AN INTRODUCTION TO CAD FOR VLSI

Stephen M. Trimberger

Kluwer Academic Publishers
Boston/Dordrecht/Lancaster

An Introduction to CAD for VLSI

by

Stephen M. Trimberger
VLSI Technology, Inc.

Kluwer Academic Publishers
Boston/Dordrecht/Lancaster

Distributors for North America:
Kluwer Academic Publishers
101 Philip Drive
Assinippi Park
Norwell, Massachusetts 02061, USA

Distributors for the UK and Ireland:
Kluwer Academic Publishers
MTP Press Limited
Falcon House, Queen Square
Lancaster LA1 1RN, UNITED KINGDOM

Distributors for all other countries:
Kluwer Academic Publishers Group
Distribution Centre
Post Office Box 322
3300 AH Dordrecht, THE NETHERLANDS

Consulting Editor: Jonathan Allen

Library of Congress Cataloging-in-Publication Data

Trimberger, Stephen, 1955-
An introduction to CAD for VLSI.

Includes bibliographies and index.
1. Integrated circuits—Very large scale integration—Design and construction—Data processing.
2. Computer-aided design. I. Title.
TK7874.T753 1987 621.381'73'0285 87-2841
ISBN 0-89838-231-9

Printed in the United States of America

to
leon and carol

PREFACE

The last decade has seen an explosion in integrated circuit technology. Improved manufacturing processes have led to ever smaller device sizes. Chips with over a hundred thousand transistors have become common and performance has improved dramatically. Alongside this explosion in manufacturing technology has been a much-less-heralded explosion of design tool capability that has enabled designers to build those large, complex devices. The tools have allowed designers to build chips in less time, reducing the cost and risk. Without the design tools, we would not now be seeing the full benefits of the advanced manufacturing technology.

The Scope of This Book

This book describes the implementation of several tools that are commonly used to design integrated circuits. The tools are the most common ones used for computer aided design and represent the mainstay of design tools in use in the industry today. This book describes proven techniques. It is not a survey of the newest and most exotic design tools, but rather an introduction to the most common, most heavily-used tools. It does not describe how to use computer aided design tools, but rather how to write them. It is a view behind the screen, describing data structures, algorithms and code organization.

This book covers a broad range of design tools for Computer Aided Design (CAD) and Computer Aided Engineering (CAE). The focus of the discussion is on tools for transistor-level physical design and analysis. These tools are applicable across many design disciplines and across all phases of the design process. They comprise a minimum set of tools needed to design working chips consistently.

This book does not describe every tool or type of tool for integrated circuit design. Some topics are omitted because of limited space and time, because they are not sufficiently different from tools that are described in the book, because they are too new and have not proven their worth or because they comprise a large and separate body of knowledge that requires a more thorough treatment than would fit into this book. I have attempted, at least, to mention in context the tools I have omitted to allow the reader to pursue them separately.

This book is the book I wish I could have read when I got started. It introduces tools across the whole spectrum of design tasks. It describes the most successful algorithms and techniques in a form that is comfortable to a novice in the field. I am somewhat disappointed with how little actually fits into this book. Readers should be warned that it is, as it says in the title, an introduction. There is much to computer aided design that is not here, but here is a starting point.

The Intended Audience

This book is intended to tell a programmer how to write software for computer aided design of integrated circuits. It does not discuss theoretical results or describe algorithms in rigorous mathematical terms. It describes computer software and does so in a language that should be comfortable to programmers. As a secondary goal, this book gives the rationale behind decisions made when developing CAD software to give the reader the ability to make these design decisions himself with consideration of his problem. These decisions can make the difference between a tool that is widely useful and one that is a simple experimental toy.

This book assumes a solid knowledge of programming and that the reader has had an introduction to data structures and algorithms. Although no integrated circuit design experience is required, the whole exercise of writing tools for integrated circuit design is pointless without some knowledge of the target domain. A programmer could learn to write design tools without the integrated circuit design expertise, but some knowledge is needed to understand the design tradeoffs.

Although this book is suitable as a textbook for advanced undergraduates, it is more suited to early graduate students in computer science who have the programming experience and have been exposed to integrated circuit design issues. This book is also useful in an industry setting where there is interest in developing or interfacing computer aided design tools. A programmer in such an environment should find the discussion in this book revealing about the software with which he is working.

Current practitioners of CAD programming are often very knowledgeable in their specific domain, but somewhat ignorant of techniques in other areas. This book will round out their breadth of knowledge of CAD tools across the range of integrated circuit design.

