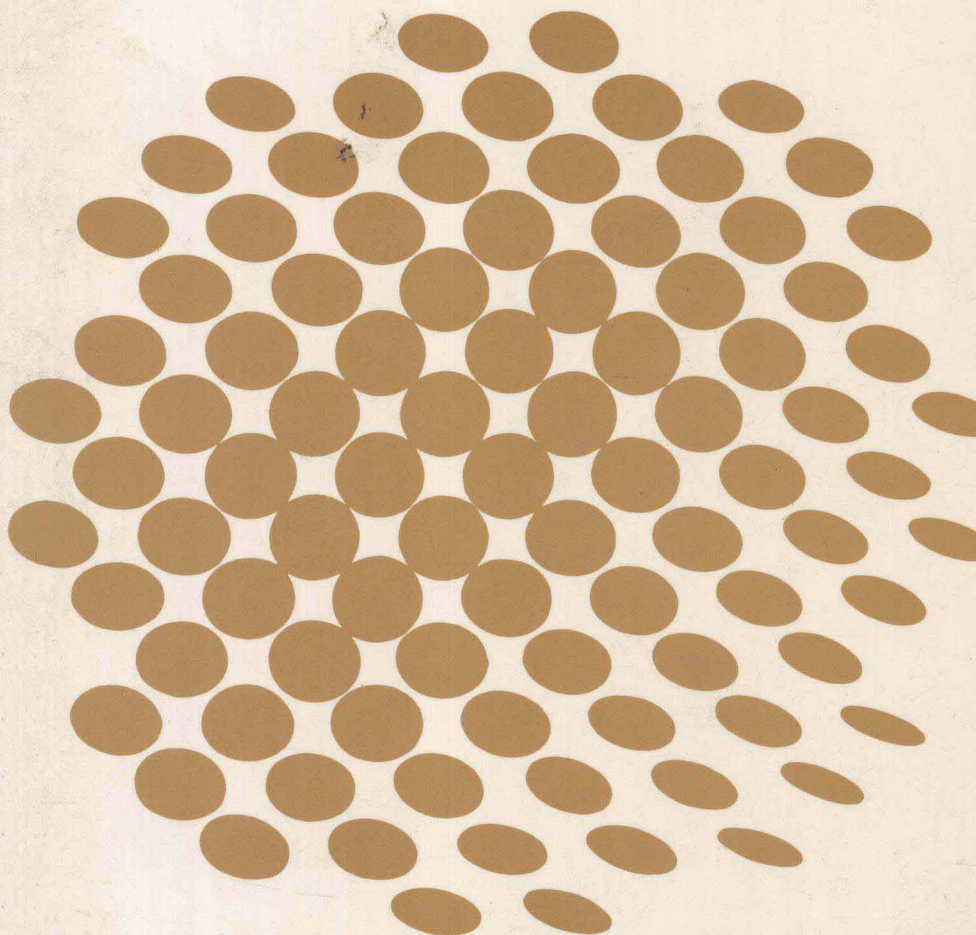


DIGITAL HARDWARE **DESIGN**

JOHN B. PEATMAN



International Student Edition

DIGITAL HARDWARE DESIGN

JOHN B. PEATMAN

School of Electrical Engineering
Georgia Institute of Technology

INTERNATIONAL STUDENT EDITION

McGRAW-HILL INTERNATIONAL BOOK COMPANY

Auckland Bogota Guatemala Hamburg Johannesburg Lisbon
London Madrid Mexico New Delhi Panama Paris San Juan
São Paulo Singapore Sydney Tokyo

In memoriam
Edward S. Donn
1939–1979

DIGITAL HARDWARE DESIGN

INTERNATIONAL STUDENT EDITION

Exclusive rights by McGraw-Hill Kogakusha, Ltd. for manufacture and export. This book cannot be re-exported from the country to which it is consigned by McGraw-Hill.

I

Copyright © 1980 by McGraw-Hill, Inc.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher.

This book was set in Melior by Progressive Typographers.

