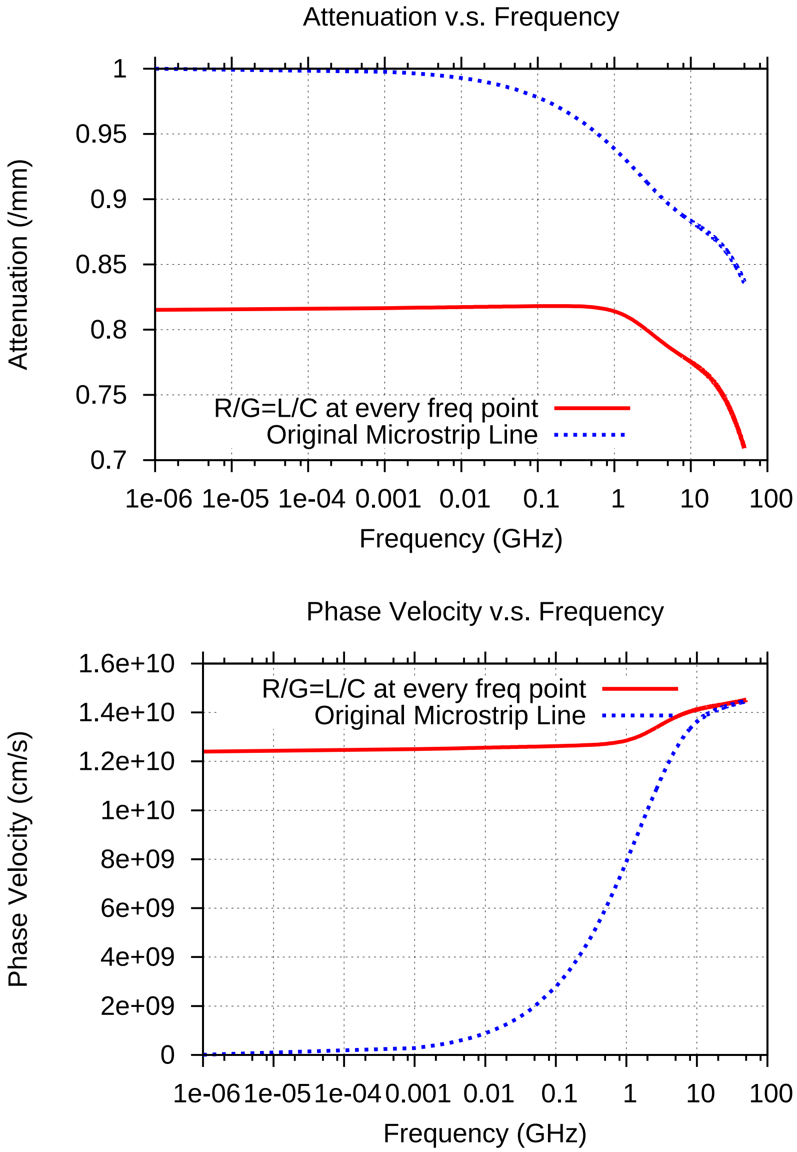


Figure II.11 Attenuation and phase velocity of the 10mm microstrip line

the schematic of Figure II.9 in Hspice. Each 0.5mm/1mm wire segment is modeled using W-element with the S-parameters. The input is 1000 bit pieces of a 10Gbps pseudo random bit sequence (PRBS). The rising/falling edges are set to be 10% of the cycle time. The eye diagram at the receiver side, as shown in Figure II.12, shows extremely clear eye opening and small jitter. Note that in this example no termination is needed for the receiver since the reflected signal attenuates to almost zero after a round trip.

Table II.3 Average power and energy per bit at different bit rate

| Bit Rate (Gbps) | 5 | 10 | 20 | 40 |
|---|---|---|---|---|
| $P_{\text{avg}}$ (mW) | 2.5624 | 2.5754 | 2.6544 | 2.7612 |
| $E_{\text{bit}}$ (pJ) | 0.5125 | 0.2575 | 0.1327 | 0.0690 |

A practical concern for the proposed shunted distortionless transmission line is the static power consumption, since in this case we have direct DC path to the ground. We show that, as long as the line operates at a reasonable data rate, static power consumption is rarely a problem. We measure the average power $P_{\text{avg}}$ of 1000 bit pieces at different data rate in Hspice, and calculate the energy per bit number using the following equation:

$$E_{\text{bit}} = P_{\text{avg}} \cdot T_{\text{cycle}} \tag{II.27}$$

where $T_{\text{cycle}}$ is the cycle time.

As we see from Table II.3, $P_{\text{avg}}$ only increases slightly as we double the data rate. This is actually a unique property of transmission line signaling. In conventional RC wires, the power is used to charge and discharge the whole wire segment. In transmission lines, the energy, once injected, actually propagates down the line, and the power dissipated is due to the line loss. As we increase the data rate, $P_{\text{avg}}$ also increases because high-frequency signals attenuate more. However, since the energy of a digital signal concentrates on the low-frequency side, increased high-frequency attenuation does not increase the overall power consumption much. The energy consumed per bit, therefore, decreases for higher bit rate.

### II.D.2 Multi-chip Module Interconnect

Multi-chip module interconnects are usually buried in the packaging substrate in the form of microstrip line. Figure II.13 shows the specification used in IBM high-end AS/400 system [44], with the modification that the signal line thickness is reduced from 4.5$\mu$m to 2$\mu$m to minimize the skin effect. We assume the resistivity of pure copper, $\rho$=1.72e-06 $\Omega$·cm. For the insulator we assume liquid

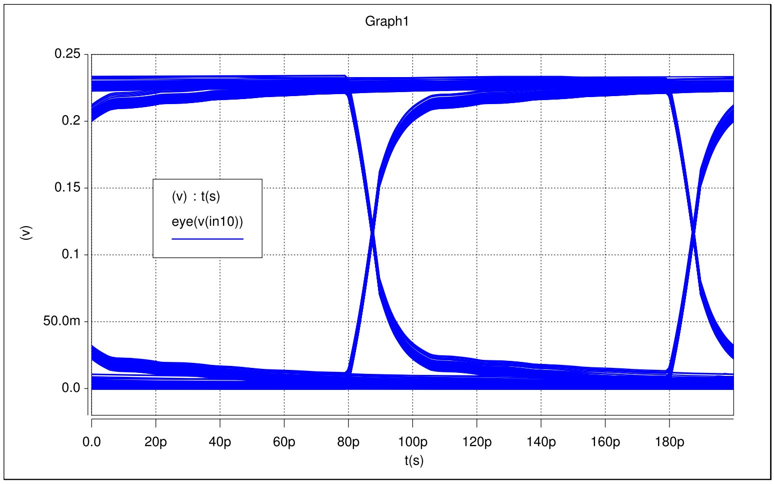


Figure II.12 Eye diagram of the 10mm microstrip line with 10 shunt resistors inserted

crystal polymer (LCP) which has a small dissipation factor of 0.0025 yet is much cheaper than ceramics [54]. The line length is 10cm.

We carry out the same procedure to evaluate the effect of adding distributed shunt resistors. The RLGC parameters of the line are first extracted by CZ2D, a 2-D extraction tool in the IBM Electronic Interconnect and Packaging suite [2]. The phase velocity and attenuation curves of the uncompensated line as well as the matched line are then calculated, which is shown in Figure II.15. It is interesting to note that for MCM stripline trace, the phase velocity curves of the uncompensated line and matched line merge at about 1GHz, while this frequency is about 10GHz for the on-chip microstrip line in Section II.D.1.

The high-frequency characteristic impedance of the line is $Z_0$=78Ω. Note that the characteristic impedance of typical PCB traces is 50Ω, thus if the MCM trace goes off to the board there will be impedance mismatch. However, if the MCM trace stays on the packaging we would rather have a larger $Z_0$ because that will minimize the attenuation. The line delay is 57.78ps/cm.

We again add 10 shunts (1 shunt per cm) to the stripline and use the matching conductance at DC. At 1MHz, we have $R$=11.07Ω/cm, $C$=0.74pF/cm

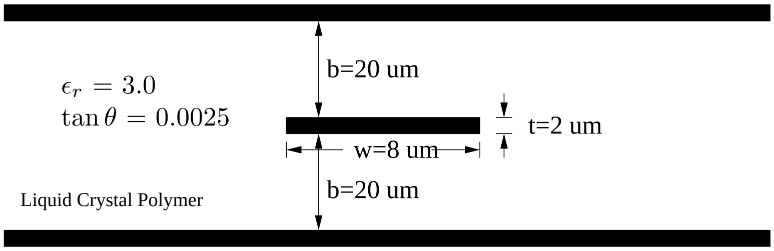


Figure II.13 Cross section of the 10cm MCM stripline trace

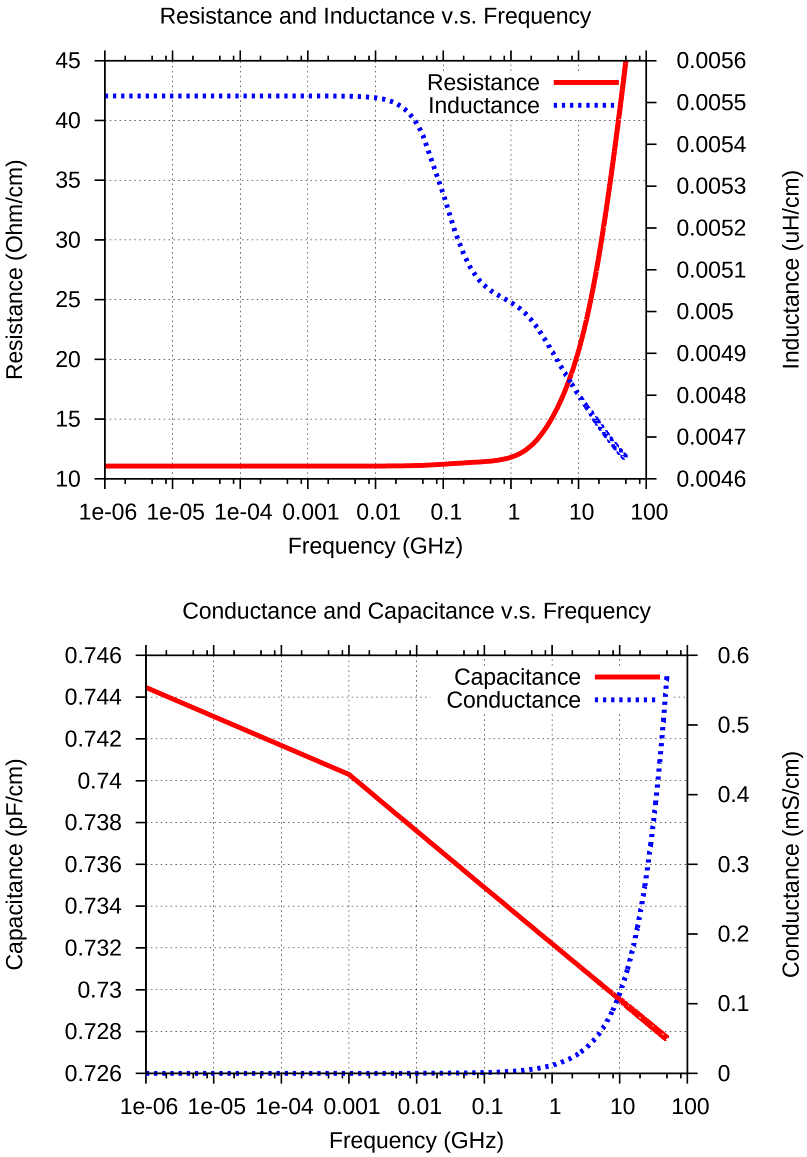


Figure II.14 RLGC values v.s. frequency for the 10cm MCM stripline

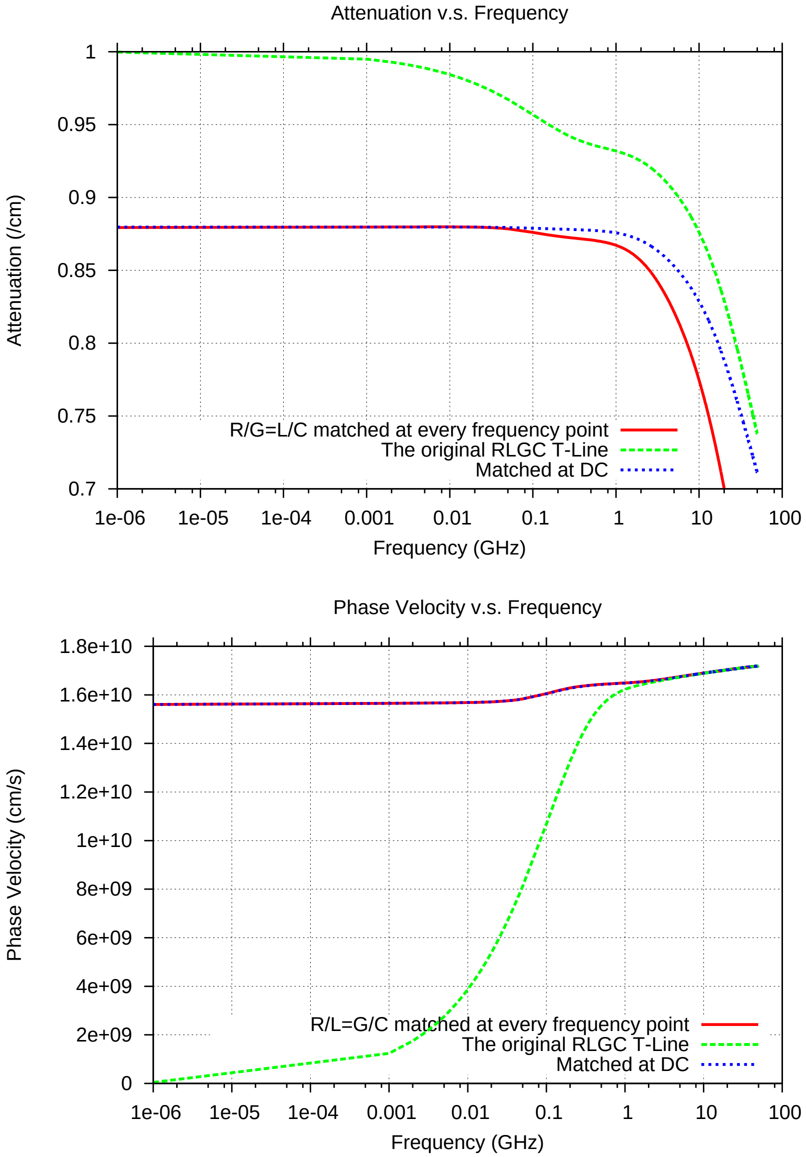


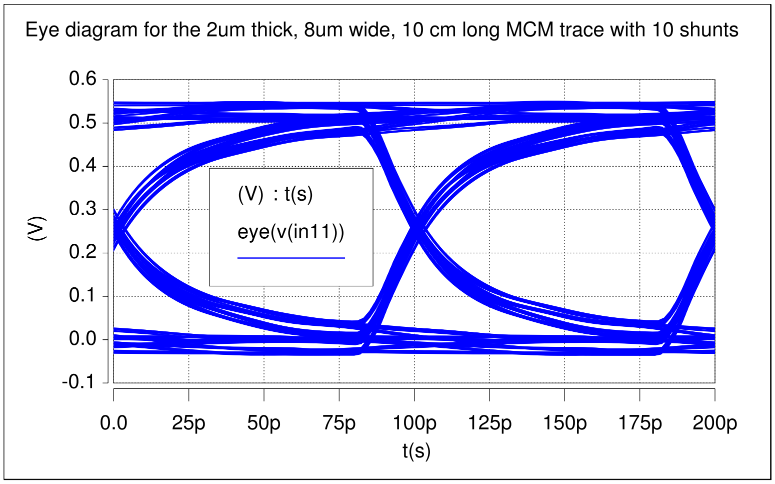
Figure II.15 Attenuation and phase velocity of the 10cm MCM stripline

and $L$=5.52e-03$\mu$H/cm. The total shunt resistance needed for meeting the Heaviside condition for the 10cm stripline is

$$R_{\text{shunt,total}} = \frac{L_{\text{1MHz}} \cdot 10cm}{(R_{\text{1MHz}} \cdot 10cm)(C_{\text{1MHz}} \cdot 10cm)} = 66.95\ \Omega \qquad \text{(II.28)}$$

This translates to 669.5$\Omega$ for each of the 10 shunt resistors.

Figure II.16 shows the the eye diagram of the 10cm stripline when ten 669.5$\Omega$ shunt resistors are evenly added. We see that under the input of a 1000-bit



Each shunt is 669.5Ω

Figure II.16 Eye diagram of the 2$\mu$m-thick MCM stripline with 10 shunts inserted

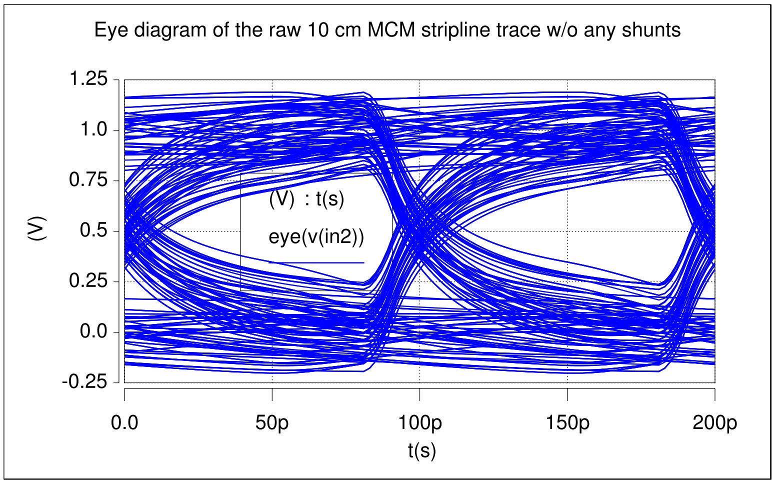


Figure II.17 Eye diagram of the 2$\mu$m-thick MCM stripline without shunts

pseudo random bit sequence at 10Gbps, the maximum eye opening voltage is 0.42V and the jitter is only 6.1ps. Without any shunt compensation, the eye diagram in Figure II.17 shows a eye-opening of 0.51V out of 1V DC saturation voltage, and the jitter is 22.5ps.

We also simulated the 10cm MCM stripline of 4.5$\mu$m thickness, and the corresponding eye-diagrams are shown in Figure II.18 and Figure II.19. For the uncompensated line, increasing the thickness exacerbates its frequency dependency,

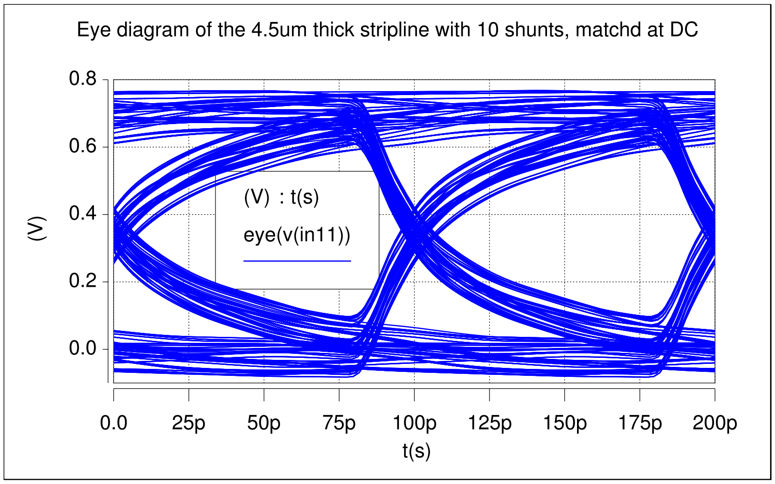


Figure II.18 Eye diagram of the 4.5$\mu$m-thick MCM stripline with 10 shunts inserted

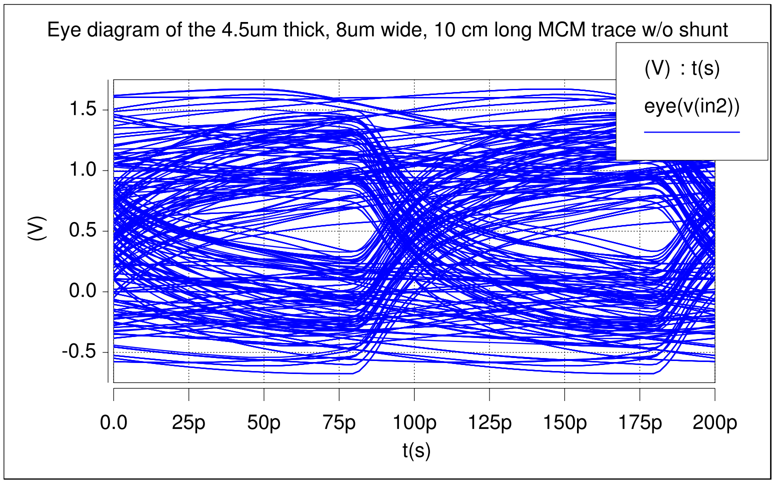


Figure II.19 Eye diagram of the 4.5$\mu$m-thick MCM stripline without shunts

which makes the eye-diagram unrecognizable. The compensated line in this case still performs quite well, and a 11.6ps jitter and an eye-opening of 0.476V are observed.

## II.E Analytical Eye-Diagram Modeling

We have studied the advantages of and proposed design guidelines for the passive compensation scheme using distributed shunt resistors. Many interesting questions regarding the shunt scheme remain. For example, for a given transmission line configuration we might wish to determine the shunt value that produces the minimum jitter as well as the sensitive of the jitter with respect to the shunt value. We might also wish to study the tradeoff between the jitter and eye-opening. Unfortunately, to answer these questions significant amount of simulation work is needed, and in many cases simulation provides little insight for the optimization tasks. To this end, We develop an analytical method to evaluate the jitter and eye-opening of an interconnect system based on given bitonic step response.

We confine our discussion to data dependent jitter only, which is the deviation of data threshold-crossing time from a reference time due to the residue memory of previous data bits. In a linear time invariant (LTI) system such kind of jitter is deterministic, and is unique to an input bit sequence. We are interested in bounding the worst case jitter over all possible input bit sequences.

### II.E.1 Bitonic Step Response Assumption

Due to the wave nature of transmission line, signals undergo multiple reflections if perfect termination is not provided. As a result, the output step response will fluctuate before it settles down to the saturation voltage. Nevertheless, because on-chip and MCM transmission lines are kind of lossy, the reflected wave will diminish to a negligible amount after two round trips. Thus, the output step response usually appears to be bitonic.

