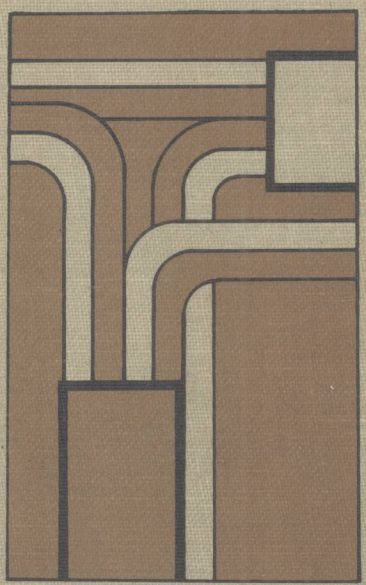
A Graphical Engineering Aid for VLSI Systems

Paul J. Drongowski



UMI Research Press
Computer Science:
Computer Architecture and Design

A Graphical Engineering Aid for VLSI Systems

by
Paul J. Drongowski
A.R. Jennings Distinguished Assistant Professor
Department of Computer Engineering and Science
Case Western Reserve University
Cleveland, Ohio

UMI RESEARCH PRESS
Ann Arbor, Michigan

Produced and distributed by
UMI Research Press
an imprint of
University Microfilms International
A Xerox Information Resources Company
Ann Arbor, Michigan 48106

Library of Congress Cataloging in Publication Data

Drongowski, Paul J., 1952-
A graphical engineering aid for VLSI systems.

(Computer science. Computer architecture and design ; no. 4)
Revision of thesis (Ph.D.)—University of Utah, 1982.
Bibliography: p.
Includes index.
1. Integrated circuits—Very large scale integration—Design and construction. 2. Computer-aided design. I. Title. II. Title: Graphical engineering aid for V.L.S.I. systems. III. Series.
TK7874.D76 1985 621.3819'5835 85-1041
ISBN 0-8357-1656-2 (alk. paper)

A Graphical Engineering Aid for VLSI Systems

Computer Science: Computer Architecture and Design, No. 4

Harold S. Stone, Series Editor

IBM Corporation
T.J. Watson Research Center
Yorktown, New York

Other Titles in This Series

No. 1	*Automated Microcode Synthesis*	Robert A. Mueller
No. 2	*A Parallel Pipeline Computer Architecture for Speech Processing*	Vassilios John Georgiou
No. 3	*Algorithms for Some Design Automation Problems*	James P. Cohoon

For my Mom and Dad,
Helen and Sigmont Drongowski

Preface

The effect of very large scale integration (VLSI) on consumer, industrial, and military electronics is pervasive. The reduced size and increased power of VLSI computing engines have encouraged the development of at least one new industry (personal computers) and revolutionized others, from industrial process control to sewing machines and microwave ovens. VLSI systems are particularly well suited for high volume applications where increased functionality at reduced cost provides a key competitive advantage.

For a product to succeed in the marketplace, it must satisfy a particular demand when the demand arises at a reasonable price. A product that is late, expensive, or inappropriate simply will fail. Computer aided design (CAD) tools can help an engineering team to exploit VLSI technology more effectively by:

—Decreasing development time (getting the product to market sooner),

—Reducing design complexity (producing a more reliable and useable product), and

—Decreasing nonrecurring engineering costs (lowering the overall price of the product).

Contemporary engineering workstations and CAD tools are beginning to fill these needs through schematic capture, simulation, layout, and analysis programs.

This book describes an engineering aid for the design of VLSI systems which emphasizes a functional (or linguistic) approach to the description, analysis, and synthesis of VLSI circuits. Tool builders and researchers alike should find several aspects of this work novel and useful. The specification and use of a graphical hardware description language (HDL) based upon representational datatypes is a significant departure from textual (or symbolic) HDLs. Algorithms that map a design to its implementation and that calculate

the expected execution time of a system are given in the appendices and should provide a basis for future work or application. For students who are interested in digital systems CAD, this book will present a comprehensive view of an experimental CAD system. The interrelationships between notation, design databases, and tools are explored. More recent results and work are reported in the epilogue.

Special thanks go to Al Davis (Fairchild Flair), Chuck Rose (Case Western Reserve and Endot Inc.), and Sam Fuller (Digital Equipment Corp.) for their long-term (and sometimes long-distance) guidance and support. Al's advice, encouragement, and technical expertise were invaluable. Since Chuck and Sam helped me to form many of my early impressions of CAD and computer architecture, they, too, have been quite influential.