The editors were Frank J. Cerra and Madelaine Eichberg;

the copy editor was Barbara Zeiders;

the designer was Anne Canevari Green;

the production supervisor was Richard A. Ausburn.

The drawings were done by J & R Services, Inc.

Library of Congress Cataloging in Publication Data

Peatman, John B

Digital hardware design.

Includes index.

1. Electronic digital computers—Design and construction. 2. Digital electronics. I. Title.

TK7888.3.P36 621.3819'582 80-10452

ISBN 0-07-049132-1

When ordering this title use ISBN 0-07-066470-6

DIGITAL HARDWARE DESIGN

PREFACE

This book is intended for the engineer who is interested in the design of digital instruments and devices. It recognizes the changes wrought in the *design process* by the advent of microcomputers and other products of the integrated circuit revolution. It differs from many books in that it develops design capability while focusing upon the larger problem of designing *maintainable* products.

The role of the microcomputer in the design of an instrument or device has become pervasive. This role is described in order to put into perspective the relationship between the microcomputer and the remainder of a design. With a clear view of the opportunities afforded by this relationship, the book concentrates upon this other half of the digital design problem. That is, it concentrates upon the design of circuitry to handle those tasks which require a faster response than is obtainable by an algorithm executed in the software of a microcomputer.

Because of the pervasive role of the microcomputer in the design process, the study of this book is best supplemented by a study of microcomputer hardware and software development plus a study of those input/output devices and considerations which can be handled directly by the microcomputer. This book has been prepared by assuming that the reader will either precede or follow its study with the study of a companion microcomputer book, such as the author's *Microcomputer-Based Design* (McGraw-Hill, 1977). Although one might question which study should come first, this book has been prepared to be useful

to readers in either situation. In addition, the characteristics of a particularly versatile one-chip microcomputer (Motorola's 6801) are described in Appendix A3 to illustrate the support that a well-conceived microcomputer can provide for instrument design.

With its emphasis upon digital design in a world that has become dominated by low-cost microcomputers, this book breaks away from a variety of design ideas which have been important in the past. Thus, it accentuates the use of gates to implement only the simplest of combinational functions. The complex functions that required the Karnaugh maps or the Quine-McCluskey method in the past arise less frequently in these times when *all* functions can be translated into the software of a microcomputer. For applications in which such treatment is too slow, this book discusses the relegation of combinational circuit complexity to *programmable* devices (e.g., PROMs, FPLAs, and PALs). This approach is justified on the commonly accepted grounds of minimizing package count and supporting maintainability with a regular circuit structure. Likewise, designing complex sequential circuits has largely given way to designing simple sequential circuits and using standard counters and shift registers.

The microcomputer has caused another revolution in the design process. There is no longer any good reason (other than the insensitivity of a designer) for creating a product which cannot thoroughly test itself. If the product malfunctions, it should help pinpoint the source of the malfunction. The techniques of designing maintainability into a product are given solid treatment in this book. These ideas are made concrete by carrying out several iterations on the design of a specific instrument (a "signature analyzer").

The consistent, reliable handling of the timing considerations arising in a design have long been the hallmark of the experienced digital designer. These timing considerations, and the design opportunities they afford, are given extensive treatment. Furthermore, most instrument designs require the careful handling of the communications between real-time circuitry having one clock source and a microcomputer having another. Several alternative approaches are developed.

Because this book considers the design of fast circuitry to augment the microcomputer part of an instrument design, it must address the problem of selecting a suitable family of logic to use for such a design. Throughout the book, the 74LSxxx family of TTL logic (introduced by Texas Instruments and second-sourced by many companies) is used. This follows present industry practice and offers the opportunity to speed up such a design with little modification other than to substitute 74Sxxx parts (or the recently introduced 74ALSxxx or 74ASxxx parts) which serve the same logic function. Such a choice underplays the role of emitter-coupled logic (such as Motorola's MECL

10,000 logic family) for extremely fast designs. It also underplays the role of CMOS logic for slow, low-power-consumption designs. However, concentration on one family of logic offers the compensating advantage of introducing familiarity with a broad range of real devices which can become building blocks for the reader in his or her designs.

