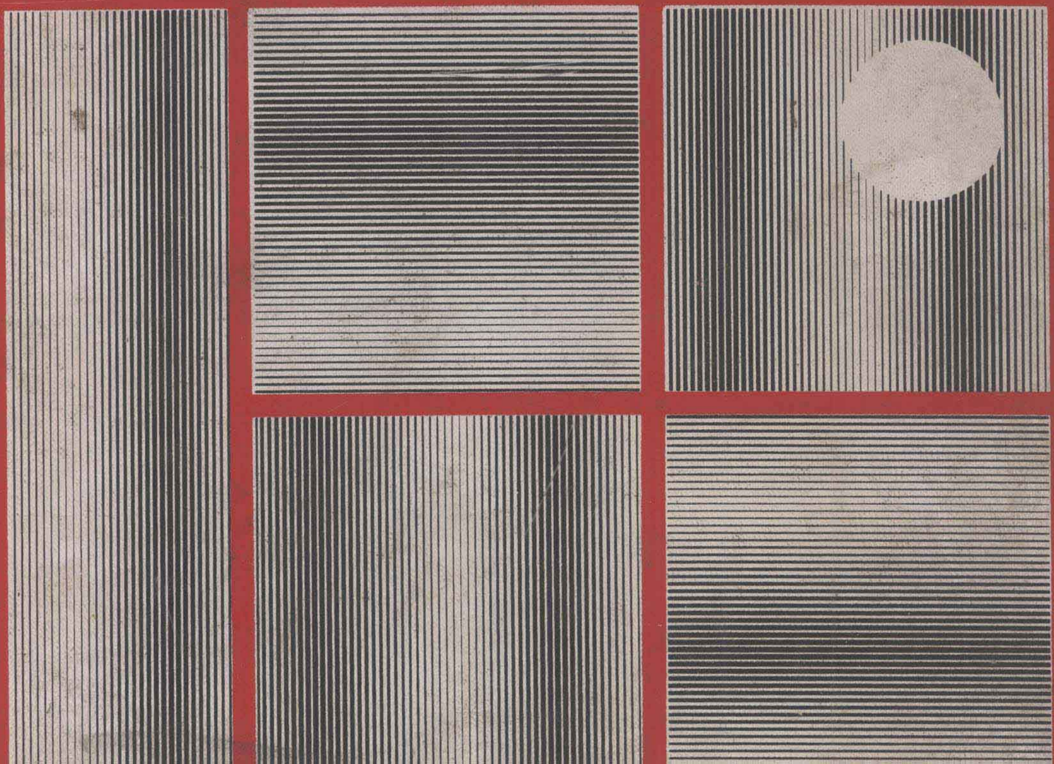
KENNETH L. SHORT

Microprocessors and Programmed Logic



Microprocessors and Programmed Logic

Kenneth L. Short

STATE UNIVERSITY OF NY
STONY BROOK, NY 11790

PRENTICE-HALL, INC., Englewood Cliffs, NJ 07632

Library of Congress Cataloging in Publication Data

Short, Kenneth L
Microprocessors and programmed logic.

Bibliography: p.
Includes index.
1. Microprocessors. I. Title.
QA76.5.S496 001.6'4 79-23167
ISBN 0-13-581173-2

Editorial/production supervision and interior design by Steven Bobker
Cover design by Edsal Enterprises
Manufacturing buyers: Gordon Osbourne and Anthony Caruso

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

Prentice-Hall International, Inc., *London*
Prentice-Hall of Australia Pty. Limited, *Sydney*
Prentice-Hall of Canada, Ltd., *Toronto*
Prentice-Hall of India Private Limited, *New Delhi*
Prentice-Hall of Japan, Inc., *Tokyo*
Prentice-Hall of Southeast Asia Pte. Ltd., *Singapore*
Whitehall Books Limited, *Wellington, New Zealand*

Preface

The single most significant development in digital systems design in recent years has been the advent of the microprocessor, a central processing unit integrated on a single chip of silicon. The processing power and economics of the microprocessor have had a tremendous impact on the way digital systems are designed and on their scope of application.

This book is about the microprocessor. It is also about its related integrated circuits and the hardware and software design of microprocessor based systems. Its purpose is to first provide the reader with a thorough understanding of the basic hardware and software concepts necessary for the design of microprocessor based systems, and, further, to provide the reader with an in-depth knowledge of specific actual devices and the attendant practical considerations and design techniques necessary to effectively design systems using them.