I would like to thank the following friends and colleagues for their constructive comments and insights: Gary Lindstrom, Bob Keller, P. Subrahmanyam, Kent Smith, Elliott Organick, Lino Ferretti, Uri Weiser, Dave Matty, Larry Seppi, and Ed Snow.

I will always be grateful for the cheering section in Cleveland: my parents, Lydia, Jack, Alicia, and Carl Boldt. Finally, I would like to thank Francie Hunt, whose unfailing patience and encouragement got me through it all then—and now.

Contents

1

Introduction

High density integrated circuitry has offered an unprecedented opportunity to extend the capabilities of modern computing tools. The design of systems based on the technology of very large scale integration (VLSI) also poses some unprecedented challenges [Sut77].

Historical evidence indicates that chip complexity has grown exponentially since about 1971. Foreseeable improvements in fabrication technology guarantee a continuing increase in circuit densities at this phenomenal rate. A recent study conducted at Intel Corporation shows that designer productivity has declined for the last several years while complexity has increased [Lat79]. In 1982, it took an estimated 60 person-years just to lay out the design of a microcomputer chip. Clearly, our computer-based tools must be drastically improved to gain increased leverage on the integrated circuit design problem.

This book presents and evaluates a method for the design of nMOS VLSI systems. This method, called d, is a discipline supporting top-down and bottom-up hierarchical design. The system to be designed is described in a graphical notation, d-n, which separates the specification of system behavior and structure from its physical design, moving VLSI design from a preoccupation with geometry toward the realm of programming. Through floor plan diagrams and mechanical delay analysis, the designer can stay in touch with the real physical concerns of space utilization and execution speed. Some experimental software tools were constructed to demonstrate the feasibility of a real, complete computer aided design (CAD) system using this method.

N-channel MOS was selected as the subject technology in this research for several reasons:

1. nMOS has sufficient component density to support the design of very complex systems, the topic of this study.

2. It is a popular technology and will be used extensively through 1985. Results from this work, therefore, would have fairly wide and timely impact on the IC design community.

3. Expertise on nMOS circuit design and device behavior is readily available.

4. The local computing environment includes some design tools for nMOS circuits, making software experiments a bit easier to conduct.

5. Other scientists are proposing schemes for the design of nMOS systems. By using the same technology, objective comparisons between methods and results can be made.

In this book, the term "VLSI" will mean "nMOS VLSI."

Chapter 2 examines some of the major issues affecting the design of VLSI systems. It presents the conventional approach to VLSI design and looks at contemporary efforts to improve our tools and techniques.

In chapter 3, the goals and guiding principles of the d methodology are outlined. The notation for d-based VLSI designs, d-n, is presented in chapter 4. A detailed, reference-style presentation of d-n is made in appendix A. The tools to support the language and the methodology are proposed in chapter 5.

The design of a cache memory system in d-n is presented in chapter 6. Chapter 7 discusses the experimental tools which were developed as part of this project and makes some suggestions for the future implementation of d. Because a tool both supports and constrains its user, the relationship between d and the VLSI designer is an important one. This subject is explored in chapter 8.

Chapter 9 states the critical conclusions which may be drawn from this work. It also presents myriad possibilities for future research and extensions.

2

VLSI Design

Background

Contemporary disciplines for system design encourage the use of both top-down and bottom-up methods. In top-down design, overall system function is successively decomposed into smaller, more manageable units. Bottom-up design is a process of synthesis in which building blocks are combined to form still larger building blocks. If the resulting system is structured such that lower level elements are not defined in terms of higher level elements, then the system is said to be hierarchical.

A design team usually begins work with an initial specification that describes the intended behavior of the system. Subsequent design activities may be separated into three domains. Behavioral design decomposes and refines the initial specification into a hierarchy of subproblems. These subproblems and their solutions are assigned to individual modules or subsystems. Structural design involves the interconnection of these modules to form buildings which may be interconnected at still higher levels in the hierarchy. Often behavioral and structural design are performed in concert. Eventually the behavioral and structural designs must be implemented in a particular technology. The actual representation of the behavior and structure of the system in this technology is the object of physical design [Van79].

Often the initial specification is found to be ambiguous or incomplete, resulting in some redesign. Changes in the initial specification will directly affect the behavioral design and indirectly, the structural design. For example, the system structural and behavioral specification may be inadequate to support some new operational requirement which was added to the specification by management after design had already started. In other cases, redesign may result from unanticipated interactions between domains. For example, structural design normally terminates when the engineers feel that each of the building blocks at the bottom of the hierarchy is "sufficiently primitive" to be directly implemented in the chosen technology. If this is not the

case, structural design must be continued. This is an example of interaction between the structural and physical domains.