Sometime during 1980 or 1981 a new logic symbol standard will be introduced by the International Electrotechnical Commission as IEC Publication 617-12 and by the Institute of Electrical and Electronics Engineers as a revised IEEE Std 91/ANSI Y32.14. The *dependency notation* employed in this symbol standard will fill a long-standing need for a way to represent MSI devices like decoders, multiplexers, and counters so that the symbol, by itself, describes the operation of the device. It will go a long way toward clarifying the circuit schematics for an instrument. In this book, we introduce this symbology little by little so that the reader can build up a repertoire of notation ideas without being overwhelmed in the process. By the end of Chapter 2 we will have developed symbols for many widely used devices. For further support, the reader is referred to Appendix A1 which has been prepared by F. A. Mann of Texas Instruments Inc. to describe the main ideas of dependency notation. The reader is also referred to the 1980 printing of Texas Instruments' *TTL Data Book for Design Engineers* (third edition) which includes the dependency notation symbol for every one of that manufacturer's TTL devices.

This book is directed toward a specific goal of engineering studies—the development of creative design capability. Digital design offers a superb opportunity to develop this capability under rather ideal conditions: the specifications of system performance can be made both real and unambiguous; the building blocks to be employed can be easily delineated (e.g., use 74LSxxx parts); and the validity of a simple design criterion permits the student to carry out design while subject to a specific, real measure of the quality of a design. To take advantage of this opportunity, most chapters close with a broad variety of problems having a design flavor.

The book will typically be used in a one-quarter or one-semester course on digital hardware design. As a senior-level text, the book can build upon earlier digital hardware studies and extend these into a strong design framework. As a junior-level text, the book can serve as an introduction to digital hardware studies, leading the student to a solid overview of the implications of digital design in today's environment.

If this book is used by students who are already familiar with logic ideas and logic devices, then Chapter 2 might be scanned quickly to pick up dependency notation ideas. Also, Section 2-12 might be scanned for familiarization with PROMs, FPLAs, and PALs. Chapter 3

lays the groundwork for maintainability ideas which recur throughout the book. Chapter 4 presents a powerful approach for describing and designing the controller portion of fast circuitry. At the same time, it introduces the signature analyzer instrument design which is carried on in Sections 5-4 and 6-1.

One way to use this book might be to accentuate the design thread provided by the signature analyzer instrument design example. For such use, the dependency notation ideas of Chapter 2 might lead quickly to a study of Chapters 3 and 4, followed by Sections 5-1 through 5-4 and 6-1. The studies might then be filled out by selecting from among the remaining sections of Chapters 5 and 6. Such a study might be augmented by obtaining the service manual for an instrument and considering its redesign in light of the maintainability ideas presented in the text. A laboratory accentuating the development of these ideas might employ a personal computer (like that of Fig. 1-18) to fill the role of the microcomputer so that software development can be handled with simplicity and flexibility.

I am grateful for the one-fourth of my life during the past eight years spent working with outstanding design engineers within Hewlett-Packard. Most recently, working with Paul Lingane and Allen Edwards of the Stanford Park Division gave me an appreciation for the signature analysis concept developed and promoted by Hans Nadig, Bob Frohwerk, Tony Chan, Gary Gordon, and Ed White of the Santa Clara Division. For an appreciation of the impact which programmable chips can have in minimizing instrument design cost, I am grateful to Charles Muench of Intelligent Systems Corp. I am indebted to my students at Georgia Tech who have served as mentors through a variety of design activities.

In another vein, I am grateful to Dr. Demetrius Paris for fostering my activities at Georgia Tech and for having the vision to promote my participation in IEEE SCC11.9, the "dependency notation group." Conrad Muller, Fred Mann, and John Russell have each been a vivid example to me of the impact which a dedicated individual can have upon an industry standard. I am grateful to Ed Donn for his insights, his friendship, and the impact of his life upon my life. Eddie planned Chapter 1. The influence of his thoughts pervade Chapter 3. Finally, I fondly acknowledge the many roles my wife, Marilyn, played in the preparation of this book, including the editing and typing of the manuscript.