**Definition II.E.1** *A step response is defined to be* **bitonic** *if it monotonically increases to its peak voltage and then monotonically decreases to its saturation voltage.*

This process can be further explained by Figure II.20. When the injected step signal reaches the receiver end, it gets completely reflected back toward the source because of open termination. At the near end, the voltage source acts as a short to the ground. Thus the reflected voltage gets bounced back again toward

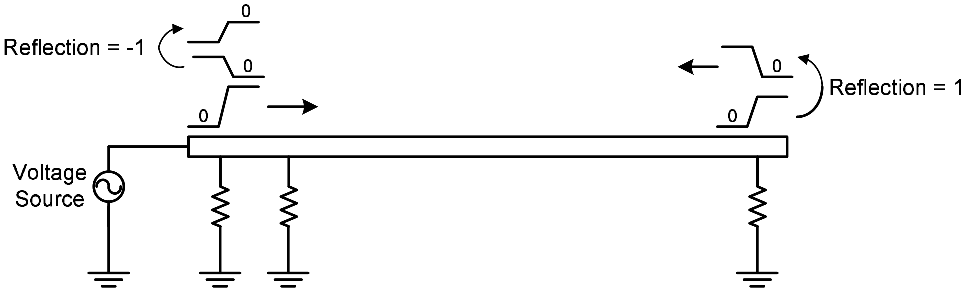


Figure II.20 Generation of bitonic step response

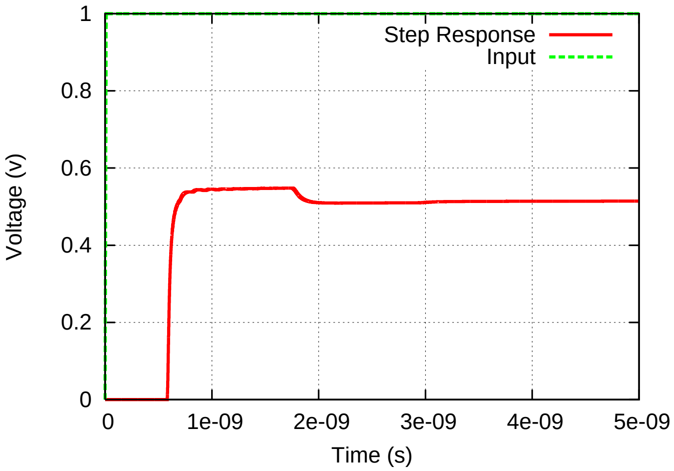


Figure II.21 A bitonic step response

the receiver, but with a inverted magnitude. This inverted voltage will create a dip in the step response at the far end. As an example, Figure II.21 shows the bitonic step response of the MCM case presented in Section II.D.2.

For a generic bitonic step response, we can divided it into pieces of cycle time T as shown in Figure II.22. We characterize the generic bitonic step response $s(t)$ with five pivotal points:

$$\begin{aligned} V_1 &= V(T) \\ V_{\mathrm{max}} &= V(T_0) = \text{maximum voltage of the step response} \\ V_{\mathrm{sat}} &= \text{final saturation voltage of the step response} \\ V_{\mathrm{tp1}} &= V(kT) \;\text{ and }\; V_{\mathrm{tp2}} = V((k+1)T) \\ &\quad \text{where }\; kT \leq T_0 \leq (k+1)T \end{aligned}$$

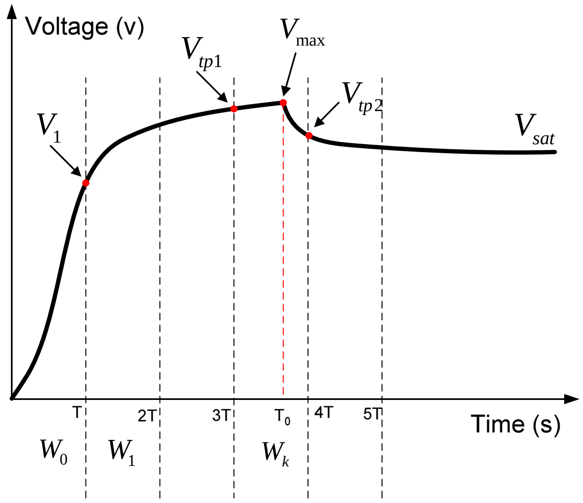


Figure II.22 A generic bitonic step response

Thus the piecewised step response can be represented by

$$s(t) = \{W_0(t), W_1(t+T), \cdots, W_k(t+kT), \cdots\}$$

Given an arbitrary bit sequence $\{a_n\}$, we can generate the output transient response from the step response using the principle of linear superimposition:

$$y(t) = s(t) + \sum_{n=1}^{\infty} b_n \cdot s(t - nT) \tag{II.29}$$

where $s(t)$ is the step response and

$$b_n = a_n - a_{n-1} \quad (a_n \in \{0, 1\}, n \geq 0) \tag{II.30}$$

and $a_{-1} = 0$, $a_0 = 1$ to guarantee a rising edge.

Clearly $\{b_n\}$ captures the edges of the input digital signal; if $b_i = 1$, there is a positive edge from $a_{i-1}$ to $a_i$, and $b_i = -1$ translates to a falling edge from $a_{i-1}$ to $a_i$. Since positive edges and negative edges must appear alternatively in a digital signal, we conclude that the sequence $\{b_n\}$ shall consists of alternative 1's and -1's if all the zeros are deleted.

Before we dive into the detail analysis of the jitter and eye-opening, it is instrumental to have a qualitative view of the eye-diagram. Remember that an

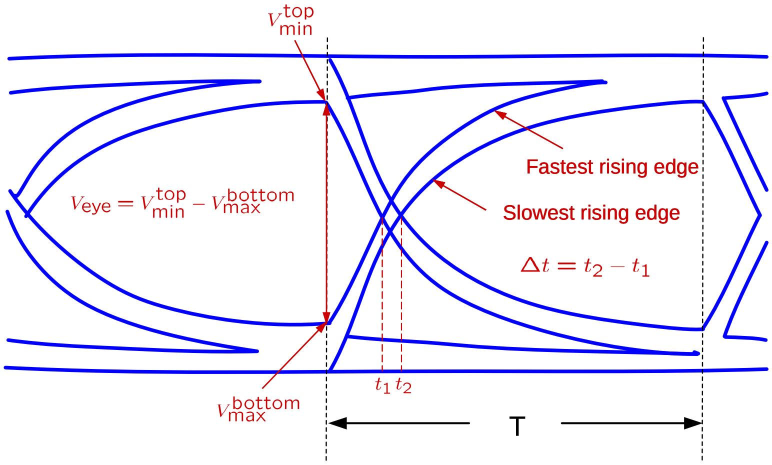

$V_{\text{min}}^{\text{top}}$
$V_{\text{eye}} = V_{\text{min}}^{\text{top}} - V_{\text{max}}^{\text{bottom}}$
Fastest rising edge
Slowest rising edge
$\Delta t = t_2 - t_1$
$t_1$ $t_2$
$V_{\text{max}}^{\text{bottom}}$
T

Figure II.23 A qualitative view of the eye diagram

eye-diagram is generated by cutting the transient response into pieces of multiple cycles and then folding them together. To estimate the eye-opening, we will need to know the minimum voltage a rising edge can get to at time $lT$, denoted as $V_{\text{min}}^{\text{top}}$, and the maximum voltage a falling edge can get to at time $lT$, denoted as $V_{\text{max}}^{\text{bottom}}$. The worst-case eye-opening is then bounded by

$$V_{\text{eye}} = V_{\text{min}}^{\text{top}} - V_{\text{max}}^{\text{bottom}} = V_{\text{sat}} - 2(max\{V_{\text{tp1}}, V_{\text{tp2}}\} - V_1) \tag{II.31}$$

To determine the jitter, we will first extract the fastest rising edge, which will cross the threshold voltage $V_{\text{sat}}/2$ earliest at $t_1$. The slowest rising edge will cross the threshold voltage latest at $t_2$. And the worst-case jitter is bounded by $\Delta t = t_2 - t_1$. Since the eye-diagram is symmetric with regard to its falling edge and rising edge, considering only rising edge is sufficient.

### II.E.2 Worst Case Eye-Opening

In order to produce $V_{\text{top}}^{\text{min}}$, $V_1$ must be superimposed with the lowest sampled voltage on a falling step response, which shall be either $V_{\text{tp1}}$ or $V_{\text{tp2}}$. Thus

$$V_{\text{min}}^{\text{top}} = V_1 + V_{sat} - \max\{V_{\text{tp1}}, V_{\text{tp2}}\} \tag{II.32}$$

Such a scenario is illustrated in Figure II.24.

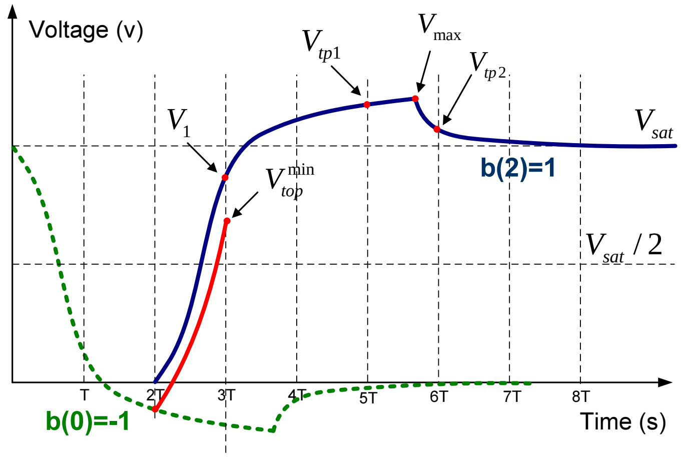


Figure II.24 Minimum rising edge voltage

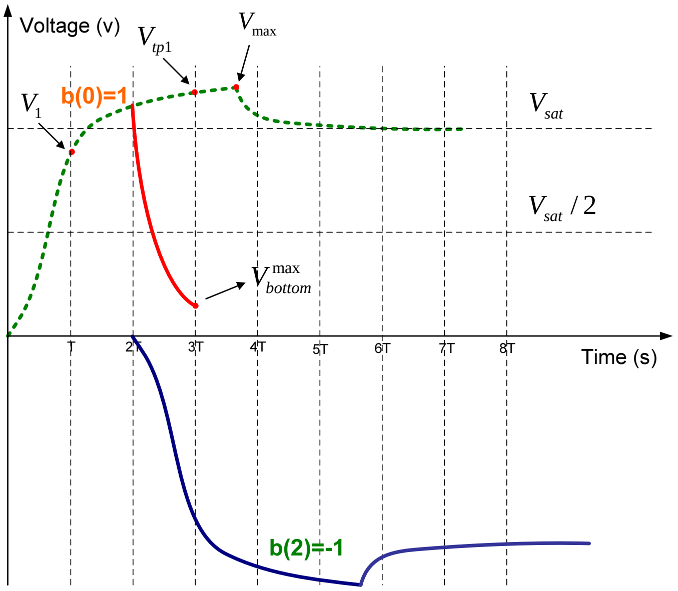


Figure II.25 Maximum falling edge voltage

Likewise, to produce $V_{\max}^{\text{bottom}}$, $-V_1$ is expected to be superimposed with the maximum sampled voltage on a positive step response, as shown in Figure II.25. Therefore

$$V_{\max}^{\text{bottom}} = \max\{V_{\text{tp1}}, V_{\text{tp2}}\} - V_1 \tag{II.33}$$

For the bitonic response in Figure II.21, we can measure that $V_1 = 507.9747e - 03$, $V_{\text{sat}} = 514.3697e - 03$ and $V_{\text{tp1}} = 547.9974 - e03 > V_{\text{tp2}}$. Hence the predicted eye-opening is $V_{\text{eye}} = 0.4343242V$. To compare, the measured eye-opening from Hspice simulation is 0.43112V, and the error by our analytical method is only 0.7%.

### II.E.3 Worst Case Jitter

For the fastest rising edge, we consider a rising-falling-rising combination shown in Figure II.26. Ideally, the fastest rising edge shall rise from the highest possible voltage and has the sharpest slope. However, to have the highest rising voltage and sharpest slope is often contradictive, and it is necessary to check all possible combinations. Let

$$R_i(t) = W_0(t) + W_p(t) - W_i(t) \tag{II.34}$$

and let $t_1^i \in (0, T)$ be the solution of the equation

$$R_i(t_1^i) = \frac{V_{\text{sat}}}{2} \tag{II.35}$$

Then $t_1 = \min\{t_1^i\}$.

Likewise, let

$$F_i(t) = W_0(t) - W_i(t) + V_{\text{sat}} \tag{II.36}$$

and let $t_2^i \in (0, T)$ be the solution of the quation

$$F_i(t_2^i) = \frac{V_{\text{sat}}}{2} \tag{II.37}$$

Then $t_2 = \max\{t_2^i\}$.

For the step response shown in Figure II.21, the above analysis method reports a jitter of 5.5ps, while the measured jitter from Hspice simulation is 5.68ps. The error is only 3.17%.

### II.E.4 Jitter and eye-opening tradeoff

We utilize the proposed analytical method to study the tradeoff between jitter and eye-opening under different shunt schemes. Other than our distributed

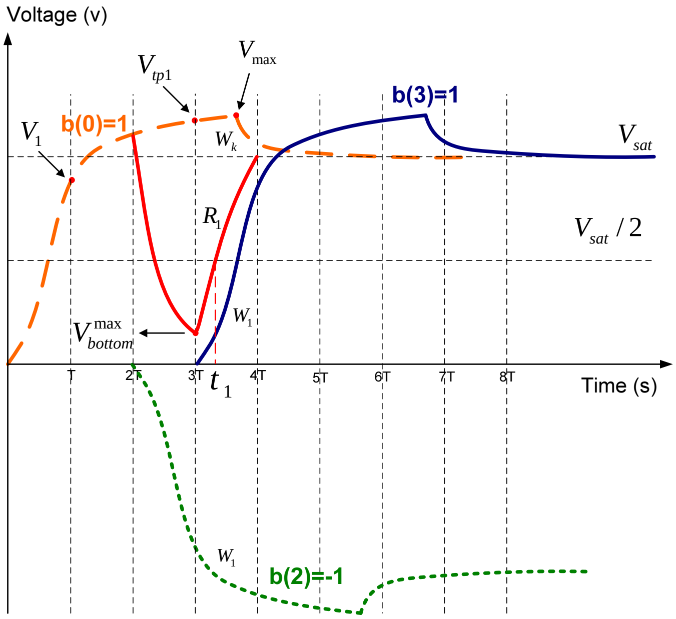


Figure II.26 Fastest rising edge

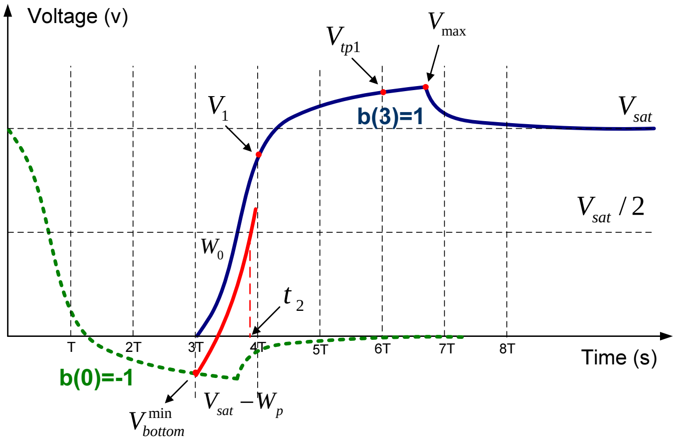


Figure II.27 Slowest rising edge

compensation scheme, another popular method to suppress the distortion is to use resistive termination only.

Four MCM stripline cases have been studied:

**Case I:** Thickness=2$\mu$m, width=8$\mu$m, half dielectric=20$\mu$m, length=10cm

**Case II:** Thickness=2$\mu$m, width=8$\mu$m, half dielectric=20$\mu$m, length=20cm

**Case III:** Thickness=4.5$\mu$m, width=8$\mu$m, half dielectric=20$\mu$m, length=10cm

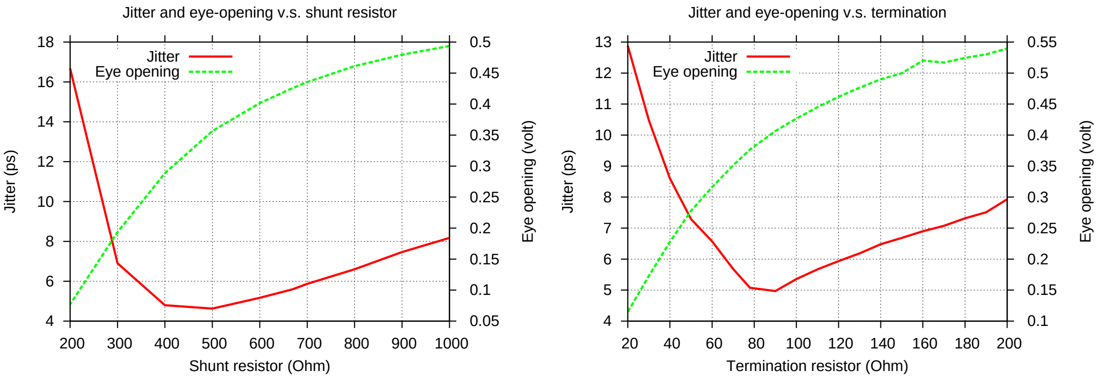


Figure II.28 Jitter and eye-opening tradeoff for the 2um-thick 10cm-long stripline

**Case IV:** Thickness=4.5$\mu$m, width=8$\mu$m, half dielectric=20$\mu$m, length=20cm

For each of the four cases both distributed shunt compensation scheme and resistive termination scheme are analyzed and compared, and the results are illustrated in Figure II.28 to Figure II.31. The shunt density remains to be 1 resistor per 1cm across all occasions. We see that three of the four cases perform better with distributed shunt scheme in terms of jitter. The greatest benefit of the distributed shunt scheme is observed for case III, in which a minimum jitter of 21.76ps is achieved with $800 sim 900\Omega$ of each shunt resistor. By contrast, the minimum jitter is 37.6ps if only resistive termination is used.

Another observation is that at the point of minimum jitter the sensitivity of the jitter is smaller toward larger shunt resistance. Thus in real designs it is advised to use a shunt value that is slightly larger than the optimum.

## II.F Summary

In this chapter we have described a novel passive compensation scheme by intentionally introducing leakage resistors to meet the Heaviside condition. The proposed shunt resistor scheme preserves the low power property of typical transmission line, and pushed the performance to the extreme. We demonstrated the efficacy of the proposed scheme using both an on-chip microstrip and a MCM

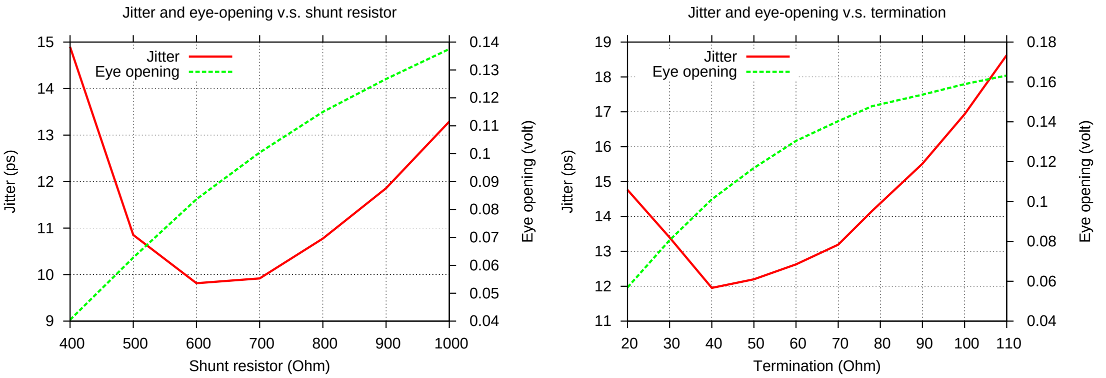


Figure II.29 Jitter and eye-opening tradeoff for the 2um-thick 20cm-long stripline

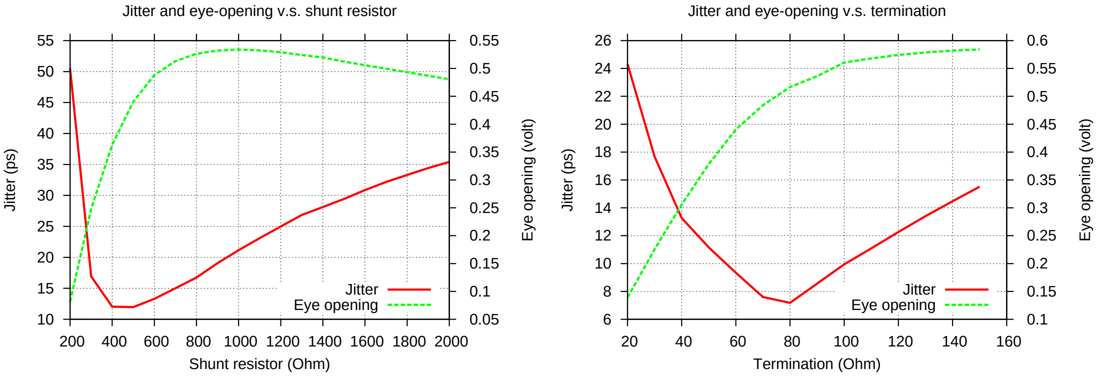


Figure II.30 Jitter and eye-opening tradeoff for the 4.5um-thick 10cm-long stripline

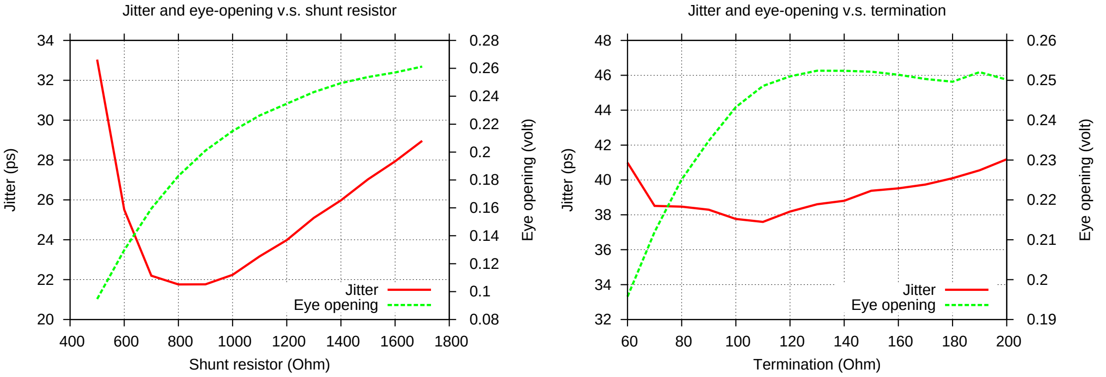


Figure II.31 Jitter and eye-opening tradeoff for the 4.5um-thick 20cm-long stripline

stripline. We also devised a new approach to analytically predict the jitter and eye-opening based on given bitonic response. This technique enables us to quickly study the tradeoff between jitter and eye-opening with respect to the shunt value, and is utilized to compare against the resistive termination method.

The content in this chapter, in part, is a reprint of the paper "Distortion minimization for packaging level interconnects" co-authored with Rui Shi, Hongyu Chen, Chung-Kuan Cheng, Alina Deutsch and George Katopis in the proceedings of 2006 EPEP conference, and the paper "Approaching speed-of-light distortionless communication for on-chip interconnect" co-authored with Rui Shi, Hongyu Chen and Chung-Kuan Cheng in the proceedings of ASPDAC 2007. The dissertation author was the primary investigator and author of these two papers.

# III Interconnect-Centric Cyclic Shifter Optimization

## III.A Introduction

Binary shifters, similar to adders and multipliers, are indispensable in high performance microprocessors, especially in those that support floating-point operations. For communication applications such as encryption and error control coding, the cyclic shifter is a critical component because rotation operations are heavily demanded. Yet the simplicity of the shifter logic misleadingly disguises its importance in circuit design; Literature on shifter design is relatively scarce compared to that of adders and multipliers, and textbooks typically cover shifter in just one or two pages [57], [46]. The main reason is that the complexity of the shifters comes from the internal wire connections which does not fit into the traditional logic-centric design methodology.

