TAB 1111

$8.95

HOW TO DESIGN, BUILD & PROGRAM YOUR OWN WORKING COMPUTER SYSTEM

e complete computer build-it book—including
sign, construction, programming, and de-bugging!

/ ROBERT P. HAVILAND

HOW TO DESIGN, BUILD & PROGRAM YOUR OWN WORKING COMPUTER SYSTEM

Other TAB books by the author:

No. 792 *Build-It Book of Miniature Test & Measurement Instruments*

No. 905 *Build-It Book of Digital Electronic Timepieces*

No. 975 *The "Compulator" Book—Building Super Calculators & Minicomputer Hardware with Calculator Chips*

HOW TO DESIGN, BUILD & PROGRAM YOUR OWN WORKING COMPUTER SYSTEM

BY ROBERT P. HAVILAND

TAB BOOKS Inc.
BLUE RIDGE SUMMIT, PA. 17214

FIRST EDITION

FIRST PRINTING—MARCH 1979
SECOND PRINTING—JANUARY 1980
THIRD PRINTING—JULY 1980
FOURTH PRINTING—MARCH 1981

Printed in the United States of America

Library of Congress Cataloging in Publication Data

Haviland, Robert P.
How to design, build and program your own working computer system.

Bibliography: p.
Includes index.
1. Electronic digital computers—Design and construction—Amateurs' manuals. I. Title.
TK9969.H38 621.3819'58'2 79-11200
ISBN 0-8306-9810-8
ISBN 0-8306-1111-8 pbk.

Contents

Introduction

So you want to build a computer. There are, in fact, a whole set of very good reasons for home computer construction.

Perhaps you are considering this approach because you want to learn computers from the inside out. You can't do this by buying an already assembled unit and setting it on the bench. In fact, you can't really do it by assembling a kit; all the thinking about how things work has already been done for you.

Or perhaps you want to have a unit that you feel free to change. Changes can usually be made in commercial units, or kits, but it is surprising how difficult the designer can make simple operations because he did not consider the need for change. Then too, a change in a commercial or a kit unit reduces the resale value. Just the opposite is true for home-built units; a unit which is kept up-to-date is worth more than one of frozen design. Just price 15-year-old electronic equipment to see this.

Perhaps you want to have features not commercially available. The current thrust of commercial microcomputer design is directed towards the packaged general purpose unit. It is oriented to programming and not towards such things as application for control, or monitoring. Currently, if you want these features you cannot buy them, you must build them.

Perhaps you are a dedicated experimenter and want to try new approaches and develop new uses. True, much work can be done with commercial and kit units, but it is surprising how difficult this can be with an already built unit. It seems that the signal you need is buried on the most inaccessible board, the one having no spare terminals.

Perhaps you want to have the pride of accomplishment. You want to be able to say, "That's my computer" and mean it. With commercial units, you can have pride of ownership, but it's not really the same thing.

Or, finally, perhaps you want to save money. On some things, this is hardly possible, since competition has brought the prices down, and since the pricing policy of many manufacturers heavily penalizes the small lot user. But on other things, and overall, a very appreciable saving is possible. If you look at advertisements and catalogues, and find the general class of computer you want, you should be able to build an equivalent unit for one-third of the cost of the commercial unit, or about one-half the cost of the kit unit. You will, of course, have to spend more time if you build, but at least you don't have to pay income tax on the time spent.

All of the above reasons for home construction are true, and any one can be adequate justification for building. They are especially true for a person just starting in the field, one who really needs to learn "what to do" as well as learn the "how and why" of operations. For the amateur in the field, home construction is very worthwhile.

For a professional in some field, home construction is less attractive, as it tends to take time away from his professional work. However, the professional who is attempting to work in a new field, or attempting to develop and extend that field, will probably find it necessary to do a large amount of design construction and test work. The reason for this is that commercial and kit equipment tends to become frozen in concept, propagating a proven design. While the advertisements may claim innovations, they are usually modifications, and true innovations rarely exist.

