HOW TO "READ" FM TUNER SPECIFICATIONS

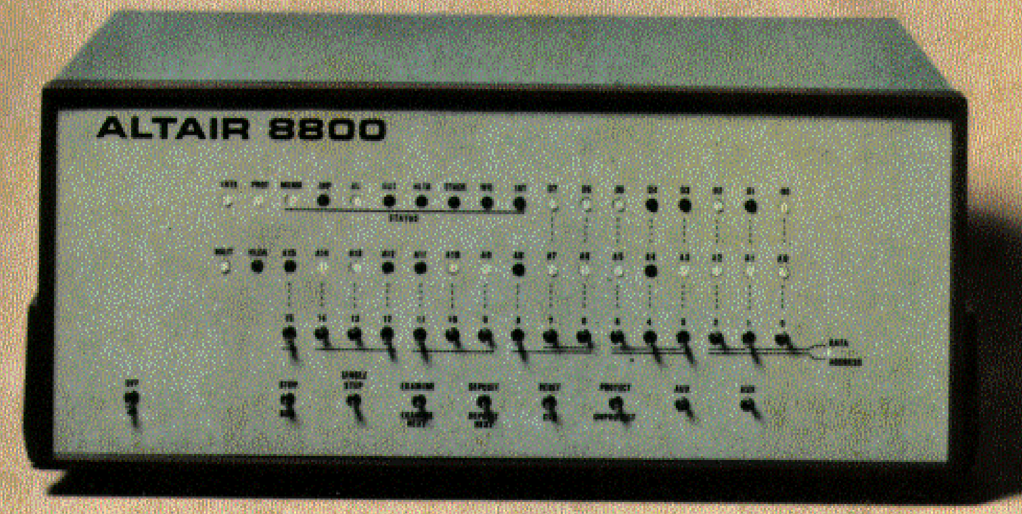
Popular Electronics

WORLD'S LARGEST-SELLING ELECTRONICS MAGAZINE JANUARY 1975/75¢

PROJECT BREAKTHROUGH!

World's First Minicomputer Kit to Rival Commercial Models... "ALTAIR 8800"

SAVE OVER $1000



ALSO IN THIS ISSUE:

- An Under-$90 Scientific Calculator Project
- CCD's—TV Camera Tube Successor?
- Thyristor-Controlled Photoflashers

TEST REPORTS:

Technics 200 Speaker System
Pioneer RT-1011 Open-Reel Recorder
Tram Diamond-40 CB AM Transceiver
Edmund Scientific "Kirlian" Photo Kit
Hewlett-Packard 5381 Frequency Counter

18101

JANUARY 1975 VOLUME 7, NUMBER 1

Popular Electronics®

WORLD'S LARGEST-SELLING ELECTRONICS MAGAZINE

FEATURE ARTICLES

CONSTRUCTION ARTICLES

COLUMNS

PRODUCT TEST REPORTS

DEPARTMENTS

POPULAR ELECTRONICS, January 1975, Volume 7, Number 1. Published monthly at One Park Avenue, New York, NY 10016. One year subscription rate for U.S., Possessions and Canada, $6.98; all other countries, $8.98. Second class Postage paid at New York, NY and at additional mailing offices. Authorized as second class mail by the Post Office Department, Ottawa, Canada and for payment of postage in cash. Subscription service and Forms 3579; P.O. Box 2774, Boulder, CO 80302.

POPULAR ELECTRONICS Including ELECTRONICS WORLD, Trade Mark Registered. Indexed in the Reader's Guide to Periodical Literature.

Ziff-Davis also publishes Boating, Car and Driver, Cycle, Flying, Modern Bride, Popular Photography, Skiing and Stereo Review.

Forms 3579 and all subscription correspondence should be addressed to POPULAR ELECTRONICS, Circulation Dept., P. O. Box 2774, Boulder, CO 80302. Please allow at least eight weeks for change of address. Include your old address, as well as new—enclosing, if possible, an address label from a recent issue.

Editorial contributions must be accompanied by return postage and will be handled with reasonable care; however publisher assumes no responsibility for return or safety of art work, photographs, models, or manuscripts.

Popular Electronics
JANUARY, 1975

PE TESTED

EXCLUSIVE!

ALTAIR 8800
The most powerful minicomputer project ever presented—can be built for under $400

BY H. EDWARD ROBERTS AND WILLIAM YATES

THE era of the computer in every home—a favorite topic among science-fiction writers—has arrived! It's made possible by the POPULAR ELECTRONICS/MITS Altair 8800, a full-blown computer that can hold its own against sophisticated minicomputers now on the market. And it doesn't cost several thousand dollars. In fact, it's in a color TV-receiver's price class—under $400 for a complete kit.

The Altair 8800 is not a "demonstrator" or souped-up calculator. It is the most powerful computer ever presented as a construction project in any electronics magazine. In many ways, it represents a revolutionary development in electronic design and thinking.

