microprocessor systems design

Edwin E. Klingman,

microprocessor systems design

Edwin E. Klingman,
President
Cybernetic Micro Systems

Library of Congress Cataloging in Publication

KLINGMAN, E E
Microprocessor systems design.

Bibliography: p.
Includes index.
1. Microprocessors—Design and construction.
I. Title.
TK7885.K544 621.3819'52 76-45190
ISBN 0-13-581413-8

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

This book is not concerned with designing toys or mere teaching models, but is directed toward the design of powerful computer systems which are orders of magnitude better than the best that major universities could afford little more than a dozen years ago. That a serious student can understand the design and application of such a system today—and then build his/her own system—would not even have been good science fiction a few years ago.

The book is intended to serve both as an introduction to computer systems in general and to the design of systems based on microprocessors in particular. The presentation is self-contained. All relevant techniques are developed, and their range of application is described. The emphasis is evenly divided between the concepts and the state-of-the-art devices with which one may fashion a computer. The design of a general purpose computer is used as a vehicle for developing all of the basic concepts required for understanding the spectrum of microprocessors available today. The choice of processor was based on the following criteria: 1) a simple basic instruction set and 2) a control structure that embodies most of the features that are employed in processors. The second criterion argues against the highly integrated second and third generation families of devices that have been designed with compatible timing. Although the trend toward integrated sets of devices will surely continue, it is absolutely necessary that the digital systems designer be capable of interfacing any of the thousands of devices which are available, and this requires an understanding of timing and control structures.

Processors of this type include the Intel 8008, RCA COSMAC, General Instruments CP-1600, Intersil IM6100, Mostek MK5065P, National PACE, and others. Of these, the Intel 8008 possesses a very simple instruction set of 48 instructions which is a subset of the 8080/8085 microprocessors's 78 instructions and the Zilog Z-80's 158 instructions. The coded state signals and multiplexed bus structure allow development of the necessary timing/control concepts required for most processor design. Historically, as the first eight bit central processor unit, it has influenced the design of several later processors.

The book is organized in three sections as follows: The first section, consisting of Chapters 1–4, treats the basic concepts of digital building blocks and presents typical examples of such. The second section, Chapters 5–8, designs a computer based around the 8008, developing computer concepts as needed. The final section, Chapters 9–14, treats specific topics in terms of the spectrum of microprocessors; i.e., concepts are developed and their implementations in various microprocessors are presented. Such topics include software structures, addressing structures, Input/Output structures, Interrupt structures, Direct Memory Access structures, and architectural structures. In addition to all relevant concepts, this portion of the book is rich in detail and the design data required for using these processors.

The philosophy of design will consist of defining a goal and then concentrating upon the simplest means of realizing the goal. The economic premise underlying this philosophy is that the typical reader is assumed to be interested in understanding, acquiring, and applying a general purpose computer in a specific problem area. Each application is therefore considered to be one of a kind. Thus the cost function to be minimized is design time in contrast to the commercial situation in which unit cost is to be minimized at the expense of design time, thereby maximizing the profit over a large number of systems.

This philosophy is felt to be realistic in the assessment of the needs of a typical reader and also to provide the most sensible approach to a very complex subject. This objective will be met primarily by choice of the most convenient and/or simplest building blocks. In this vein a choice between random access memory blocks, for example, would be made in favor of the simple static RAM rather than the slightly cheaper but much more complex dynamic RAM if both RAMs satisfy the requirements of the design.

The treatment of software is consistent with the assumed 'applications oriented' bias of the reader. The most powerful tool for applying computers in most environments is the *interpreter*, and a simple, expandable interpreter is developed which is capable of serving as the basis for a number of applications. In addition, the concepts concerning assemblers, macro-assemblers, and conditional assembly are developed in Chapter 9 and in the appendices.

Although written primarily for computer scientists, electronic engineers, and physicists, the book is designed to be useful in many disciplines. Information processors are multi- and inter-disciplinary and, as such, are of interest to growing numbers of people. Once the principles of digital building blocks have been grasped, they should prove as useful to an experimental psychologist as to an experimental physicist.

