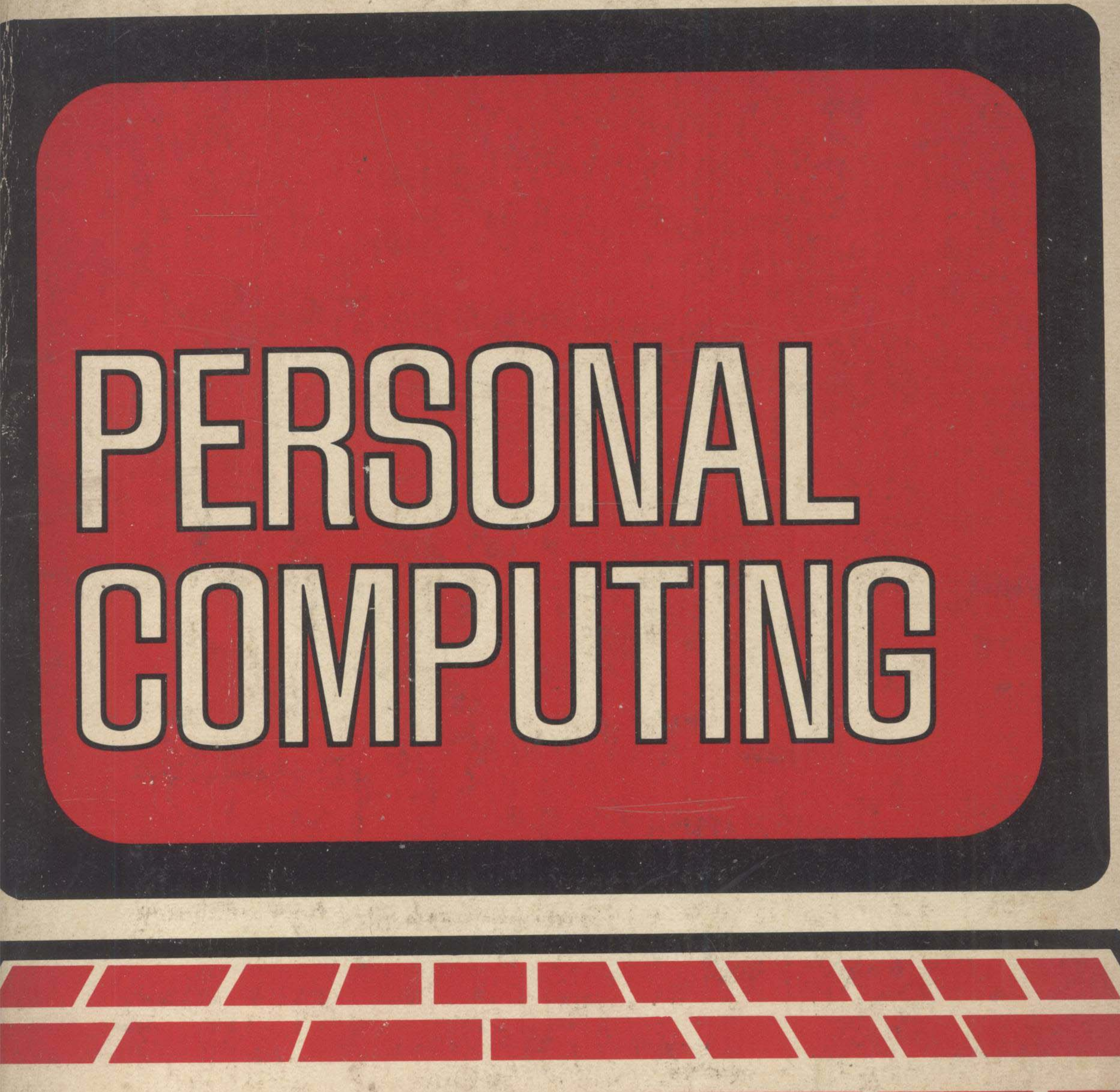
A REWARD BOOK

$10.95
($12.50 in Canada)



PERSONAL COMPUTING

im Huffman

Build your own computer!
This down-to-earth guide shows you how--and how to use it, too!

Jim Huffman

PERSONAL COMPUTING

RESTON PUBLISHING COMPANY, INC.
A Prentice-Hall Company
Reston, Virginia

Library of Congress Cataloging in Publication Data

HUFFMAN, JIM.
Personal computing.

Bibliography: p.
Includes index.
1. Microcomputers. 2. Microprocessors. I. Title.
QA76.5.H773 001.6'4'04 79-722
ISBN 0-8359-5516-8
ISBN 0-8359-5515-X pbk.