The Altair 8800 is a parallel 8-bit word/16-bit address computer with an instruction cycle time of 2 µs. Its central processing unit is a new LSI chip that is many times more powerful than previous IC processors. It can accommodate 256 inputs and 256 outputs, all directly addressable, and has 78 basic machine instructions (as compared with 40 in the usual minicomputer). This means that you can write an extensive and detailed program. The basic computer has 256 words of memory, but it can be economically expanded for 65,000 words. Thus, with full expansion, up to 65,000 subroutines can all be going at the same time.

The basic computer is a complete system. The program can be entered via switches located on the front panel, providing a LED readout in binary format. The very-low-cost terminal presented in POPULAR ELECTRONICS last month can also be used.

PROCESSOR DESCRIPTION

Processor: 8 bit parallel
Max. memory: 65,000 words (all directly addressable)
Instruction cycle time: 2 µs (min.)
Inputs and outputs: 256 (all directly addressable)
Number of basic machine instructions: 78 (181 with variants)
Add/substract time: 2 µs
Number of subroutine levels: 65,000
Interrupt structure: 8 hardwire vectored levels plus software levels
Number of auxiliary registers: 8 plus stack pointer, program counter and accumulator
Memory type: semiconductor (dynamic or static RAM, ROM, PROM)
Memory access time: 850 ns static RAM; 420 or 150 ns dynamic Ram

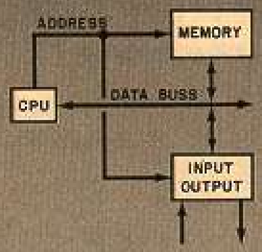


Fig. 1. Basic block diagram of computer parts and operation.

In an upcoming issue, we will describe how to build a low-cost CRT-type terminal that can be used with the computer and can also be mated with any time-sharing computer by telephone.

About the Computer. A computer is basically a piece of variable hardware. By changing the bit pattern stored in the memory, the hardware (electronic circuitry) can be altered from one type of device to another. When the bit pattern, and thus the hardware, is changed, we have what is referred to as "software." Any type of variable instruction (programming)—such as Basic, Fortran, Cobol, Algol—is generally classified as software.

To cause it to vary the hardware, you must communicate with the computer. In the case of the 8800, this is done by setting the bit pattern on the front-panel switches in accordance with a set of instructions (provided with the Intel 8080 LSI chip). For example, the 8800 computer will automatically add when a specific bit pattern (10000010) is received. By setting address and data switches, a complete program of up to 78 steps in the basic computer can be inserted into the processor. If extensive programming is to be performed, an assembler or higher language is used. With an assembler, the person doing the program simply types the word "add" on the device. (In Basic and Fortran, a + is used instead.)

Fundamental programming concepts are simple enough to master in a relatively short time. However, to become an efficient programmer requires a lot of experience and a large amount of creativity.

The block diagram of the basic 8800 computer (or any computer, for that matter) is shown in Fig. 1. It consists of the following subsystems:

CPU. The heart of the computer is the CPU, or central processor unit. See fig. 2. The CPU performs all the calculations, generates system timing, and makes all decisions. Of particular importance are the decisions the CPU makes concerning what device should have access to the data buss. It makes these decisions by sending status information at the beginning of each computer cycle, telling the memory and the input/output what to expect for the rest of the cycle.

The CPU contains the program timer, sometimes called the P counter. This device keeps track of the current location in the memory that the processor is using. Also located in the CPU is the arithmetic unit.

The CPU used in the 8800 computer, the Intel 8080 LSI chip, is relatively expensive in quantities of one. It was selected, however, because it serves to create a minicomputer whose performance competes with current commercial minicomputers. In practice, a lower-performance processor would have been adequate for the majority of the tasks the user might wish to initially define. But the problem with the lesser-power approach is that relatively little money would be saved, and it would be doomed to near-future obsolescence for practical purposes. Our intent here was to produce a processor with more than enough power to handle any job.

Still another consideration was programming. The larger the instruction set, the easier the computer is to program. The 8080 chip has 78 instructions, which is almost twice that of the next power level CPU available (Intel's 8008), which is really designed for use as a buffer.

The CPU contains eight general-purpose registers, P counter, arithmetic unit, accumulator, stack pointer, instruction decoder, and miscellaneous timing and control circuits. The arithmetic unit is of special interest because it contains the circuitry required to perform arithmetic in both decimal and binary formats.

The stack pointer is the register that keeps track of the subroutine addresses. The 8800 computer is capable of performing an almost unlimited number of subroutines, a feature not available with other microprocessors and absent in many minicomputers.