Digital design is obviously of use to engineers who need to apply computer based systems to specific problems, particularly the monitor/control problems of transportation, factory automation, or communication systems and to the physicists, chemists, and biologists who need to automate their laboratories for tight control of experimental parameters and programmed data acquisition and analysis. Others who may wish to utilize digital building blocks include a growing number of artists who appreciate the dynamic medium afforded by the computer, particularly in conjunction with, but not limited to, video displays. As mentioned above, the experimental psychologist should welcome the computer to his laboratory, and a large number of nonexperimental psychologists today feel that an understanding of the computer is a valuable asset to those who think about thinking. Architects who wish to design adaptive and interactive living environments will find the necessary tools described herein. Selected portions of this text can form a course for business oriented students who need to understand the operation and effect of automation in sufficient detail to plan for, rather than just respond to, events.

The unique feature of this book lies in its *complete* treatment of real, nontrivial, digital systems made possible by the emergence of Large Scale Integrated (LSI) microprocessors. By treating microprocessors at the system level and treating subsystems at the component level, it becomes possible to cover entire systems in one book and to place the extremely detailed treatment of particular systems in a conceptual context of sufficient scope to apply to both current and future systems.

In order to gain some appreciation of microprocessors and the rate at which technological change is occurring today, we may contrast typical microprocessor characteristics with the first major processor that I had the opportunity to work on: an IBM 650. In 1960 this system leased for about $10,000 per month. If we consider only parts cost, a microprocessor-based system will today range from $50 to $1000. Although these processors differ vastly in architecture and implementation (vacuum tubes versus LSI) it is possible to draw a rough comparison between them.

The IBM 650 had 60 instructions; most microprocessors possess 70 to 100. The 650 possessed a 20-millisecond instruction execution time; today's microprocessors possess two micro-second instruction execution times—ten thousand times faster! Although the power requirements of the 650 are unknown, it did require heavy-duty air conditioning facilities, whereas microprocessors use only a few watts and are roughly one hundred times smaller in volume. The 650 used 2000 words of drum memory; most microprocessors address 65,000 words of semiconductor memory.

Although this book covers state-of-the-art digital design and in places predicts some developments to be expected in the next few years, it is impossible to predict very far downstream. A thorough coverage of the hardware available today and the trends in hardware is presented, and concepts that are unlikely to change are expounded. But rather than list innumerable applications of computer systems, which are already well-documented elsewhere, the stress here is on the capability of such systems. Each reader is assumed to bring with him his own applications requirements.

Chapter One presents an overview of information processing and digital design in an attempt to shed some light on the significance of the transformations occurring in the field of computer science.

Chapter Two introduces binary arithmetic, Boolean algebra, logic gates, and Venn diagrams. The TTL family characteristics are presented and discussed. Combinatorial logic is described and examples are presented. A brief summary of the building blocks to be used throughout the book concludes Chapter Two.

Chapter Three introduces flip-flop storage elements. Clocked flip-flops are discussed; and the R-S, T, D, J-K Master-Slave flip-flops are presented in detail. The development of memory cells and their organization into arrays is covered. Shift registers are derived and discussed.

Chapter Four deals with coding and MSI and LSI building blocks. The space/time concept of coding is developed in detail. A representative digital building block from each class of building blocks is treated. This chapter is essential to any course in computer design, as these are the blocks from which the system will be built. The final sections describe the interfacing between families of blocks.

Chapter Five treats the instruction set of the CPU. The internal architecture of the CPU is analyzed in detail as preparation for a meaningful treatment of the instructions. Examples are developed in the form of short programs both in the machine language and assembly language.

Chapter Six treats the state transition and timing for a central processor unit. This material is balanced between the general concepts that are applicable to most processors and the specific details that are necessary for design work. Emphasis is placed on time/space multiplexing and on the need for, and interpretation of, status information. State timing and the state transition diagram are covered.

Chapter Seven begins the hardware design of a computer system. The circuitry that effects information transfers between the CPU and memory is designed. Particular attention is paid to an analysis of timing. Often several alternatives are considered before arriving at a final design.

Chapter Eight analyzes the I/O instructions and develops the I/O port selection circuitry.

Chapter Nine traces the use of symbols from the elemental hardware level through macro-assemblers and interpreters.