Overview of the Book

The book adopts an informal style. Where precision is necessary, it is in the form of pieces of code. Ideas are motivated and developed from simple solutions to more complex ones. Later chapters build on earlier ones, but each chapter is understandable and complete on its own.

The book starts with an overview of the design process and of CAD tools and proceeds to summaries of parsing and interactive graphics techniques. The goals of these introductory chapters are to provide background material and to emphasize the parts of those fields that are applicable to CAD. If the reader is familiar with these fields, he may safely skip the introductory chapters. Integrated circuit design, parsing and computer graphics are fields of study in their own right and the summaries given in this book do not cover those fields in great depth.

Later sections of the book describe graphical techniques, procedural techniques and analysis; each chapter is dedicated to one type of tool. Each introduces the tool and describes its implementation in a developmental way. Simple options are discussed and their drawbacks are used to provide the rationale for the complexity of the final system. These chapters are intended not only to give the reader the knowledge of the methods and techniques of building tools, but the reasons for the complexity of those tools. It is hoped that the reader will be able not only to implement these tools, but to make the decisions necessary to further develop the tools described in this book.

At the end of each chapter I briefly discuss related tools. The assignments at the end of the chapter serve two purposes: first, to focus the reader's thinking to issues that he should consider during the design and second, to give the reader experience developing CAD code in a reasonable framework. Every chapter has a categorized bibliography to direct further investigation of the topics discussed in the chapter.

I have attempted to provide all the information necessary for a reader who is unfamiliar with CAD to understand all the tools. Some readers, especially those who are already practicing CAD programmers, may find the presentation tedious and overly-detailed in places. I deemed it better to include the detail and allow the advanced reader to skim, rather than to leave out the detail and force the novice to dig.

Acknowledgements

Primary thanks go to the management and the employees of VLSI Technology, especially Al Stein, Henri Jarrat and Doug Fairbairn who kindly accepted this drain on my time.

Additional thanks are due to Jim Rowson, who assisted in the original outline five years ago and who provided daily encouragement and suggestions and who was often the first to read rough drafts. I wish to thank the VLSI Design Technology group who described algorithms, read initial drafts and corrected mistakes and misperceptions, especially Bill Walker, Ken Van Egmond, Tom Schaefer, Paul McLellan, Dick Lang and David Chapman. Thanks also to the internal reviewers, Tom Bulgerin, Suresh Dholakia, Nancy Gomes, Leslie Grate, Mike Grossman, Mark Hartoog, David Hsu, Chris Kingsley, Mike Kliment, Jim Lipman, Antonio Martínez, Scott Nance, Charles Ng, Shawn Purcell, Jim Rowson, Bill Salefski, Laura Smith, Russ Steinweg, and Bob Shur.

This book was formatted and printed entirely on VLSI Technology's documentation preparation system with figures and examples from the editors of VTItools. Many thanks to the people who developed the documentation system and kept it running, particularly Paul McLellan, Art Cabral, SE Tan, Debby Hungerford, Paul Gazdik and Flo Paroli.

CONTENTS

Preface xiii

Chapter 1. Integrated Circuit Design 1

Introduction to Integrated Circuit Design 1
The Integrated Circuit Design Process 6
Variances from This Model of the Design Process 11
Managing the Complexity of a Design 12
Exercises 15
References 15

Chapter 2. Parsing 17

Reading Files 17
Describing a Data Format 18
An Example 24
Parser Overview 25
Implementation of the Parser 26
Internal Parser Procedures 29
The Token Scanner 31
Building a Data Structure 32
Data Structures 33
The Semantics Module 34

Memory Usage Issues 38
Writing Layout Files 38
Exercises 39
References 41

Chapter 3. Graphics 43

Output Devices 43
Graphical Output Primitives 45
Virtual Screen Coordinates 45
User Coordinates 47
Clipping and Viewports 47
Summary of Graphics Output Features 56
Summary of Coordinate Systems 56
Color 57
Graphical Input 61
Dynamic Displays 62
Exercises 65
References 67

Chapter 4. Plotting Layout 69

The Core of a Plotter 69
The Command Loop 70
Loading a File 71
Displaying a Cell 72
Necessary Additions 73
Desirable Features 74
Practical Considerations 75
Considerations for Pen Plotters 75
Considerations for Raster Plotters 77
Exercises 79
References 80