It is well-known that in terms of design style there are two types of shifters for circuit designers to choose from: *array shifter* or *logarithmic shifter* [57]. The former is noted for its high speed because in theory every data signal only passes through one transmission gate. However, in practice the capacitance presented to the input signals increases linearly with the word length, and the number of transistors grows quadratically with the word length. Moreover, for array shifter an additional decoder for control signals is required. These factors make the array shifter less appealing for large word length. As an attractive alternative, the logarithmic shifter completes the shifting in $log_2(N)$ stages, where $N = 2^n$ is the word length. At each stage the data signals either uniformly pass through or shift by $2^i$ bits, where $0 \leq i \leq n-1$. However, the logarithmic shifter has its own weakness. The amount of inter-stage interconnect wires roughly double from one stage to the next. The situation is even worse with the aggressive technology scaling due to which the interconnect is dominant for both delay and power consumption.

In terms of functionality, the shifters can be categorized as *cyclic shifter*, *unidirectional right shifter* and *bidirectional shifter*. Mixture of these functions in one shifter module is also commonly seen. An unidirectional right shifter can be arithmetic or non-arithmetic. In an arithmetic right shifter, the higher bits of the shifted word are extended with the MSB (most significant bit) of the original word, while the higher bits in a non-arithmetic shifter are padded with zeros. A bidirectional shifter can be constructed by placing a swapping stage both before and after the unidirectional right shifter.

There were few attempts to improve the shifter by optimizing the interconnect wires. M. A. Hillebrand et al. [25] proposed to half the wire length in a cyclic shifter (rotator) by permuting the cell positions in the intermediate stages. In [48], a two-dimensional folding strategy is suggested for the layout of a cyclic shifter. However, since the shifter is usually aligned with the adder and multiplier in a datapath module and can not decide the module width on its own, the approach is hardly useful in practice. At the logic level, ternary shifting [53] is proposed to replace binary shifting so that the number of stages is reduced. In essence, this method uses 3-to-1 MUXes (multiplexer) to build the shifter network. In [58] Yih et al. proposed a multilevel approach to reduce the number of transmission gates in the barrel shifter.

We propose two methods to alleviate the interconnect burden in the logarithmic cyclic shifter (rotator). Our main contributions can be summarized as follows.

- We propose fanout splitting to decouple the shifting and non-shifting behaviors at each stage so that the non-shifting paths can be saved from switching when the data signals are configured to pass through, and vice versa. In practice, this idea is realized by replacing the MUXes in a traditional design with DEMUXes (demultiplexer) which have two outputs governing the shifting and non-shifting paths separately, giving rise to the term "fanout splitting". We show that by doing so the accumulated wire load on the critical path is reduced from $O(N \log_2(N))$ to $O(N)$. Moreover, the switching probabilities on the inter-stage wires of the new design is lower than that of the conventional MUX-based design, resulting in smaller overall switched capacitance.

- We apply cell order optimization to our DEMUX-based design to further reduce the delay. We formulate the optimization as an Integer Linear Programming (ILP) problem and solve it using commercial ILP solver CPLEX 9.1. For 32- and 64-bit cases a sliding window heuristic is devised to tackle the complexity issue.
- Using the ILP formulation framework we also study the power-delay tradeoff for cell ordering. We show that linear order placement is inherently inefficient in terms of longest path delay.

The analysis results show that for 64-bit case the total delay is reduced by 67.1%, and the dynamic power consumption is reduced by 17.6%, when both fanout splitting and cell order optimization are applied.

The rest of this chapter is organized as follows: In Section III.B we formally state the shifter design problem. Section III.C gives the motivation of our work and describes the fanout splitting technique in detail. The cell order optimization and its ILP formulation are presented in Section III.D. In Section III.E we give both analysis and ASIC implementation results. Section III.F concludes this chapter.

## III.B Notations and Problem Statement

We confine our discussion to the logarithmic cyclic shifter. In the rest of this chapter, "shifter" refers to the logarithmic cyclic shifter unless otherwise stated. We may sometimes use rotator interchangeably.

A shifter takes as inputs an N-bit binary data word $\mathrm{D}[\mathrm{N}-1:0]$ and an n-bit control word $\mathrm{S}[n-1:0]$, and produces an output word $\mathrm{Z}[\mathrm{N}-1:0]$ with $\mathrm{N}=2^n$ for some positive integer $n$. The binary value of the control word is denoted by $|\mathrm{S}| := \sum_{i=0}^{n-1} \mathrm{S}[i] \cdot 2^i$. The shifter implements the function ROTATE:

$$\begin{aligned} \mathrm{Z}[\mathrm{N}-1:0] &= \textsc{rotate}(\mathrm{D}[\mathrm{N}-1:0], \mathrm{S}[n-1]) \\ &:= (\mathrm{D}[|\mathrm{S}|-1:0], \mathrm{D}[\mathrm{N}-1:|\mathrm{S}|]) \end{aligned}$$

where $(\mathrm{A}, \mathrm{B})$ represents the concatenation of two binary strings A and B.

Since the logarithmic shifter is an interconnect intensive component, the main design objective is to come up with a scheme that reduces both power dissipation and delay induced by the wires.

## III.C Fanout Splitting Shifter Design

### III.C.1 Motivation

Figure III.1 displays an example of a conventional 8-bit rotator using MUXes (hereafter referred to as MuxShifter). The MUXes are arranged in an array where the MSB cells are on the left. Assuming there is an input buffer stage, the whole MUX network including the input buffers can be mapped onto a grid graph $G = (V, E)$ where $V := \{(column, row) = (i, j) | (i, j) \in \langle 0, 1, ..., N - 1\rangle \times \langle 0, ..., n\rangle\}$. From each node $(i, j)$ with $j < n$ there are two outgoing edges: $(i, j) \rightarrow (i, j + 1)$ and $(i, j) \rightarrow (i - 2^j \bmod N, j + 1)$. Following the terminology used in [25], we call $(i, j + 1)$ the *unshifted child* and $(i - 2^j \bmod N, j + 1)$ the *shifted child* of $(i, j)$. Likewise, each node $(i, j)$ with $j > 0$ has two incoming edges

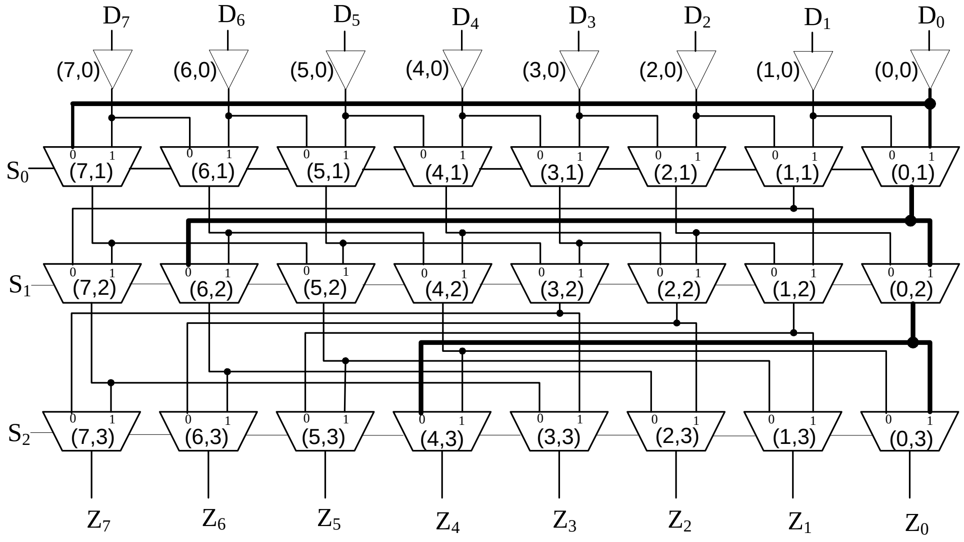


Figure III.1 8-bit conventional MUX-based shifter

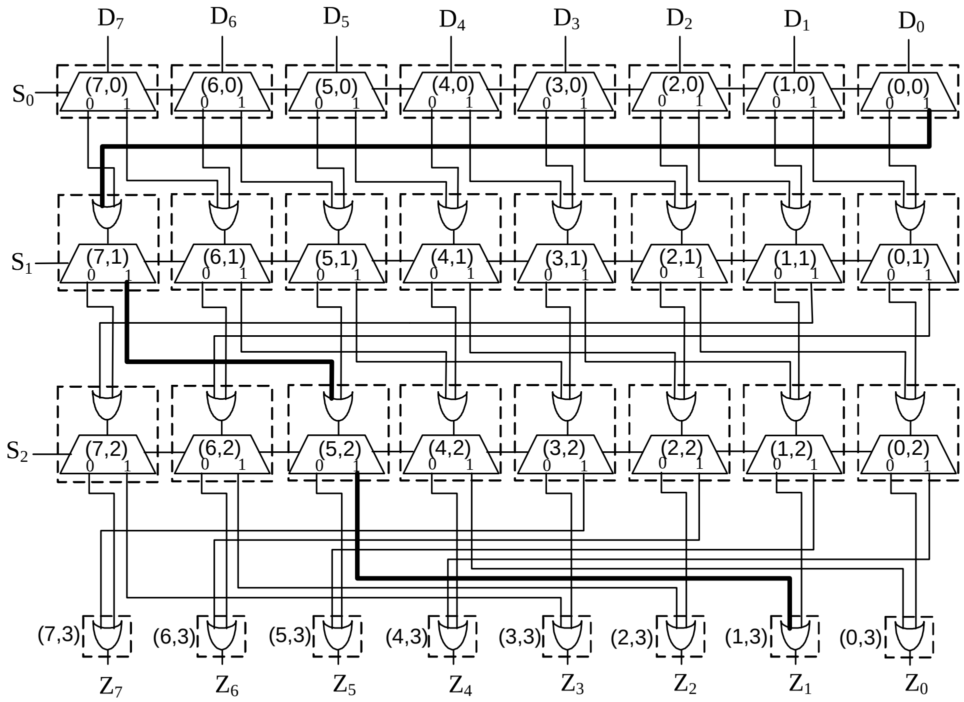


Figure III.2 Proposed DEMUX-based shifter

from $(i, j-1)$ and $(i + 2^j \bmod N, j-1)$, the former called the *unshifted parent* and the latter called the *shifted parent*.

The issue with MuxShifter is that the way the MUXes are interconnected is unaware of the functionality of the shifter. The two edges coming out of node $(i, j)$ are electrically hard-wired together. In the grid graph, this can be modeled by assigning weight $2^j + 1$ to both $(i, j) \rightarrow (i, j + 1)$ and $(i, j) \rightarrow (i - 2^j \bmod N, j + 1)$. As a result, when the shifter is configured to shift 0 bit, all the lateral wires have to be switched as well, unnecessarily consuming more power. This is clearly against the intuitive switch-only-when-needed low power design principle.

Even worse, the longest lateral wires at each stage in MuxShifter are accumulated on the critical path. Consider the path from $D_0$ to $Z_0$ in Figure III.1. The accumulated wire load (highlighted by heavy lines) on this path is 20, assuming 1 unit wire length per column and per row. In general, the accumulated wire load on the critical path for MuxShifter is $\Omega(N \log_2(N))$ [25] where $N$ is the word length.

### III.C.2 Fanout Splitting Approach

Motivated by the above observations, we propose to use DEMUXes to build the shifter in lieu of the MUXes in conventional designs. Each DEMUX has two outputs, the left one is the unshifted signal while the right one is the shifted signal. At the next stage, we precede each DEMUX with an OR gate which collects the shifted and unshifted signals from the previous stage. Such a design is shown in Figure III.2. Since a DEMUX has separate outputs, this technique is termed "fanout splitting". We call the new design DemuxShifter.

Similar to MuxShifter, DemuxShifter can be represented by exactly the same grid graph. The difference is that the two edges coming out of node $(i, j)$ are now independent. Edge $(i, j) \rightarrow (i, j+1)$ has weight 1 while edge $(i, j) \rightarrow (i-2^j \bmod N, j + 1)$ has weight $2^j + 1$.

Because of fanout splitting, the shifting and non-shifting behaviors are now decoupled. This directly reduces the accumulated wire load on the critical path. For the 8-bit case shown in Figure III.2, the wire load of the critical path is now 16. In fact, we have the following conclusion:

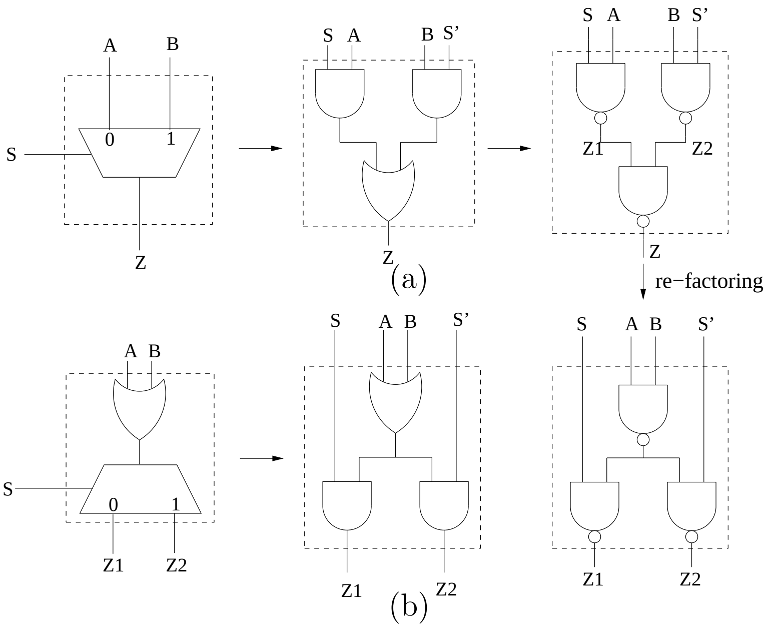


Figure III.3 The gate level implementation of MUX and switch)

*Claim 1*: The wire length of the critical path in DemuxShifter is $O(N)$.

**Proof:** We show a lower bound for the critical path wire length. From level $j$ to $j+1$ there are two types of horizontal wires. Those go to the lower bits have length $2^j$. Those wrap around to the higher bits have length $N-2^j$. The key observation is that any path from a topmost node $(i_1, 0)$ to a bottommost node $(i_2, n)$ shall contain at most one horizontal wire that wraps around. Therefore,

$$
\begin{aligned}
&\text{critical path wire length}\\
&\le \max_{0\le j\le n-1}\Big\{\big(\sum_{\substack{k=0\\k\neq j}}^{n-1} 2^k\big) + N - 2^j + n\Big\}\\
&= \max_{0\le j\le n-1}\Big\{\big(\sum_{k=0}^{n-1} 2^k\big) + N - 2^{j+1} + n\Big\}\\
&\le 2N - 3 + n
\end{aligned}
$$

This proves the claim.

■

To understand the gate complexity, we give the gate implementations of the MUX and the OR+DEMUX conglomerate (essentially a 2-input/2-output

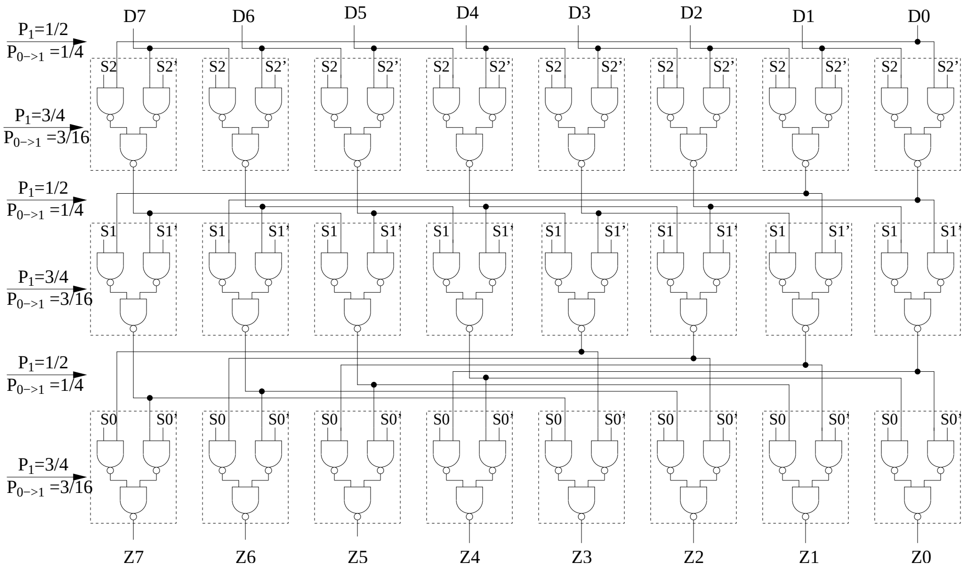


Figure III.4 8-bit MuxShifter using NAND gates

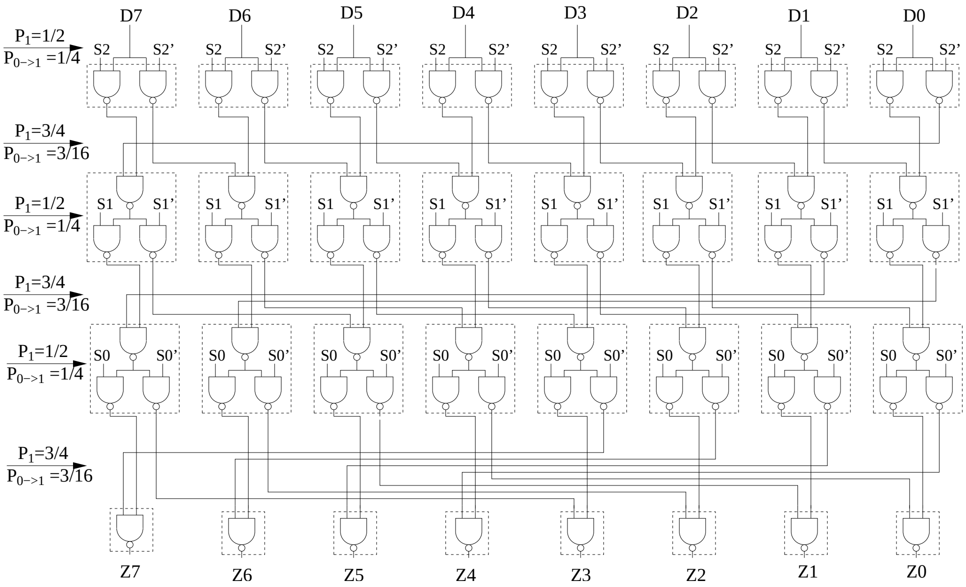


Figure III.5 8-bit DemuxShifter using NAND gates

switch box) in Figure III.3. Interestingly, at gate level the fanout splitting design is really like re-factoring the OR function in the MUXes later to be combined with the logic of the next stage. Note the NAND gate version of the cross switch does not really implement a cross switch; it is derived from the NAND gate version of the MUX by utilizing the re-factoring observation. Figure III.4

and Figure III.5 illustrate the NAND gate implementations of MuxShifter and DemuxShifter, respectively. From the figures we see the fundamental difference between MuxShifter and DemuxShifter is the organization of the inter-stage wires. They both have the same gate complexity $O(N \log_2(N))$.

At a first glance, the total inter-stage wire length in DemuxShifter is the same or even a little bit more than that of MuxShifter. Nevertheless, as we will show in the next, the switching probability on the wires decreases from 1/4 to 3/16, which leads to less overall power consumption.

It is well-known that the dynamic power consumption of a circuit is proportional to the effective switched capacitance $C_{\text{eff}}$,

$$P_{\text{dynamic}} = C_{\text{eff}} V_{\text{supply}}^2 / 2 = (C_{\text{load}} P_{0 \to 1}) V_{\text{supply}}^2 / 2 \tag{III.1}$$

where $P_{0 \to 1}$ is the switching probability on the load capacitance. In general, predicting the signal switching probabilities in a circuit is difficult because of signal correlation due to re-convergent paths. However, the shifter is a strictly levelized design and does not contain re-convergent fanouts, hence its switching probabilities can be easily calculated.

Define the 1-probability $P_X$ to be the probability that a signal $X$ is *one*. The $0 \to 1$ switching probability of signal $X$ can be written as

$$P_{X|0 \to 1} = P_X(1 - P_X) \tag{III.2}$$

We start by assuming that all the primary input signals $[D_7\text{:}D_0]$, $[S_2\text{:}S_0]$ are equal probable at 0 or 1. Consider the NAND gate implementation of a MUX in Figure III.3(a). From the truth table (Table I) we see that

$$P_{Z1} = P_{Z2} = 3/4; P_Z = 1/2$$

We then propagate the 1-probabilities downward across the shifter network, and calculate the switching probabilities using Equation (III.2). The MuxShifter (Figure III.4) and DemuxShifter (Figure III.5) are annotated with both 1-probabilities and switching probabilities. From the figures we clearly see that DemuxShifter has lower switching probabilities on the inter-stage long wires.

Table III.1 Truth table of the MUX using NAND gates

| S | S' | A | B | Z1 | Z2 | Z |
|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 |

## III.D Cell Order Optimization Using Integer Linear Programming

M. A. Hillebrand et al. [25] observed that the longest delay path in a cyclic logarithmic shifter can be reduced by permuting the cells within each row in the intermediate stages, and they applied this technique to a MUX-based design. Apparently, our DEMUX-based shifter can benefit from this technique as well. Different from their constructive approach, we formulate the cell permutation optimization as an Integer Linear Programming (ILP) problem, and use the state-of-the-art CPLEX solver to obtain the results.

### III.D.1 ILP Formulation

Assuming the word length $N = 2^n$, an $N$-bit DemuxShifter consists of $n + 1$ levels. Level 0 is the input stage while level $n$ is the output stage (see Figure III.5). The cell order of the input and output stages are fixed to be linear with MSB on the left and LSB on the right. For each intermediate stage, a set of binary decision variables specifying the permutation order is introduced.

- $x^l_{ij} \in \{0, 1\}$: 1 if and only if logic cell $i$ is placed at physical position $j$ on level $l$ $(0 \le i, j \le N - 1, 1 \le l \le n - 1)$.

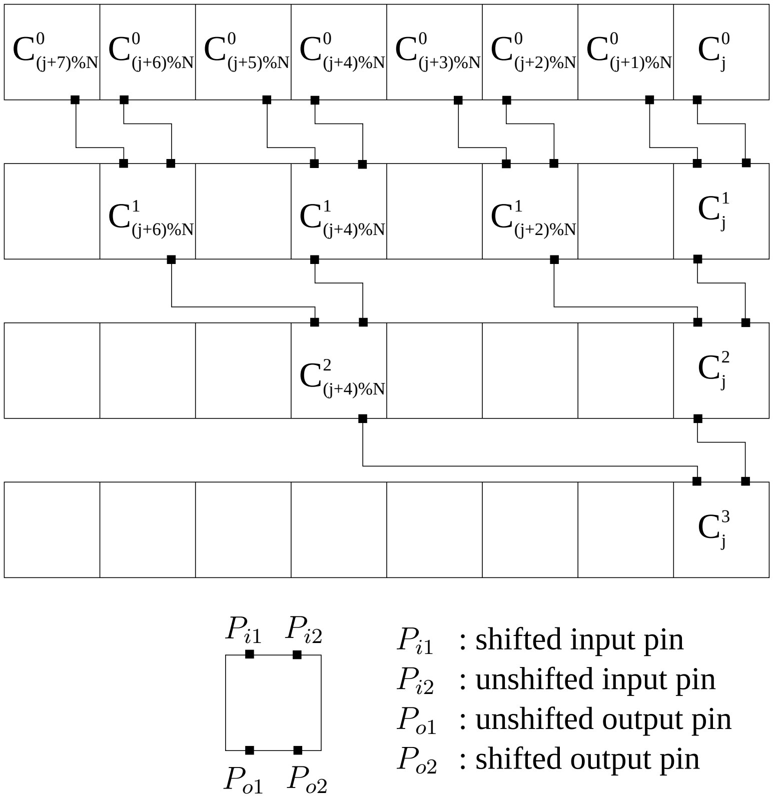


Figure III.6 Delay paths ending at $C_j^3$

Since each logic cell occupies one and only one physical position, the following constraints naturally hold:

$$\sum_{i=0}^{N-1} x_{ij}^l = 1 \quad (0 \le j \le N-1, 1 \le l \le n-1) \tag{III.3}$$

$$\sum_{j=0}^{N-1} x_{ij}^l = 1 \quad (0 \le i \le N-1, 1 \le l \le n-1) \tag{III.4}$$

Note that the set of decision variables $x_{ij}^l$ and constraints (III.3), (III.4) completely define the permutation solution space. The rest of the problem is to represent the objective, i.e., minimizing the longest path.