General considerations aside, the design of a VLSI-based digital system brings along its own set of problems, largely due to the nature of the implementation technology itself. (The remainder of this section is an overview of the nMOS process and its physical implications for digital system design.) The ultimate product of the VLSI design process is a set of masks which are used to pattern successive layers of circuit materials on a silicon substrate. The physical process of forming the patterns and layers is called fabrication. The choice of materials, layers, and patterns depends upon the kinds of circuit devices to be employed and the fabrication process to be used to manufacture the chips. For example, the popular n-channel MOS (nMOS) technology consists of four layers of materials (n-channel silicon, polysilicon, aluminum metal, and p-bulk) separated by insulating layers of silicon dioxide. Any of these conducting paths may cross and openings may be formed in the insulating layers to jump the circuit between layers. Wherever polysilicon (or just "poly" for short) crosses an n-channel region, an nMOS transistor is formed. These transistors are the basic active circuit elements and may be interconnected by two and one-half layers of wiring.[1] The masks produced by the VLSI design process determine the physical position and size of the transistors and wires on the chip—the so-called "chip geometry."

Since the transistors and wires are composed of semiconducting materials their electrical behavior follows the natural principles of device physics. Polysilicon, n-channel, and metal wires act as capacitors and resistors. Hence, circuits which drive these wires are charging minute capacitors and are incurring signal delays. The resistivity of the wire path determines the amount of signal drop across the wire, affecting noise immunity and logic thresholds. Both the resistance and capacitance of a wire are proportional to its length and depend upon the wire material.

Behavior of the nMOS transistor is determined by the physical dimensions of the crossing n-channel and poly paths. The amount of current which a transistor can source or sink depends upon the length and width of the transistor gate (the intersection of the poly and n-channel region). The capacitance of the nMOS transistor gate is a very significant factor in circuit delays and is also determined by the transistor geometry.

On-chip capacitances rarely exceed 1 picofarad and, therefore, make modest current demands on most of the circuitry. When a signal must be driven off the chip, however, the capacitive load increases to approximately 100 picofarads. Therefore, on-chip signals can be switched 100 times as fast as off-chip signals because delay is roughly proportional to load capacitance. In addition, much larger transistors and multistage amplifiers must be used to drive the signal into the bigger load. Hence, there is a space (smaller driver size) and time (shorter

signal delay) advantage to be gained by packing as much functionality as possible onto a single chip [Dav79].

Signal delays affect system timing. Systems that are fully synchronous depend upon a global clock to sequence control events. In order to drive the clock signal into the relatively long control wires that pervade every region of the chip without excessive delay, a large clock generator and buffer must be used. In a highly complex digital system such as a VLSI circuit, it is hard to guarantee that the clock period will be sufficiently long to prevent hazards due to signal skew. There are simply too many assumptions about timing behavior to be completely verified.

A more practical approach is the self-timed system concept of Seitz [Sei79, Sei80]. In this kind of organization, the separate subsystems are responsible for keeping their own notion of time (e.g., a local clock). Assumptions about synchronous behavior, therefore, are localized at the subsytem level and communication between subsystems is performed asynchronously. This eliminates errors due to signal skew.

Transistors and wires cannot be placed on the chip haphazardly. Due to certain limitations in the fabrication process, physical device dimensions are restricted by minumum size and spacing constraints called design rules. Sizes and separations are specified as multiples of the minimum feature size. The feature size is largely determined by the accuracy of the etching, implantation, and alignment steps in the fabrication process. Some of the design rules have their basis in device electronics. For example, an unrelated (unconnected) poly wire must not overlap a parallel n-channel diffusion wire or an unwanted transistor will be created. It is not an easy task to guarantee adherence to the design rules when a system has 500,000 rectangles such as a chip in the Intel iAPX 432 family [Lat79].

Not all the chips on a particular wafer will operate correctly after fabrication. The percentage of working chips is commonly referred to as the yield. Most chip failures are due to defects in the silicon wafer, mask flaws, or over (under) etching the layers. One model for estimating chip failure assumes that flaws are uniformly and randomly distributed across the wafer surface. By minimizing chip area, yield can be improved for a given design [Mea80]. Additional failures can be expected after packaging as a result of bonding errors between the chip and the wire carrying the external signal to the package pin. By keeping the number of pins low, the overall yield (after packaging) can be increased. Clearly, chip area and yield must be balanced against functionality and the reduction of off-chip communication.