However, before making a commitment to yourself to start on a computer construction program, think it over carefully. Even a basic computer is not a simple device, and in its fully expanded form, it becomes extremely complex. To build a computer "from the ground up" requires a considerable investment in time, and an appreciable investment in money. It also takes skill, and skill in several fields. To build to your own design requires an appreciable amount of ability in logic design, in construction, in troubleshooting, and in the marriage of logic circuits and devices designed long before computers become a reality. Unless you feel comfortable with all these fields, home computer construction is not really recommended for you.

If you do not have skill in all the necessary fields, there are some things you can do to get the necessary background. Suppose you lack building skill. Correct this by starting with design and construction of simple projects. See, for example, Haviland, 1976, 1977-1, 1977-2.* After these simple steps, improve your construction skills, perhaps by building a complex kit, or perhaps by building an item of your own design. If your problem is unfamiliarity with computer principles, correct this by taking a logic course at a local school or by correspondence. Work out logic problems, for example, in Navy, 1972 or in Ward, 1975. Join one of the computer clubs. Then go ahead.

Perhaps instead of skill, your problem is time, or money. If it is simply time, buy a basic computer, or perhaps a minimum kit such as the SC/MP, KIM, or ELF. Add capability to this as time and opportunity permits. If money is the problem, buying, of course, is completely unattractive. One approach is to build the most basic design, possibly an ELF or an SC/MP, and to expand this slowly. Another is to work along the lines outlined in this book, with the intention of stopping when the basic working computer is obtained. With slow, careful purchasing, excellent bargains can be found which makes this a cost effective approach.

Suppose, following the above suggestions, that you do decide to go ahead with a computer construction project, and specifically, the one covered in this book. If you do this, what will you have? Part way through the construction's chain, you will have a basic, expandable computer in operation. If you complete all of the construction projects, you will have a good working computer which can be still further expanded.

In addition to this expandable computer, you will have a thorough understanding of how the computer works, why it was designed in the way it was, and what additional design features could be added. Further, you will have acquired considerable skill in construction, in troubleshooting, and in logical analysis. If you follow the suggestions, you will also have developed a very considerable skill in programming not only this computer, but any other type.

---

*References are listed at the end of this volume, and are quoted in chapters by author and date.

At this point you are on your own. You probably will want to expand capability still further. The simplest, and most likely way, is by expanding the memory capacity of the computer. This allows larger problems, and also more in the way of stored programs. Expansion can also be done by increasing the computational capability, by adding several calculating functions, or by paralleling processors. Other modes of expansion are:

- Addition of external devices
- A greater variety of input/output equipment
- More modern terminals
- Remote terminals
- Different types of bulk memory storage

The exterior expansion can also be in the direction of greater capability for control of external devices, for example, control of a complete radio station or a home workshop by computer operation.

The various chapters in this book cover a single line of approach. For example, the design is based on a particular central processor design, which, in turn, is based on selection of a particular chip. There is a single bus. A particular set of input and output equipment was chosen for the design. Associated with this are a particular set of computer functions, many of which are based on the input/output devices chosen.

It is not necessary to follow the same path, although following it will simplify design and construction problems. For example, another processor chip could be used. The change, of course, does not end at this particular place, but must be carried through all successive parts of design and construction. This is the reason for the statement that it will be easier to follow the approach used here. But, again, it is not necessary; the design shown here is one of innumerable possible designs.

One reason for not following the design as shown would occur if additions are to be made to an existing small computer installation. In this case, the design concept should be changed as needed. Perhaps this is simply a matter of selection of a connector, or just a change in shape of a printed circuit board.

In this connection, it should be noted that most of the designs shown in this book have been laid out with flexibility of use in mind. Virtually all circuits and leads are fully buffered, with attention given to loading. Addresses may be changed over a wide range. And many circuits provide both direct and inverted signals, selectable by jumpers. These features have been provided for future flexibility of the designs shown here, but they also provide flexibility for adapting the designs to other computer types.

In addition to flexibility with respect to design, attention has also been given to physical flexibility. For example, the chassis in which the printed circuit cards are mounted could be made top loading, instead of front loading. The present design was laid out for expansion, first to double the present number of board spaces, then further by construction of additional cabinets. Power capability and circuitry for peripherals are also expandable.