Denote a cell with logic index $i$ at level $l$ as $C_i^l$. Pick a cell $C_i^0$ at the input stage and a cell $C_j^n$ at the output stage, there is an unique delay path from $C_i^0$ to $C_j^n$. Put another way, from cell $C_j^n$ there are exactly $N$ path tracing back to the input level. The scenario for the 8-bit case is shown in Figure III.6. The logic indices of the cells on each delay path are annotated on the graph, while the physical positions of the cells are for illustration purpose only — they can be anywhere in their respective rows. Thus for an $N$-bit shifter, there are in total $N^2$ delay paths. The length of the longest path, call it $T_{\max}$, is given by

$$\begin{aligned} T_{\max} = \max\Big\{&\text{length of the delay path from} \\ &C_i^0 \text{ to } C_j^n | 0 \le i, j \le N\Big\} \end{aligned} \qquad \text{(III.5)}$$

which can be expanded into $N^2$ linear constraints as

$$\begin{aligned} T_{\max} \ge &\text{ length of the delay path from } C_i^0 \text{ to } C_j^n \\ &\text{for } 0 \le i, j \le N-1 \end{aligned} \qquad \text{(III.6)}$$

with the objective

$$\text{obj}: \text{ minimize } \mathrm{T_{max}} \qquad \text{(III.7)}$$

Each input-to-output delay path consists of $n$ wire segments. We show the technique to represent the length of a single wire segment as a set of linear constraints. Without loss of generality, suppose $C_{i_1}^0$ is connected to $C_{i_2}^1$, yet their respective physical locations are unknown. The length of the wire segment connecting them can be written as

$$d = \Big|\sum_{j=0}^{N-1} j \cdot x_{i_1 j}^0 - \sum_{j=0}^{N-1} j \cdot x_{i_2 j}^1\Big| \qquad \text{(III.8)}$$

which can be converted into linear constraints

$$d - \sum_{j=0}^{N-1} j \cdot x_{i_1 j}^0 + \sum_{j=0}^{N-1} j \cdot x_{i_2 j}^1 \ge 0 \qquad \text{(III.9)}$$

$$d + \sum_{j=0}^{N-1} j \cdot x_{i_1 j}^0 - \sum_{j=0}^{N-1} j \cdot x_{i_2 j}^1 \ge 0 \qquad \text{(III.10)}$$

Note that constraints (III.9) and (III.10) only set the lower bound for $d$, hence are not entirely equivalent to Equation (III.8). However, since our objective is to minimize $T_{\max}$, and $T_{\max}$ is non-decreasing w.r.t. $d$, we conclude that the lower bound for $d$ must be achieved in the optimal solutions.

Once we know how to represent the length of a single wire segment, it is straightforward to express the length of an input-to-output delay path. In modelling the delay paths we have neglected the gate delay and vertical wire length, since they equally contribute to all paths.

In a nutshell, our *minimum delay* ILP formulation comprises constraints (III.3), (III.4), (III.6) and objective (III.7). We also consider the *minimum power* formulation which seeks to minimize the total wire length $T_{\text{total}}$ subject to a given constraint on maximum delay.

$$\text{obj}: \ \text{minimize } \mathrm{T}_{\text{total}} \tag{III.11}$$

$$\text{s.t.} \quad \mathrm{T}_{\max} \leq \text{Constant} \tag{III.12}$$

The minimum power formulation helps to study the tradeoff between delay and power, since total wire length is an indicator of the dynamic power consumption.

### III.D.2 Sliding Window Scheme

The CPLEX solver uses a branch and bound method with linear relaxation [28], thus the running time greatly depends on the problem size. In our ILP formulation, the number of decision variables is $(n-1)N^2$, which grows more than quadratically. As a result, the 8-bit case was solved in seconds while the 16-bit case took almost one day to find an optimal solution. For 32- and 64-bit cases, treating the problem as a whole becomes infeasible. We now describe a sliding window heuristic to tackle the complexity issue of our ILP formulation.

The sliding window scheme optimizes the shifter in multiple passes. In each pass, a sliding window is superimposed on the shifter layout, as shown in Figure III.7. Only cells within the sliding window are allowed to permute freely, with the goal to minimize $T_{\max}$ (min-delay formulation) or $T_{\text{total}}$ (min-power formulation). The window then moves from left to right and top to bottom until it reaches the bottom-right corner. The procedure terminates when there is no improvement between two passes.

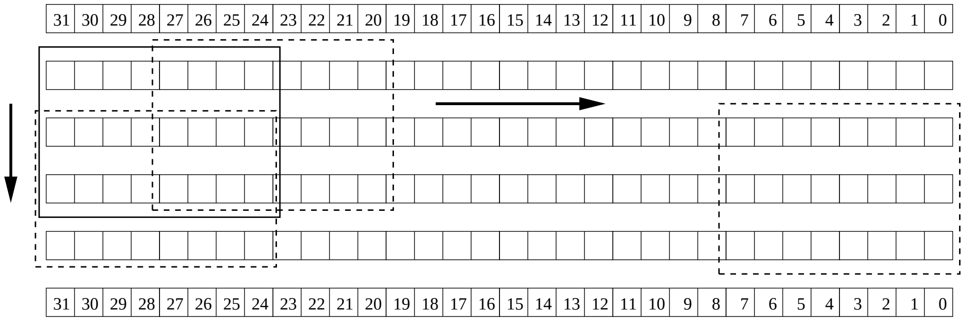
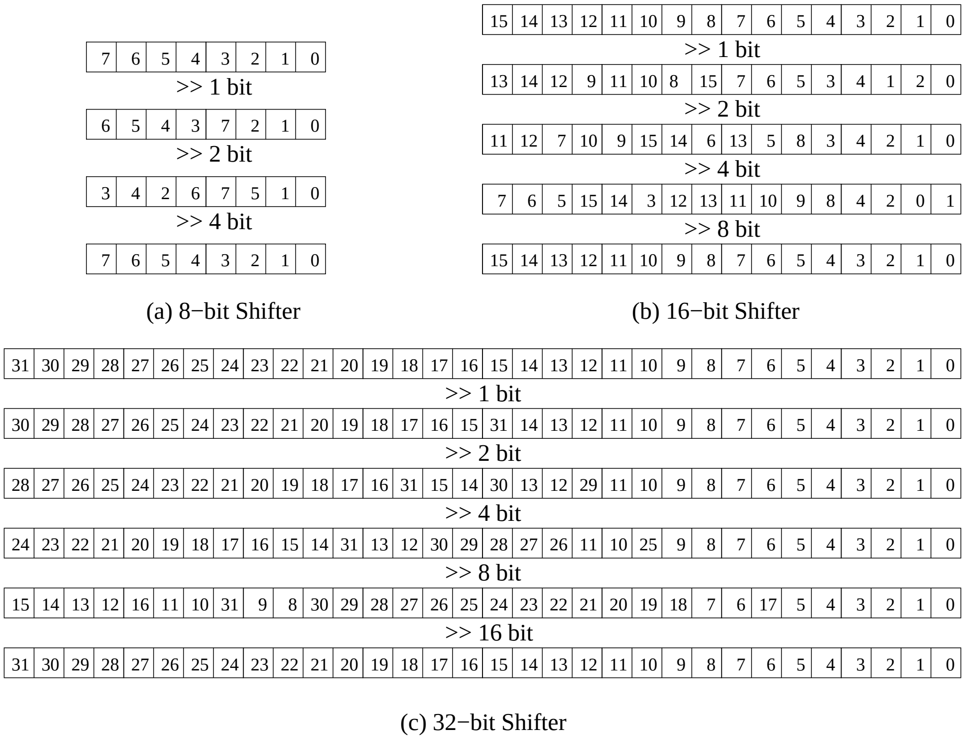


Figure III.7 Sliding window scheme for the 32-bit case



7 6 5 4 3 2 1 0
>> 1 bit
6 5 4 3 7 2 1 0
>> 2 bit
3 4 2 6 7 5 1 0
>> 4 bit
7 6 5 4 3 2 1 0

(a) 8−bit Shifter

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
>> 1 bit
13 14 12 9 11 10 8 15 7 6 5 3 4 1 2 0
>> 2 bit
11 12 7 10 9 15 14 6 13 5 8 3 4 2 1 0
>> 4 bit
7 6 5 15 14 3 12 13 11 10 9 8 4 2 0 1
>> 8 bit
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

(b) 16−bit Shifter

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
>> 1 bit
30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 31 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
>> 2 bit
28 27 26 25 24 23 22 21 20 19 18 17 16 31 15 14 30 13 12 29 11 10 9 8 7 6 5 4 3 2 1 0
>> 4 bit
24 23 22 21 20 19 18 17 16 15 14 31 13 12 30 29 28 27 26 11 10 25 9 8 7 6 5 4 3 2 1 0
>> 8 bit
15 14 13 12 16 11 10 31 9 8 30 29 28 27 26 25 24 23 22 21 20 19 18 7 6 17 5 4 3 2 1 0
>> 16 bit
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

(c) 32−bit Shifter

Figure III.8 Optimal cell orders for minimum delay

Within each pass four parameters control the optimization process: window width WW, window height WH, horizontal sliding step HS and vertical sliding step VS. Tradeoff must be made to balance the runtime and result quality when selecting these parameters. In our experiments we have empirically found that the following set of parameters to be satisfying:

- WW = 8 columns

- WH = 3 rows
- HS = 4 columns
- VS = 1 row

In particular, HS is set to 4 columns so that the cells have large chance to move across the entire row; and WH is set to 3 rows to allow adequate interaction between rows, thus avoiding being trapped in the local minima. The same set of parameters is used for both 32- and 64-bit cases.

Figure III.8 shows delay-optimal cell orders for 8-bit, 16-bit and 32-bit cases. Note the solutions for 8-bit and 16-bit are global optimum while the solution for 32-bit is suboptimal with the sliding window scheme.

## III.E Experimental Results

### III.E.1 Analysis Results

We implemented the sliding window scheme in C using the CPLEX Callable Library [29]. For both min-delay formuation and min-power formulation, the optimized cell order is fed into a delay and power evaluation routine. The evaluation routine takes in a set of parameters based on a technology independent layout model, which we will describe in the next, and produces the estimation results. The whole process is illustrated in Figure III.9.

We use the logical effort method [52] for fast, technology independent delay/power estimation for both MuxShifter and DemuxShifter. For a single stage gate, logical effort measures its delay in unit of $\tau$, the delay of an ideal inverter with no parasitic driving an identical inverter.

$$D = D_{\text{abs}}/\tau = gh + p \tag{III.13}$$

where $g$ is the *logical effort* of the gate, which is defined as the ratio of the input gate capacitance to the input capacitance of an inverter with the same unit effective

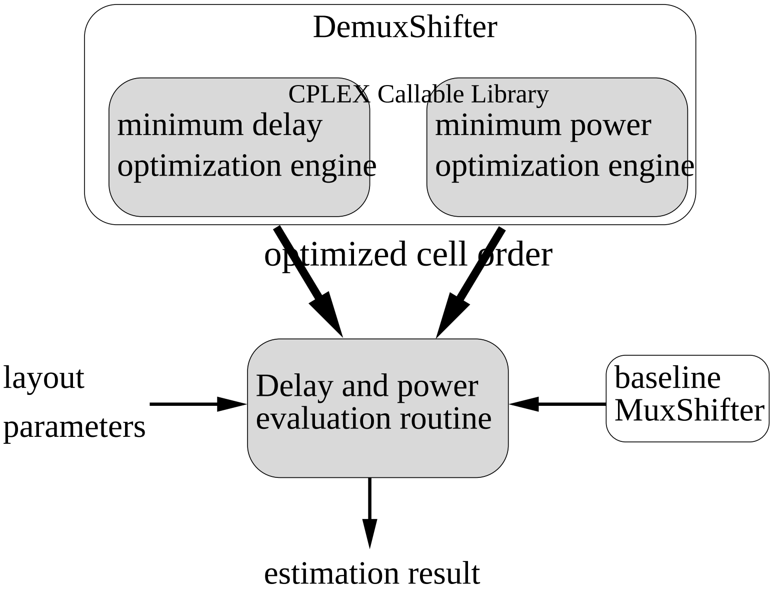


Figure III.9 Estimation flow

resistance; $h$ is the *electrical effort* of the gate, that is, the ratio of load capacitance to input capacitance. $p$ characterizes the parasitic (intrinsic) delay of the gate. For a static CMOS NAND gate, $g = 4/3$ and $p = 2$.

We consider the wire load effect by incorporating wire capacitance into $h$. Thus the electrical effort of a NAND gate is written as

$$h = h_{\text{fanout}} + h_w \cdot l_w \tag{III.14}$$

where $h_{\text{fanout}}$ is the number of load gates and $l_w$ is the length of the driven net normalized to the width of a MUX/DEMUX cell, or simply the number of columns the wire spanned. Naturally, $h_w$ is the electrical effort contributed by the wire per column spanned.

Assuming all the NAND gates in the shifter are 2x of the minimum size uniformly and the MUX/DEMUX cell width is $80\lambda$, we directly borrow the technology parameters from [27], and use the following formula to estimate $h_w$ for each

Table III.2 Device/Wire parameters

| $2\lambda$ | 0.18 | 0.15 | 0.13 | 0.10 | 0.07 |
|---|---|---|---|---|---|
| $C_g$ | 0.234 | 0.220 | 0.135 | 0.072 | 0.066 |
| $C_a$ | 0.0589 | 0.0540 | 0.0462 | 0.0720 | 0.0611 |
| $C_f$ | 0.0302 | 0.0248 | 0.0183 | 0.0141 | 0.0148 |
| $C_x$ | 0.0583 | 0.0494 | 0.0428 | 0.0453 | 0.0416 |
| $h_w$ | 0.63 | 0.92 | 1.06 | 1.54 | 1.03 |

$2\lambda$($\mu$ m): technology feature size;
$C_g$(fF): input capacitance of minimum size inverter;
$C_a$(fF/$\mu m^2$): unit area capacitance of wire;
$C_f$(fF/$\mu$m): unit fringing capacitance of wire;
$C_x$(fF/$\mu$m): unit coupling capacitance of wire.
Note: All wire capacitance are obtained assuming 2x minimum width and 2x minimum spacing.

technology node:

$$\begin{aligned} h_w &= \frac{\text{Capacitance of wire per column spanned}}{\text{Input capacitance of a 2x NAND gate}} \\ &= \frac{(C_a \cdot 4\lambda + C_f + C_x) \cdot 80\lambda}{C_g \cdot 2 \cdot 4/3} \end{aligned} \tag{III.15}$$

From Table II we see that it is safe to assume $h_w = 1$ across different technology nodes, as it is used in [20].

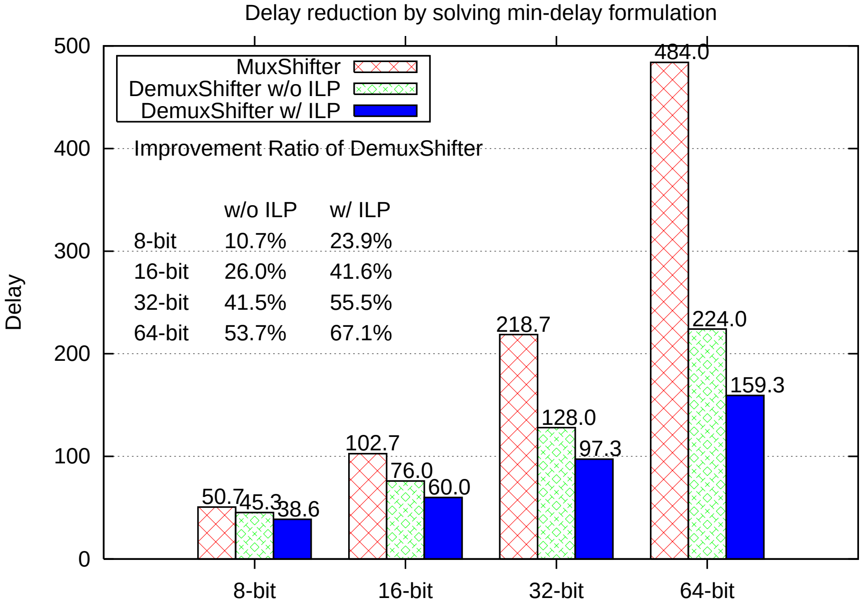
Note that in the analysis we have assumed that the resistive RC delay of the wires is negligible. This assumption is supported by the work of Huang and Ercegovac [27].

The overall path delay is then simply the sum of the stage delays:

$$D_{\text{path}} = \sum_i D_i = \sum_i g_i h_i + \sum_i p_i \tag{III.16}$$

Since the shifter does not contain re-convergent paths we can simply search the $N^2$ input-to-output paths and pick the longest one.

The dynamic power is estimated by summing up all the switched capacitance $C_{\text{eff}}$ in the shifter, including gate input capacitance and wire capacitance. The input gate capacitance of a 2x NAND gate is $8C_g/3$, which is also the capacitance of the wire of one column span.

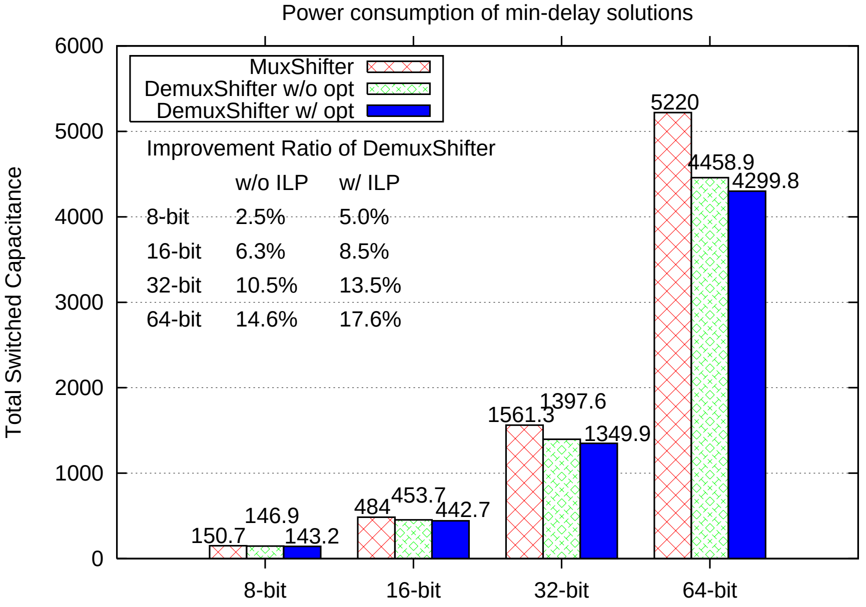


Improvement Ratio of DemuxShifter

|  | w/o ILP | w/ ILP |
|---|---|---|
| 8-bit | 10.7% | 23.9% |
| 16-bit | 26.0% | 41.6% |
| 32-bit | 41.5% | 55.5% |
| 64-bit | 53.7% | 67.1% |

The power number is in unit of $C_g$ (intrinsic delay of a minimum sized inverter).

Figure III.10 Delay comparison of MuxShifter and DemuxShifter

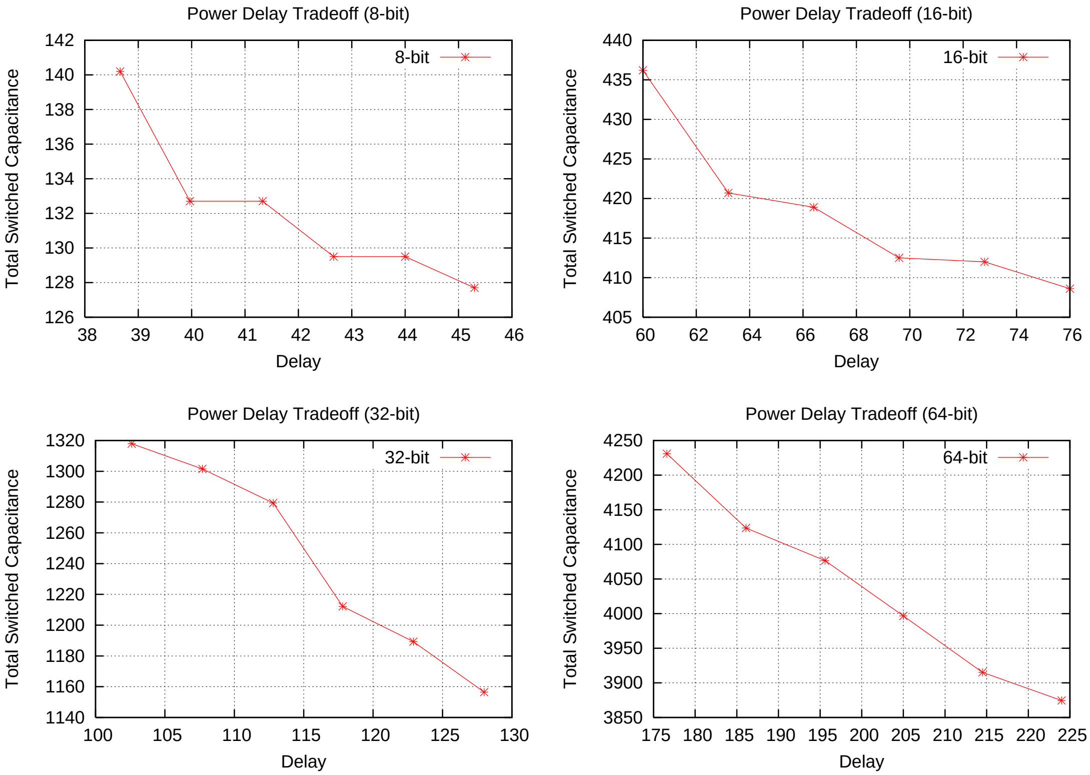


Improvement Ratio of DemuxShifter

|  | w/o ILP | w/ ILP |
|---|---|---|
| 8-bit | 2.5% | 5.0% |
| 16-bit | 6.3% | 8.5% |
| 32-bit | 10.5% | 13.5% |
| 64-bit | 14.6% | 17.6% |

The power number is in unit of $C_g$ (input capacitance of a minimum sized inverter).

Figure III.11 Power comparison of MuxShifter and DemuxShifter



The power is in unit of $C_g$ and delay is in unit of $\tau$.

Figure III.12 Power-delay tradeoff by solving the min-power formulation

Figure III.10 shows the delay estimation results. Comparing to conventional MuxShifter, the DemuxShifter without cell order optimization reduces the delay by 41.5% for 32-bit case and 53.7% for 64-bit case. When cell order optimization is applied, the reduction further increases to 55.5% and 67.1%, respectively.