The instruction decoder is the core of the variable-hardware concept. It

PARTS LIST

C1, C5 to C13—0.1-µF disc capacitor
C2—0.01-µF disc capacitor
C3, C4—100-pF disc capacitor
C14 to C20—0.001-µF disc capacitor
IC1—8080 central processing unit IC (Intel)
IC2 to IC5—74L74 IC
IC6 to IC14—8T97 IC
IC15, IC17—7402 IC
IC16, IC32, IC33—7404 IC
IC18 to IC20, IC51—74123 IC
IC21—7473 IC
IC22 to IC24, IC50—7400 IC
IC25—7430 IC
IC26—7410 IC
IC27 to IC31, IC39 to IC41—7405 IC
IC34, IC35—8111 IC
IC36—74L30 IC
IC42 to IC49—74L00 IC
IC52—7406 IC
IC53 to IC58—8111 IC (optional)
LED1 to LED36—Panel-type, red light-emitting diode
(Note: Following are resistors ½-watt, 10% tolerance)
R1, R3, R9 to R31, R56—1000 ohm
R2, R4, R7, R8—330 ohm
R5, R33 to R37—2200 ohm
R6—7500 ohm
R32—100 ohm
R38, R48 to R55—10,000 ohm
R39—200,000 ohm
R40 to R47—470 ohm
R57 to R92—220 ohm
S1 to S16, S25—Spst miniature toggle switch
S17 to S24—Spdt spring-loaded, momentary-action miniature toggle switch
XTAL—2-MHz crystal
Misc.—Metal case; power supply (see text); line cord; multiconductor ribbon cable; mounting hardware; solder; etc.
Note: The following items are available from MITS, Inc., 6328 Linn N.E., Albuquerque, NM 87108 (Tel.: 505-265-7553): partial kit No. 8800PK (includes pc boards and all electronic components (but not case, switches, or power supply), $298; complete kit No. 8800K (contains all parts, including ready-to-use case, switches, and power supply), $397; Completely assembled and tested Model 8800A computer (includes 90-day warranty), $498. Prices do not include postage or delivery charge. Both kits include detailed assembly and operating manual. A FREE set of etching and drilling guides, component-placement diagrams and miscellaneous information is available from the kit supplier (send self-addressed 8½" × 11" envelope with 40c postage). Check supplier or manufacturer for latest ICI price, available separately.