A Prentice-Hall Company
Reston, Virginia 22090

10 9 8 7 6 5 4 3 2 1

PRINTED IN THE UNITED STATES OF AMERICA

PERSONAL COMPUTING

Preface

The purpose of this book is to introduce you to the world of personal computers. If you are already somewhat familiar with these devices, this text will provide some of the background and the details you might have missed. If you are uncertain as to how these fascinating creations operate, you can use this book to fill in the details. This is intended to be a near universal text. However, things are changing so rapidly in the personal computer field that it is hard to cover all the bases. We cover personal computers in terms as general as possible; then, by bringing in specific terms, we are able to apply the general information. In fact, in this text you will not only receive general and applied theory, but you will get the plans for a complete working microcomputer as well. This computer can be built for very low cost and is used as a type of demonstrator to show direct applications of the theories covered in the text.

Of course, you really don't have to build the computer given in this book. You may already have a system and merely need some further enlightenment on its operation. Or perhaps the facts given here will allow you to properly evaluate the system you have been thinking about purchasing.

For whatever reason you may be reading, you are dealing with a fascinating and ever-changing field. You are a pioneer. This is a new frontier as surely as was radio in its infancy. Have you ever wished you were alive in the days when men in their basements and garages strained to hear voices over crystal sets and pioneered radio and television broadcasting? You have a second chance.

The future is yours. Only you will be able to discover how you will be able to mate this new technology to the needs of today and tomorrow. Personal computers have come of age. They are real and they are impacting on society. But, by their very names, they will find their greatest application as we learn to apply them in our own personal situations. I cannot tell you (and you cannot tell me) what your most important application will be.

Each person's needs and demands are his own, and it is best for everyone if we gain an understanding of our personal computing system so we can devise our own interpersonal relationship. If all this sounds like you are about to enter a science-fiction-like world where you will "invent" your own little cybernetic pal, it is. If we are to understand and interrelate in a world of rapidly changing technology, we must understand what we are doing and come to grips with the reality of it all. This book is indeed about your first crude cyborg: your personal computer system.

PERSONAL COMPUTING

Contents

Computer Systems

Man's first computer was his fingers (and perhaps his toes). Sadly, he could only add and subtract with this computer. This computer remembered the last count as long as he didn't get a cramp or make a fist. In 1946 the first totally electronic digital computer, ENIAC (for Electronic Numerical Integrator And Calculator) was developed at the University of Pennsylvania. It used miles of wire and 18,000 vacuum tubes; it failed every few hours and put out enormous volumes of heat using 150 kilowatts. Compared to today's computers it was too slow at 300 multiplications per second and far too expensive. Furthermore, ENIAC, in its 40-by-20-foot room, and all its grandchildren for years to come were giant devices that gulped in a tremendous volume of data, juggled it around in their insides, and spat out the results of their manipulations on ultra-expensive high speed printers and other machinery that would give your local auto mechanics nightmares. All this whirring, buzzing, and blinking gave us the tremendous age of the computer, and these computers cost the people who bought them a lot of money.

Not only were the first computers costly, they were big and bulky and demanded a lot of care just to keep things running. Since these computers were so big, bulky, and expensive, they came to be used almost exclusively by the big industrial companies. Most computers even had to have their own special rooms with private air conditioning because of the tremendous amount of heat generated by all the vacuum tubes. But despite the drawbacks, a binary digital computer made possible feats of data processing and problem solving that had previously been impossible.

Through the succeeding generations of computers, speed was the main factor to consider, since all the processing was being done in simple binary on/off conditions. More speed meant simply that more money had to be spent in developing a more sophisticated technology. As it cost more to develop this new computer technology, the companies that were in the forefront had to make more profit. They charged whatever the market would bear in order to have the money left over to come up with new technologies. Thus, already high-priced computers became even higher priced. Yet somewhere in the back of most of these designers' minds was the dream of the day when people could afford their own personal computers. That dream must have seemed a million years from fulfillment.

The technology race continued. Only the big companies were the customers of the computer companies. Only the biggest corporations could afford the outlandish prices they had to pay to get the latest electronic "brains." Because of the demands the big corporations made of the computer manufacturers, technology leaned toward pleasing the "bigs." Computer salespeople adapted their sales techniques to the bigs. Peripheral or add-on equipment manufacturers were building products geared to the big corporations.

