DIGITAL DESIGN WITH STANDARD MSI & LSI

Digital Design with Standard MSI and LSI

THOMAS R. BLAKESLEE

Consulting Engineer

A WILEY-INTERSCIENCE PUBLICATION

JOHN WILEY & SONS

New York • London • Sydney • Toronto

Library of Congress Cataloging in Publication Data

Blakeslee, Thomas R 1937–
Digital design with standard MSI and LSI.

"A Wiley-Interscience publication."
Includes bibliographies and index.
1. Logic circuits. 2. Digital integrated circuits.
I. Title.

TK7868.L6B55 621.3819'58'35 74-23930
ISBN 0-471-07937-5

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

Preface

Logic design has traditionally been concerned with mathematically trying to reduce to a minimum the number of diodes and flip-flops in a logic system. Although this has served us well through the progression from vacuum tubes to transistors to small-scale integrated circuits (ICs), today's large- and medium-scale ICs have made this design criterion irrelevant. We can now buy 1024 flip-flops (in a MOS RAM) for less than it cost a few years ago for a single flip-flop. The MOS read-only memory equivalent of many hundreds of logic gates likewise costs about the same as a single gate cost a few years ago. In addition, microprocessors have made it possible today to build complex systems entirely from *standard* LSI components.

Although rapid progress continues to make faster, cheaper, and more complex ICs available, the *basic* change in direction has already occurred. This book is an attempt to develop a design approach relevant to the new order that has emerged. The basic principles developed should therefore be more and more applicable as the IC revolution continues. It is already quite clear, for example, that microprocessors will continue to grow faster, cheaper, and more like real computers. The digital designer must therefore have a greater knowledge of programming. The techniques developed for using higher levels of integration and minimizing package count should grow more useful as the cost differential between large- and small-scale ICs continues to shrink.

This book emphasizes system design, because the LSI and MSI building blocks are really system components, that can be traded off with other mechanical and electrical components for an optimum overall result. Since the cost of the logic itself is becoming a negligible part of total system cost, it is increasingly important to attack the system problem as a whole. Though the book concentrates on the use of *standard* LSI and MSI products, most of the principles are equally valid where product volume makes the use of custom LSI circuits practical.

Briefly, the design technique centers around using standardardized "bargain components" to handle most system requirements, then filling in, as needed, with small-scale ICs. The primary aim is *minimizing IC package count,* rather than the traditional minimization of the number of flip-flops and gate inputs. Many practical considerations normally ignored by logic texts are also included, because ignorance of these problems causes consequences far more disastrous than just wasting a few gates.

I have tried to make the book useful to working professionals and students alike by covering basic concepts, terminology, and abbreviations in a glossary rather than in the text. In this way the professional and the advanced student do not have to be bored with concepts they already know, yet the beginner can learn the terms, as needed, by looking them up in the glossary.

The first two chapters establish and justify, by example, the design philosophy used throughout the remainder of the book. While the emphasis in Chapters 3 to 5 is on using MSI and LSI circuits where possible, the basic techniques of traditional logic design are also presented since they are still needed to "fill in the cracks" between MSI and LSI circuits and for very small subsystems. An intuitive state assignment method is presented that allows minimizing the total logic without obscuring system operation with mathematical abstractions.

Chapters 7 and 8 cover programmed logic and programming techniques and describe software and hardware for the Intel and National Semiconductor microprocessors. The presentation is general enough to apply to other microprocessors and to minicomputers. Programmed logic is treated as just another approach to logic design. Chapter 9 discusses the time dimension as a method of making multiple use of circuits for bit serial or time-shared operation. Since circuit speeds are increasing and LSI circuits are interconnection limited, this is becoming an increasingly important technique.

Chapters 6 and 10 consider the nasty realities of the real world such as race conditions, hangup states, noise, reflections, and cross talk. Chapter 11 covers some of the techniques, such as negative feedback and incremental operation, that are used to simplify modern input/output devices. Several successful real computer peripheral devices are described as examples. Chapter 12 discusses, from a strictly practical point of view, the use of statistics in reliability and buffer overflow calculations. Finally, Chapter 13 discusses the social consequences of engineering and the possibilities of using the fantastic capabilities of ICs to improve the quality of life.