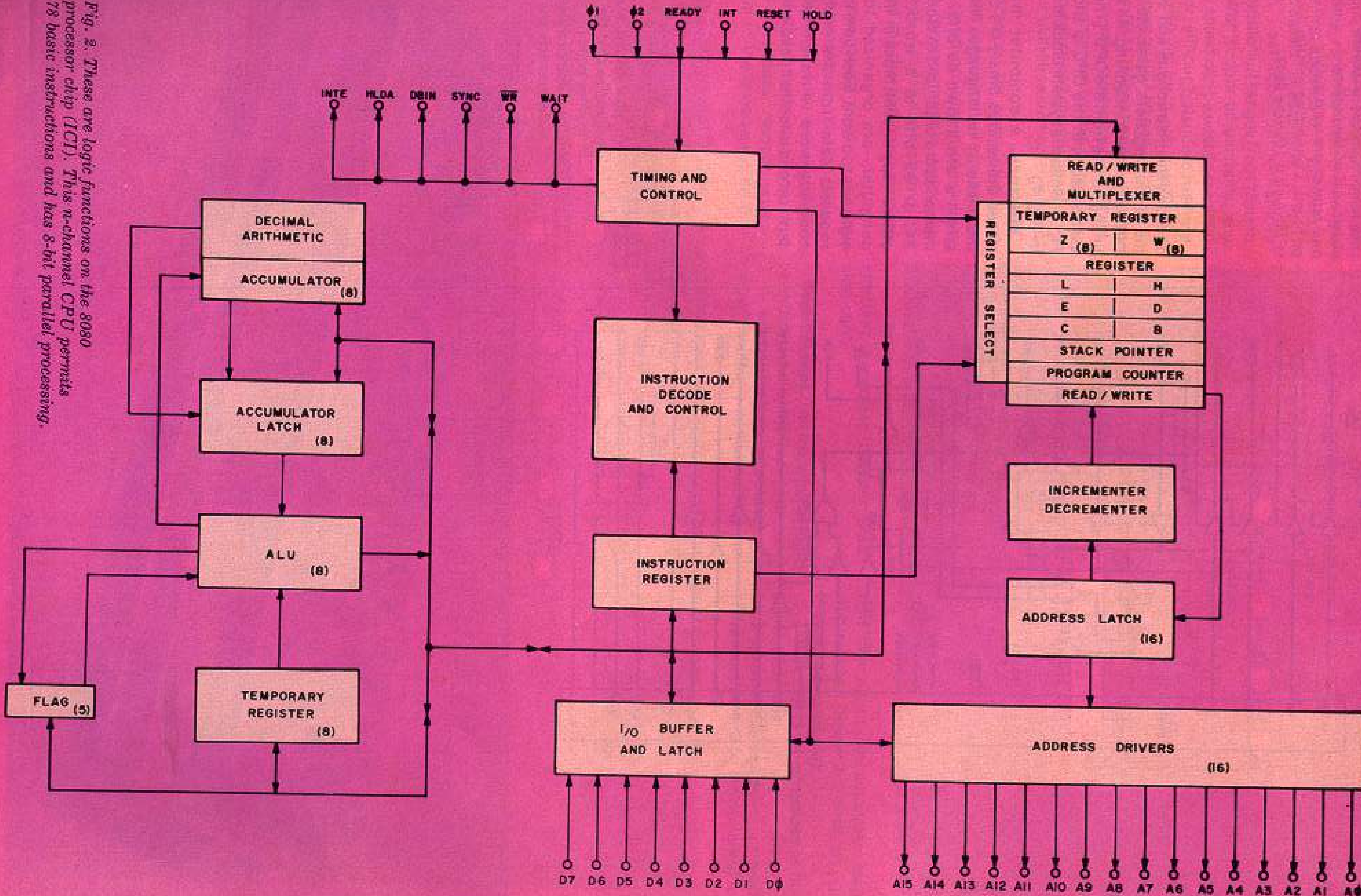


Fig. 2. These are logic functions on the 8080 processor chip (IC1). This n-channel CPU permits 78 basic instructions and has 8-bit parallel processing.

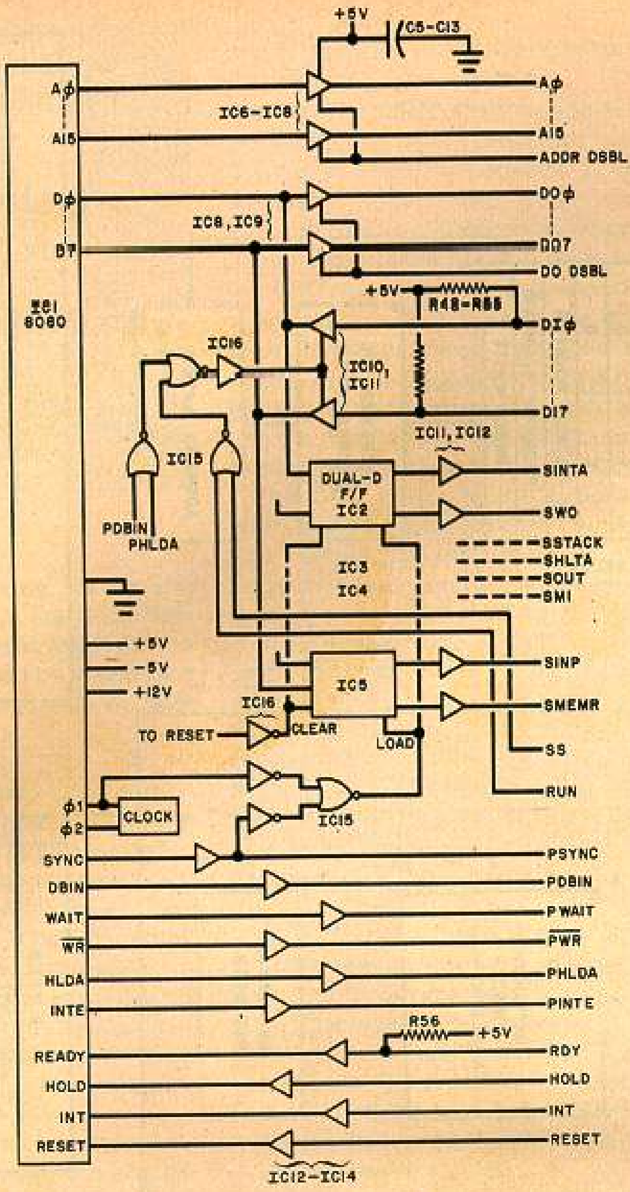
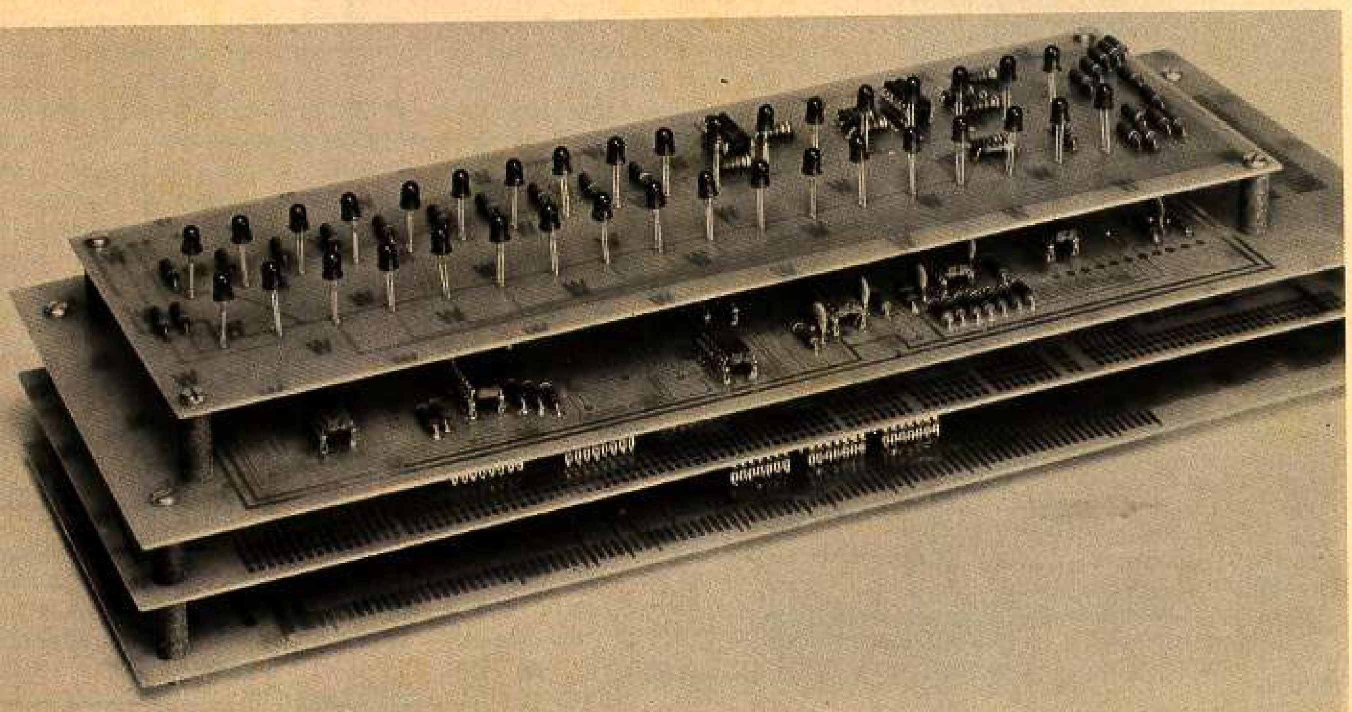