Since the bigs had many employees, they wanted printers that could print paychecks faster. They wanted storage devices, like tapes, that were fast and would hold vast amounts of data. They wanted faster terminals with capabilities for easier operation. The bigs also wanted software, or computer programs, that allowed ease of operation by unskilled people. None of these motives were the same ones the pioneer "computer hobbyists" would have in later years. But the technologies that made computers and their peripherals faster would also end up making them cheaper and more sophisticated. As computer users, or prospective users, we owe a lot to the pioneers, the big corporations who demanded technology from manufacturers without much regard to cost.

But, despite our debt to them, we also have much to "unlearn." For instance, the computer salesman with his classical approach to selling a computer by talking about its speed, amount of memory, etc., is going to take a back seat in the personal computer market. He will be replaced by the marketer who explains how pretty the box will look on the kitchen cupboard. In fact, the personal computer salesman may not even mention that the computer is a computer. He may not want to scare the potential customer with thoughts of some giant electronic brain with whirring tapes and card readers. Instead of crowing about the amount of memory or the speed, the future salesman will say, "Here is such-and-such company's black box. For only $299.95, it will balance your checkbook in fifty seconds. It will remember up to three hundred checks at a time, too."

Education is going to have to conform to the "new rules of personal computing," too. Most instructors came up through the traditional computing ranks. They are entering the educational field with approaches and backgrounds relevant to the "bigs." No one is saying that all instruction is outdated or that all instructors are using outdated and outmoded ideas. It is just that it is a very human characteristic to resist change. Personal computing education is going to have to become super flexible because the world of personal computers is a world of rapid change. The personal computer is being nearly revolutionized by new developments almost daily. College and technical school instructors are going to have to ride the tide of change. Of course, for many, the first change will be the biggest. That change for them will be the change to thinking of a personal computer instead of a classical computer.

It is most people's guess that education normally is at least a few years behind technology. With few exceptions, most schools not specializing in technical education, yet offering computer-type curriculums will have the hardest time changing. Keep in mind that we are talking about personal computers here and we are looking to the day when everyone in a household will have his own computer. Thus, the teaching of the interrelationship of the personal computer to the individual will be as important at a liberal arts college as at a technical school. So-called "old school" computer educators are likely to take the classical computer approach to teaching personal computers. This text is intended to impress on you that this approach doesn't fit the personal computer system. Typically, in today's educational environment, a computer is better if it is faster and bigger (in other words, more powerful). This is not necessarily true in personal computing. In fact, most experts agree that cost alone will become the major factor in considering a personal computing system. All agree that there is an attractive cost area of less than $1,000 that will make ownership of a personal computer an affordable reality. A cost of $500 (today's dollars) or below will practically insure the personal computer's success as a major consumer product like the TV and the calculator.

It is cost that has kept the average person from computers up to this point. Even the cost of programming will continue to keep away the student. Not that the student of personal computing will have a problem with costs, it is just that his instructor remembers that only a few years ago (when the instructor was receiving his education) computer time was very expensive. Computer time is the time the computer uses to perform the operations you command it to do. While the bits and bytes are handled by the electronic hardware, they are arranged by programming the computer using changeable commands. These commands are called "software" since they are changeable and are written out on paper before being entered in the computer. With really big computers, running some software can cost as much in computer time as a small personal computer system now costs.

Because of this cost situation, computer programming has assumed a classical approach, too. Programmers were faced with some unique situations that have influenced and will continue to influence their approach to writing software for computers. Some knowledge of classical hardware and software is necessary to an understanding of this situation and how to deal with it.

HARDWARE

The electronic digital computer was different from the older analog computers in that it used and stored data in two simple states (thus the name "binary"). Analog computers, on the other hand, were widely used by the military in such things as radar systems and other control areas where some function such as following a target with an antenna had to be performed. In the analog computer, if storage had to be performed, it was done with a capacitor. The capacitor worked according to its property of energy storage—if a certain analog level were to be remembered, it was fed to a capacitor where it set the charge on the capacitor. Operational amplifiers were developed to provide the functions performed by the analog computer. For instance, multiplying by "N" was performed by running the analog signal through an amplifier with a known gain of "N" times. You could even add by introducing a DC voltage and letting the analog signal ride the DC level. Subtracting simply meant adding a negative voltage. To invert, you used a common cathode amplifier (like a common emitter) with a gain of one. To perform integration and differentiation, you would use RC networks with your amplifier system.