A unique feature of this book is its utilization of a single microprocessor, the 8085A, as the example used to illustrate fundamental concepts. Use of a single microprocessor as the instructional example allows an increased depth of coverage of the operation, features, and limitations of a real device.

In addition, it allows use of a single consistent set of signals and signal names for interfacing the many logical devices which constitute a microprocessor system. Use of a single set of signal names simplifies the reader's task in understanding hardware interfacing concepts and the functional operation of various LSI devices.

The 8085A is a general purpose 8-bit microprocessor. This microprocessor was chosen because of its widespread use in industrial applications and its widespread support. This support manifests itself in the form of documentation, application notes, and software and hardware development aids for the 8085A. In addition, there exists a large family of peripheral LSI devices which are designed to be compatible with the 8085A. Because of its wide applicability, the reader will find that the knowledge gained about the 8085A and its support devices is immediately applicable to many actual designs in industry.

A further advantage of studying a single microprocessor in depth is that the reader not only learns of the features of the device but also learns that with these features come attendant limitations which must be dealt with in any practical application. It should be noted that a reader following a device specific instructional approach will find that once a specific microprocessor and its application in digital system design has been mastered, it is relatively easy to understand the operation and application of other microprocessors from a study of the manufacturers' user manuals and application notes. This has been the experience of university and industry students who have followed this approach in the author's microprocessor courses over the past several years.

The design of microprocessor systems requires a knowledge of both hardware and software. It is assumed that the reader has a basic knowledge of digital hardware at the gate and flip-flop levels. This material can be found in any introductory book on digital systems design. The software concepts in this book are illustrated using assembly language for the 8085A. However, prior knowledge of assembly language programming is not necessary. Some general knowledge of computer programming in a high level language is desirable.

The goal has been to introduce the necessary hardware and software concepts in an elementary, systematic, and integrated fashion and to logically build upon these concepts. While the study of no single text can provide mastery in a subject area, it is believed that this text will provide the reader with a solid foundation for the development of proficiency in the design of microprocessor systems.

This book covers the topics recommended for inclusion in the course DL-3 Microprocessor Systems as part of a computer science and engineering curriculum as proposed by the Model Curriculum Subcommittee of the IEEE Computer Society.[1]

Acknowledgments

A number of people have contributed to this book being written. While I cannot thank them all, I would like to express my appreciation to Mr. Paul

[1]"A Curriculum in Computer Science and Engineering Committee Report." IEEE Service Center, 445 Hoes Lane, Piscataway, NJ 08854, January, 1977.

Becker, Mr. Don Buchout, Dr. Velio Marsocci, Mr. Casey Powell, Mr. Rich Reeder, and Dr. David Smith.

I would also like to thank the many students who, in using rough drafts of this text as class notes, have provided comments and suggestions which have led to the improvement of the text.

A very special thanks to Lee Cushman whose editing, organizational, and typing skills were critical to the writing of this book.

Kenneth L. Short

Contents

1 Introduction

A technological advance which is affecting the practice of logic design is the existence of the LSI microprocessor; a data flow and control on one to several LSI chips. In this case the logic designer's building blocks are data flows, control stores, and read/write memory chips. He arranges the chips and programs the control store.

*Glen G. Langdon, Jr.**

**Logic Design: A Review of Theory and Practice.* New York: Academic Press, Inc., 1974.

1.1 THE IMPACT OF LSI ON LOGIC DESIGN

The basic goal of logic design is a system that functions as required and is reliable, easy to maintain, and cost effective. As a rule, simplicity of design is the key to the attainment of this goal. Whereas an overly complex design might meet the first requirement, it undoubtedly would fall short in the other areas.

In practical design, of course, cost effectiveness is of great importance. Generally speaking, the most cost effective design is usually the simplest and, because of its simplicity, more reliable and easier to maintain.

The total cost of a design at the system level includes expenditures for development, components, tooling, assembly, testing, system repair and maintenance, as well as a spare parts inventory. [1,2] This total cost is directly proportional to the number of components in a system, and the number of components has, in the past, been directly proportional to the number of gates and flip-flops. Conventional switching theoretic techniques of digital systems design are geared toward minimizing the number of gates and flip-flops, thereby minimizing the number of components in order to minimize system cost. See Figs. 1.1-1(a) and 1.1-1(b). These minimization techniques were developed originally for systems which used relays to implement gates and flip-flops. As technology advanced, vacuum tubes, then discrete component solid state devices replaced relays. Since the number of components was still proportional to the number of gates and flip-flops, even in designs constructed from vacuum tubes or discrete component solid state devices, the switching theoretic design techniques were still effective in minimizing system cost.