THOMAS R. BLAKESLEE

Woodside, California
August 1974

Contents

CHAPTER ONE

PHILOSOPHY Adapting the Job to the Bargain Components

When we consider the incredible range of seemingly unrelated tasks being done by **digital* integrated circuits** (ICs), it seems that digital logic is, indeed, about to take over the world. Since most of these jobs are not basically digital or even electrical, why are they done so much more efficiently with digital logic? By exploring the answers to this question, we can gain an understanding of the real reasons for the tremendous savings offered by the digital approach and thereby learn how to fully exploit its potential.

1.1 STANDARDIZATION

Standardization is the key to the fantastic material wealth we have today. When a large number of identical (or nearly identical) items are manufactured on a production line, tremendous savings are possible. A large production volume makes it feasible to invest heavily in special production equipment and set up an efficient production line. The gulf between the bargains available in high-volume standard products and the cost of things that cannot be produced

* Boldface type indicates the first use of a term in the Glossary.

in volume has grown steadily since Henry Ford started the trend. Today you can buy a complete typewriter, with all its intricate parts, for less than it would cost to have one of the key levers made in a machine shop. Think of what it would cost to have the hundreds of intricate parts in a $10 wristwatch custom made by a machinist!

The "batch" process used to manufacture ICs is perhaps the ultimate example of this kind of efficient production. Thousands of repetitions of the same circuit are produced on a single **wafer** of silicon only 3 in. in diameter (see Fig. 1-1). Each time a step in the manufacturing process is done, it is done for thousands of circuits in a single operation. This means that ICs can be made *very* economically, *but only if we can use a large volume of identical circuits.* Except for the final packaging, the labor required to make one circuit is the same as that to make a thousand! Of course, the processing of the wafers themselves is most efficient if done on a high-volume production line basis. If every wafer produced contains 1000 circuits, just 100 wafers a day gives us 100,000 integrated circuits! It is obvious, then, that we must use an IC type in tremendous volume to reap the full benefits of this production technique. If we are making a portable radio, a television set, or a pocket calculator, this is no problem—we can just design a custom circuit and the volume will be high simply because the volume of our product is high.

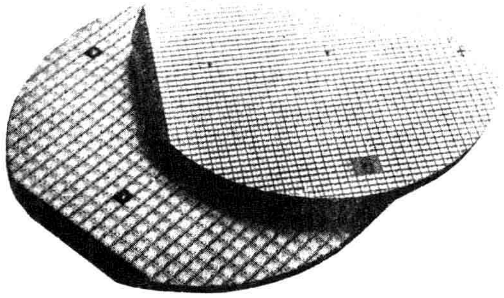
1.2 STANDARD DIGITAL CIRCUITS

Most digital systems, unfortunately, are produced in only moderate quantities. The only way we can really reap the benefits of high-volume IC production is if the same basic integrated circuits can be used in many *different* systems throughout the industry. This is a reality today only because of a very special quality possessed by **binary** logic.