John B. Peatman

CONTENTS

PREFACE	vii
1 DESIGNING FOR USEFULNESS	1
1-1 A technological revolution	1
1-2 Toys versus tools	7
2 PRELIMINARY CONSIDERATIONS	21
2-1 Codes	22
2-2 Logic functions	29
2-3 Logical problem formulation	33
2-4 Voltage levels and noise immunity	36
2-5 Graphic representation of logic	37
2-6 TTL gates	42
2-7 Implementing small functions	45
2-8 Loading rules	49
2-9 Dependency notation: Decoders	51
2-10 Dependency notation: Multiplexers	58
2-11 Open-collector and three-state outputs	61
2-12 Programmable combinational circuits	72
2-13 Synchronous sequential circuitry	101
2-14 Synchronous building blocks	114
2-15 The microcomputer as a component	123

3	DESIGNING FOR MAINTAINABILITY	140
3-1	Self-test, push-to-test, and fault isolation	140
3-2	Hardware organization	147
3-3	A design criterion	153
3-4	Information compression	156
3-5	Signature analysis	162
3-6	Design accommodations for maintainability	172
3-7	One approach to maintainable sequential circuit design	186
3-8	Design accommodations for ATE	196
4	ALGORITHMIC STATE MACHINES	212
4-1	Design of a signature analyzer	214
4-2	ASM building blocks	221
4-3	ASM charts	223
4-4	Alternative implementations	228
5	CIRCUITRY WITH A SINGLE CLOCK SOURCE	246
5-1	Timing considerations of logic	246
5-2	Maximum clock rate: Synchronous design	260
5-3	Avoiding race conditions	269
5-4	Buffering of synchronized data	271
5-5	Multiple-phase clocking	292
5-6	Maximum allowable clock skew	302
5-7	Pseudosynchronous clocking	303
6	SEPARATELY CLOCKED CIRCUITS	314
6-1	Switching between clock sources	314
6-2	Communication via RAM	319
6-3	Handshaking	324
6-4	Serial data transmission	325
6-5	Interconnected master-slave "smart" subsystems	331
6-6	Buffering data transfer rates	346
6-7	One-shots	362
	APPENDIXES	373
A1	LOGIC SYMBOLOGY AND DEPENDENCY NOTATION by Frederic A. Mann	373
A2	SYSTEM IMPLEMENTATION	399
A3	MOTOROLA 6801/68701 ONE-CHIP MICROCOMPUTER	410
	INDEX	429

1

DESIGNING FOR USEFULNESS

1-1 A TECHNOLOGICAL REVOLUTION

“Once upon a time” the applications of digital technology were constrained by the cost, size, and heat dissipation of the technology itself. An early vacuum-tube computer filled a large room. Its air conditioner filled the adjacent room. As shown in Fig. 1-1, product planning meant room filling.

Today, product planning is entirely reoriented. In contrast to being driven by the abilities of technology, today’s electronics world is driven by the needs of markets. This world says “if you can define it, you can design it,” quickly followed by “but let’s make sure somebody wants it.”

In a 1979 address, Dr. C. Lester Hogan, vice-chairman of Fairchild, cast light upon our present state by reflecting back to the 1950s. At that time the five leading electronic-device manufacturers were