Figure 1-1 shows a simple mathematical function performed by an analog computer. You can see how the operational amplifiers provide outputs. Although the data was hard to interpret, you could perform analysis against time by using an oscilloscope or you could perform so-called steady-state, or signal-less, analysis by looking at meters connected to the output of the last operational amplifier. The system was, of course, prone to error. It was hard to reproduce exact operating conditions from day to day. The kind of operational amplifiers that could retain their characteristics were very, very expensive. Nevertheless, for a time there were no alternatives except to carry out calculations by hand. Sometimes hand calculations were faster than setting up the functions on an analog computer and running the programs. Certainly one would not "go to the computer" for an answer to a simple question. These systems were good only for problems involving many

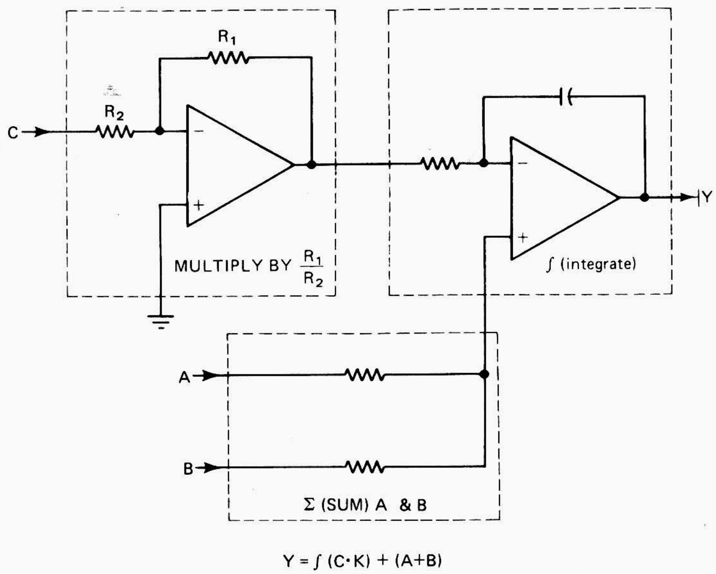


FIGURE 1-1. Simple math function running on a very simple analog computer. The result *Y* is controlled mathematically by the other functions as they are set up in the operational amplifiers.

operations, or "iterations" as they are called, varying only certain conditions.

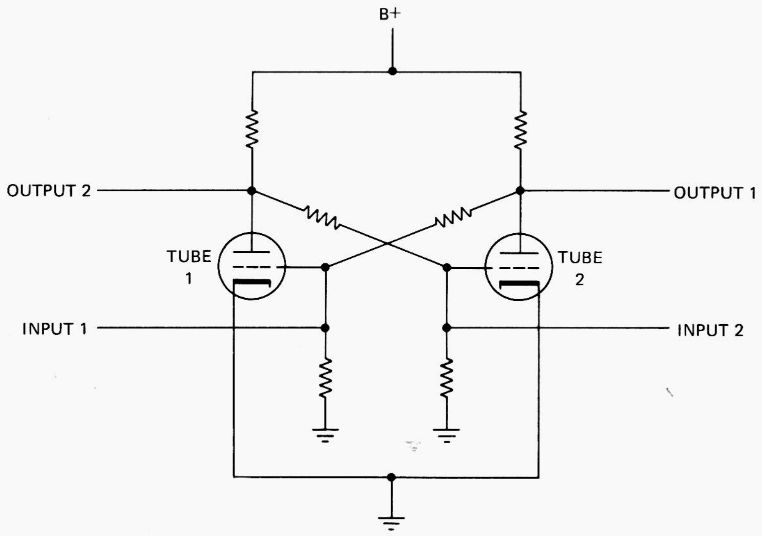
Digital computers, on the other hand, offered new alternatives to the involved setup time and the general expense of analog computers. Digital computers reduced everything to a series of yes/no decisions. Electronically, these yes/no decisions could be duplicated by on and off. Since there were only two states for the stages of the computer, they were called "binary computers." Even though it may seem somewhat crude to try to reduce operations to binary, the relative speed at which the electronic circuits could operate allowed the computer to make its yes/no evaluations many times in a second. Though a critical problem with a lot of steps might take a few minutes or even hours to run, it still became practical to solve complex problems within hours—problems that previously would have taken an entire staff of mathematicians days or weeks to solve. The electronic circuits didn't make mistakes either, as long as they were working correctly and as long as the problem was entered correctly.

As far as making the machines work correctly, it was much easier to duplicate circuits where tubes were either conducting or shut off. Expensive operational amplifiers with their complicated schemes to eliminate drift errors, etc., were no longer needed. Remembering something in "digital" was easier, too. Instead of a capacitor that was constantly losing its charge, memory could be performed by an electronic circuit known as a "flip-flop" that would stay either "flipped" or "flopped" depending on the conditions of its input, as long as the power to the circuit was on. Figure 1-2 shows a flip-flop circuit.