Chapter Ten treats the addressing structures that are associated with microprocessors. The first half of the chapter provides detailed coverage of the address presentation mechanisms currently in use while the second half covers address formation mechanisms now used in microprocessors. The duality of procedure/data structure is discussed and then used as a framework for the introduction of computed addressing techniques.

Chapter Eleven provides an extensive coverage of Input/Output structures and techniques. Examples using Teletypes and analog-to-digital converters attempt to balance hardware, software, and timing concepts. The status-checking techniques lead quite naturally to interrupt structures which are treated in the following chapter.

Chapter Twelve presents interrupt structures as an event-driven symbol replacement technique. The wide range of interrupt structures and techniques that are available with microprocessors is given detailed treatment. A single level interrupt system is expanded to a general multi-level interrupt system, and the software that is required is developed in detail.

Chapter Thirteen treats direct memory access structures and techniques and covers microprocessor DMA subsystem design, both integral and additive.

In Chapter Fourteen, treating architectural themes, the concepts of bundled and orthogonal structures are defined and used to place current architectures in perspective, as well as to predict future trends. This chapter ends with a brief introduction to multi-processing concepts as applied in microprocessor architectures.

For three years prior to publication of this text, I taught survey courses and seminars on microprocessors, consulted on microprocessor developments, and designed commercial systems. These activities brought me into contact with well over 1,000 engineers and programmers working in every imaginable application area. Although it is impractical to list all of these people by name, they have together contributed far more to this book than any individual. Yet there are specific individuals who deserve mention: Dr. Robert Naumann, who supported my minicomputer and microprocessor work at NASA. Drs. Ron McNutt and Bill Short, who arranged for me to teach a microprocessor course at Athens College in late 1972. Dr. John Peatman at Georgia Tech, who encouraged this undertaking at a critical time. Jerry L. Ogdin, Dr. Roger Camp, John Harris, and Scott McPhillips, with whom I conducted some of the first seminars, have influenced my thinking on microprocessors. I have had helpful conversations with Tampy Thomas at Intersil and Hash Patel at National. The support given me by Intel Corporation has been tremendous, and I am indebted to many Intel employees, particularly Juris Brempelis, Dane Elliot, Ken McKenzie, Hal Feeney, Jim Lally, and Dr. Bill Davidow. I am also grateful for enlightening and enjoyable conversations with Chuck Peddle and Will Mathys of MOS Technology, Inc. Dr. John Gault of North Carolina State has been most helpful in reviewing the manuscript and suggesting a number of changes in organization. I feel that the changes have improved the presentation and I appreciate his efforts. Finally I wish to thank my production editor, Margaret McAbee, for her gracious assistance in producing this book.

EDWIN E. KLINGMAN

contents

11 *I/O structures and techniques* 282

12 *interrupt structures* 339

13 *direct memory access: structures and techniques* 390

1 microprocessor systems design: an overview

"A science of design. . . would depend upon the relative simplicity of the interface as its primary source of abstraction and generality."

Herbert Simon
The Sciences of the Artificial,
The MIT Press, 1969

The utilitarian value of technology lies in the effectiveness of tools and techniques in accomplishing desired goals. The most effective and widely applicable technology today is that of micro-electronics, in which the ability to fabricate integrated circuits on a dielectric base has culminated in the monolithic microprocessor, or "computer on a chip." Our goal is to develop an understanding of and a facility with these integrated circuit "building blocks" that are revolutionizing the field of digital design. Our vehicle will be the design of a general-purpose digital computer that uses a basic Large-Scale Integrated (LSI) circuit central processor unit. This system will allow the introduction of both hardware and software structural features that enhance the power of an information processor. Emphasis will be placed on the design of real-time interactive systems.

ADVANTAGES OF MICROPROCESSORS

Unquestionably the major advantage of microprocessors is that of cost. They are now inexpensive enough for thousands of applications, and prices are dropping as their capabilities increase.

In applications limited to pure number crunching associated with large-scale scientific calculations, and those of file management associated with large-scale data bases, the information processors must of necessity be large scale. This is primarily due to the fact that long word lengths and high execution speeds facilitate long calculations and rapid data searches. These factors are of lesser importance for smaller calculations and data banks, and even less so in monitor/control applications. They will become still less important as the word lengths and speeds of microprocessors increase.