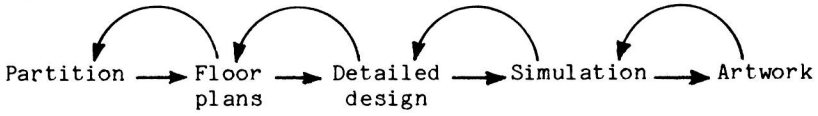
From the preceding discussion, the physical nature of VLSI systems can be seen to be a very heavy influence on the behavioral and structural design of these systems. During design, the initial specification must be taken from a very high level description of system behavior to the actual geometry of transistors

and wires. Wire lengths and transistor sizes directly affect the execution speed of the system, the amount of functionality which can be squeezed onto the chip (density), and even the yield which can be expected after fabrication and packaging. Structural design must accommodate not only the placement of modules, but also the routing of wires between modules. Wire length must be kept to a reasonable distance to prevent signal delays from becoming too long or the size of drive transistors from becoming too large. In addition to system level issues, there is the matter of managing all the details which must "play together" in order to form a working system. Design rules must be followed and transistor sizes chosen appropriately. All interconnections between subsystems must be made correctly. Clearly, VLSI design is an exercise in complexity management—a problem begging for a structured design discipline.

Current Practice

Existing VLSI design methods and tools are heavily oriented toward the generation of artwork (fabrication masks) and the validation of system design through simulation. Figure 1 illustrates the typical industrial design process.

Figure 1. VLSI Design Process as Practiced by Industry



The design team begins with an initial specification and decomposes the problem into subproblems. During this stage of design, the system structure is drawn as a set of block diagrams showing the major subsystems, their interfaces, and the interconnections between the subsystems. Frequently, the total design problem is too big to fit on a single circuit chip. The decomposition process must, therefore, take subsystem sizes and off-chip communication times into account. Decomposition is usually followed by a sizing step in which the physical dimension of each subsystem is estimated according to its function. If a group of subsystems cannot fit on a chip, they must be subdivided further. If subdivision is impossible, the chip design must be abandoned!

Once a suitable structure has been found, the block diagram is translated into a floor plan indicating the relative position and size of each subsystem. The detailed design of the subsystems is then started. First, the major subsystem components such as programmable logic arrays (PLA), read only memories (ROM), random access memories (RAM), registers, and arithmetic units are identified. Preliminary subsystem layouts are drawn leaving room for the interconnections, control logic, and provisions for circuit testing. The

preliminary subsystem layouts are compared with the floor plan. If the initial size estimates are wrong, and they usually are, the floor plan must be rearranged to accommodate the changes [SmiPC].

Computer graphics has been a valuable tool for planning and detailed design. Revisions can be made rapidly and the designer quickly discovers what will and will not work. Many of these systems incorporate automatic design rule checking (DRC) to catch "syntax" errors as early as possible.

After a subsystem has been fully detailed, its operation can be simulated. Due to the amount of computing resources required, electrical simulation cannot be used to analyze circuits with more than 20 to 30 transistors. The design must be translated to a gate or register transfer (RT) level description to be simulated as a complete system since the ratio of real to simulated time is much lower for this kind of simulation.

With gate or RT level simulation, information about system timing is sometimes lost. Using the gate and register delays which have been determined through electrical simulation, the individual circuit delays are combined into an aggregate delay for the subsystem. This is done manually with all the attendant problems of human fallibility. Further, this strategy does not account for delays due to wire capacitance. If either the simulation results or the delay analysis are unsatisfactory, the detailed subsystem design must be changed. Eventually, a satisfactory design is found, the layout is finalized, the design rules are checked one last time, and pattern generation (PG) tapes are produced from the circuit geometry. The PG tapes are then used to make the glass circuit masks for the fabrication process.

Contemporary Approaches

This section describes contemporary research efforts to produce better VLSI design tools.

Caltech

The VLSI research at Caltech is funded by a consortium of industrial organizations and is perhaps the largest of the university efforts.

Much of the Caltech work has been inspired by Carver Mead. In their popular book, Mead and Lynn Conway have proposed a method for VLSI design that combines top-down design with a notation and technique for the design of low to medium complexity, special purpose circuit cells [Mea80]. Using this method, the engineer decomposes the system into modules from which a floor plan for the chip is drawn. The modules consist primarily of highly regular circuit structures such as PLAs, RAMs, ROMs, and special