Fig. 3. The logic associated with the CPU (IC1) is shown at left. All of the buffers and latches are on a single pc board. Connecting wiring is through a 100-line buss.

decodes the instructions and sets up the various registers, gates, etc., in the CPU for proper functioning. All system timing comes from the CPU. (The logic associated with the CPU is shown in Fig. 3, while the system clock is shown in Fig. 4.)

Memory. A computer memory stores the various binary 0's and 1's that make up its language. These 0's and 1's are known as *bits*. Some memories are organized to store 4, 16, 24, or 32 bits to a *word*, while others—specifically those in the 8800 computer—are organized to store eight bits to a word. Each time the CPU requests data from the memory, a complete word is transmitted. The term *byte* is interchangeable with the

Printed circuit boards are designed so that the various mating pads are aligned. Multi-conductor ribbon cable interconnects the boards.



term word in an 8-bit processor. (The basic 8800 memory is shown in Fig 5.)

The time required from when the address first appears until the data is stable is called "access time." In most modern semiconductor-memory minicomputers, it ranges from 15 ns to 30 µs. With proper adjustments, any memory speed can be used in the 8800 computer, although standard memory time is 850 ns for a static random-access memory (RAM) and 420 ns for a dynamic RAM. Higher-speed memories will not appreciably affect the performance of the computer, while slower-speed memories will result in an overall reduction in system speed.

In addition to semiconductor RAM's, the processor will also service ROM's (read-only-memories) and PROM's (programmable read-only memories). Access time should be reinforced for the particular memory used.

Any conventional memory can be used in the computer if the input loading on the buss does not exceed 50 TTL loads and if the buss is driven by standard TTL loads. Normal expansion loads to the buss would be one standard low-power load per expansion card.

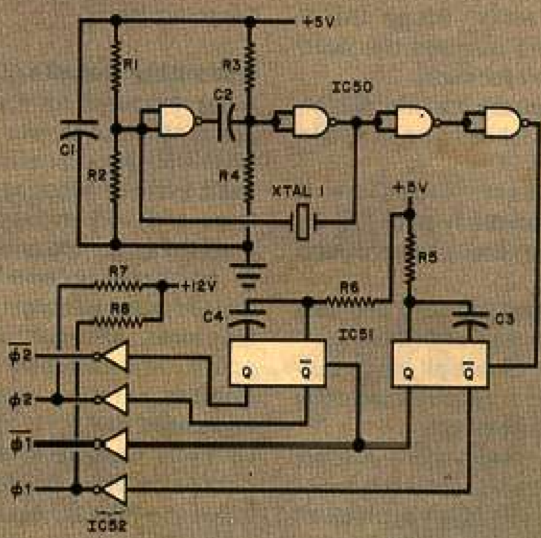


Fig. 4. Computer operation is controlled by signals from this 2-MHz clock circuit.

Front Panel. The front-panel logic permits the following functions:

1. STOP: Stops the processor immediately after it completes the current instruction. An automatic stop occurs when power is turned on (interrupts are disabled).