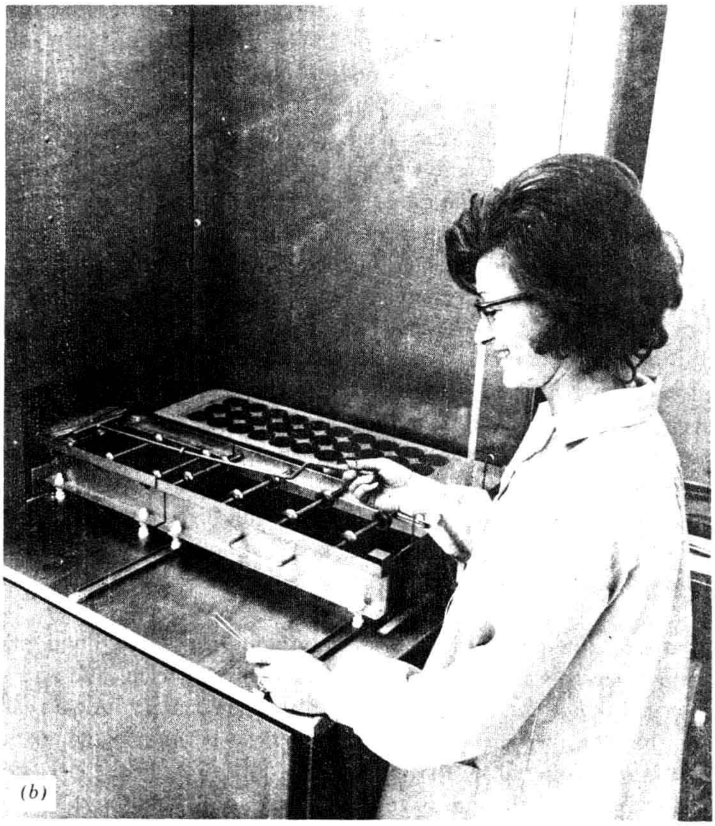
The binary number system is the simplest possible because digits can have only one of two values: 1 or 0. Because of this, the number of possible combinations is quite restricted. For example, the multiplication tables, which we spent years memorizing in the decimal (10 valued) number system, are trivial in the binary system.* The whole system of binary (Boolean) algebra can be developed, with proofs, in a few pages (see Refs. 1 and 2). It is this simplicity that has made standardization of digital ICs possible. It is actually possible to build *any* logic system, including a large computer, entirely from a single **gate** circuit type and a single **flip-flop** circuit type.† In the early days of ICs this is exactly what was done. By building the entire computer out of NAND gates

* The complete multiplication tables are $1 \times 1 = 1$, $1 \times 0 = 0$, and $0 \times 0 = 0$. Addition is equally simple: $1 + 1 = 0$ (and carry 1), $1 + 0 = 1$, and $0 + 0 = 0$.

† Actually, it *could* be done with a single type of gated flip-flop, but this is somewhat wasteful.



(a)



(b)

Fig. 1-1. The key to IC economy: 30 wafers, each capable of producing hundreds of ICs, can be processed simultaneously (Applied Materials AMG-500 Reactor System).

and *J–K* flip-flops, quite a sizable volume of these two circuit types could be consumed by a single company making only 100 or so computers.

This ultimate in standardization is practical only with digital logic. High-volume mechanical parts can be made very economically, as the $5 alarm clock proves, but too many variations are possible in mechanical components to achieve the kind of standardization we now have in digital ICs. Clock gears, for example, can have any number of teeth and be any of an infinite number of sizes. Catalogs of standard gears do exist, but they have pages and pages of tables of different gear sizes and numbers of teeth. With such a large selection, it is impossible to produce standard gear components for general use in anywhere near the required volume.

With digital ICs standardization is easy. The logic equivalent of the speed-reducing gear train in a clock is a chain of identical, standard, flip-flops—each of which reduces the speed by a factor of 2. These flip-flops are identical to the ones used in a computer, a tape unit, or any other logic device. For this reason, it is quite practical to build just one clock, out of standard digital components (in fact many hobbyists have done it), but it would take thousands of dollars worth of custom-machined parts to make just one mechanical clock.

We thus have the key to the digital logic takeover of the world: *standardized bargain components*. The advantages of standardization are great even when the job does not fit the components. The example of the clock illustrates this. Since we ultimately want a mechanical representation of the time, it certainly seems logical to make the clock mechanical in the first place. Although the standard flip-flops are nice, they produce nothing but electrical 1 and 0 outputs, which then have to be decoded and converted into something we can see. This is a lot of extra trouble, but the result is much less expensive.

The tremendous savings from using the standardized components more than offset the inefficiency of adapting the components to the application. This is not an exceptional case. As a matter of fact, the gulf between the capability per dollar of ICs and any other approach makes it almost always true that, if we can somehow adapt the job to these components, the result will be more economical.

Now that we know why digital logic is taking over so many nondigital jobs, we can generalize this principle into a general design technique and slogan: "*Fit the job to the bargain components.*" All that is required for this technique to work is that the inefficiency in adapting to the bargain component is less than the advantage offered by its use. In the case of integrated circuits versus mechanical components the gulf is very wide indeed. A 100:1 cost advantage is not unusual where the quantities are small. This means that we would be ahead if our efficiency is greater than 1%! Since the technology gap between mechanical parts and ICs is growing all the time, this design technique, which is quite effective now, should prove even more effective in the future.