A Digital Design Methodology for Optical Computing

Miles Murdocca

The MIT Press
Cambridge, Massachusetts
London, England

This book was set in Times Roman.
Printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Murdocca, Miles.
A digital design methodology for optical computing / Miles Murdocca.
p. cm.
Includes bibliographical references.
ISBN 0-262-13251-6
1. Optical data processing. 2. Computers, Optical. 3. Digital integrated circuits. I. Title.
TA1630.M87 1990 90-5480
621.39'1—dc20 CIP

A Digital Design Methodology for Optical Computing

Preface

Nearly half of the content of this book comes from my Rutgers doctoral dissertation. My original intent was that this book should be an expanded Ph.D. thesis in the form of a monograph that promotes the message: "Use regular free-space interconnects for optical computing." My aim was to help steer the field of optical computing in this direction by providing convincing arguments that complicated, exotic means for interconnecting optical logic gates are unwarranted. That's the purpose of a monograph – a narrow, deep expository on a single subject. I'm grateful to the reviewers who convinced me to make it into the broader book you are now reading, because the interested reader hopes to gain more than a single insight from a book dealing with such a new technology.

The intended audience for this book includes computer engineers, computer scientists, physicists, and other scientists who have an interest in digital optical computing. This book is written from the perspective of a computer designer, with an emphasis on a design methodology that simplifies the optical hardware without adding complexity to the design process. I have done my best to include other approaches to general purpose digital optical design that cover the more celebrated areas of research such as hybrid optical/electronic approaches and guided wave interconnects. Unlike the Feitelson survey *Optical Computing* (The MIT Press, 1988) which gives an overview of the field, I have presented the material in this text with distinct biases toward regularity in design and simplicity in methods. I argue against methods that introduce complexity in the optical hardware or add difficulty to the design process. My position is that high performance computers *in any technology* must exhibit regularity in structure and must submit to simple design techniques. This must be the case if our goal is to manage the enormous complexity of next-generation computers.

Inspiration is not born in a vacuum. The reader should not pore over the text and gaze at the figures in amazement that a single mind could have dreamed up such wonders. It didn't. Two years after I joined AT&T Bell Laboratories, the world's largest corporation divested itself and entered an age of competition in the long distance telecommunications market. The climate within AT&T changed from benevolent monopoly to market-driven competitor. That might have been the case for the whole company, but efforts were made to protect the research area from the front lines. Optical computing took the limelight as an example that Bell Labs research still thrived, and enjoyed considerable support. I've been lucky to have been involved with optical computing at this time, enjoying the freedom of exploring my own interests in optical computing while enjoying interaction with the development side of the business. The interaction between research and development at Bell Labs fostered a sense of direction and purpose that I feel has led the frontier in optical computing technology. Part of the culture of research at Bell Labs is the maintenance of an academic atmosphere. Some people joke that their real employer is The Institute of Electrical and Electronics Engineers (IEEE), a scientific organization that publishes professional journals and sponsors technical meetings. A result of this thinking is a visible presence at professional meetings, through which I am grateful to have met many of the inventors at other institutions whose works are cited in this text.

In the last few years, the Air Force Office of Scientific Research (AFOSR) funded a startup project at Rutgers exploring the application of digital optical computing to large structured problems like content addressable memory (CAM). My involvement with the Rutgers group allowed me to make greater progress toward the ideas presented in this book than if my time was divided among many areas of research. The Rutgers group identified problems with electronic systems that led to the optical content addressable memory (CAM) and Connection Machine case studies described in Chapter 5. The Rutgers AFOSR sponsored project maintained ties with the Bell Labs groups which allowed large architectural problems to be studied at Rutgers without taking on the additional burden of building the optical hardware.

The text is organized into six chapters that should be read in order. The first chapter provides motivations for using optics in digital computing, and gives a brief history of the field. The breadth of coverage is incomplete with respect to the field as a whole and is biased toward the methodology supported in this text, and I refer the reader to Feitelson's survey for a more complete background. I don't think there is enough breadth to teach an entire course from this text, although it would make a good supplemental text for a VLSI design course, a seminar on advanced computer architectures, or a broader course on optical computing.

The status of optical hardware discussed in Chapter 2 is a moving target because advances are made on a daily basis. A few trends noted in the chapter are not likely to grow obsolete so quickly however, such as the fabrication of regular arrays of identical logic gates and the need for small, easily manufacturable optical systems. Issues are treated in a topical manner because I don't want to lessen the emphasis on regularity in design and simplicity in methods by providing distractions on device specifics that may have little relevance in a few years.

Chapter 3 describes alternative approaches to designing a general purpose digital optical computer and introduces the reader to the main approach described in Chapter 4, which is the methodology supported in this book. The approaches discussed are representative of current work going on in the field but are by no means complete. I don't want to confound or distract the reader by offering too much information, rather, my intent is to expose the reader to enough material to appreciate the methodology presented in Chapter 4.