A distinct advantage of microprocessors over large processors is that, whereas large processors are shared by many people and tasks, microprocessors are dedicated to one person and/or one task. The attendant decrease in operating system complexity is enormous. The software required to keep track of multi-user environments presents huge overhead costs on top of rather formidable hardware costs. These costs are absent in dedicated microprocessor systems. The prefix *micro-* used in the preceding discussion should be interpreted in terms of size and cost rather than capability. The microprocessors of today are superior in most aspects to the major processors of a dozen or more years ago. The decrease in size and cost of processors by several orders of magnitude has correspondingly increased the range of practical applications of processors by several orders of magnitude.

ESSENTIAL FUNCTIONS OF PROCESSORS

Before proceeding to discuss the design of processor-based systems, it is appropriate to define a processor in terms of its function. The essential function of a processor can be specified by five operations:

1. READ the input symbol x.
2. COMPARE x with z, the internal state of the processor.
3. WRITE the appropriate output symbol y.
4. CHANGE the internal state z to the new state z'.
5. REPEAT the above sequence with a new input symbol x'.

The power inherent in this set of operations is well known to those familiar with automata theory. Heinz von Foerster has given an exceptional introduction to the processor for physicists by pointing out that Maxwell's Demon can be simply described by five equivalent operations. Maxwell's Demon was the mechanism postulated by James Clerk Maxwell to cause heat to flow from a cold container to a hot container in apparent violation of the Second Law of Thermodynamics. This was to be achieved by positioning a "demon" or mechanism between the two containers to channel selectively the most energetic or "hottest" molecules from the cold gas into the hot container. The operations performed by the Demon are:

1. READ the velocity, v, of the approaching molecule with mass m.
2. COMPARE the energy $(mv^2/2)$ with the mean energy $\langle mv^2/2 \rangle$ (or temperature T) of the cooler container (intend state T).

3. OUTPUT this molecule (to the hotter container) if $(mv^2/2)$ is greater than $\langle mv^2/2 \rangle$; otherwise close the channel.
4. CHANGE the internal state T to the new (cooler) state T'.
5. Repeat the above sequence for a new approaching molecule m'.

Maxwell's Demon was designed a century ago, long before the advent of quantum mechanics and information theory. The concept of entropy as a measure of disorder in a system is today central to many fields, and the ability to increase order *locally* at the expense of energy is widely recognized. Thus, the microprocessor, operating from a power supply or energy source, can retard the increase of entropy in a local system to an arbitrarily slow rate. This remarkable ability characterizing the information processor can be used to optimize or increase the efficiency of almost any imaginable process.

The primary problem facing a world of finite *resource* and finite *sinking* capabilities is that of waste. The pollution that accompanies all life was, in the prehuman world, recycled naturally. Some human societies such as the Hopi Indians, having lived on the same plateau for approximately 5000 years, also maintained an ecologically efficient relationship. The invention of agriculture, the city, and industrial civilization, combined with unnatural population densities, have aggravated and emphasized the problem of waste. It may be in this area that the information processor, in its capacity as an entropy retarder, will be of most value in the immediate future.

The ecological applications of information processors to retard the growth of entropy (i.e., minimize waste) range from the system simulations of Jay Forrester et. al. at MIT to process regulation systems designed to increase efficiency. Increased efficiency in any process equates to less waste. There can be little doubt that as energy costs rise and processing costs decline the applications of microprocessors to increase efficiency will continue to grow.

The concept of increasing entropy, like most of the concepts of physics, applies to closed systems. In an open system it is possible to decrease entropy or hold it constant. The application of processors in adaptive or learning processes brings about a decrease in entropy by removing redundancy in a system. Learning processors or pattern recognition systems remove redundancy by disclosing structure in a system. The development of learning systems must unquestionably be considered a process of endowing our environment with intellegence. That it is currently a low order of intelligence is of little consequence. We will not pursue this theme as the vast possibilities that open up are mind-boggling, but subject to considerable controversy. We, therefore, limit our discussion to increases in efficiency of processes.