Chapter 5. Layout Editor 81

Overview of a Layout Editor 81
Turning the Plotter Into an Editor 82
Pointing As an Alternative to Typing Positions 84
Filters 85
Editor State 85
Modes and Modeless Editors 86
Menus As an Alternative to Typing Commands 88
Data Structures 88
Essential Features 97
Desirable Features 100
User Interface 102
The Database Alternative To Reading and Writing Files 108
Symbolic Layout 109
Schematic Editor 113
Exercises 115
References 117

Chapter 6. Layout Language 119

Embedded Language 120
A Simple System 120
A More Usable System 121
Layout Language Procedures 122
Data Structures 124
Example 124
Symbolic Layout 125
The Example Again 129
Using Bounding Boxes and Connectors 130
Reading Layout Files 131
Using Language Facilities 131
Parameterized Cells 133

The Layout Language Module As a LayoutParser 135
A Procedural Netlist Generator 136
Drawbacks of a Layout Language 136
Exercises 137
References 138

Chapter 7. Layout Generators 141

Parameterized Cells 141
PLA Generator 149
Introduction to Silicon Compilation 159
Datapath Compiler 159
Introduction to Placement and Routing 168
Exercises 177
References 179

Chapter 8. Layout Analysis 183

Overview and Background 183
Design Rules 184
Object-Based DRC 185
Edge-Based Layout Operations 187
Handling Intersecting Edges 191
Polygon Merging 191
Arbitrary Boolean Operations On Layout Layers 198
Resizing Layout 199
Using Bloat and Shrink to Perform a Design Rule Check 202
A More Efficient DRC 203
Connectivity 210
Performance Optimization Considerations 212
Reporting DRC Errors 213
Ambiguous Corner Checks 214
Glitches 216

Technology File 218
Fast Sorting For Edge Files 221
Circuit Extraction 221
Options 227
Mask Tooling 229
Other Algorithms 229
Exercises 231
References 232

Chapter 9. Simulation 235

Types of Simulators 236
A Simple Behavioral Simulator 237
Time and the Event Queue 239
Modelling Shared Busses 241
A Logic Simulator 242
Controlling and Observing the Simulation 246
Managing the Event Queue 249
Simulating Transistors 250
Transistor Simulation Example 261
Debugging Commands 264
Enhancements to the Transistor Simulation Algorithms 264
Static Checks 269
Multi-Level Simulation 272
High-level Input 274
Testing Integrated Circuits 275
Fault Simulation 276
Exercises 278
References 280

Index 283

CHAPTER 1

INTEGRATED CIRCUIT DESIGN

We begin our investigation of design tools with a short discussion of integrated circuit design and the design process. We discuss the kinds of data that users manipulate and the tools they need during the design process.

Introduction to Integrated Circuit Design

Integrated circuit manufacturing consists of a series of steps, each of which adds or removes material from an area on a flat surface, typically of silicon. A designer describes a circuit as set of two-dimensional patterns on different *mask layers*, each of which represents one of the manufacturing steps. Other processing steps affect the entire wafer and the actual masks that are used in manufacturing may be combinations of the designer's mask layers, but we will take the design point of view and ignore these manufacturing details. We will consider only the layers that the designer uses to specify the chip.

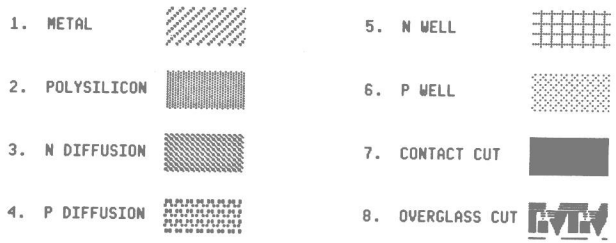
In this section, we discuss the fundamentals of integrated circuit design. The purpose of this section is not to enable you to design circuits, but to provide you with an understanding of the basic tasks for which we develop tools. For more detail on design and manufacturing issues, see Mead and Conway (1980) or Mukherjee (1986). For a further discussion of design methodology, see Lattin (1979), vanCleemput (1979), Trimberger et al. (1981) or Niessen (1983).

Mask Layers

We describe the layers in a dual-well CMOS (Complementary Metal Oxide Semiconductor) manufacturing process. Different CMOS processes may use different layers, but the design issues are the same.