However, in the early 1960's, ***Small Scale Integration, SSI***, provided small scale integrated circuits with as many as 12 gates integrated on a single silicon chip and packaged as a single multilead component. With integrated circuits, ICs, the number of components in a system was no longer proportional to the number of gates and flip-flops but was simply equivalent to the number of IC packages in the system. Thus, the reduction of system cost was dependent on the reduction of the total number of IC packages required. Although system designers could still apply the same switching theoretic techniques, simplification of a circuit which reduced the required number of gates or flip-flops resulted in a savings only if it also reduced the number of IC packages.

As IC technology advanced further, ***Medium Scale Integration, MSI,*** provided circuits with a logic complexity of 13 to 99 equivalent gates per package, and ***Large Scale Integration, LSI***, provided circuits with a logic complexity of 100 or more equivalent gates per package, seriously compromising the effectiveness of conventional switching theoretic design techniques in reducing system cost. Fabrication techniques for MSI and LSI circuits, which make it as inexpensive to put 100 or more gates on a chip as 10 destroyed the gate-flip-flop/component-count relationship, and now, for systems implemented with

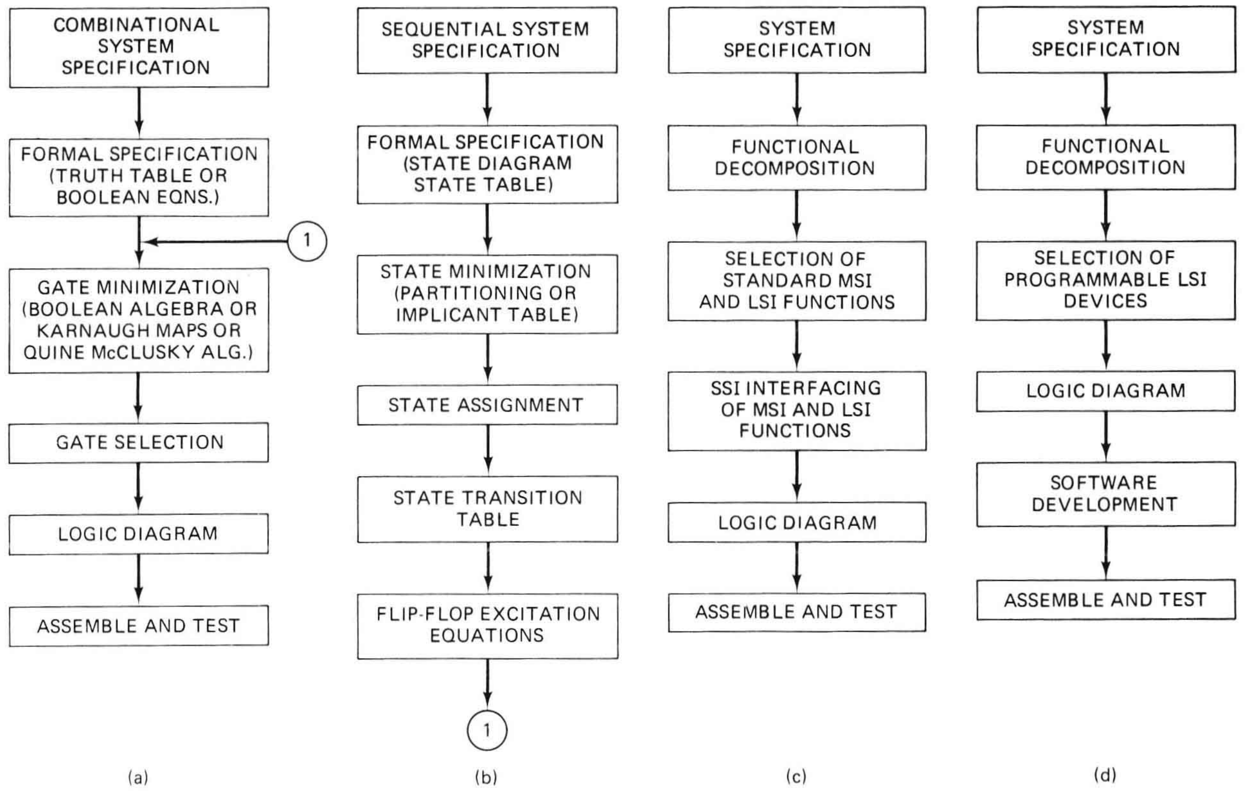

COMBINATIONAL SYSTEM SPECIFICATION
FORMAL SPECIFICATION (TRUTH TABLE OR BOOLEAN EQNS.)
1
GATE MINIMIZATION (BOOLEAN ALGEBRA OR KARNAUGH MAPS OR QUINE McCLUSKY ALG.)
GATE SELECTION
LOGIC DIAGRAM
ASSEMBLE AND TEST
(a)
SEQUENTIAL SYSTEM SPECIFICATION
FORMAL SPECIFICATION (STATE DIAGRAM STATE TABLE)
STATE MINIMIZATION (PARTITIONING OR IMPLICANT TABLE)
STATE ASSIGNMENT
STATE TRANSITION TABLE
FLIP-FLOP EXCITATION EQUATIONS
1
(b)
SYSTEM SPECIFICATION
FUNCTIONAL DECOMPOSITION
SELECTION OF STANDARD MSI AND LSI FUNCTIONS
SSI INTERFACING OF MSI AND LSI FUNCTIONS
LOGIC DIAGRAM
ASSEMBLE AND TEST
(c)
SYSTEM SPECIFICATION
FUNCTIONAL DECOMPOSITION