A major goal in all designs and layouts has been simplicity of construction and test. The chassis can be constructed with hand tools, but of course a drill press, a jigsaw, and a small metal brake will save time. Circuit boards are designed for manual one-of (one-of-a-kind) construction, and can be fabricated with hand tools only, although a jigsaw and a drill press are recommended. Circuit boards have been designed for initial test with relatively simple equipment. A vacuum tube voltmeter can accomplish all test functions; of course, an oscilloscope is considerably better and is recommended.

In addition to the material on design and on hardware construction, a number of aspects of programming the resulting computer are covered. These are intended to be a bridge between the hardware and a fully programmed operation, between hardware constructor and programmer. However, it should be noted that this volume does not include any programs as such. An ample number of programs are available in the open literature, programs which cover all aspects from simple short routines to complete higher level languages. Many of these programs are intended specifically for the processor unit used in the prototype. Others are intended for other processors but can be readily adapted for the specific processor used. Full references are given. If the suggestions as to programming are followed in parallel with the development outline, the constructor should wind up with a complete working system, both hardware and software. Both would be expandable towards more calculating power and towards both more general and more specific applications.

One caution seems to be in order. Every effort has been made to avoid error. Despite this effort, errors do creep in. In addition, it is very easy to miss fine elements of logic. Because of these factors, it is urged that you check and cross-check each step very carefully.

This is the way to catch errors. It is also the way to catch such errors as you may have accidentally introduced, possibly by misunderstanding of an instruction, or possibly by missing a connection or routing the connection wrong. We all try to avoid these little problems, but they do occur. In this connection, I would appreciate notification of any errors found to allow correction in future editions.

Robert P. Haviland

Chapter 1
A Home-Built Computer—The Basic Approach

A system as large as a computer requires many decisions. The most immediate and probably the most important decision of all is the general approach to be followed. In building the prototype described in this volume, the first decision made was that this prototype should be flexible in use. It should grow to a true, general purpose computer, capable of calculations and data handling. In addition, it should also be capable of such functions as control of external devices, and should have flexibility for future addition of both devices to be controlled and data handling devices.

The prototype was conceived as being a computer for personal use, not intended for business or mass data handling operation. This is a fairly important factor, since in personal use it is possible to substitute time for computing power. Speed, while not a negligible factor, is not nearly as important in a personal computer as in one which must handle mass data. This is an important factor in keeping costs of personal computers low.

Also, since this was to be a personal computer, simplicity seemed to be a necessary item. For example, it seems best to keep functions separate. There should be no processing on a memory board; the memory board should serve for memory only. Similarly, input/output circuitry should be kept separate from processing circuitry and from memory. This approach seems to have considerable advantages when considering future applications. It is much easier to develop design flexibility if functions are clearly separated by type.

As an example of the value of this separation of function, suppose that you have a group of applications which involve time, perhaps absolute time in the sense of day, hour and minute, or perhaps relative time in the sense of time intervals. With a well-designed system having separate functions, it is easy to add a time-generation board, using the central processor to read the time information from this board and to distribute it to other using boards. The time board, for example, could be constructed around one of the readily available clock chips, this providing both absolute and relative information.

Again, for example, this general approach permits addition of one or more separate boards for arithmetic operations. Each of these can be based on one of the low-cost calculator chips, the type being selected to match the nature of the arithmetic being done. The calculator chip could even be one of the programmable type, which approach a microcomputer in computational power. In one way of thinking, the design flexibility would be sufficient to permit one computer to control another one.

The basic elements for this concept are shown in Fig. 1-1. The computer would have three primary bus systems, one for addresses, one for data, and one for control signals, plus a power bus. There would be three basic elements connected to these buses. One, the heart of the unit, would be the central processing unit (CPU), which would accomplish the logic and arithmetic operations needed. The second, the memory unit, would hold the instructions and data needed for current operations. And the third, the input/output (I/O) unit, would provide communications to "the outside world."