RCA, Sylvania, Raytheon, Westinghouse, Amperex

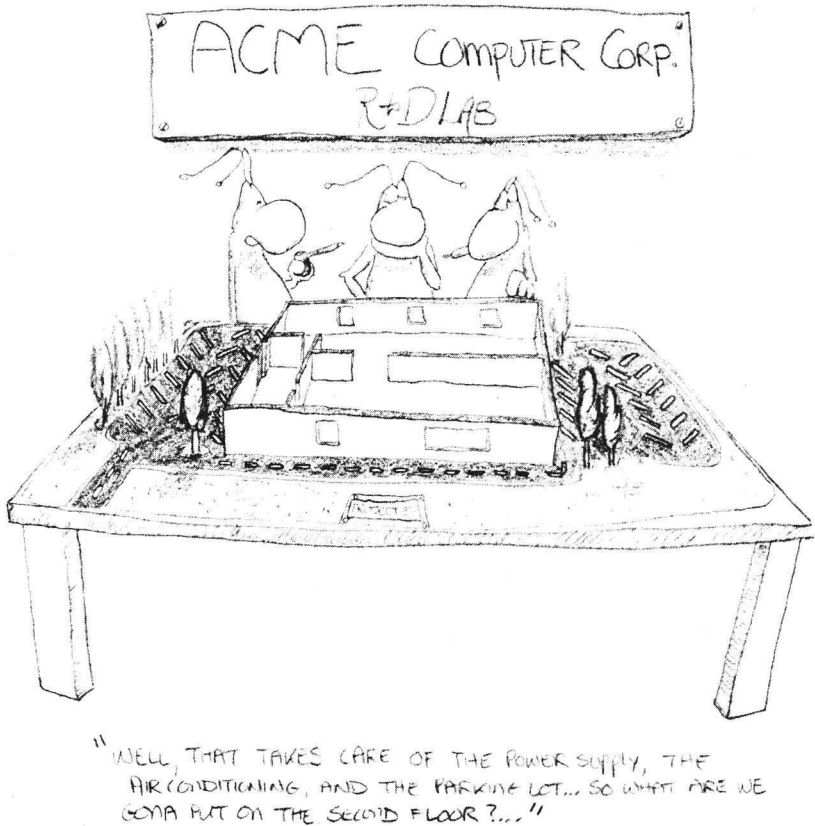


Figure 1-1. A 1950s product planning meeting. (Rand Renfroe.)

Since then the revolution in electronics technology has changed this list to

Texas Instruments, Motorola, Fairchild, National Semiconductor, Intel

We are just entering another period of technological revolution. Soon, integrated circuit (IC) manufacturers will be able to put a million components on a single chip. It is not clear what we should be doing with all of this capability. However, it seems apparent that each of the following IC developments will create significant demand:

- 1 The development of ever more "powerful" microcomputers, such as the Motorola 6801 one-chip microcomputer shown in Fig. 1-2 and discussed in Appendix A3.