Chapter 4 is the monograph that was all this book was originally intended to be, the presentation of a methodology of designing digital circuits for an optical computer. This is the backbone of the text, that shows how to design regularity into digital optical circuits with simple methods, that yield high performance without introducing significant complexity into the target machine. One of the discussions offers a profound capability that regular free-space interconnects afford us, which is the ability to completely reconfigure the gate-to-gate interconnects of a computer on every time step, at little additional hardware cost. How do we exploit the potential of such a machine? This chapter stands apart from the rest of the text in describing a complete methodology for designing digital optical circuits, and can serve as a good complement to a conventional digital design course dealing with advanced architectures.

Chapter 5 describes three case studies designed with the approach presented in Chapter 4. A parallel sorting network design makes a particularly strong case for regularity. The Rutgers CAM and optical Connection Machine case studies support this methodology by overcoming the pinout and other input/output (I/O) constraints of conventional Very Large Scale Integration (VLSI).

Chapter 6 touches on some philosophical issues of computing, such as why it is desirable to create more powerful computers. A brief summary reinforces the ideas discussed in earlier chapters.

I have found that clarity is the most gifted tool a technical writer may possess. Acronyms are only used in this book when they have been clearly defined in other published literature; I have created no new acronyms in this text, and I have defined all old acronyms on their first use and also on subsequent uses if a sizeable lapse occurs between uses. In the interest of promoting current writing style, I have referred to all persons with genderless words except where the gender of a person is commonly known and pertinent to the discussion. I have avoided using words that are overly colorful or unusual so that the reader can move quickly through the text without stumbling over words. I hope that my efforts at writing simply and clearly for the reader's benefit are successful.

Miles Murdocca
AT&T Bell Laboratories
Holmdel, New Jersey
and
Rutgers University
New Brunswick, New Jersey

Acknowledgements

I couldn't have done this work alone, and I gratefully acknowledge the support of many people and institutions for their influence in my thinking and in the preparation of this book. Those supporters, listed alphabetically, are: The Air Force Office of Scientific Research, Ravi Athale, AT&T Bell Laboratories, Karl-Heinz Brenner, Ike Chuang, Tom Cloonan, Nick Craft, Alex Dickinson, Maralene Downs, Chris Gabriel, Lee Giles, John Storrs Hall, Alan Huang, Jürgen Jahns, Scott Knauer, Vijay Kumar, Yong Lee, Tony Lentine, Saul Levy, Haw-Minn Lu, my parents Dolores and Nicholas Murdocca, Bob Prior and The MIT Press, Irv Rabinowitz, the Rutgers Computer Science Department, Tod Sizer, Lou Steinberg, Charlie Stirk, Norbert Streibl, Binay Sugla, Tony Terrano, Stu Tewksbury, Sue Walker, Larry West, all of the people who acted as readers of early drafts, and Jack Jewell, Mike Prise, Rick McCormick, and Norm Whitaker for not leaving me stranded in the Grand Canyon, and to the unnamed hiker who pointed us to the right trail when Norm and I were blustering up the wrong one a few hours before my talk in Nevada.

There are surely other people and institutions who have contributed to this book, either directly or indirectly, whose names I have inadvertently omitted. To those people and institutions I offer my tacit appreciation and apologize for having omitted explicit recognition here.

I think I had it easier than most, with the support of my employer, my school, my family, and especially my wife Ellen, who has watched me sit in front of this silly computer every day and night in amazement that anyone would do it by choice. For your patience and encouragement, this is for you Ellen.

Contents

Chapter 1
Introduction

There are rewards in building regularity into complex systems that are appreciated only after time has passed and the systems approach maturity. For example, consider the difference in street layouts of Boston and New York. Boston started out as a relatively small town that grew in incremental steps to become a metropolitan center of the northeastern United States, with a seemingly spaghetti-like system of roads that confounds the passing automobile driver. People familiar with Boston sometimes say: "If you don't know where you are driving in Boston, then don't drive in Boston." New York City on the other hand, with a grid-like road system over most of Manhattan is more easily navigated by the common driver, and local eddies are more easily smoothed with traffic light synchronization that adjusts for changing volume. Regularities are most appreciated when looking back at systems that grow to great sizes without choking from their masses.

Optics can improve regularity in computer design but it may not be obvious at first why a light based technology should have an advantage over an electronics based technology. After all, light and electricity both travel at the speed of light so we can't expect an optical computer to run any faster than an electronic computer, unless there are some limiting problems inherent in one technology that are not inherent in the other. That is the case as will be discussed in the next section, and the key to these ideas is *regularity*.