Figure III.11 shows the power estimation results. For 32-bit and 64-bit cases, the dynamic power improvement is 13.5% and 17.6%, respectively. This is less than the reduction of switching probabilities on the wires (from 1/4 to 3/16) because the gate power is also included.

To study the power-delay tradeoff in the shifters, we also solved the min-power ILP problem (Equation (III.11) and Equation (III.12)) for our proposed DemuxShifter, and the results are shown in Figure III.12. It seems that the power (total switched capacitance) is much harder to optimize. For example, in the 64-bit case relaxing the delay constraint from 176.6$\tau$ to 224$\tau$ only results in 8.4% reduction in power. Nevertheless, the results make sense, because intuitively the shifting path and non-shifting path could be conflict objectives when moving a cell. This indicates that the linear cell order is inherently inferior in terms of worst case delay.

Note that in all the experiments, we have assumed uniform sizing for the DemuxShifter. In fact, one can size up the NAND gate that drives the longer wire in each DEMUX cell to further improve the delay. This enhancement is applicable to both the permuted and non-permuted DemuxShifter, and it is interesting to study which scheme benefits more from sizing.

### III.E.2 ASIC Implementation Results

We have also implemented the proposed DemuxShifter and the conventional MuxShifter using TSMC 90nm standard cell library, for a better estimation of improvement. To avoid the time-consuming full-custom layout process while keep the advantage of bit-slice structure, we utilize the relative placement feature provided by Synopsys Physical Compiler. Figure III.13 illustrates the outline of the relative placement design flow, the core step of which is to use `rp_reader` to specify the relative physical order of cells in the design. By turning on `set_sizing_only` option, Physical compiler will preserve the relative placement and does only sizing and buffering to optimize the design.

To make a fair comparison, we used multiplexer cell `MXL` from the standard cell library to build the MuxShifter. The `MXL` cell consists of twelve transistors which is equivalent to three `NAND2` gates. On the other hand, the DemuxShifter is built entirely with `NAND2` gates as shown in Figure III.2. We also manually added buffers for the control signals to minimize tampering of layout by Physical compiler.

Table III.3 shows the critical path delay and power consumption for both

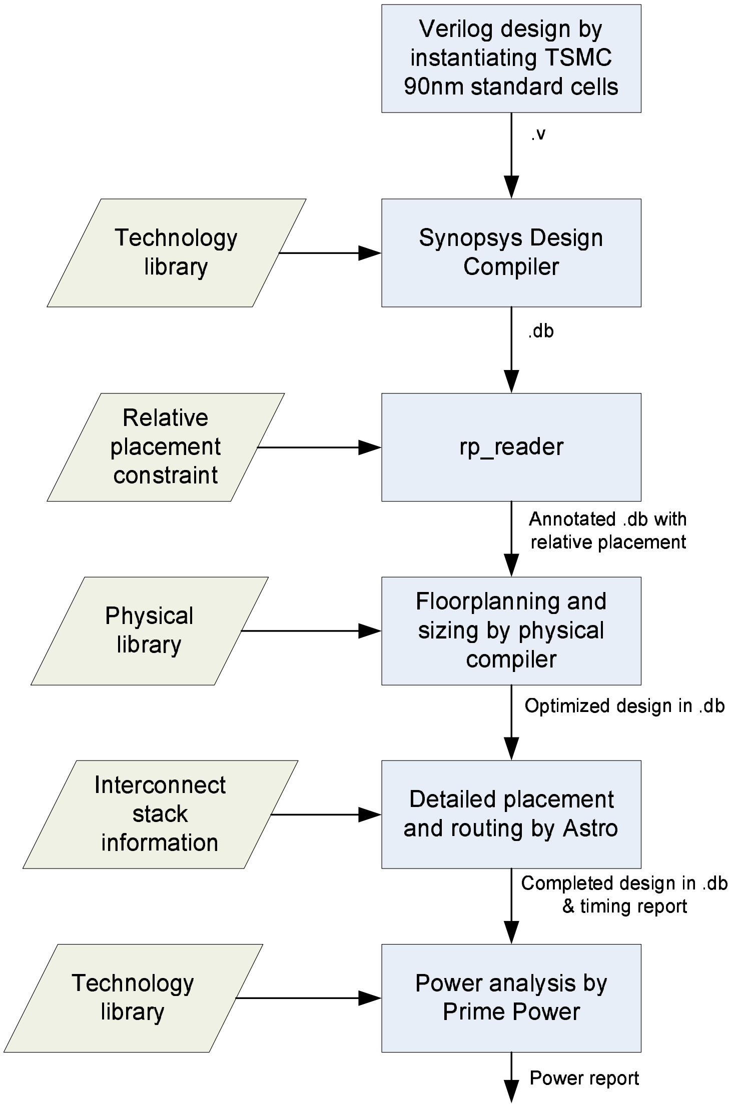

Verilog design by instantiating TSMC 90nm standard cells
.v
Technology library
Synopsys Design Compiler
.db
Relative placement constraint
rp_reader
Annotated .db with relative placement
Physical library
Floorplanning and sizing by physical compiler
Optimized design in .db
Interconnect stack information
Detailed placement and routing by Astro
Completed design in .db & timing report
Technology library
Power analysis by Prime Power
Power report

Figure III.13 Relative placement design flow

designs. The delays were carefully broke down into three parts, namely that contributed by the gates, wire capacitance and crosstalk, respectively. From the table we see clearly that the delay improvement comes primarily from the reduction of wire load. It is also shown that the net power is roughly two times of the gate power in conventional `MuxShifter`, and a total of 8.2% reduction in power can be achieved by reducing the switching activity on the nets.

Table III.3 ASIC design results of 64-bit MuxShifter and DemuxShifter

| | | MuxShifter | DemuxShifter | Imp. |
|---|---|---|---|---|
| Global longest path delay (ns) | gate only | 0.7243 | 0.7243 | 0% |
| | gate+wire load | 1.5534 | 1.1135 | 28% |
| | gate+wire load +xtalk | 1.9599 | 1.4503 | 26% |
| Longest data in/data out path delay (ns) | gate only | 0.4839 | 0.4717 | 2.5% |
| | gate+wire load | 1.2520 | 0.8473 | 32.3% |
| | gate+wire load+xtalk | 1.5986 | 1.0882 | 31.9% |
| Power (Watt) @ 500MHz | cell power | 5.165e-04 | 5.129e-04 | 0.7% |
| | net power | 1.112e-03 | 9.833e-04 | 11.6% |
| | total | 1.629e-03 | 1.496e-03 | 8.2% |

## III.F Summary

We have presented two orthogonal methods to improve the critical path delay and power consumption in logarithmic cyclic shifters. The first method, called fanout splitting, relies on using DEMUXes to reduce accumulated wire load on the critical path, as well as the switching probabilities of the inter-stage wires. We then optimize the cell order of the intermediate stages by integer linear programming (ILP) to further reduce the delay. Using the ILP formulation we also study the delay-power tradeoff in the shifter, and our results show that linear cell placement is inherently inferior in terms of critical path delay.

For simplicity of the analysis, the delay and power models are based on the logical effort method which ignores the dynamic characteristics of the circuits such as glitching power. To remedy this deficiency, we have implemented the designs in TSMC 90nm standard cells using a relative placement design flow, and the results correlate to the analysis quite well. To get the most accurate results an extensive full-custom layout/extraction/hspice simulation design flow is desired, as is conducted in [4]. Nevertheless, our estimation is meaningful in that it provides a fast high level comparison.

The content of this chapter, in part, is a reprint of the material as it appears in the paper "An interconnect-centric approach to cyclic shifter design using fanout splitting and cell order optimization" co-authored with Yi Zhu, Chung-

Kuan Cheng and David Harris in the proceeding of ASPDAC 2007. The dissertation author was the primary investigator and author of this paper.

# IV Zero-deficiency Prefix Adder

## IV.A Introduction

In this chapter, we solve the open problem of constructing zero-deficiency prefix circuit of the minimum depth. Although the parallel prefix problem has a higher level of abstraction, we will confine our discussion of the subject in the context of binary addition, that is, parallel prefix adders.

Binary adder probably is the most fundamental and most well-studied arithmetic component in VLSI Design. Research papers on binary adders are countless, yet the solution space of the adders is so large that it is always tantalizing to seek a better solution. Nevertheless, many adder structures fall into the category of parallel prefix adder, which formulates the carry propagation in an adder as a prefix problem. The parallel prefix adder formulation provides a neat framework to study many adder properties such as delay, area, fanout, power as well as the tradeoffs between them, thus has received much research attention. Knowles' classification [33] and Harris' taxonomy [21] of prefix adders are excellent references for getting a flavor of prefix adder design space.

Among various tradeoffs in prefix adders, timing/area tradeoff is of the utmost interest. It is shown that in a prefix computation graph, a small constant reduction in time of computation from $2\log n$ to $\log n$ increases the area required to embed an $n$ node prefix computation graph significant from $O(n\log n)$ to $O(n^2)$ [51]. However, the analysis is only asymptotic and it is not clear the exact curvature of the tradeoff curve. Put it another way, we are interested in finding the exact point on the tradeoff curve from which the area starts to grow drastically, and this is the subject of this chapter.

The rest of the chapter is organized as follows. We will first review how binary addition can be formulated as a parallel prefix problem in Section IV.B. The concept of zero-deficiency, which is central to the discussion of this chapter, is then raised in Section IV.C.1, followed by a brief discussion of previous work on prefix adders in Section IV.C.2. The main contribution of this chapter lies in Section IV.D, where a new class of zero-deficiency prefix adder which achieves the minimum depth is proposed. We continue to discuss the extensions of the proposed prefix adder in Section IV.E. Section IV.F concludes this chapter.

## IV.B Binary Addition as a Parallel Prefix Problem

The prefix problem, which mostly gains research attention with the emergence of parallel computing, is actually the abstraction of many practical applications such as binary addition, radix sort, linear recurrences solving, polynomial evaluation, etc. [36]. Formally, the prefix problem is defined as follows:

**Definition IV.B.1 (Prefix Problem)** *Given $n$ inputs $x_1, x_2, \ldots, x_n$ and an arbitrary binary associative operator $\bullet$, compute the prefix results $Y_i = x_i \bullet x_{i-1} \bullet \cdots \bullet x_1$ for $1 \leq i \leq n$.*

Since the binary operator $\bullet$ is associative, it is convenient to introduce the *partial prefix result* $y_{[i:j]}$ using the following recurrence formula:

$$y_{[i:i]}{=}x_i, 1 \leq i \leq n \tag{IV.1}$$

$$y_{[i:j]}{=}y_{[i:k]} \bullet y_{[k-1:j]}, 1 \leq j < k \leq i \leq n \tag{IV.2}$$

Using the above representation, $Y_i$ is nothing but a special kind of partial prefix result:

$$Y_i = y_{[i:1]} \tag{IV.3}$$

Nevertheless, $Y_i$'s are usually referred to as *final prefix results* so as to be distinguishable from other intermediate partial prefix results.

Note that the associative operator $\bullet$ in general may not be commutative. Thus in computing $Y_i$ it is imperative to keep the order of the primary inputs $x_i$ as it is defined.

Binary addition, although does not look much like the prefix problem at the first glance, has the same prefix property in its carry chain design. We shall first start with the very basic ripple-carry adder. Let us assume the adder takes in two $n$-bit binary numbers $A = a_n a_{n-1} \cdots a_1$, $B = b_n b_{n-1} \cdots b_1$ and a 1-bit carry-in $c_0$. The ripple carry adder produces the $n$-bit sum $S = s_n s_{n-1} \cdots s_1$ and carry-out $c_n$ using the following recurrence formulae:

$$\begin{aligned} s_i &= a_i \oplus b_i \oplus c_{i-1} && \text{(IV.4)} \\ c_i &= a_i b_i + a_i c_{i-1} + b_i c_{i-1} \\ &= (a_i \oplus b_i) c_{i-1} + a_i b_i && \text{(IV.5)} \\ & \text{for} \quad 1 \leq i \leq n \end{aligned}$$

Alternatively, we can define the carry generation signal $g_i$ and carry propagation signal $p_i$ at each bit position,

$$\begin{aligned} g_i &= a_i b_i && \text{(IV.6)} \\ p_i &= a_i \oplus b_i && \text{(IV.7)} \end{aligned}$$

so that Equation IV.4 and IV.5 can be rewritten as

$$\begin{aligned} s_i &= c_{i-1} \oplus g_i && \text{(IV.8)} \\ c_i &= g_i + p_i c_{i-1} && \text{(IV.9)} \end{aligned}$$

Equation IV.9 shows that the carry-out $c_i$ is either generated at the current bit (when $g_i = 1$), or it is passed from $c_{i-1}$ (when $p_i = 1$), but never both.

The concept of carry propagation and carry generation can be easily extended to multiple consecutive bits,

$$G_{[i:j]} = \begin{cases} g_i & \text{if } i = j \\ G_{[i:k]} + P_{[i:k]}G_{[k-1:j]} & \text{if } n \geq i > j \geq 1 \end{cases} \tag{IV.10}$$

$$P_{[i:j]} = \begin{cases} p_i & \text{if } i = j \\ P_{[i:k]}P_{[k-1:j]} & \text{if } n \geq i > j \geq 1 \end{cases} \tag{IV.11}$$

By introducing an associative operator $\bullet$, the computation of a pair of $(G, P)$ signals and carry signals can be expressed as:

$$(G, P)_{[i:i]} = (g, p)_i \tag{IV.12}$$

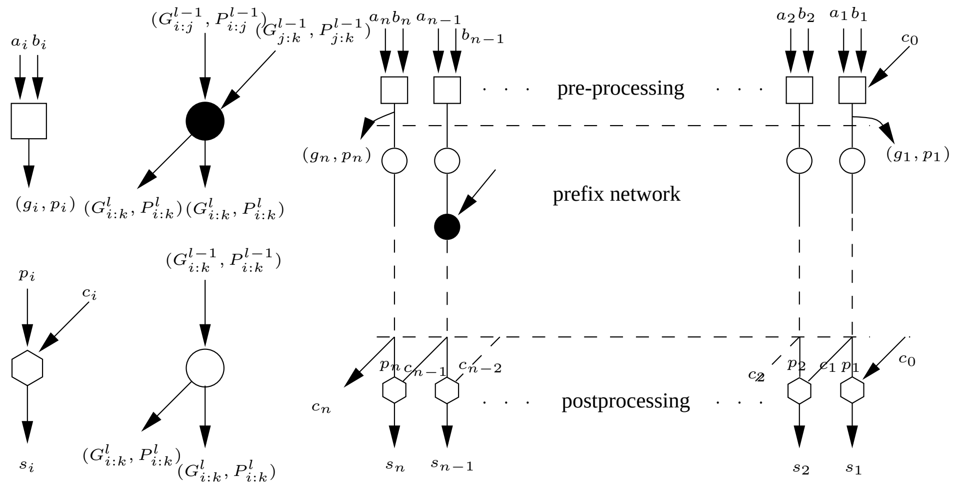
$$\begin{aligned} (G, P)_{[i:j]} &= (G, P)_{[i:k]} \bullet (G, P)_{[k-1:j]} \\ &= (G_{[i:k]} + P_{[i:k]}G_{[k-1:j]}, P_{[i:k]}P_{[k-1:j]}) \end{aligned} \tag{IV.13}$$

$$c_i = G_{[i:1]} \tag{IV.14}$$

Noting that $c_i = G_{[i:1]}$, we see that the carry computation is exactly a prefix problem as defined by Equation IV.1, IV.2 and IV.3.

Using the prefix formulation, a binary adder can be decomposed into three stages: pre-computation, prefix network and post-processing. The pre-computation stage generates all the single bit carry propagation signals and carry generation signals, which are fed into the prefix network to calculate the carry signals $c_i$. The post-processing stage finally calculates all the sum bits. This is illustrated in Figure IV.1.

Since the pre-computation and post-processing stages consist only local bitwise operations, they are pretty much standard across various designs. By contrast, the prefix network presents the major difficulty in adder design. In the rest of this chapter, we will only illustrate the prefix network in the discussions.



(a) Logic operators

(b) Prefix adder structure

Figure IV.1 Binary addition as prefix computation

## IV.C Review of Zero-deficiency Adders

### IV.C.1 The Zero-deficiency Concept

Typically, the prefix network is visualized by a directed acyclic graph, referred to as the *parallel prefix circuit*[1], which takes in the $n$ inputs $x_i$ $(1 \leq i \leq n)$ and produces the $n$ final prefix results $Y_i$ $(1 \leq i \leq n)$ in parallel. The computation nodes in the prefix circuit are arranged in levels representing the timing of the results. Each computation node has two inputs, which can be either primary inputs or partial prefix results from previous levels, and produces either another partial or a final prefix result. All the computation nodes that generate $y_{[i:j]}$ with the same $i$ are in the same column. As an example, Figure IV.2 shows a 16-input Sklansky prefix circuit[2] [49] which produces all the prefix results in 4 levels, level 0 being the primary inputs. The primary inputs are represented by solid squares while the computation nodes are represented by solid black nodes. The white nodes are simply placeholders which do nothing but pass through the data. In the rest of this paper, we always mean computation nodes when we say "nodes",

[1] We will interchangeably use the term *parallel prefix circuit* and *parallel prefix adder.*

[2] The Sklansky adder is also known as Ladner-Fischer adder in some literature since it is a special form of a large class of prefix adders proposed by Ladner and Fischer [35].

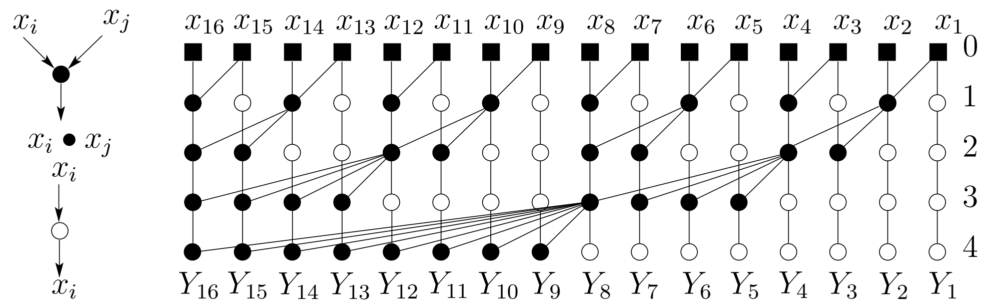

$x_i$
$x_j$
$x_i \bullet x_j$
$x_i$
$x_i$
$x_{16}$ $x_{15}$ $x_{14}$ $x_{13}$ $x_{12}$ $x_{11}$ $x_{10}$ $x_9$ $x_8$ $x_7$ $x_6$ $x_5$ $x_4$ $x_3$ $x_2$ $x_1$
0
1
2
3
4
$Y_{16}$ $Y_{15}$ $Y_{14}$ $Y_{13}$ $Y_{12}$ $Y_{11}$ $Y_{10}$ $Y_9$ $Y_8$ $Y_7$ $Y_6$ $Y_5$ $Y_4$ $Y_3$ $Y_2$ $Y_1$

Figure IV.2 An example of parallel prefix circuit: Sklansky's structure

unless otherwise stated.

It is then natural to define the size and depth of the prefix circuits as follows:

**Definition IV.C.1** *Denote a prefix circuit of $n$ inputs as $G(n)$. In the absence of confusion, we say the width of $G(n)$ is $n$. The size of $G(n)$, denoted by $s_{G(n)}$, is the number of computation nodes in $G(n)$. Similarly, the depth of $G(n)$, denoted by $d_{G(n)}$, is the number of levels in $G(n)$.*

The main problem in designing prefix circuits has been identifying the exact tradeoff between the size and the depth. Snir proved that $s_{G(n)} + d_{G(n)} \geq 2n - 2$ holds for arbitrary prefix circuits [50], based on a concept he introduced called zero-deficiency:

**Definition IV.C.2** *The deficiency of a prefix circuit $G(n)$ is defined as*

$$def(G(n)) = s_{G(n)} + d_{G(n)} - (2n - 2) \tag{IV.15}$$

*$G(n)$ is said to be of* **zero-deficiency** *if $def(G(n)) = 0$.*

Snir's theorem indicates that the solution space for parallel prefix circuits should look like Figure IV.3. For a given width $n$, the maximum depth of the prefix circuits is $n - 1$ (serial prefix circuit) while the minimum depth is $\lceil \log n \rceil$ (Sklansky's circuit [49]). For loose depth constraints we can observe a linear tradeoff between the depth and size which is exhibited by the zero-deficiency prefix circuits. However, if the depth constraint is too tight, the size of the prefix circuits

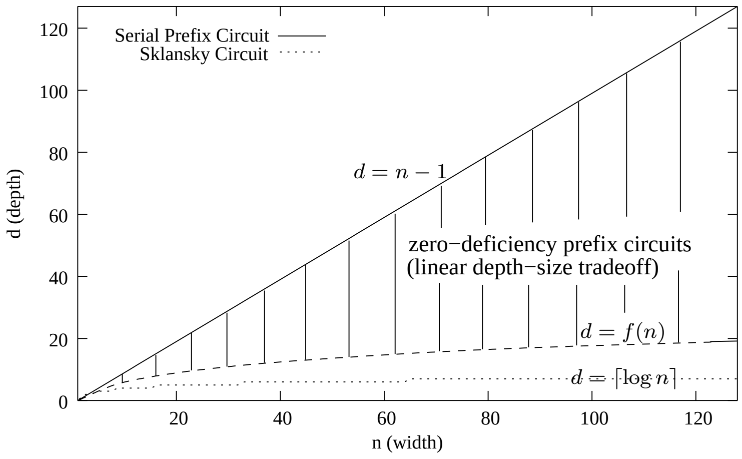


Figure IV.3 Depth-Size tradeoffs of parallel prefix circuits

will grow dramatically and zero-deficiency prefix circuits no longer exist. A direct example of this is the Sklansky's structure in Figure IV.2, which has depth $k$ and size $k2^{k-1}$ for width $2^k$. Nevertheless, even when zero-deficiency is not achievable, there is certainly a lower bound for the size under the given depth constraint. Hence, we introduce the concept of *depth-size optimal*:

**Definition IV.C.3** *A parallel prefix circuit is defined as depth-size optimal if it achieves the minimum size for the given depth constraint.*

We shall stress the difference between zero-deficiency and depth-size optimal. A zero-deficiency prefix circuit must be depth-size optimal, but the converse may not hold. In Figure IV.3, the region between curves $d = f(n)$ and $d = \lceil \log n \rceil$ is where zero-deficiency prefix circuits do not exist, but depth-size optimal prefix circuits do. It remains an open question to find the zero-deficiency prefix circuits of minimum depth.