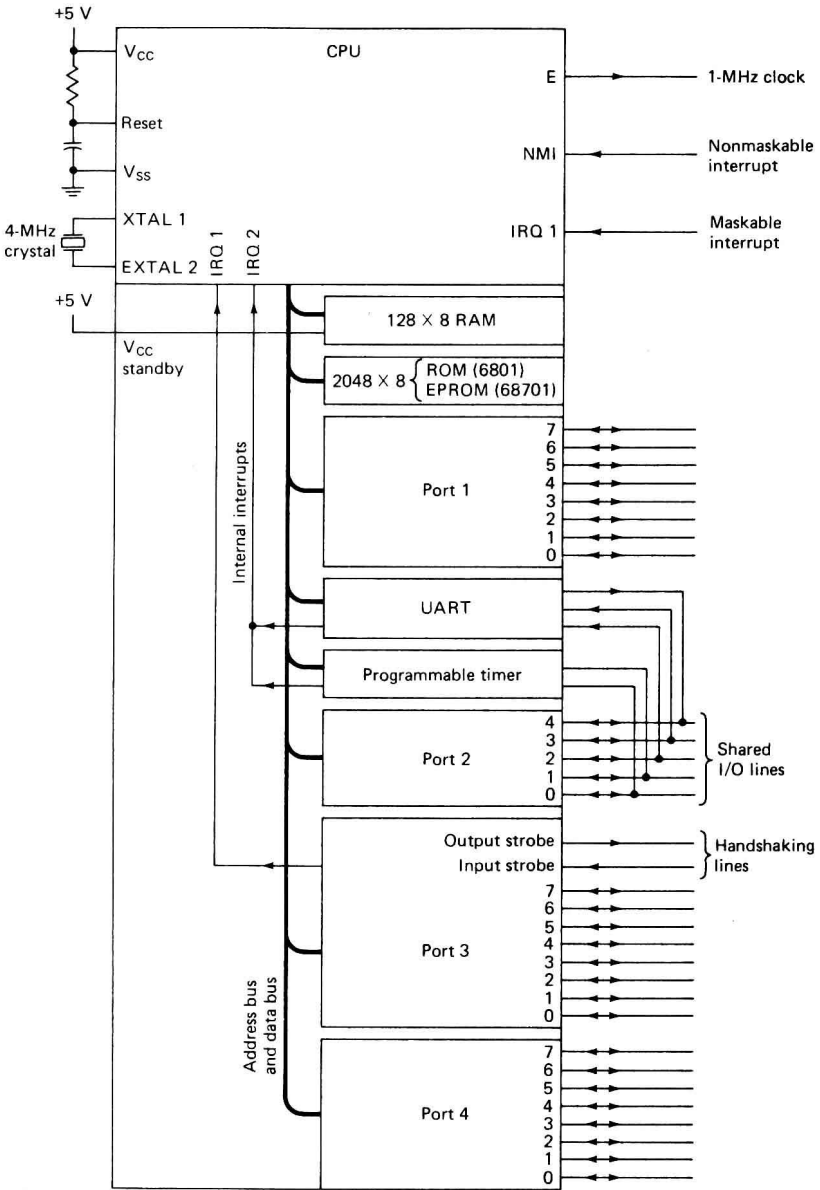


Figure 1-2. Motorola 6801 one-chip microcomputer. (Motorola, Inc.)

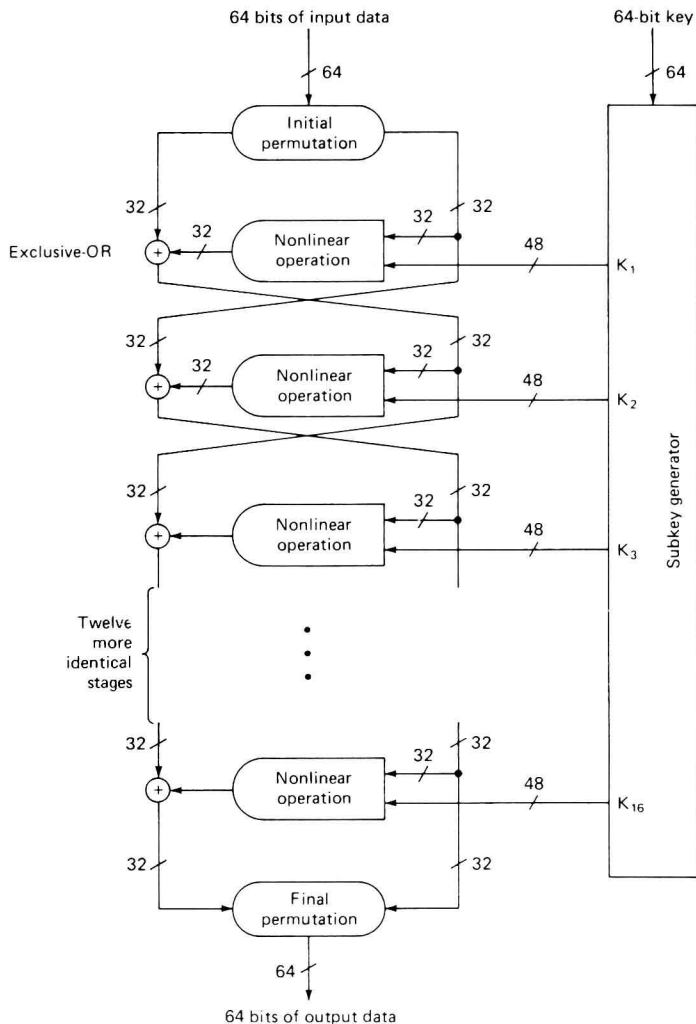


Figure 1-3. Data encryption using the National Bureau of Standards' Data Encryption Standard, FIPS PUB 46.