1.1 Why optical computing?

We should consider that there is a large disparity between the speed of the fastest electronic switching components and the speed of the fastest digital electronic computers. Figure 1.1 illustrates the problem, in showing that transistors exist that can switch in 5ps while the fastest computers run at clock rates on the order of a few nanoseconds. What are the causes of this slowdown for increasing hardware complexity, and how can we get the system speeds closer to the device speeds?

Transistor	5ps
Ring oscillator	30ps
Logic gate	120ps
Chip	1ns
System	5ns

Figure 1.1: *The difference in speed between one of the fastest transistors and one of the fastest computers is a factor of ~1000. (Table provided by the courtesy of Alan Huang, AT&T Bell Laboratories.)*

Limitations of electronics, which is the contending technology for interconnection at this time, include [66]:

- Electromagnetic interference at high speed
- Distorted edge transitions
- Complexity of metal connections
- Drive requirements for pins
- Large peak power levels
- Impedance matching effects

Electromagnetic interference arises because the inductances of two current carrying wires are coupled. Sharp edge transitions must be maintained for proper switching but higher frequencies are attenuated greater than lower frequencies, resulting in sloppy edges at high speeds. The complexity of metal connections on chips, on circuit boards, and between system components affects connection topology and introduces complex fields and unequal path lengths. This translates to signal skews that are overcome by slowing the system clock rate so that signals overlap sufficiently in time. Large peak power levels are needed to overcome residual capacitances, and impedance matching effects at connections require high currents which result in lower system speeds.

Guided wave methods suffer from none of the disadvantages of electronics mentioned here so they are gaining acceptance in large systems such as AT&T's

5ESS [2] central office switch. However, topological complexity introduced by bending tolerances for fibers, volume requirements, and skew effects pose serious enough limitations to preclude the use of guided wave technology for complex gate-level interconnects, unless radical improvements to the technology are made.

A technology based on optics offers solutions to these problems if we can exploit the advantages of optics without introducing new complexity or new limitations that render the use of optics ineffective. Advantages of using free-space optics for interconnection include [66]:

- High connectivity through imaging
- No physical contact for interconnects
- Non-interference of signals
- High spatial and temporal bandwidth
- No feedback to the power source
- Inherently low signal dispersion

High connectivity can be achieved by imaging a large array of light beams onto an array of optical logic devices. There is no need for physical interconnects (unless fibers or waveguides are used) so that connection complexity is simplified and drive requirements are reduced. Optical signals do not interact in free space, which means that beams can pass right through each other without interference. This allows for a high density of signals in a small volume. High bandwidth is achieved in space because of the non-interference property of optical signals, and high bandwidth is achieved in time because propagating wavefronts do not interact. There is no feedback to the power source as in electronics, so that there are no data dependent loads. Finally, inherently low signal dispersion means that the shape of a pulse as it leaves it source is virtually unchanged when it reaches its destination.

From the viewpoint of a computer designer, probably the single most significant advantage of optics over electronics is *communication*. Optical logic gates can be oriented normal to the surface of an optical chip so that light beams travel in parallel between arrays of optical logic devices rather than through pins at the edges of chips as in electronic integrated circuits (ICs) as illustrated in Figure 1.2.

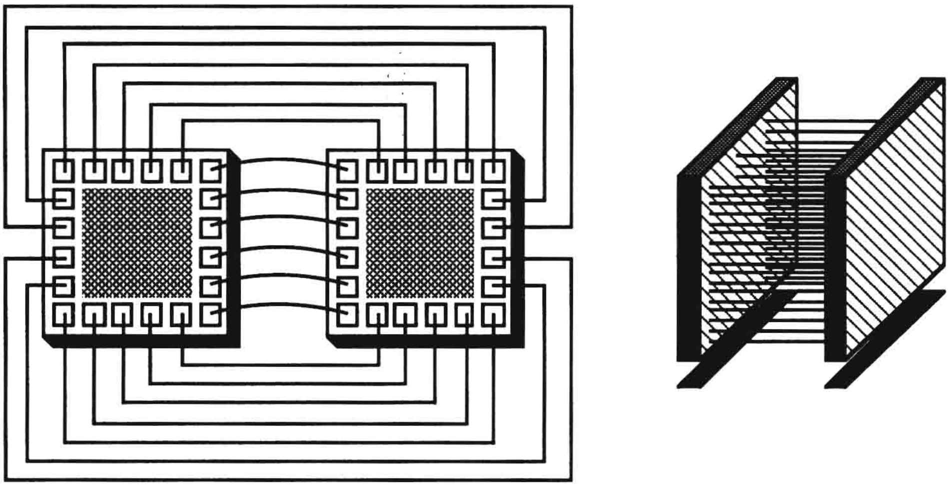

Figure 1.2: *Edge connections between electronic chips (left) and parallel connections between optical chips (right).*

Free-space storage, lack of signal skew, and dense communication are just a few of the properties of free-space optics that can be exploited in the implementation of a digital computer. A wired approach or a guided wave approach for interconnection in an advanced architecture can quickly dominate the cost of the entire machine and can pose the most serious speed limitation. Irregular free-space interconnects reduce spatial bandwidth (the amount of information that can be passed through a lens) in holographic implementations because a minimum spacing must be maintained in the input plane to avoid crosstalk in the output plane. The use of regular free-space interconnects allows the entire chip area to be devoted to active switching rather than to communication. Device failure is not nearly as critical because circuits can be redesigned after the positions of faulty components are known. Thus a free-space approach to interconnection offers significant advantages over other means of interconnection.