We hesitate to be specific about microprocessor applications simply because the list is endless. We prefer rather to consider two categories in which efficiencies are achievable: physical and social. Under physical we include efficiencies in transportation, manufacturing, and such areas as physics, chemistry, and biology laboratories, as well as medical applications. The social category consists of communications efficiencies achievable in education, government, management, and entertainment. Aside from the obvious computer/TV cable linkups and other purely communications-oriented applications, there is a less well-publicized aspect known to every person lucky enough to have unrestricted access to a modern minicomputer: the computer is beyond doubt THE SUPERTOY and is capable of providing as such a new, expressive medium for creative drive. It is entirely plausible that of all the uses to which information processors may be put, this—in the long run—may be the most important to mankind.

THE NEW DUALISM

The preceding discussion introduces the idea of inseparable opposites, growth/decay, referred to as a dualism or dualistic pair. The use of the concept *recycle* in this discussion illustrates the most important aspect of this dualism, i.e., the dualistic opposites are mutually transformable. The products of decay contribute to the growth process, which in turn maintains the decay process.

This idea of opposites that *define* each other, i.e., have no meaning alone, and yet are mutually transformable, we refer to as *dualism* or dualistic process. The relation between dualistic opposites is one of complementarity rather than negation.

There seems to be an increasing awareness of the complementarity in dualistic descriptions of the world: on/off, up/down, light/dark, hot/cold, happy/sad, and on and on. The realization that these inseparable opposites *define* each other has come with an increased awareness of, and interest in, Eastern consciousness and understanding of Western science.

Einstein's Dualisms

The fundamental dualism of the twentieth century has been that symbolized by $E = mc^2$, i.e., the matter/energy dualism. The unexpected discovery that matter could be transformed into energy, and that energy could be transformed into matter, has changed both practical and philosophical aspects of our life.

Yet another dualistic transformation of categories once believed to be mutually exclusive was conceived and formalized by Albert Einstein. The relativistic transformation in which the space/time dualism appears relates spatial to temporal quantities, and vice versa, through a four-dimensional equation. This transformation has been experimentally confirmed and practically applied by scientists in several fields of physics.

A less well-publicized complementarity is that symbolized in the Schroedinger equation of quantum physics. Reconciling deterministic world views with nondeterministic, this famous equation rigorously and deterministically states the laws governing the quantum mechanical wave function while the wave function itself is *probabilistic*. i.e., indeterministic. A few practical consequences of this interpretation are found in atomic energy, the laser, and the integrated circuits of solid-state physical electronics. Micro-electronic technology is a direct consequence of the probabilistic interpretation of the wave function determined by Schroedinger's equation. The philosophical consequences of this interpretation may be no less significant, if less recognized.

An even more radical dualistic transformation is coming to be understood and is every bit as earthshaking as the fundamental dualisms of the twentieth century: space/time, probabilistic/deterministic, matter/energy, particle/wave. The *hardware/software* dualism will have much more profound effects, both practical and philosophical, than these. This dualism can be stated as:

> *Any software process can be transformed into an equivalent hardware process, and any hardware process can be transformed into an equivalent software process.*

The nature of this dualism is at the very root of PROCESS and is the defining relationship for cybernetic processes. Neither hardware nor software has independent existence; therefore, the transformation can never be complete. There must always be software to direct hardware processes, and there must always be hardware to process software algorithms. This statement is experientially evident. We accept it as a metaphysical *given*. The analogous requirement in the matter/energy dualism states that there is a relativistic mass associated with every field, and that the energy/matter transformation can occur only in the presence of a momentum-conserving mass field.

The short history of the hardware/software trade-off has been primarily an economic trade-off. In this respect, the governing rule was: hardware is fast, software is cheap; so we accepted the simple, slow, predominantly software systems. The evolution occurring for the past quarter-century has seen the economic ground rules change. Before proceeding with the discussion of this change, it is well to define our terms more closely.

Algorithm: An algorithm, or algorithmic activity, is a procedure for carrying out a task.

Hardware: A relatively permanent physical embodiment of an algorithmic process.

Software: A relatively impermanent, informational embodiment of an algorithm.

Information: The entity that makes the difference between knowing and not knowing that which makes possible a decision.