### IV.C.2 Previous Work

The bulk of parallel prefix adders can be largely classified into non-zero-deficiency adders and zero-deficiency adders, roughly delimited in time by Snir's 1986 paper [50]. Many well-known adder structures discovered in the early days

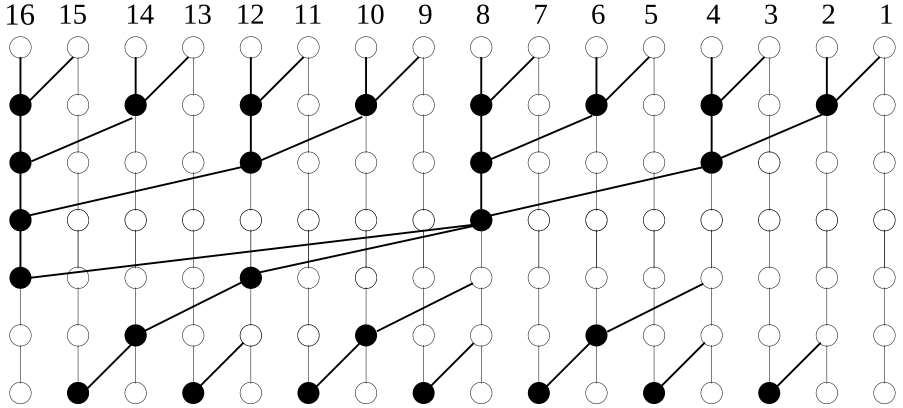


Figure IV.4 16-bit Brent-Kung prefix adder

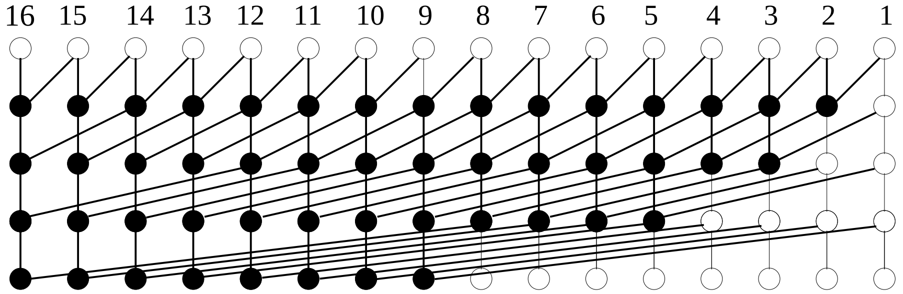


Figure IV.5 16-bit Kogge-Stone prefix adder

of adder research actually have non-zero deficiency; these include the Sklansky adder [49] (which we have seen in Fig. IV.2), Brent-Kung adder [9] and Kogge-Stone adder [34].

Table IV.1 summarizes the depth/size and deficiency of the aforementioned three classical prefix adders. Note that the Kogge-Stone adder, which is considered by many the fastest adder in practice, has the largest deficiency.

The category of non-zero-deficiency adder also includes many other variants. For example, the Han-Carlson adder [19] is a hybrid of the Brent-Kung adder and Kogge-Stone adder. In [35], Ladner and Fischer proposed a divide-and-conquer approach to construct prefix adders using recursive partition. More importantly, they for the first time established an upper bound for the minimum size of a prefix adder under given depth constraint. Their upper bound was later on improved by Fich [15].

Table IV.1 Deficiencies of three classical prefix adders

| | Sklansky | Brent-Kung | Kogge-Stone |
|---|---|---|---|
| Depth | $\log_2 n$ | $2\log_2 n - 2$ | $\log_2 n$ |
| Size | $n \log_2 n/2$ | $2n - 2 - \log_2 n$ | $n \log_2 n - n + 1$ |
| Deficiency | $(n+2)\log n/2 - 2n + 2$ | $\log_2 n - 2$ | $(n+1)\log_2 n - 3n + 3$ |

In addition to constructive prefix adder designs, some researchers focus on heuristic improvement, usually by transforming a baseline adder structure. In [16], Fishburn introduced a local operation called peephole transformation to optimize the critical path in an adder structure. Bilgory defined the cost of a prefix adder[3] as the minimum number of rows required after compaction; he then proposed heuristics to construct min-cost prefix adders for a given word length $n$, depth $d$, initial value availability $e$, and fanout $f$ [8]. In [61], Zimmermann proposed a non-heuristic algorithm for prefix circuit optimization using depth-controlled compression and expansion. Although his approach in many cases produces depth-size optimal or near optimal prefix circuits, optimality of his results is not guaranteed.

Sparked by Snir's 1986 paper, zero-deficiency prefix adders gains much popularity. In [50], Snir himself gave a class of zero-deficiency prefix circuits by concatenating a Brent-Kung circuit with a serial prefix circuit. His design requires that $d_{G(n)} \geq 2\lceil \log n \rceil - 2$. Following the same idea, Lin and Shih were able to improve the minimum depth to as small as $2\lceil \log n \rceil - 5$ by fine-tuning the widths of the Brent-Kung circuit and the serial circuit [42]. Another kind of zero-deficiency circuit of small depth is the $LYD(n)$ circuit which comprises a Brent-Kung circuit, a delay-balanced two-level serial circuit, and two one-level serial circuits [37].

[3] Bilgory however calls the prefix problem as suffix problem.

## IV.D Zero-deficiency Prefix Adder of Minimum Depth

### IV.D.1 Properties of Zero-deficiency Prefix Adders

Snir's original proof for $s_{G(n)}+d_{G(n)} \geq 2n-2$ is by mathematical induction and does not reveal the structural information of zero-deficiency circuits. We notice that there are some nice ideas in [15] and [36] which can be used to devise an elegant proof for Snir's theorem. In this section, we polish these ideas and prove an enhanced version of Snir's lower bound theorem.

**Theorem IV.D.1** *For an $n$-input prefix circuit $G(n)$, denote the depth of its last prefix output $Y_n$ as $d^M_{G(n)}$. Then*

$$s_{G(n)} + d_{G(n)} \geq s_{G(n)} + d^M_{G(n)} \geq 2n - 2 \tag{IV.16}$$

**Proof:** Let us consider the node that generates the last final prefix result $Y_n$, and use Figure IV.6 as an illustration. This output node has to cover all the primary inputs, hence the structure that generates $Y_n$ must be an upside-down tree with all the primary inputs being its leaves. Let us name it the **primal fan-in tree**. Note that some nodes in the primal fan-in tree may have fan-outs to prefix outputs other than $Y_n$, but those fan-outs and their succeeding nodes do not constitute the primal fan-in tree. Put another way, one can identify the primal fan-in tree by starting from the node $Y_n$ and tracing its inputs all the way back to the primary inputs. Remember that the binary operator is not commutative in general, and the order of the primary inputs is fixed. Thus the primal fan-in tree is essentially a binary alphabetical tree. To illustrate, the edges of the primal fan-in tree in Figure IV.6 are highlighted using heavy lines. The size of the primal fan-in tree is exactly $n-1$, and its depth is $d^M_{G(n)}$.

On the other hand, the first primary input $x_1$ has to be fed into every final prefix output, either directly or indirectly. Consequently, there is another tree rooted at $x_1$ with the final prefix outputs $Y_i$ $(1 \leq i \leq n-1)$ as its leaves. Let us

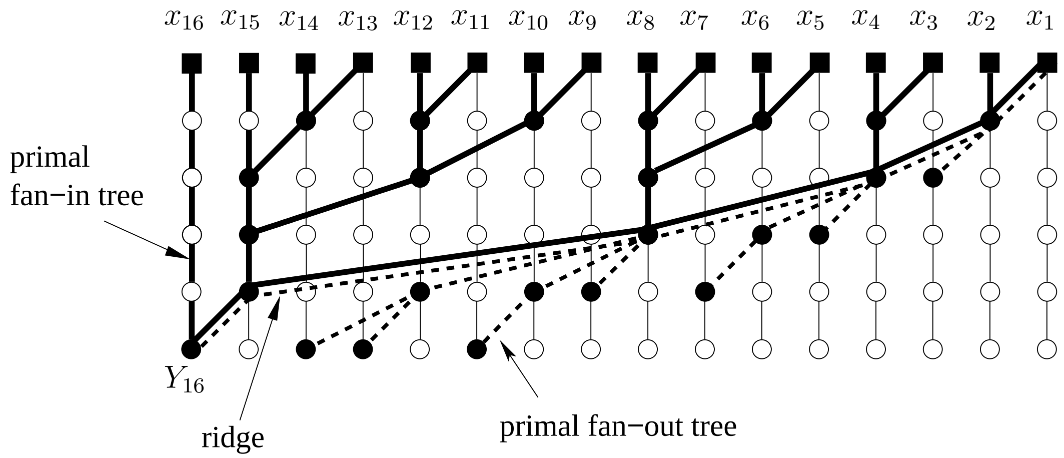

$x_{16}$ $x_{15}$ $x_{14}$ $x_{13}$ $x_{12}$ $x_{11}$ $x_{10}$ $x_9$ $x_8$ $x_7$ $x_6$ $x_5$ $x_4$ $x_3$ $x_2$ $x_1$
primal fan−in tree
$Y_{16}$
ridge
primal fan−out tree

Figure IV.6 The primal fan-in tree, primal fan-out tree and ridge of a parallel prefix circuit

call it the **primal fan-out tree**. The primal fan-out tree may not necessarily be a binary tree, since one node may fan out to more than two nodes of next level. Nevertheless, it is clear that the primal fan-out tree has at least $n-1$ nodes, which are just the $n-1$ final prefix outputs $Y_i$ $(1 \leq i \leq n-1)$. In Figure IV.6, the edges of the primal fan-out tree are shown by heavy dotted lines.

Notice that the primal fan-in tree and the primal fan-out tree actually overlap on the path from the first primary input $x_1$ to the last prefix output $Y_n$, which we call the **ridge** of the prefix circuit. There are at most $d^M_{G(n)}$ nodes on the ridge. Up to now we have had $(n-1)+(n-1)-d^M_{G(n)} = 2n-2-d^M_{G(n)}$ nodes in the prefix circuit. Since the primal fan-in tree and primal fan-out tree shall exist in every prefix circuit, this is indeed the minimum number of nodes for a general prefix circuit. Hence we have

$$s_{G(n)} + d_{G(n)} \geq s_{G(n)} + d^M_{G(n)} \geq 2n-2$$

■

In fact, each final prefix output $Y_i$ $(1 \leq i \leq n)$ is generated by a binary alphabetical tree. Only the one generates $Y_n$ is called primal because it covers all the primary inputs. Likewise, from each primary input $x_i$ $(1 \leq i \leq n)$ sprouts a

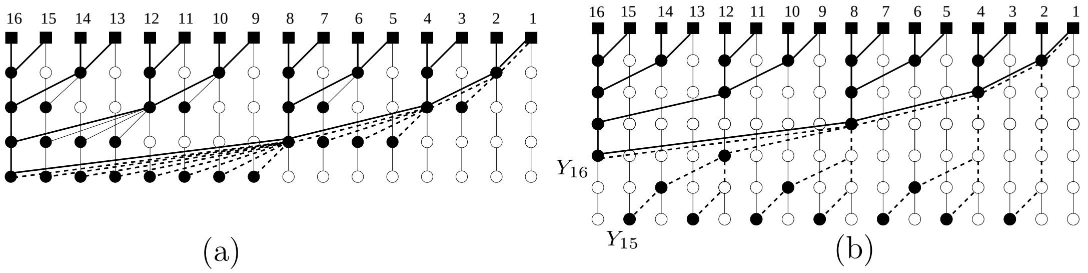


Figure IV.7 Sklansky and Brent-Kung adder decomposed

fan-out tree to $Y_j$ $(i \leq j \leq n)$. Only the one rooted at $x_1$ is called primal because it feeds all of the final prefix outputs.

It can be inferred from the above proof that a zero-deficiency prefix circuit has no nodes other than those on its primal fan-in tree and primal fan-out tree. For example, the Sklansky's prefix circuit [49] in Figure IV.7(a) has 6 nodes in addition to its primal fan-in tree and primal fan-out tree, thereby it is not a zero-deficiency circuit. Furthermore, for a zero-deficiency prefix circuit the last output $Y_n$ must also be the latest output. For instance, the Brent-Kung circuit [9] shown in Figure IV.7(b) is not a zero-deficiency circuit because $Y_{16}$ is two levels ahead of $Y_{15}$, although Brent-Kung circuit consists of only the primal fan-in tree and primal fan-out tree.

In summary, Definition IV.D.2 gives a formal treatment of the concepts involved in the proof of Theorem IV.D.1, while Corollary IV.D.3 gives the necessary and sufficient conditions for a prefix circuit to be of zero-deficiency.

**Definition IV.D.2** *For a prefix circuit $G(n)$, the binary alphabetical tree that generates the last final prefix output $Y_n$ is called the* ***primal fan-in tree*** *of $G(n)$. The broadcasting tree that propagates the first primary input $x_1$ to the final prefix outputs $Y_i$ $(1 \leq i \leq n)$ is defined as the* ***primal fan-out tree*** *of $G(n)$. The common part of the primal fan-in tree and the primal fan-out tree, that is, the path from the first primary input to the last final prefix output, is defined as the* ***ridge*** *of $G(n)$.*

**Corollary IV.D.3** *A prefix circuit $G(n)$ of depth $d$ is of zero-deficiency if and* *only if*

1. *$G(n)$ comprises only the primal fan-in tree and the primal fan-out tree, which overlap on its ridge;*

2. *$d^M_{G(n)} = d$, i.e., the last prefix output $Y_n$ is also the latest output;*

3. *The ridge has $d$ nodes, one node per level.*

### IV.D.2 Proposed Zero-deficiency Prefix Adders

In this section, we propose a new class of zero-deficiency prefix circuits, called $Z(d)$, which have the minimum depth among all zero-deficiency prefix circuits. We managed to do so from an alternative point of view, that is, by constructing a zero-deficiency prefix circuit of maximum width for a given depth. Our new philosophy leads to a systematic way of prefix circuit construction, as opposed to previous ad-hoc approaches.

We will first construct a class of parameterized trees called $T^k(t)$ trees which will be used to form the primal fan-in tree of the $Z(d)$ circuit. We then define the $A^k(t)$ trees which will be used to form the primal fan-out tree of $Z(d)$. A $Z(d)$ circuit is constructed by assembling $T^k(t)$ trees and $A^k(t)$ trees together.

The $T^k(t)$ trees follow a recursive definition as described in Figure IV.8. Note that Figure IV.8(a), (b) and (c) together define $T^k(t)$ over $1 \leq k \leq t$. A few more observations can be made by investigating the graphical definition carefully:

- $T^1(t)$ in Figure IV.8(a) are just the serial prefix circuits.

- The width of a $T^k(k)$ tree is $2^k$ for any $k \geq 1$.

- A $T^k(t)$ tree is a binary tree of depth $t$ for $1 \leq k \leq t$.

The last two observations above can be easily verified by mathematical induction based on the recursive definition.

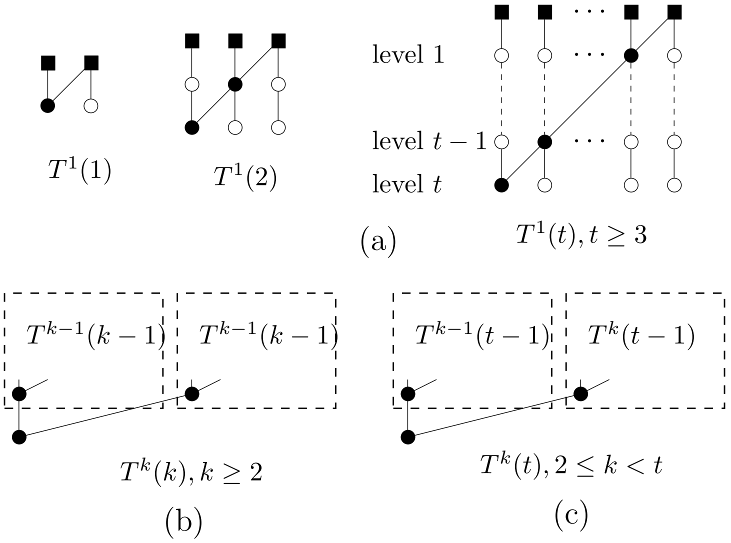


Figure IV.8 The recursive definition of $T^k(t)$ tree

Also it is interesting to note that the parameter $k$ in some sense denotes the "order" of a $T^k(t)$ tree, i.e., the level of recursions that one need to go through when tracing $T^k(t)$ back to the primitive construct $T^1(1)$. It can be proved, again by induction, that $T^k(t)$ has $k$ computation nodes on its most significant column. However, the proof is omitted here since it is not crucial in the following discussion.

As an example, Figure IV.10(a) shows the $T^3(5)$ tree whose depth is 5, and how it is constructed from $T^2(4)$ and $T^3(4)$, conforming to Figure IV.8(c). One can even trace the composition of $T^2(4)$ to $T^1(4)$ and $T^2(3)$, and similar for $T^3(4)$.

Algorithm 1, which is essentially a direct algorithmic translation of Figure IV.8, formally presents how a $T^k(t)$ tree is generated.

Following nearly the same recursive way of construction as the $T^k(t)$ trees, we define the $A^k(t)$ trees as shown in Figure IV.9. The algorithmic description of an $A^k(t)$ tree is presented in Algorithm 2. By simple induction, we can infer that for an $A^k(t)$ tree, $k+1$ is the tree depth while $t+1$ is the fan-out of the root. We also give an example of the $A^3(5)$ tree shown in Figure IV.10(b). Comparing $A^3(5)$ with $T^3(5)$ helps to point out the similarity between $A^k(t)$ and $T^k(t)$. In fact, we

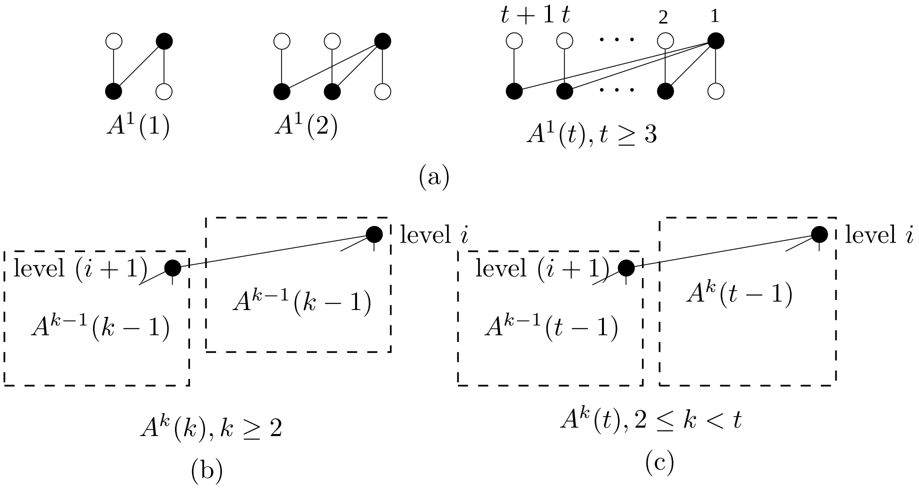


Figure IV.9 The recursive definition of $A^k(t)$ tree

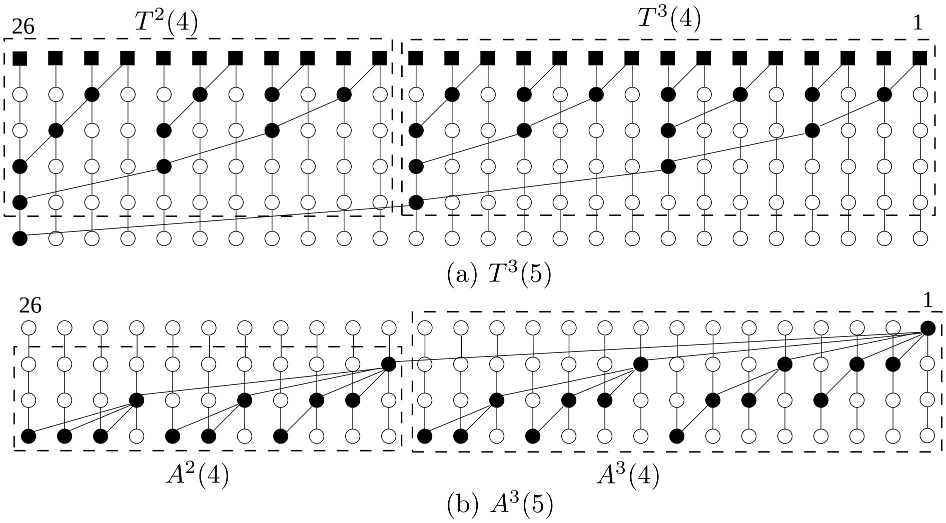


Figure IV.10 Examples of the $T^k(t)$ tree and $A^k(t)$ tree

notice that

- $A^1(t)$ and $T^1(t)$ have the same width for $t \geq 1$ (see Figure IV.8(a) and Figure IV.9(a));
- $A^k(t)$ and $T^k(t)$ shall have the same recurrence formula of the width for $2 \leq k \leq t$ (compare Figure IV.8(b) with Figure IV.9(b), and Figure IV.8(c) with Figure IV.9(c)).

Therefore, it is straightforward to conclude that $T^k(t)$ and $A^k(t)$ have the same

```
Function: TtreeGen($k$,$t$)
Input    : $k \in \mathbb{Z}+$, $t \in \mathbb{Z}+$
Output   : $T^k(t)$ Circuit

if $t < k$ then
    return -1;
    // $T^k(t)$ does not exist.
else if $k = 1$ then
    Generate $T^1(t)$ according to Figure IV.8(a);
    return $T^1(t)$;
else if $k = t$ then
    $T^{k-1}(k-1) \leftarrow$ TtreeGen $(k-1, k-1)$;
    Construct $T^k(k)$ according to Figure IV.8(b);
    return $T^k(k)$;
else
    $T^{k-1}(t-1) \leftarrow$ TtreeGen $(k-1, t-1)$;
    $T^k(t-1) \leftarrow$ TtreeGen $(k, t-1)$;
    Construct $T^k(t)$ according to Figure IV.8(c);
    return $T^k(t)$;
end
```

**Algorithm 1**: Generation of the $T^k(t)$ circuit.

width.

Summarizing the above discussion, we have the following theorem:

**Theorem IV.D.4** *Given $k \in \mathbb{Z}+$ and $t \in \mathbb{Z}+$ s.t. $1 \leq k \leq t$, the following properties regarding $T^k(t)$ and $A^k(t)$ hold:*

1. *$T^k(t)$ and $A^k(t)$ have the same width, which is*

$$N(k,t) = \sum_{i=0}^{k} \binom{t}{i} \textit{ for } 1 \leq k \leq t \tag{IV.17}$$

2. *$T^k(t)$ is a binary tree of depth $t$ and size $N(k,t) - 1$;*

```
Function: AtreeGen($k$,$t$)
Input    : $k \in \mathbb{Z}+$, $t \in \mathbb{Z}+$
Output   : $A^k(t)$ Circuit

if $t < k$ then
    return -1;
    // $A^k(t)$ does not exist.
else if $k = 1$ then
    Generate $A^1(t)$ according to Figure IV.9(a);
    return $A^1(t)$;
else if $k = t$ then
    $A^{k-1}(k-1) \leftarrow$ AtreeGen $(k-1, k-1)$;
    Construct $A^k(k)$ according to Figure IV.9(b);
    return $A^k(k)$;
else
    $A^{k-1}(t-1) \leftarrow$ AtreeGen $(k-1, t-1)$;
    $A^k(t-1) \leftarrow$ AtreeGen $(k, t-1)$;
    Construct $A^k(t)$ according to Figure IV.9(c);
    return $A^k(t)$;
end
```

**Algorithm 2**: Generation of the $A^k(t)$ circuit.

3. *$A^k(t)$ has a depth of $k+1$ and a size of $N(k,t)$, exactly one computation node per column.*

**Proof:** According to the definition of $T^k(t)$, $N(k,t)$ obeys the following recurrence formula:

$$N(k,t) = N(k,t-1) + N(k-1,t-1) \quad \text{for } 2 \leq k < t \tag{IV.18}$$

and boundary conditions:

$$N(1,t) = t \quad \text{for } t \geq 1 \tag{IV.19}$$

$$N(k,k) = 2^k \quad \text{for } k \geq 2 \tag{IV.20}$$

Although $N(k,t)$ is defined over $1 \leq k \leq t$ only, mathematically it is valid to extend $N(k,t)$ to the entire first quadrant of $t \geq 0, k \geq 0$. Interestingly if we set the new boundary conditions to be

$$N(0,t) = N(k,0) = 1 \quad \text{for } t \geq 0, k \geq 0 \tag{IV.21}$$

Then the extended $N(k,t)$ sequence still satisfies the recurrence equation (IV.18) for $t \geq 1, k \geq 1$, as well as the original boundary conditions (IV.19) and (IV.20).