There are limits imposed by the methods supported in this book. Strictly speaking, electronics is not worse than optics for short distance communication, on the order of a centimeter. Photons do not interact with other photons directly, and typically involve electrons even in an "all-optical" approach, and this conversion between electrons and photons consumes time and energy. Electronics is a mature, inexpensive technology that allows for a high density of switching components. Photonics on the other hand is less mature and requires tight imaging tolerances and constant power consumption for modulator-based optical logic gates. The position is argued here that an all-optical computer will result in a simpler design than a hybrid optical/electronic approach, and that this simplicity is more significant for managing complexity in digital optical computing than an inherent superiority at the quantum level. Still, the reader should be mindful of the advantages of electronic digital computing, and that nearly five decades of electronic technology should not be completely cast aside.

Another limitation of the optical approach described in the rest of this book is that optical logic gates are spaced a few microns apart on optical chips but require several hundreds of microns, and possibly several centimeters of interaction distance for lenses, gratings, and other imaging components. The use of microoptic techniques (monolithically fabricated lenses) can be applied to reduce or eliminate this problem but for initial systems an infant technology like microoptics should not be relied upon as the basis for connections. This means that switching speeds on the order of 100ps or faster cannot be exploited in a tight feedback loop because the interaction distance forces a minimum separation between arrays of devices. Fast switching speeds can still be accommodated because free space can provide a delay memory allowing propagating wavefronts of information to be maintained between arrays. In a pipelined system, this minimum separation is not significant when the pipeline remains filled. Applications that can take advantage of the bandwidth and parallelism of free-space propagation without suffering from the minimum device separation along the propagation axis include signal processing, digital switching (as in a telephone central office switch), and matrix-vector multiplication. If microoptic techniques mature to accommodate the free-space interconnection needs of the systems described here, then the applications that digital optical systems are suited for can broaden to include nearly every application that is limited by speed or connectivity.

In consideration of these arguments we might conclude that optics is a better technology for digital computing, but we need to support this claim by showing how to apply this technology to an entire system, and measure the cost and performance of the system that results. The chapters that follow give more specific evidence that regularity in structure and simplicity in design methods are suitable goals for digital optical computing by showing how such a machine can be constructed and what new advantages can be exploited.

1.2 A brief history of optical computing

Digital computing with the use of optical components was considered at least as early as the 1940's by von Neumann [118]. If lasers were available at the time, the first digital computers may well have used optics. In the early 1960's and throughout the 1970's and 1980's, optics technology was employed for computing Fourier transforms of military images in matched filtering operations. Synthetic aperture radar (SAR) signal processing [19, 34, 40] matches images in stored photographic form with input images, at a very high rate. Spectrum analysis is performed with acousto-optic signal processing [102]. Both of these applications are performed optically when bandwidth needs exceed electronic capability.

Studies at IBM [69] showed that digital optics would not surpass digital electronics with technology available in the foreseeable future, and arguments in favor of optical computing were hard to support except in a few niche applications like matched filtering. There were renewed interests in the late 1970's as advances were made in optical transmission and optically nonlinear materials. Limits of electronic digital circuits became more apparent as the need for communication bandwidth became more severe [48, 49, 50], and attention returned to optics. A current survey on the field of optical computing can be found in Feitelson's *Optical Computing* [25]. Other works that cover broad aspects of the field can be found in References [3, 6, 35, 51, 106].

There is a great volume of literature on various aspects of optical computing such as optical bistability, optically nonlinear materials, architectures, number systems, methodologies, *etc*. There is far too much work to categorize it all here, so the focus is on projects that have had the greatest influence on the design of general purpose digital optical computers. Except for neural networks, discussion on analog optical computing is intentionally omitted. Error accumulation and accuracy limit the extent analog computing can enjoy in both electronic and optical technologies. There will always be special applications for analog optics such as signal processing, matrix-vector multiplication, and neural networks so the significance of analog optics is not meant to be lessened here, but the role analog optics is likely to play must be placed in perspective with digital optics.

Huang's symbolic substitution [49], which is a parallel method of binary pattern replacement has had a significant influence on the field of optical computing.