In addition to these elements which form the core of the computer, more elements could be added as required or desired. Typical ones are shown also in Fig. 1-1, as unconnected blocks. One of these could be a second processor, or perhaps an arithmetic unit based on the calculator chips, as just discussed. A time unit could be an alternate or addition to these. When the computer is used for control of external elements, separate control units would be added, with such data and signal conversions as needed, to match external devices. Finally, there would be extended provision for storage for such items as masses of data of programs which are rarely used, or

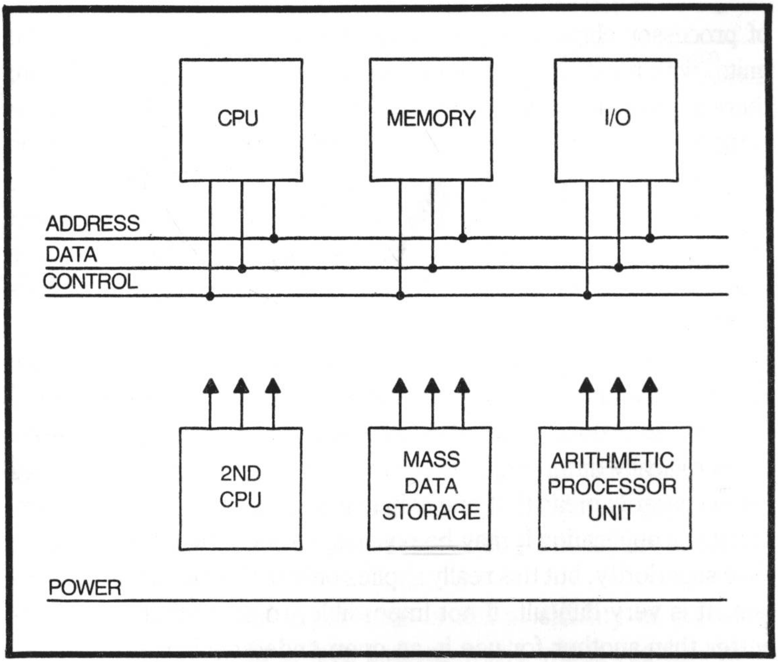

Fig. 1-1. Concept of a home-built computer. Basic elements are shown on top row. Lower row shows typical elements which can be added. Four buses interconnect all units.

of programs required when the general usage of the computer is changed. And, of course, the memory and input/output units can always be expanded by building duplicate or extended versions of the basic memory and input/output elements.

This approach appears to have particular advantages for the home constructor. For one thing, each additional block to be added can be a nearly independent project of its own, each being constructed and tested as an independent device. When operable, the element can be connected into the computer, with the remaining operations involved only in making sure that the standard signal conditions of the three computer buses have been satisfied. In essence, this separation of functions reduces the design of a large complex computer to the much simpler problem of design and construction of a series of independent but related projects.

CENTRAL PROCESSOR SELECTION

With the basic approach in mind, prototype design can proceed to the choice of a simple processor. At this time there is quite a range of processor chips available, ranging from four-bit to sixteen-bit units, with most being designed for eight-bit, one-byte operation; there are indications that thirty-two-bit processors will be available in the future. Characteristics of four typical chips are shown in Table 1-1. This gives the address, data, and control bus structure of the chip, and some information on its operations, especially with respect to cycle time and number of instructions available. These chips are representative. See Bursky, 1978 for an extensive review of types available.

Choice amongst these four specific chips, or even among the larger number available, is not really easy. There is almost no logical basis for selection. True, the designs are different, and in some cases radically different. However, the fact of the difference does not necessarily mean that one chip is generally better than another. In a given application it may be possible to show that one chip does have superiority, but this really applies only to that particular application. It is very difficult, if not impossible, to show that one chip is better than another for use in an open ended design.

Some sketch designs were made for several of the most popular chips. These seem to show that the complex chips, generally the chips with a large instruction repertoire, require more supplementary chips to provide functions which are either necessary or which seem important. A separate analysis of the actual use of instructions in programming showed the interesting fact that a very small number of instructions carries most of the programming work. In a given program as many as one-half of the available instructions may not be used at all and another one-fourth might be used only one or two times. The analysis also showed that all chips available had sufficient basic instructions: the operations possible with the complex chips can be accomplished with the simpler ones. The difference is that two or more instructions may be required with the simple chips to accomplish a desired result. However, the overall loss in speed due to this need for more instructions did not appear to be as great as might seem true at first glance, since the specialized instructions are so rarely used This finding reinforced the thought that it would be at least difficult and probably impossible to show that one chip was really better than another.

There are, however, two other factors of importance. A home-built computer should use components readily available to the home builder at low cost. At present, this indicates that most of the