- 2 The transfer of any significant part of the microcomputer programming burden from designers to on-chip, "canned" software. For example, the microcomputer of Fig. 1-2 can be expanded into a multichip microcomputer configuration. It might include a "real-time operating system" developed by Motorola and supplied within the 6801 chip itself. The rest of the software would

be developed separately by an instrument† designer and reside in a separate memory chip.

- 3 The “casting into silicon” of pervasive algorithmic processes. The National Bureau of Standards’ “Data Encryption Standard” shown in block diagram form in Fig. 1-3 is one example that has already happened. It supports data security measures for the commercial sector of our society.
- 4 The inclusion within an integrated circuit of the ability to be queried by another device as to its “health,” to test itself, and to respond that it is either working or malfunctioning. “Smart” peripheral controller chips organized around a one-chip microcomputer can include this capability via extra on-chip software incorporated solely for this purpose. Alternatively, chips developed to implement specific algorithms can include the hardware organization, together with on-board testing algorithms, to execute this function.
- 5 Standardization efforts which permit designers to develop custom ICs without first acquiring in-depth knowledge of IC technology. Programmable read-only memories (PROMs) were a first step in this direction, followed by field-programmable logic arrays (FPLAs), discussed in Chap. 2. A more recent opportunity has been the development of the *master-slice* approach to custom IC design. Here an instrument designer generates an interconnection mask for an array of standard devices within a chip, such as that of Fig. 1-4. With this approach, an IC manufacturer can define most of the critical parameters of the chip, leaving the interconnects to the user. In this way a user with limited exposure to semiconductor technology can produce a specialty IC at relatively low cost.

The list above projects opportunities for dramatic growth at the *integrated circuit* level. Moving to a higher level of organization, we can identify further opportunities for dramatic growth:

- 1 The identification of smart, replaceable *modules* that can be queried and then test themselves and respond. These modules can include a one-chip microcomputer on a printed circuit (PC) board, a PC board in an instrument, an instrument in an automatic test system, or the automatic test system itself. For example, the instruments shown in Fig. 1-5 are interconnected using the GPIB, the general-purpose interface bus defined by IEEE Std 488-1978.

† We will use the term *instrument* in a general sense to mean any self-contained, stand-alone device, not a rack full of equipment.