2. RUN: Starts the processor at the current address.

3. EXAMINE: Causes the data stored at the location (set by the switches) to

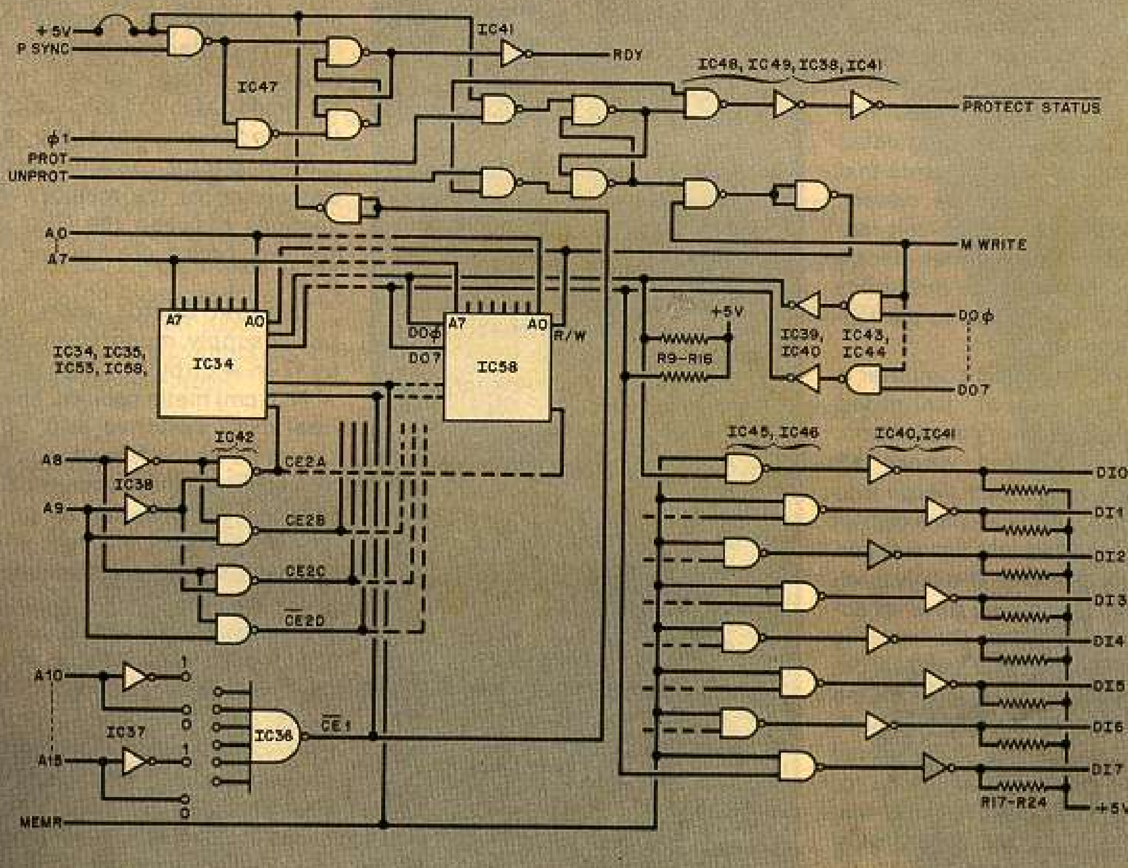


Fig. 5. The basic memory contains up to eight 256 x 4 RAM's.

be displayed in binary by LED's.

4. EXAMINE NEXT: Steps the P counter once and displays the word stored at the next location.

5. DEPOSIT: Causes the information preset by the switches to be stored in the memory.

6. DEPOSIT NEXT: Stops the P counter and loads the memory.

7. SINGLE STEP: Steps the program one instruction.

8. RESET: Clears the CPU and sets up a starting address of 0.

9. CLEAR EXT: Clears all input/output devices; occurs automatically when power is turned on.

10. PROTECT/UNPROTECT: Allows selective protection/unprotection of blocks of memory. When a block of memory is protected, it is impossible to write over that block, but its contents can be read out.

There are 36 LED status indicators on the front panel, 16 of which are used for the address buss, 8 for the system status latches, and 8 for the data buss. The four remaining LED's are used for indicating memory-protect, interrupt, system-wait, and hold status.

Power Supply. Four power sources are required to operate the computer: +5 volts at 2 amperes, −5 volts at 500 mA, −12 volts at 500 mA, and +8 volts at 6 amperes. The first three are regulated, while the last is unregulated. The three regulated lines power the processor. The unregulated line powers the peripheral cards that can be used to expand the system, each of which has its own 5-volt regulator on board. This reduces electrical noise and obviates the possibility of total system failure due to the failure of only one regulator.

Expansion. The basic computer is designed for almost unlimited peripheral and memory expansion, using a buss system where all input/output connections merge into a common line. Hence, an external card can be plugged into any slot and it will function properly. The only qualification is that each card have an address decoder to allow the specific card to take what data it needs from the common buss and put data on the buss as required. The processor buffers are designed to drive 300 external cards, which should be adequate for most applications. Bear in mind that only 17 cards will yield 65,000 words of memory.