1. *Metal.* The primary wiring layer, typically aluminum.
2. *Polysilicon.* Poly-crystalline silicon, usually abbreviated *poly*, forms the gates of transistors. It is also a conductor and used for wiring.
3. *N+ Diffusion.* Silicon diffused with impurities to create a large excess of Negative charge carriers (electrons). It forms the source and drain of MOS N-channel transistors. It is a conductor and is used for wiring.
4. *P+ Diffusion.* Silicon diffused with impurities to create a large excess of Positive charge carriers (electron holes). It forms the source and drain of MOS P-channel transistors. It is a conductor and is used for wiring.
5. *N-Well.* A region of the silicon that has been doped to have a small excess of Negative charge carriers. It surrounds P-channel transistors.
6. *P-Well.* A region of the silicon that has been doped to have a small excess of Positive charge carriers. It surrounds N-channel transistors.
7. *Contact Cut.* A hole cut in the insulating oxide to allow metal to connect electrically to polysilicon and the diffusions.
8. *Overglass Cut.* A hole cut in the protective overglass layer to allow bonding wires to connect to metal bonding pads.

Designers use colors to distinguish the different layers, but there is no standard color scheme. Throughout this book, we will show pictures of circuit layouts in black and white using different patterns to represent different layers.



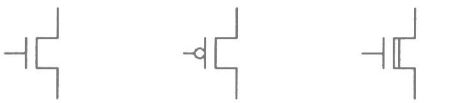
Wires

We can make wires in metal, polysilicon or the diffusions. Metal is preferred, because it has lower resistance and capacitance than the others, but polysilicon and diffusion may be used as well. We form a contact between metal and polysilicon with a contact cut. The structure including the metal, the contact cut and the polysilicon or diffusion is called a *contact.* An integrated circuit manufacturing process may have two or more metal layers separated by an insulating oxide. Designers have additional masks corresponding to the additional metal layers and connections between them, called *vias.*

Transistors

A polysilicon wire cuts a diffusion wire where they cross forming a MOS transistor. The polysilicon becomes the gate of the transistor and breaks the diffusion into two electrical nodes, the transistor source and drain. There are two kinds of transistors in CMOS. If we form a transistor on P-diffusion in an N-well, we get a *P-channel transistor.* When we form a transistor on N-diffusion in a P well, we get an *N-channel transistor.*

A MOS transistor acts like a switch. When there is a high voltage on the gate of an N-channel transistor, the switch is closed, making the connection between the source and drain. When there is a low voltage on the gate, the switch is open and source and drain are separated. A P-channel transistor works the same way except that a high voltage opens the switch and a low voltage closes it. In NMOS, there is a third kind of transistor, a depletion transistor. A *depletion* transistor is always closed, but has high resistance and is used as a resistor.



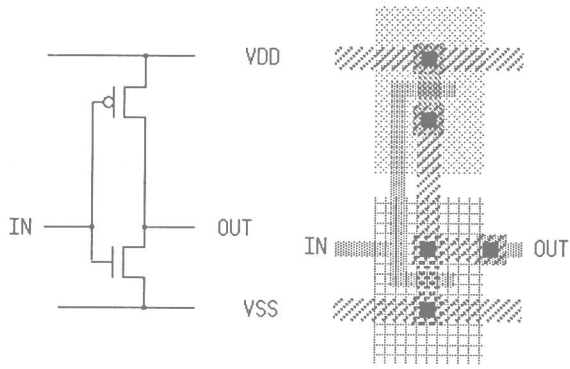
Electrical Details

An N-channel transistor turns on when there is a high voltage on its gate with respect to the substrate under the transistor. During design, a designer must connect the area under the N-transistors to VSS and the area under the P-transistors to VDD to provide the reference voltage for the

transistors. These connections are called *substrate contacts* or *well ties* and not only allow transistors to operate correctly, but also prevent a catastrophic failure of CMOS circuits called *latchup*. Integrated circuit designers will certainly notice the absence of substrate contacts in all examples in this book. They have been intentionally omitted to enhance the clarity of the presentation of the tools.

Logic Gates

In MOS technologies, the high voltage power supply is called *VDD*, the low voltage is *VSS* or rarely, *GROUND*. We build logic gates in CMOS by making a *pullup structure* with transistor switches that connect the output to VDD when we want the output to go high and a *pulldown structure* that connects the output to VSS when we want it to go low. We make a CMOS inverter like the following figure.



VDD is connected to the metal wire on top, VSS is connected to the wire on the bottom. The input signal enters in polysilicon on the left and forms the gate of two transistors, an N-transistor on the bottom and a P-transistor on the top. The bottom N-transistor turns *on* when IN is high, connecting OUT to VSS. The top P-transistor turns *on* when IN is low, connecting OUT to VDD.

Connections in series make AND-type connections, connections in parallel make OR-type connections. To connect the output to VSS when two signals are both high, we connect two N-transistors in series to VSS. To connect the output to VSS when either of two signals are high, we connect two N-transistors in parallel to the output.