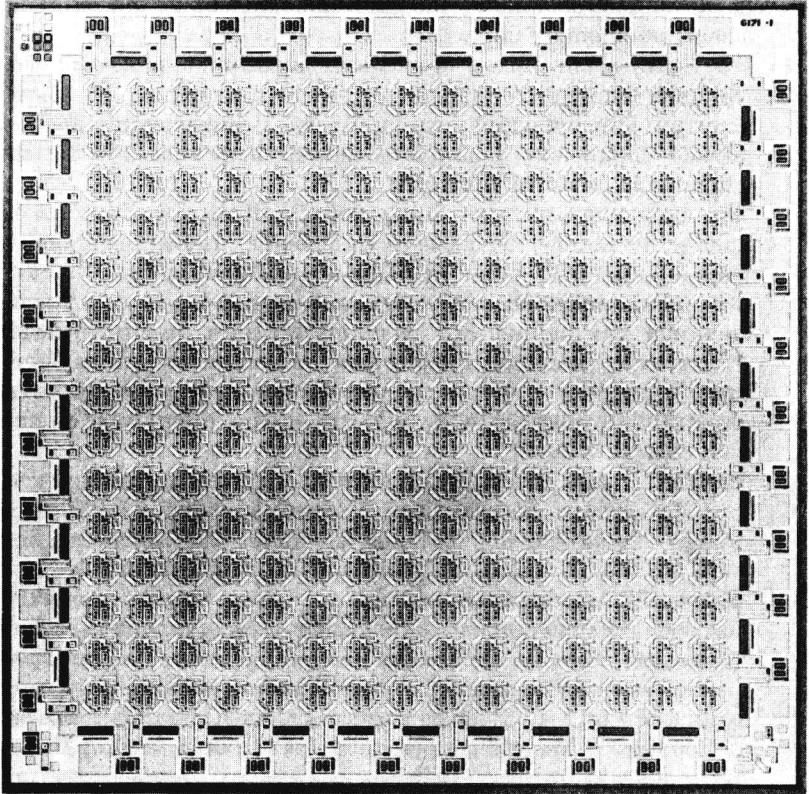


Figure 1-4. Master-slice chip for custom integrated circuits. (Interdesign Inc.)

The programmable controller on the bus cannot only test itself, it can send commands to all instruments on the bus to test themselves and then report back the results. The microcomputer controller within an instrument can likewise test itself and then command any “smart” subsystems (controlled by separate microcomputers) to test themselves and to report back. In this manner, each level in the hierarchy can obtain the self-test information it needs in order to pass along a status report on its health to the next-higher level in the hierarchy.

- 2 The standardization efforts supporting “concatenation of complexity,” illustrated in Fig. 1-6. We use this term to identify different levels of activity, within any one of which a meaningful contribution can be made, but without requiring strong interaction with the levels just above or below.

The general-purpose interface bus (GPIB) serves as a broad

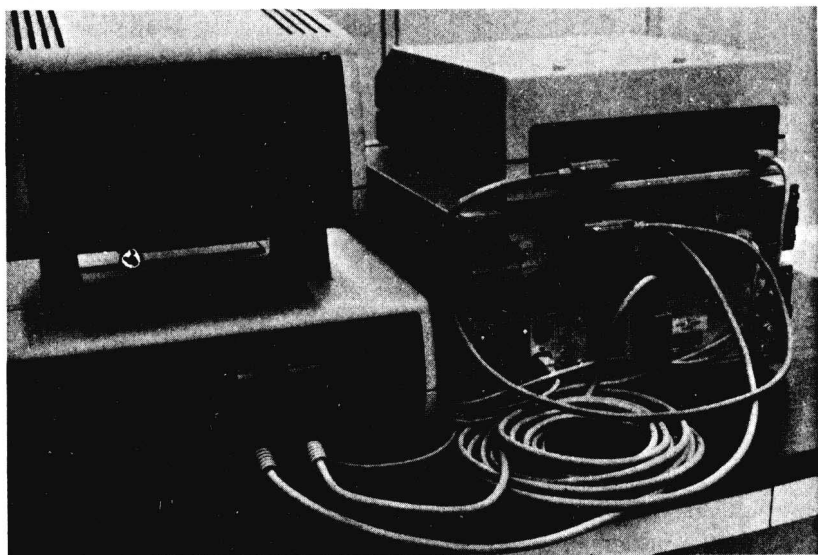


Figure 1-5. Instruments interconnected on the GPIB, general-purpose interface bus. (Hewlett-Packard Co.)

example of the ability of standardization to create markets. For example, the sales of test and measuring instruments by Hewlett-Packard for the first half of 1979 showed a 40 percent increase over the corresponding sales during the first half of 1978. John Young, the president of Hewlett-Packard, has reported that one-third of these sales were related in one way or another to the bus.

- 3 The design of new products which iterate upon the opportunities made possible by other recently developed products. Again to cite the example of Hewlett-Packard, John Young has reported that over half of their sales can be attributed to products introduced within the past three years. New product development plays a crucial role in the growth (or decline) of a company's activities. It is probably also true that a company's creative efforts are being counterproductively spent when it tries to create future growth through the protection of past accomplishments (via secrecy and litigation). In a rapidly evolving technological world, winning demands continuous creativity.

1-2 TOYS VERSUS TOOLS

A toy is an interesting collection of features, but it does not really do much for you. A tool is a helper. It may even help in ways you did not realize you would need.