[Editor's Note: At this writing, a number of different peripheral devices are in various stages of design or undergoing tests]

Assembly Details. The basic computer employs four printed circuit board assemblies, each of which contains one functional element of the basic system. Because the boards are large and very complex, we are not publishing etching and drilling guides or component-placement diagrams. Instead, you can obtain a set of guides, diagrams, an instruction set, buss points, and miscellaneous information by sending a stamped self-addressed 8½" ×11" manila envelope with 40c postage to MITS, Inc. (See note below Parts List for address.) Request the PE8800 package.

The front panel display board accommodates the 36 LED indicators and their associated drivers. Address line inputs *A0* through *A16*, data lines *D0* through *D7*, and the various status lines originate on the CPU board. The boards have been designed so that the various mating pads on both are aligned. Multi-conductor flexible ribbon cable interconnects the boards.

The front panel control board contains the circuitry for the interfacing between the control switches located on the front panel and the CPU. In addition to the interconnections to the actual processor, this board accepts memory address switches *A0* through *A15* (also on the front panel). The first eight of these switches (*D0* to D7) are used to put data into the CPU. The EXAMINE/EXAMINE NEXT, DEPOSIT/DEPOSIT NEXT, SINGLE STEP, and RUN/STOP switches are also wired directly to the front panel control board.

The third board contains the Intel 8080 central processing unit LSI chip, two-phase clock and buffers, and the various lines going to the buss. (The buffers are tri-state, high-input-impedance, high-output-level devices.) This board also has four dual-D flip-flops wired as latches for the eight bits of status information. All input and output wiring to and from the CPU board is via a 100-line buss.

The basic memory board contains 256 eight-bit words of random access memory (RAM). It is directly expandable to 1000 words. This board also contains the input/output data-gating, address-decoding, memory-wait, and memory-protect circuits. The memory-wait circuit allows the memory time to stabilize the output data to the processor, while the memory-protect circuit prevents accidental overwriting of the memory. All connections between the CPU and the memory board are via the 100-line buss.

The four boards, along with the power supply, mount in an 18-in. deep by 17-in. wide by 7-in. high (45.7 × 43.2 × 17.7-cm) metal cabinet. The various operating switches and LED indicators go on the front panel. When all this is done, the computer cabinet's interior will appear to be almost empty. However, the internal cabling system is arranged with connectors to accommodate 17 more boards within the case, all connected to the main buss lines. The added boards can be used for memory, input/output devices, control devices, etc. All you have to do is plug the boards into the connectors and the computer does the rest.

Part 2 of this article, next month, will describe the operation of the computer and present some sample programs. ◈

SOME APPLICATIONS FOR THE ALTAIR 8800 COMPUTER

Listed below is only a small sampling of the thousands of possible applications for the computer. The Altair 8800 is so powerful, in fact, that many of these applications can be performed simultaneously. It can be used as a:

- Programmable scientific calculator
- Multichannel data acquisition system
- Automatic control for ham station
- Sophisticated intrusion alarm system with multiple combination locks
- Automatic IC tester
- Machine controller
- Digital clock with all time-zone conversion
- High-speed I/O device for large computer
- Digital signal generator
- Automated automobile test analyzer
- On-board mobile controller
- Autopilot for planes, boats, etc.
- Navigation computer
- Time-share computer system
- "Smart" computer terminal
- Brain for a robot
- Pattern-recognition device
- Printed matter-to-Braille converter for the blind
- Automatic drafting machine
- Automatic controller for heat, air conditioning, dehumidifying
- Controller for sound systems
- Digital filter
- Signal analyzer

BUILD THE ALTAIR 8800 MINICOMPUTER
PART TWO

Practical use of the computer, including programming

BY H. EDWARD ROBERTS AND WILLIAM YATES

LAST MONTH, we discussed the various subassemblies used in the basic Altair 8800 computer, went into details on how it is assembled, and listed a few applications. Here, we will describe a test program to be used in checking operation and then focus on practical uses and go through a software example to familiarize you with some operating procedures.

Test Program. The following simple program is used for initial testing of the computer's operation. It also illustrates how a program is loaded and run. The selected program will add two numbers stored at address locations 128 and 129 and store the result at address location 130. The procedure is as follows:

1 Set the power switch to ON and momentarily toggle the RESET switch. (Note: Excluding the power switch, all bottom-row switches on the front panel are spring-loaded, momentary-action types. The switches automatically return to their center-off positions when released from either of their operate positions. When instructed to operate any of the bottom row switches, momentarily throw it to the position indicated and release it.)

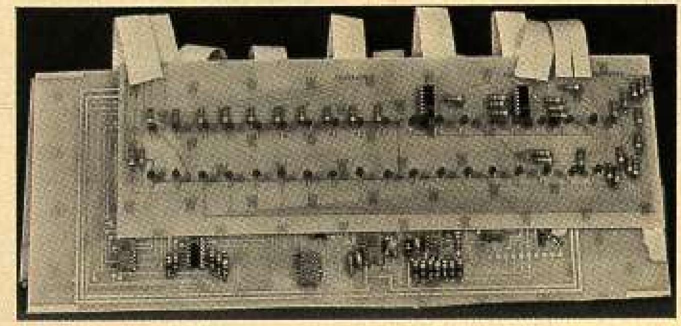
2 Set address switches *A0* through *A15* all to the 0 positions (down). Operate the EXAMINE switch, which should cause address LED's *A0* through *A15* to extinguish to indicate that location 0 is ready. (Some of the data LED's, *D0* through *D7*, might be illuminated, indicating the current contents at location 0.)