When the voltage of input 1 is positive, it causes the tube to saturate. The output of the saturated tube goes negative (its output drops toward zero volts), and this more negative signal causes tube 2 to cut off. The positive voltage output from cutoff tube 2 now holds tube 1 in saturation even though the signal that caused all the flipping and flopping to happen in the first place has long since gone away. Thus, the memory action. Now, if another positive input comes along to input 1, nothing will happen to the flip-flop. But if a positive voltage comes along on the input 2 side of the flip-flop, tube 2 will turn on and tube 1 will turn off. Tube 1 will hold tube 2 on and the negative output of tube 2 will hold tube 1 off. Our flip-flop "flipped" before, now it has "flopped." It will again retain this state until acted upon by the proper input signal to flip it back. The flip-flop holds its previous conditions or state until another input makes it change state.

Even if you are moving things around at a very high rate of speed,

FIGURE 1-2. Functional drawing of the circuit for a flip-flop. When one of the tube conducts, the other is cut off. The tubes then stay in that state until an input comes in to the flip-flop to change its state.



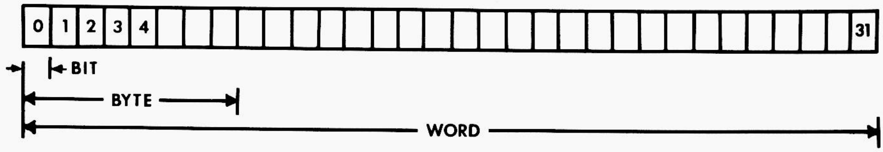
running a lot of binary signals around in a computer can be a little slow. As soon as you start moving a large quantity of data within the machine, you spend a lot of time with simple on/off circuit conditions. Therefore, the on/off states of "bits" are arranged in parallel so that actually the machine is performing operations on many bits at the same time. This is the configuration of the bits within the machine. The computer is configured with a bit, byte, and word structure. As already explained, a bit is an on/off state. When many of these bits are used together at the same time they create a machine "word."

Words are the tools that make sense to the computer. A word is an arrangement of data bits such as that shown in Figure 1-3 which is acted upon all at once by the computer. Note that a word may further be broken into a "byte" (like bite), which is the smallest part of the computer's word that can contain a complete character. There is a standard code for data characters called the "American Standard Code for Information Interchange," ASCII (Asky) for short. All alphabetical and numerical characters are represented by either 7 or 9 bits. The complete ASCII code is given in the appendix at the back of this book. One of the bits of the ASCII code is a parity bit, or a bit that helps in error correction when transmitting ASCII. The computer can drop the parity bit internally. The typical byte will be 6 or 8 bits, rather than 7 or 9.

Thus, we may find the internal structure of a given computer to be configured as a 32-bit word with four 8-bit bytes. A typical "8-bit" microcomputer has an 8-bit byte, but also an 8-bit word. There is only one character per word in this device.

Now, if we are to store meaningful information in our computer, we must have one flip-flop per bit. For a 32-bit computer you would have 32 flip-flops just to store one data word! What's more, to move data from one place to another in the computer, you will have to have 32 interconnecting wires with a common ground. These wires are usually arranged to feed bus wires from one point to another inside the computer. Thus, we find the use of the terms "data bus" and "address bus" in the computer. The data bus is what is in the computer, or the

FIGURE 1-3. Thirty-two-bit computer "word," showing "bit," "byte," and word structure.



contents. The address bus tells where the data is located. It is like a map. You can tell right now that there is a lot of wiring and other circuitry involved in hooking together all the internal pieces of the computer. If for no other reason, early computers were doomed to be expensive.

Figure 1-4 shows some peripherals along with the computer (processor) as the hardware in the typical large computer center. Peripherals are devices that serve the computer, such as card readers, tape units, printers, etc.

Figure 1-5 shows the internal parts of a typical computer in its "classical" configuration. All computers contain five fundamental circuits: input, output, processing, storage, and control. The circuits make up the computer hardware. The circuits perform the processes called for in the instructions (the computer software).

The input and output circuits move data to and from the computer. The Input/Output (I/O) data signals may be serial (a bit at a time), parallel (several bits at once), or a combination of both. The processing circuits, sometimes called the "arithmetic (ə•rith′mə•tik) circuits," perform operations on a computer datum. The arithmetic operations can range

FIGURE 1-4. Typical peripherals in a large computer center. Although the computer itself makes up the "brains" of the operation, the peripherals offer interconnections to the outside world.