We can then construct a generating function using the extended $N(k,t)$ as the corresponding coefficients:

$$F(x,y) = \sum_{\substack{k\geq 0\\ t\geq 0}} N(k,t)x^k y^t \tag{IV.22}$$

We can further derive that

$$\begin{aligned}
F(x,y) &= \sum_{k\geq 0} x^k + \sum_{t\geq 1} y^t + \sum_{\substack{k\geq 1\\ t\geq 1}} N(k,t)x^k y^t \\
&= \sum_{k\geq 0} x^k + \sum_{t\geq 1} y^t + \sum_{\substack{k\geq 1\\ t\geq 1}} (N(k,t-1) + N(k-1,t-1))x^k y^t \\
&= \sum_{k\geq 0} x^k + y\sum_{t\geq 0} y^t + y\sum_{\substack{k\geq 1\\ t\geq 0}} N(k,t)x^k y^t + xy\sum_{\substack{k\geq 0\\ t\geq 0}} N(k,t)x^k y^t \\
&= \sum_{k\geq 0} x^k + yF(x,y) + xyF(x,y) \qquad \text{(IV.23)}
\end{aligned}$$

Thus

$$F(x,y) = \frac{1}{1-y(1+x)} \cdot \sum_{k\geq 0} x^k = \sum_{k\geq 0} (y(1+x))^k \cdot \sum_{k\geq 0} x^k \tag{IV.24}$$

Recall that $N(k,t)$ is the coefficient of term $x^k y^t$ in $F(x,y)$. The only term containing $y^t$ in the right-hand side of (IV.24) is $y^t(1+x)^t$, and

$$(1+x)^t = \sum_{i=0}^{t} \binom{t}{i} x^i \tag{IV.25}$$

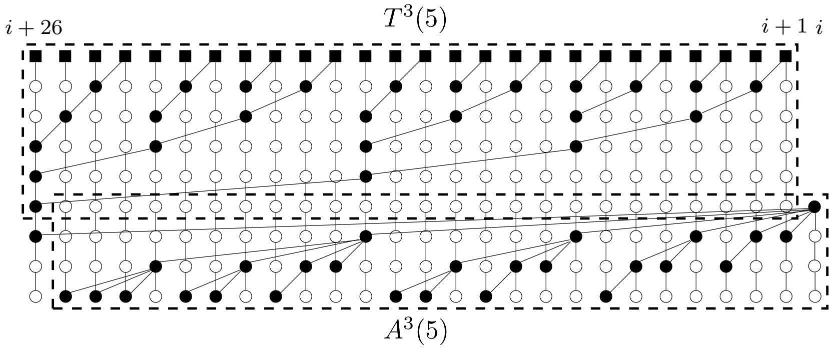


Figure IV.11 Assembling $T^3(5)$ and $A^3(5)$

Clearly, only the first $k+1$ terms of the expansion of $(1+x)^t$ will generate $x^k$ when multiplied by proper terms in $\sum_{k\geq 0} x^k$. Therefore

$$N(k,t) = \sum_{i=0}^{k} \binom{t}{i} \quad \text{for } 1 \leq k < t \tag{IV.26}$$

Property (2) and (3) can be proved directly using mathematical induction which is omitted here.

■

Note that when $t = k$, the width of $T^k(k)$ is $N(k,k) = \sum_{i=0}^{k} \binom{k}{i} = 2^k$, which is in accordance with our previous observation.

It is interesting to note that $T^3(5)$ and $A^3(5)$ can be assembled together to form a prefix circuit, as shown in Figure IV.11. If we consider the root of $A^3(5)$ as the final prefix output $Y_i$, from the figure we can tell that the assembled structure produces all the final prefix outputs $Y_j$ for $i+1 \leq j \leq i+26$. In general, we can always combine a pair of a $T^k(t)$ tree and an $A^k(t)$ tree to form a prefix circuit as shown in Figure IV.12.

**Theorem IV.D.5** *Assemble $T^k(t)$ and $A^k(t)$ trees together as shown in Figure IV.12(a), and denote the resulted structure as $TA^k(t)$:*

1. *$TA^k(t)$ generates the final prefix outputs $Y_j$ $(i+1 \leq j \leq i+N(k,t))$ if the root of $A^k(t)$ is the final prefix output $Y_i$;*

2. *The depth of $TA^k(t)$ is $k+t$;*

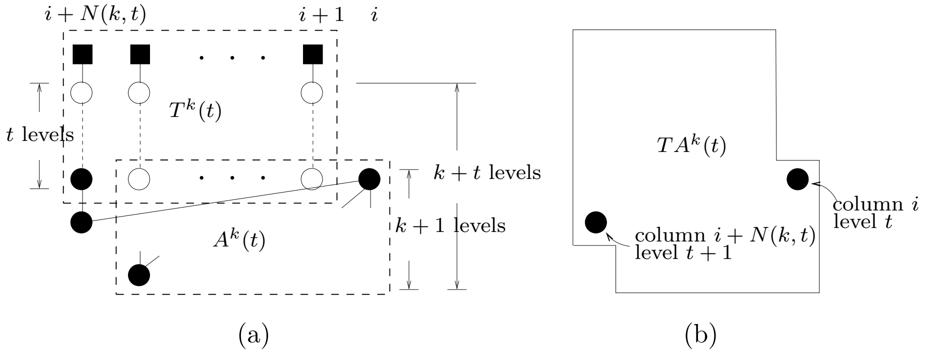


Figure IV.12 Assembling $T^k(t)$ and $A^k(t)$ into prefix circuit $TA^k(t)$

*3. The size of* $TA^k(t)$ *is* $2N(k,t)$*.*

**Proof:** We will first prove (1) by mathematical induction. The base cases of $TA^1(1)$ and $TA^2(1)$ can be verified with ease.

Using the definitions of $T^k(t)$ and $A^k(t)$, the $TA^k(t)$ circuit defined in Figure IV.12(a) can be decomposed into the one shown in Figure IV.13. Note that if we move $A^k(t-1)$ one level toward $T^k(t-1)$ then essentially they form $TA^k(t-1)$. Therefore by inductive assumption, the final prefix outputs $Y_j$ for $i+1 \leq j \leq i+N(k,t-1)$ are available.

Likewise, moving $A^{k-1}(t-1)$ two level closer to $T^{k-1}(t-1)$ would yield $TA^{k-1}(t-1)$. Since the root of $A^{k-1}(t-1)$ is the final prefix output $Y_{i+N(k,t-1)}$, the final prefix outputs $Y_j$ for $i+N(k,t-1)+1 \leq j \leq i+N(k,t)$ are available too.

Second, from Figure IV.12(a) we can immediately tell that the depth of $TA^k(t)$ is $(k+t)$ since $T^k(t)$ and $A^k(t)$ have one level overlapping.

Finally,

$$\begin{aligned}\text{size of } TA^k(t) &= 1 + (\text{size of } T^k(t)) + (\text{size of } A^k(t)) \\ &= 1 + (N(k,t) - 1) + N(k,t) \\ &= 2N(k,t)\end{aligned}$$

■

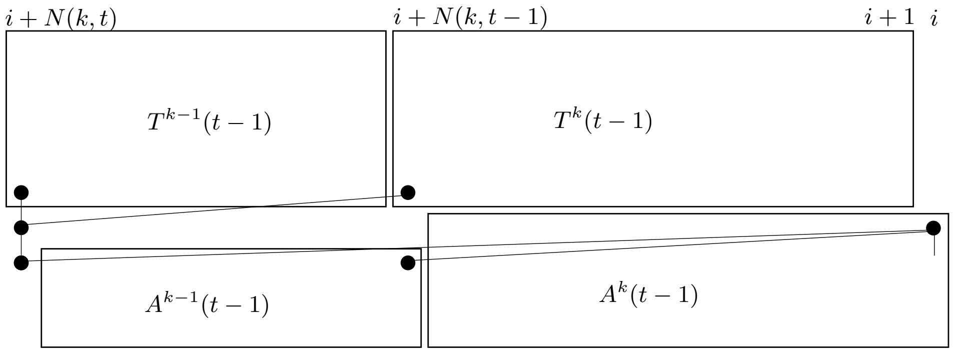


Figure IV.13 A decomposed view of $TA^k(t)$

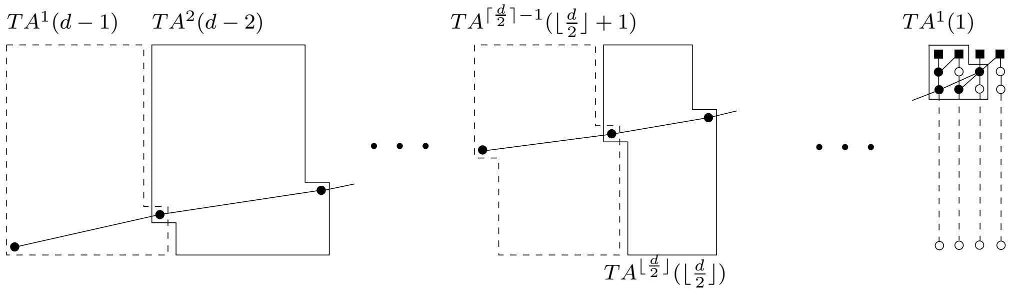

$TA^1(d-1)$
$TA^2(d-2)$
$TA^{\lceil \frac{d}{2} \rceil -1}(\lfloor \frac{d}{2} \rfloor + 1)$
$TA^1(1)$
$TA^{\lfloor \frac{d}{2} \rfloor}(\lfloor \frac{d}{2} \rfloor)$

Figure IV.14 A new class of zero-deficiency prefix circuits $Z(d)$

The proposed $Z(d)$ circuit is constructed by concatenating $TA^i(i)$ $(1 \leq i \leq \lfloor d/2 \rfloor)$ and $TA^{d-i}(i)$ $(\lfloor d/2 \rfloor < i \leq d-1)$ in the increasing order of $i$, as illustrated in Figure IV.14. When concatenating two neighboring circuits $TA^{k_1}(i)$ and $TA^{k_2}(i+1)$, the most significant output of $TA^{k_1}(i)$ merges with the least significant input of $TA^{k_2}(i+1)$, allowing the former to feed the latter. Therefore, starting from $TA^1(1)$, the $Z(d)$ circuit works really like a line of dominos, generating prefix results for every column.

As an example, Figure IV.15 shows the $Z(d)|_{d=8}$ circuit, which is constructed from $TA^1(7)$, $TA^2(6)$, $TA^3(5)$, $TA^4(4)$, $TA^3(3)$, $TA^2(2)$ and $TA^1(1)$. Algorithm 3 describes the generation of the $Z(d)$ circuit, while Theorem IV.D.6 gives the width of the $Z(d)$ circuit.

**Theorem IV.D.6** *The width of the $Z(d)$ circuit, which we denote by $N_Z(d)$, is*

$$N_Z(d) = F(d+3) - 1 \textit{ for } d \geq 1 \tag{IV.27}$$

*where $F(k)$ is the well-known Fibonacci series defined by the recurrence*

$$F(1) = F(2) = 1, F(k) = F(k-1) + F(k-2) \quad for\ k \geq 3$$

**Proof:** From Figure IV.14 we see that the width of $Z(d)$ is just the sum of the widths of all its constituent $TA^k(t)$ circuits plus 2.

$$\begin{aligned} N_Z(d) &= 2 + \sum_{i=1}^{\lfloor \frac{d}{2} \rfloor} N(i,i) + \sum_{i=\lfloor \frac{d}{2} \rfloor + 1}^{d-1} N(d-i,i) \\ &= 2^{\lfloor \frac{d}{2} \rfloor + 1} + \sum_{i=1}^{\lceil \frac{d}{2} \rceil - 1} N(i, d-i) \qquad \text{(IV.28)} \end{aligned}$$

What remains is to derive a closed form for the right-hand side of the above equation. In writing $N_Z(d)$ we will distinguish the cases of $d$ even and $d$ odd.

$$N_Z(2d) = 2^{d+1} + \sum_{i=1}^{d-1} \sum_{j=0}^{i} \binom{2d-i}{j} \qquad \text{(IV.29)}$$

$$N_Z(2d+1) = 2^{d+1} + \sum_{i=1}^{d} \sum_{j=0}^{i} \binom{2d+1-i}{j} \qquad \text{(IV.30)}$$

Taking the difference of $N_Z(2d+1)$ and $N_Z(2d)$, and utilizing the well-known binomial equation $\binom{n}{k} - \binom{n-1}{k} = \binom{n-1}{k-1}$, we have

$$\begin{aligned} N_Z(2d+1) - N_Z(2d) &= \sum_{i=1}^{d} \sum_{j=0}^{i} \binom{2d+1-i}{j} - \sum_{i=1}^{d-1} \sum_{j=0}^{i} \binom{2d-i}{j} \\ &= \sum_{j=0}^{d} \binom{d+1}{j} + \sum_{i=1}^{d-1} \sum_{j=0}^{i} \left( \binom{2d+1-i}{j} - \binom{2d-i}{j} \right) \\ &= (2^{d+1} - 1) + \sum_{i=1}^{d-1} \sum_{j=1}^{i} \binom{2d-i}{j-1} \\ &= 2^{d+1} + \sum_{i=1}^{d-1} \sum_{j=0}^{i} \binom{2d-i}{j} - 1 - \sum_{i=1}^{d-1} \binom{2d-i}{i} \\ &= N_Z(2d) - \sum_{i=0}^{d} \binom{2d-i}{i} + 1 \qquad \text{(IV.31)} \end{aligned}$$

Therefore

$$2N_Z(2d) - N_Z(2d+1) = \sum_{i=0}^{d} \binom{2d-i}{i} - 1 \tag{IV.32}$$

Similarly, we have

$$2N_Z(2d+1) - N_Z(2d+2) = \sum_{i=0}^{d} \binom{2d+1-i}{i} - 1 \tag{IV.33}$$

It is easy to show by induction that

$$\sum_{i=0}^{d} \binom{2d-i}{i} = F(2d+1) \tag{IV.34}$$

$$\sum_{i=0}^{d} \binom{2d+1-i}{i} = F(2d+2) \tag{IV.35}$$

where $F(r)$ denotes the $r^{th}$ Fibonacci number. Now let us substitute Equations (IV.34) and (IV.35) into Equations (IV.32) and (IV.33) respectively, and then unify these two equations, we obtain

$$N_Z(d+1) = 2N_Z(d) - F(d+1) + 1 \tag{IV.36}$$

Again, it can be shown by induction that

$$N_Z(d) = F(d+3) - 1 \text{ for } d \geq 1 \tag{IV.37}$$

■

```
Function: ZGen(d)
Input    : $d \in \mathbb{Z}+$
Output   : Zero-deficiency prefix circuit $Z(d)$
if $d = 1$ then
    $Z(1) \leftarrow T^1(1)$;
    return $Z(1)$;
else
    for $i = 1$ to $\lceil \frac{d}{2} \rceil - 1$ do
        $T^i(d-i) \leftarrow$TtreeGen $(i, d-i)$; // call algorithm 1;
        $A^i(d-i) \leftarrow$AtreeGen $(i, d-i)$; // call algorithm 2;
        Stitch $T^i(d-i)$ and $A^i(d-i)$ into $TA^i(d-i)$ as shown in
        Figure IV.12(a);
    end
    for 1 to $i = \lfloor \frac{d}{2} \rfloor$ do
        $T^i(i) \leftarrow$TtreeGen $(i, i)$; // call algorithm 1;
        $A^i(i) \leftarrow$AtreeGen $(i, i)$; // call algorithm 2;
        Stitch $T^i(i)$ and $A^i(i)$ into $TA^i(i)$ as shown in
        Figure IV.12(a);
    end
    Stitch all the T, A trees together to form $Z(d)$ as shown in
    Figure IV.14;
end
```

**Algorithm 3**: Generation of the $Z(d)$ circuit.

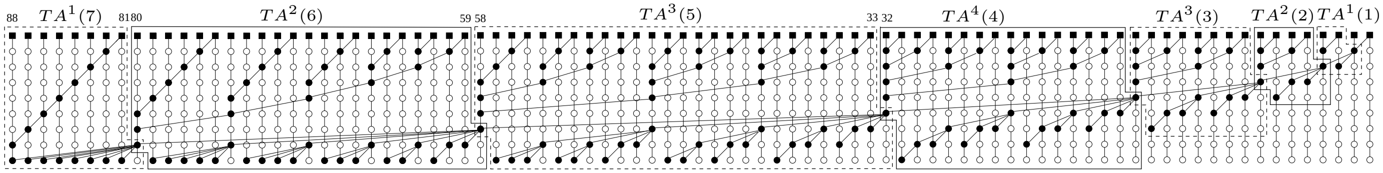


Figure IV.15 An example of the $Z(d)$ circuit: $Z(d)|_{d=8}$

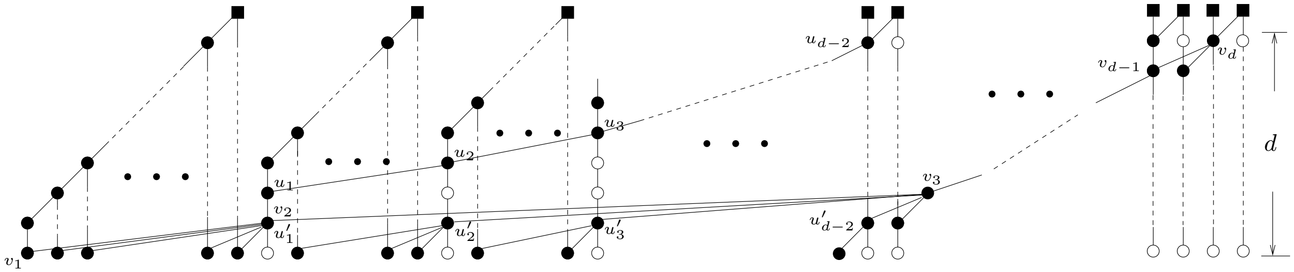


Figure IV.16 Zero-deficiency prefix circuit of maximum width in action.

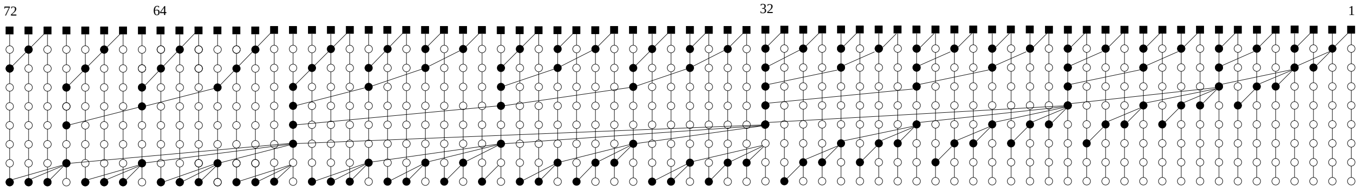


Figure IV.17 Variant of $Z(d)|_{d=8}$ with limited fanout 4

In order to prove the optimality of the $Z(d)$ circuit, we shall first show that the $Z(d)$ circuit is indeed of zero-deficiency, which is addressed in Theorem IV.D.7, and prove that it does have the minimum depth among all possible zero-deficiency circuits, which is addressed in Theorem IV.D.8. Actually, the intuition behind constructing the $Z(d)$ circuit is that we can try to find the zero-deficiency prefix circuit of maximum width for a given depth, instead of finding the zero-deficiency prefix circuit of minimum depth for a given width. Of course, the two points of view are actually equivalent.

**Theorem IV.D.7** *The parallel prefix circuit $Z(d)$ shown in Figure IV.14 is of zero-deficiency.*

**Proof:** The depth of $Z(d)$ is automatically $d$ by the way we constructed it. The size of $Z(d)$ is the sum of the sizes of all the constituent $TA^k(t)$ circuits. Note that every two neighboring $TA^k(t)$ circuits have one node in common, and there are in total $d-2$ common nodes. Therefore,

$$\begin{aligned} s_{Z(d)} &= \sum (\text{size of } AT^k(t) \text{ circuits}) - (d-2) \\ &= 2(N_Z(d)-2) - (d-2) \\ &= 2N_Z(d) - d - 2 \end{aligned}$$

That is:

$$s_{Z(d)} + d = 2N_Z(d) - 2$$

Therefore, $Z(d)$ is of zero-deficiency.

■

**Theorem IV.D.8** *$Z(d)$ has the maximum width for a given depth $d$ among all zero-deficiency prefix circuits.*

**Proof:** We will only sketch the arguments. We would like to show that given a depth $d$, the widest zero-deficiency circuit that we can build is actually $Z(d)$.

Suppose $G(n)$ is a zero-deficiency circuit of depth $d$. According to Corollary IV.D.3, the ridge of $G(n)$ has $d$ nodes on it, one node per level. Starting from the last final prefix output $Y_n$, let us number the nodes on the ridge as $v_1$,$v_2$,$\cdots$,$v_d$ in order, as shown in Figure IV.16, and correspondingly denote the columns where they reside in as $col_{v_1}$, $col_{v_2}$, $\cdots$, $col_{v_d}$. Node $v_i$ is at level $d-i+1$.

1. For columns between $col_{v_1}$ and $col_{v_2}$, the only way to get the prefix results is to take node $v_2$ as the input and produce the results at level $d$. Accordingly, the only possible structure above the $v_1 - -v_2$ connection is a serial prefix circuit, and its maximum possible width is $d$.

2. Now consider the columns between $col_{v_2}$ and $col_{v_3}$. The structure above the $v_2 - -v_3$ connection is an alphabetical binary tree, and its root is the node directly above $v_2$. For this tree to have a span as wide as possible, the path from its root to its least significant input must increase exactly one level each time. The total number of nodes on this path is $d-2$, and every node is a partial prefix result. Let us label these nodes as $u_1$, $u_2$,$\cdots$, $u_{d-2}$, and their corresponding columns as $col_{u_1}$, $col_{u_2}$,$\cdots$,$col_{u_{d-2}}$. $Z(d)$'s prefix results at columns $col_{u_i}$ $(i = 1\cdots d-2)$ can be generated at level $d-1$ by directly taking $v_3$ as the input.

3. Again, label these output nodes at level $d-1$ as $u'_1$, $u'_2$,$\cdots$, $u'_{d-2}$ (note $u'_1$ is just $v_2$). The outputs of the columns between $u'_1$ and $u'_2$ have no other choice but to take $u'_2$ as the input directly. Accordingly, the structure above the $u_1 - -u_2$ connection must be a serial prefix circuit, and its maximum width is $d-2$. Likewise, the structure above the $u_2 - -u_3$ connection is also a prefix circuit whose maximum width is $d-3$, and so on. Therefore, the binary tree above the $v_2 - -v_3$ connection is identical to the $T^2(d-2)$ tree.