SELECTION OF PROGRAMMABLE LSI DEVICES
LOGIC DIAGRAM
SOFTWARE DEVELOPMENT
ASSEMBLE AND TEST
(d)

Figure 1.1-1. Steps in digital system design: (a) combinational and (b) sequential design using gates and flip-flops (c) design using standard MSI and LSI circuits (d) design using programmable LSI circuits.

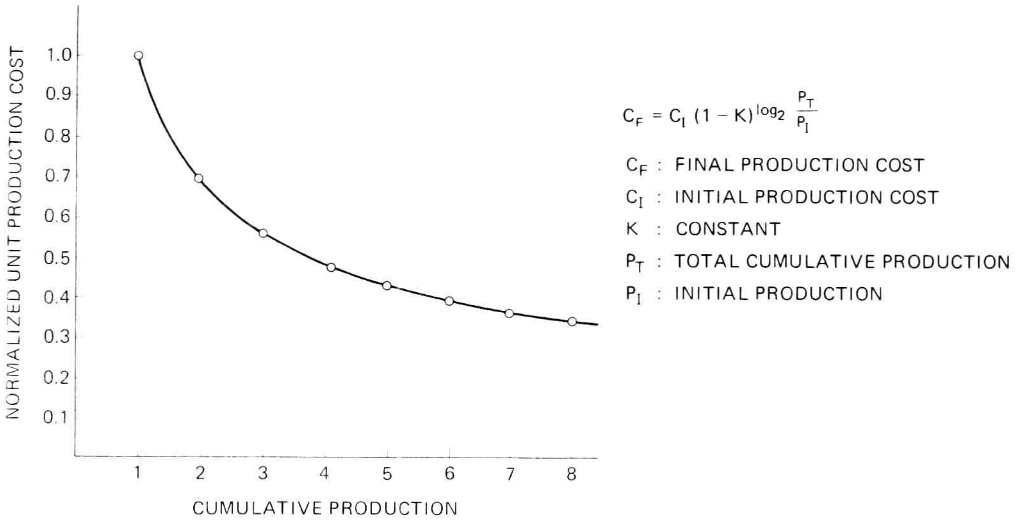


Figure 1.1-2. Semiconductor learning curve.

integrated circuits, total cost is directly proportional to the number of IC packages.[1]

Maximum utilization of LSI results in decreased system costs because LSI circuits provide more logic per package and require less power consumption. And reducing the number of lead connections by replacing many SSI and MSI packages with fewer LSI packages provides greater reliability, since a common source of failure for an IC is at a lead connection to the chip.

The economics of IC manufacture demand that entire functions requiring large numbers of gates be fabricated on a single chip. [3] The complexity of the functions is limited primarily by the chip area required for their fabrication. This area must not exceed that compatible with a high manufacturing yield. ***Yield*** is the percentage of acceptable ICs resulting from the manufacturing process. To limit the chip area, the number of external connections to the chip, which provide input and output data and control signals, must also be limited.

When medium and large scale integrated functions are produced in large quantities, the production cost can be amortized so that the price of these functions is far less than that of the equivalent functions implemented with SSI circuits. Utilization of standard MSI and LSI functions, although they might include gates which are unused in a particular application, is usually more economical than designing minimized logic implemented at the gate level. See Fig. 1.1-1(c).

[1]Exceptions to this relationship occur in LSI systems which use microprocessors. In some applications software development costs may have a greater impact on total system cost than the hardware component count.