We can continue the above analysis to the lower columns in a similar way. It turns out that in order to make the whole zero-deficiency circuit as wide as possible, the binary tree above the $v_3 - -v_4$ connection must be the $T^3(d-3)$

Table IV.2 Widths of $LS(n)$, $LYD(n)$ and $Z(d)$ circuits

| depth | $W_{LS}$ | $W_{LYD}$ | $W_Z$ | depth | $W_{LS}$ | $W_{LYD}$ | $W_Z$ |
|---|---|---|---|---|---|---|---|
| 3 | 7 | 7 | 7 | 13 | 260 | 308 | 986 |
| 4 | 11 | 12 | 12 | 14 | 383 | 446 | 1596 |
| 5 | 16 | 20 | 20 | 15 | 517 | 576 | 2583 |
| 6 | 23 | 33 | 33 | 16 | 575 | 843 | 4180 |
| 7 | 33 | 54 | 54 | 17 | 1030 | 1101 | 6764 |
| 8 | 47 | 77 | 88 | 18 | 1535 | 1625 | 10945 |
| 9 | 66 | 95 | 143 | 19 | 2055 | 2139 | 17710 |
| 10 | 95 | 135 | 232 | 20 | 3071 | 3176 | 28656 |
| 11 | 131 | 169 | 376 | 21 | 4104 | 4202 | 46367 |
| 12 | 191 | 242 | 609 | 22 | 6143 | 6264 | 75024 |

tree, and the structure under the $v_3 - -v_4$ connection is just an $A^3(d-3)$ tree, and so on. Thus, the resulting circuit is exactly $Z(d)$ as we defined in Figure IV.14.

■

Table IV.2 shows the widths of $Z(d)$ circuits for $3 \leq d \leq 22$, with the results of Lin's design [42] and the $LYD(n)$ circuit [37] listed for comparison. Each number is read as the maximum width up to which a zero-deficiency prefix circuit of that type under the given depth can be constructed. Clearly, our design dominates the other two, especially when the depth is large.

## IV.E Extensions

It is now clear that the curve of function $d = f(n)$ in Figure IV.3 is just the inverse function of $N_{Z(d)} = F(d+3) - 1$ given in Theorem IV.D.6. Formally, we have

**Theorem IV.E.1** *The minimum depth of any n-input zero-deficiency prefix circuit is given by*

$$d_{min}(n) = \min\{t : F(t) \geq n+1\} - 3 \qquad \text{(IV.38)}$$

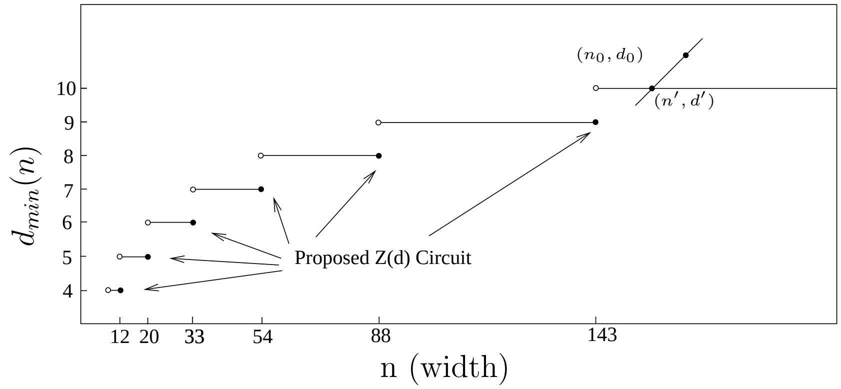


Figure IV.18 Minimum depth of zero-deficiency prefix circuits as a function of width $n$

Figure IV.18 sketches the shape of $d_{min}(n)$, which is a piecewise constant function. Note that the $Z(d)$ circuits described in Algorithm 3 only represent the extreme cases of the zero-deficiency circuits, namely, the rightmost points of the step segments in Figure IV.18. It is of practical interest to construct zero-deficiency circuits for every width-depth pair $(n, d)$ such that $d \geq d_{min}(n)$. Fortunately, this is not difficult to do based on the proposed $Z(d)$ circuit, and can be tackled in two steps.

First, we need to consider all the $(n_0, d_0)$ pairs residing on the curve of $d_{min}(n)$. For these points, we can simply construct $Z(d)|_{d=d_0}$ first, whose width is $F(d_0 + 3) - 1$, and discard the most significant $F(d_0 + 3) - 1 - n_0$ columns. For example, a 64-input prefix circuit of depth 8 can be obtained by discarding the leading 24 columns of $Z(d)|_{d=8}$. Care must be taken to make small adjustments on the new most significant columns after discarding.

For $(n_0, d_0)$ pairs such that $d_0 > d_{min}(n_0)$, we first draw a 45 degree line though $(n_0, d_0)$, intersecting $d_{min}(n)$ at $(n', d')$, as shown in Figure IV.18. We can then construct zero-deficiency circuit for $(n', d')$, and concatenate it with a $(d_0 - d')$-input serial prefix circuit.

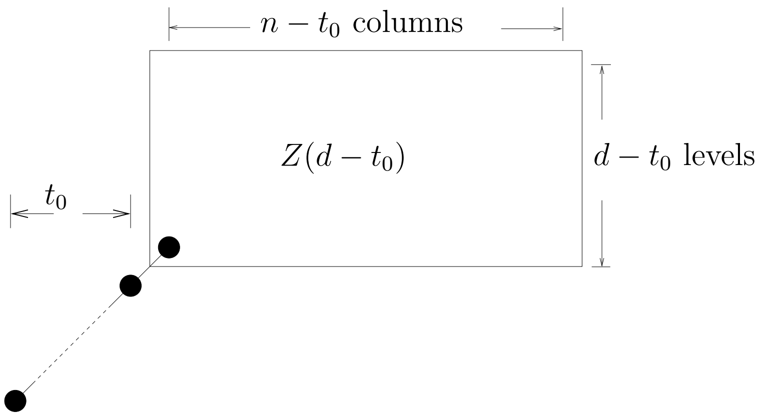
The above idea translates into Algorithm 4, which for an arbitrary $(n, d)$ pair generates a zero-deficiency prefix circuit.

```
Function: ZXGen(n,d)
Input    : $n \in \mathbb{Z}+$, $d \in \mathbb{Z}+$
Output   : A zero-deficiency prefix circuit ZX(n,d) of width $n$ and
           depth $d$
if $d < d_{min}(n)$ then
    return -1;
else
    Find $t_0$ s.t. $d - t_0 = d_{min}(n - t_0)$;
    $Z(d - t_0)$ ←ZGen $(d - d_0)$; // Call Algorithm 3;
    $ZX(n - t_0, d - t_0)$ ←truncating the $t_0$ most significant columns
    of $Z(d - t_0)$;
    $ZX(n, d)$ ←Concatenate $ZX(n - t_0, d - t_0)$ with a $t_0$-input
    serial prefix circuit as shown in Figure IV.19;
    return ZX(n,d);
end
```

**Algorithm 4**: Generation of zero-deficiency circuit for arbitrary $(n, d)$ pair

Another possible extension problem is to generalize the $Z(d)$ circuit and consider fan-out constraints. Lin et. al. has done a lot of work in this respect, that is, constructing zero-deficiency[4] prefix circuits of limited fan-out, especially 2 and 4 [38] [41] [39] [40]. Their approaches are largely constructive and cannot be generalized for arbitrary fan-out constraints. However, we may have a different way of constructing fan-out limited prefix circuits by directly pruning the out branches of large fan-out nodes in our proposed $Z(d)$ circuit. A preliminary study shows that by doing this way we can generate a prefix circuit of up to 72-bit for depth constraint 8 and fan-out limit 4, as shown in Figure IV.17. By contrast, under the same constraints the maximum widths that can be achieved by H4 [41], Z4 [39] and WE4 [40] are 64, 67 and 57 respectively.

[4] Lin et. al. had used depth-size optimal to denote the concept of zero-deficiency in their papers. However, we explicitly distinguish depth-size optimal and zero-deficiency, as stated in Section IV.C.1.



$t_0$ is a positive integer such that $d - t_0 = d_{min}(n - t_0)$.

Figure IV.19 Zero-deficiency prefix circuit for arbitrary $(n, d)$ pair

## IV.F Summary

In [61], Zimmermann proposed a non-heursitic algorithm to generate parallel prefix circuit of small size under a given depth constraint. His algorithm consists of two steps: compression and expansion. The compression step starts with a serial prefix circuit and compact it as much as possible, effectively reducing the depth at the cost of increasing size. The expansion step tries to reduce the size by bouncing back the depth subject to the depth constraint. His algorithm in many cases can produce zero-deficiency prefix circuit. However, since his algorithm is greedy in nature, there is no guarantee on the optimality. In contrast, the proposed $Z(d)$ circuit and its generalization has provable optimality for $d \geq d_{min}(n)$, and shows its advantage when the depth constraint is really tight. For example, given width 54 and depth 7, our design has 99 nodes, while Zimmermann's algorithm gives a design of 104 nodes [60].

Another merit of the proposed structure is that, when the depth constraint is loose, Algorithm 4 tends to compact all the nodes to lower levels and columns. Such a scheme can achieve low power in the context of prefix adder design, because cells of higher logic levels usually have higher activity rates which are proportional to dynamic power consumption. Zimmermann's algorithm, however, tends to bring nodes to higher levels in the expansion step. A detailed analysis on power characteristic of the proposed structure is planned for future work.

On the other hand, Zimmermann's algorithm can handle cases of integer, non-uniform input arrival times (AT), while the proposed approach only considers the case of uniform AT. Hence it is interesting to extend our idea to produce *provably* good prefix circuits under non-uniform AT. A conceivable way to attack this problem is to partition the whole circuit into consecutive sub-blocks, each with an uniform AT profile, and solve the sub-blocks individually and sequentially. In this case, each sub-block may receive a delayed first input, which is generated by the previous sub-block.

In this chapter, we have proposed a new class of zero-deficiency prefix circuits $Z(d)$ which has the minimum possible depth for a given width. We described the construction of $Z(d)$ and proved its optimality in detail. We also extend $Z(d)$ circuits to consider general width and depth specifications. Thus for $d \geq d_{min}(n)$ (as established in Theorem IV.E.1), the optimal depth-size tradeoff problem for prefix circuits is completely resolved.

This chapter, in part, is a reprint of the paper "On the construction of zero-deficiency parallel prefix circuits with minimum depth" co-authored with Chung-Kuan Cheng and Ronald Graham in ACM Transactions on Design Automation of Electronic Systems, Vol. 11, Issue 2, pp.387-409, April 2006. The dissertation author was the primary investigator and author of this paper.

# V Conclusions

## V.A Summary of Contributions

In this dissertation, we have focused on novel electrical serial link design and datapath optimization simultaneously toward realizing high-performance and low-power VLSI systems in the nanometer era. We demonstrated that not only system level interconnects are of crucial importance, but also datapaths can benefit considerably from treating interconnects with proper care.

For system level (on-chip global level and above) interconnects we proposed and demonstrated the superiority of a passive compensation scheme. Transmission line structures support wave propagation which naturally resolves the communication latency problem that perplexes the conventional RC wires. Nevertheless, transmission lines suffer from the ISI-induced distortion which greatly limits the bandwidth. The proposed method approaches distortionless (i.e. unlimited bandwidth) communication by using evenly distributed resistors to flatten out the frequency response of the interconnect. Furthermore, we developed an analytical method for predicting the jitter and eye-opening based on arbitrary bitonic responses. This technique allows us to quickly explore the design space such as shunt spacing/value to achieve the best result.

For datapath, we have chosen cyclic shifter as a representative to corroborate our interconnect-centric design methodology. Two approaches were put forward to alleviate the interconnect complexity in shifter network. The fanout-splitting technique reduces the critical path wire load from $O(n \log n)$ to $O(n)$, and saves dynamic power on the wires. The ILP-based cell order optimization can be orthogonally applied to further minimize the amount of wires.

We also studied the problem of depth-size tradeoff in prefix adders, and designed a new class of zero-deficiency adders of minimum depth. Since the size (i.e. the number of nodes) of a prefix adder is an indicator of its power consumption, the proposed structure achieves high performance with only marginal area/power cost.

## V.B Future Work

A number of interesting questions are left open for future research. For designing distortionless transmission line, it would be instrumental to quantify and compare the relative impacts of phase velocity and attenuation on signal quality. This will further our understanding of the performance limitators for interconnects at various scales.

For cyclic shifter an possible extension is to apply the fanout splitting technique and cell order optimization to the ternary shifter [53].

For the proposed zero-deficiency prefix adder, we would like study its variants when maximum fanout, number of wire tracks and maximum wire span are taken into account. These factors are of practical interest if we are to make the design of real use.

# Bibliography

[1] Ibm broadband transmission-line characterization using short-pulse propagation, 2006.

[2] Ibm electromagnetic field solver suite of tools, 2006.

[3] Moore's law, 2006.

[4] K. P. Acken, M. J. Irwin, and R. M. Owens. Power comparisons for barrel shifter. In *ISLPED*, pages 209–212, 1996.

[5] Ehsan Afshari and Ali Hajimiri. Nonlinear transmission lines for pulse shaping in silicon. *IEEE Journal of Solid-State Circuits*, 40(3):744–752, 2005.

[6] Semiconductor Industry Association. *International Technology Roadmap for Semiconductors.* 2005.

[7] Halil Burhan Bakoglu. *Circuits, Interconnections, and Packaging for VLSI.* Addison-Wesley, 1990.

[8] Avinoam Bilgory and Daniel D Gajski. A heuristic for suffix solutions. *IEEE Transactions on Computers*, 35(1):34–42, January 1986.

[9] R. P. Brent and H. T. Kung. A regular layout for parallel adders. *IEEE Transactions on Computers*, 31(3):260–264, March 1982.

[10] Richard T. Chang, Niranjan Talwalkar, C. Patrick Yue, and S. Simon Wong. Near speed-of-light signaling over on-chip electrical interconnects. *IEEE Journal of Solid-State Circuits*, 38(5):834–838, May 2003.

[11] Guoqing Chen and Eby G. Friedman. Low-power repeaters driving rc and rlc interconnects with delay and bandwidth constraints. *IEEE Transactions on VLSI*, 14(2):161–162, February 2006.

[12] Chung-Kuan Cheng, John Lillis, Shen Lin, and Norman Chang. *Interconnect Analysis and Synthesis.* Wiley Interscience, 2000.

[13] William J. Dally and J. Poulton. Transmitter equalization for 4-gbps signaling. *Micro, IEEE*, 17(1):48–56, January 1997.

[14] William J. Dally and John W. Poulton. *Digital System Engineering.* Cambridge University Press, 1998.

[15] F. E. Fich. New bounds for parallel prefix circuits. In *Proc. of the 15th Annual ACM Symposium on Theorey of Computing (STOC '83)*, pages 100–109, New York, NY, USA, May 1983. ACM Press.

[16] J. Fishburn. A depth-decreasing heuristic for combinational logic; or how to convert a ripple carry adder into a carry-lookahead adder or anything in-between. In *DAC*, pages 361–364, 1990.

[17] Michael P. Flynn and JOshua Jaeyoung Kang. Global signaling over lossy transmission lines. In *ICCAD*, pages 985–992, November 2005.

[18] Shinichiro Gomi, Kohichi Nakamura, Hiroyuki Ito, Kenichi Okada, and Kazuya Masu. Differential transmission line interconnect for high speed and low power global wiring. In *CICC*, pages 325–328, September 2004.

[19] T. Han and D. A. Carlson. Fast area-efficient adders. In *Proc. of the 8th Symposium on Computer Arithmetic*, pages 49–56, May 1987.

[20] D. Harris and I. Sutherland. Logical effort of carry propagate adders. In *Proc. of the Asilomar Conf. on Signals Systems and Computers*, volume 1, pages 873–878, 2003.

[21] David M. Harris. A taxonomy of parallel prefix networks. In *Proc. of the 37th Asilomar Conference on Signals, Systems and Computers*, volume 2, pages 2213–2217, November 2003.

[22] Masanori Hashimoto, Akira Tsuchiya, Akinori Shinmyo, and Hidetoshi Onodera. On-chip global signaling by wave pipelining. In *Proc. of the IEEE 13th Topical Meeting on Electrical Performance of Electronic Packaging*, pages 311–314, October 2004.

[23] Masanori Hashimoto, Akira Tsuchiya, Akinori Shinmyo, and Hidetoshi Onodera. Performance prediction of on-chip high-throughput global signaling. In *Proc. of the IEEE 14th Topical Meeting on Electrical Performance of Electronic Packaging*, pages 79–82, October 2005.

[24] Oliver Heaviside. Electromagnetic induction and its propagation. *XL. The Electrician XIX*, pages 79–81, 1887.

[25] M. A. Hillebrand, T. Schurger, and P.-M. Seidel. How to half wire lengths in the layout of cyclic shifter. In *Proc. of the Intl. Conf. on VLSI Design*, pages 339–344, 2001.

[26] Ron Ho, Ken Mai, and Mark Horowitz. Efficient on-chip global interconnects. In *ISVLSI*, pages 271–274, June 2003.

[27] Z. Huang and M. Ercegovac. Effect of wire delay on the design of prefix adders in deep submicron technology. In *Proc. of the Asilomar Conf. on Signals Systems and Computers*, volume 2, pages 1713–1717, 2000.

[28] ILOG Inc. *ILOG CPLEX 9.1 Reference Manual*, April 2005.

[29] ILOG Inc. *ILOG CPLEX Callable Library 9.1 Reference Manual*, April 2005.

[30] Hiroyuki Ito, Junpei Inoue, Shinichiro Gomi, Hideyuki Sugita, Kenichi Okada, and Kazuya Masu. On-chip transmission line for long global interconnects. In *IEDM*, pages 677–680, December 2004.

[31] Anup P. Jose, George Patounakis, and K. L. Shepard. Near speed-of-light on-chip interconnects using pulsed current-mode signalling. In *Proc. of the IEEE Symposium on VLSI Circuits*, pages 108–111, June 2005.

[32] Jinsook Kim, Weiping Ni, and Edwin C. Kan. A novel global interconnect method using nonlinear transmission lines. In *CICC*, pages 617–620, September 2005.

[33] Simon Knowles. A family of adders. In *Proceedings of the 14th IEEE Symposium on Computer Arithmetic*, pages 30–34, Los Alamitos, CA, April 1999. IEEE Computer Society Press.

[34] P. Kogge and H. Stone. A parallel algorithm for the efficient solution of a general class of recurrence relations. *IEEE Transactions on Computers*, C-22(8):786–793, August 1973.

[35] R. E. Ladner and M. J. Fischer. Parallel prefix computation. *Journal of the Association for Computing Machinery*, 27(4):831–838, October 1980.

[36] S. Lakshmivarahan and S. K. Dhall. *Parallel Computing: Using the Prefix Problem.* Oxford University Press, New York, 1994.

[37] S. Lakshmivarahan, C. M. Yang, and S. K. Dhall. On a new class of optimal parallel prefix circuits with (size+depth)=$2n-2$ and $\lceil \log n \rceil \leq depth \leq (2\lceil \log n \rceil - 3)$. In *Proc. of the 1987 Int. Conf. on Parallel Processing*, pages 58–65, August 1987.

[38] Y. C. Lin. Optimal parallel prefix circuits with fan-out 2 and corresponding parallel algorithms. *Neural, Parallel and Scientific Computations*, 7(1):33–42, March 1999.

[39] Y. C. Lin and J. N. Chen. Z4: A new depth-size optimal parallel prefix circuit with small depth. *Neural, Parallel and Scientific Computations*, 11(3):221–235, September 2003.

[40] Y. C. Lin and J. W. Hsiao. A new approach to constructing optimal parallel prefix circuits with small depth. *Journal of Parallel and Distributed Computing*, 64(1):97–107, January 2004.

[41] Y. C. Lin, Y. H. Hsu, and C. K. Liu. Constructing h4, a fast depth-size optimal parallel prefix circuit. *Journal of Supercomuting*, 24(3):279–304, March 2003.

[42] Y. C. Lin and C. C. Shih. A new class of depth-size optimal parallel prefix circuits. *Journal of Supercomputing*, 14(1):39–52, July 1999.

[43] Nir Magen, Avinoam Kolodny, Uri Weiser, and Nachum Shhamir. Interconnect-power dissipation in a microproessor. In *Proc. of the 2004 International Workship on System Level Interconnect Prediction*, pages 7–13, 2004.

[44] E. D. Perfecto, A. P. Giri, R. R. Shields, H. P. Longworth, J. R. Pennacchia, and M. P. Jeannerte. Thin-film multichip module packages for high-end ibm servers. *IBM J. RES. DEVELOP.*, 42(5):597–605, September 1998.

[45] David M. Pozar. *Microwave Engineering*. Wiley, third edition, February 2004.

[46] J. M. Rabaey, A. Chandrakasan, and B. Nikolic. *Digital Integrated Circuits - A Design Perspective*. Prentice-Hall Publisher, second edition, December 2002.

[47] Stefan Rusu. Trends and challenges in vlsi technology scaling towards 100nm (presentation slides). In *Proc. of the 27th European Solid-State Circuits Conference (ESSCIRC)*, pages 194–196, September 2001.

[48] P.-M. Seidel and K. Fazel. Two-dimensional folding strategies for improved layouts of cyclic shifters. In *ISVLSI*, pages 277–278, 2004.

[49] J. Sklansky. Conditional sum addition logic. *IRE Trans. Electron. Comput*, EC-9(6):226–231, June 1960.

[50] M. Snir. Depth-size trade-offs for parallel prefix computation. *Journal of Algorithm*, 7(2):185–201, June 1986.

[51] Binay Sugla and David A. Carlson. Extreme area-time tradeoffs in vlsi. *IEEE Transactions on Computers*, 39(2):251–257, February 1990.

[52] I. Sutherland, B. Sproull, and D. Harris. *Logical Effort: Designing Fast CMOS Circuits*. Morgan Kaufmann, February 1999.

[53] G. M. Tharakan and S. M. Kang. A new design of a fast barrel switch network. *IEEE Journal of Solid-State Circuits*, 27(2):217–221, 1992.

[54] Dane C. Thompson, Olivier Tantot, Hubert Jallageas, George E. Ponchak, Manos M. Tentzeris, and John Papapolymerou. Characterization of liquid crystal polymer (lcp) material and transmission lines on lcp substrates from 30 to 110 ghz. *IEEE Trans. on Microwave Theory and Techniques*, 52(4):1343–1352, April 2004.

[55] Akira Tsuchiya, Yuuya Gotoh, Masanori Hashimoto, and Hidetoshi Onodera. Performance limitation of on-chip global interconnects for high-speed signaling. In *CICC*, pages 489–492, 2004.

[56] Akira Tsuchiya, Masanori Hashimoto, and Hidetoshi Onodera. Design guideline for resistive termination of on-chip high-speed interconnects. In *CICC*, pages 613–616, 2005.

[57] N. H.E. Weste and D. Harris. *CMOS VLSI Design : A Circuits and Systems Perspective*. Addison-Wesley Publisher, third edition, May 2004.

[58] S.-J. Yih, M. Cheng, and W.-S. Feng. Multilevel barrel shifter for cordic design. *Electronics Letters*, 32(13):1178–1179, June 1996.

[59] Ling Zhang, Hongyu Chen, and Chung-Kuan Cheng. On-chip interconnect analysis and evaluation of delay power and bandwidth metrics under different design goals. In *Unpublished manuscript*.

[60] R. Zimmermann. Java applet for adder synthesis. In *http://www.iis.ee.ethz.ch/ zimmi/ applets/ prefix.html*.

[61] R. Zimmermann. Non-heuristic optimization and synthesis of parallel-prefix adders. In *Int. Workshop on Logic and Architecture Synthesis*, pages 123–132, December 1996.