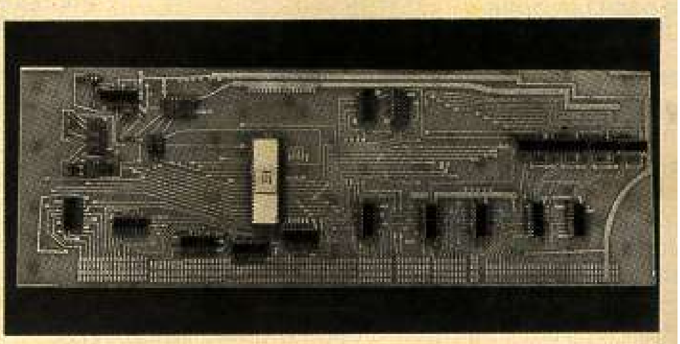
3 Next, store the load accumulator instruction at location 0 by using the binary number for 58 (00111010). Set this binary input up by using switches *D0* through *D7*, with a 1 represented by the switch in the up position and a 0 with the switch in the down position. Hence the switch sequence for 00111010 would be: *D7* down, *D6* down, *D5* up, *D4* up, *D3* up, *D2* down, *D1* up, *D0* down. Store this number at location 0 by operating the DEPOSIT switch. The *D0* through *D7* LED's should now match these settings, with a lighted LED indicating a 1 and a darkened LED indicating a 0. None of the *A0*-*A15* LED's should be on indicating location 0. The load accumulator instruction now tells the computer that the next two entries will be an address number (16 bits). Upon program execution, the data stored at that address number will be transferred to the accumulator.

4 Address numbers, such as address 128, are expressed in 16-bit binary format. The least-significant bits (last eight) are stored in the first memory location following the load accumulator instruction, while the most-significant bits are stored in the second memory location. Set *D0* through *D7* for 10000000 (128) and operate the DEPOSIT NEXT switch. This number is now stored, in binary form, at memory location 1. (*A0* LED should be lit indicating location 1.) Set *D0* through *D7* all to 0 and operate the DEPOSIT NEXT switch. The all-zero binary number is now stored at memory location 2 (*A1* LED is lit) and the computer has been instructed to put the contents of address 128 into the accumulator.

5 To add a second number to the current number stored in the accumulator, the computer must be instructed to transfer the current number to one of the general-purpose registers. In this example, we will use register B. The instruction used is "move A to B," where A is the accumulator. The code for this instruction is 01000111, set up with switches *D0* through *D7*. Operate the DEPOSIT NEXT switch. The instruction "move A to B" is now stored at memory location 3. (*A1* and *A0* lit.)

6 Now, instruct the computer to load the data from address 129 into the accumulator. This procedure is identical to that outlined in steps 3 and 4 above. Set switches *D0* through *D7* for 00111010 and operate the DEPOSIT NEXT switch. The load accumulator instruction is now stored at memory location 4. (*A2* lit.) Set *D0* through *D7* for





10000001 (129) and operate the DEPOSIT NEXT switch to store this number at memory location 5. (*A2, A0* lit) Then set *D0* through *D7* all to 0 and operate the DEPOSIT NEXT switch to store the all-zero number at memory location 6 (*A2, A1* lit).

7 Store the add instruction at memory location 7 by setting *D0* through *D7* for 10000000 (128) and operating the DEPOSIT NEXT switch. When executed, this instruction adds the number in the accumulator to the number stored in register B and places the result in the accumulator (*A2, A1, A0* lit).

8 To store the result at address 130, first store the instruction at memory location 8 by setting *D0* through *D7* for 00110010 and operating the DEPOSIT NEXT switch (*A3* lit). Set *D0* through *D7* for 10000010 and operate the DEPOSIT NEXT switch. The least-significant eight bits of address 129 are now stored at memory location 9 (*A3, A0* lit) Set *D0* through *D7* to 0 and operate the DEPOSIT NEXT switch. The most-significant eight bits of address 129 are now stored at memory location 10 (*A3, A1* lit).

9 A program that adds the contents of address 128 to the contents of address 129 and stores the result in address 130 has now been loaded into the computer. With the use of a "jump" instruction, you can now create a program loop that will direct the computer back to memory location 0 and allow repeating this addition procedure continuously for as long as desired. Store the jump instruction at memory location 11 by setting *D0* through *D7* for 11000011 and operating the DEPOSIT NEXT switch (*A3, A1, A0* lit). Set *D0* through *D7* to 0 and operate the DEPOSIT NEXT switch *twice*. The 16-bit address 0 is now stored at memory locations 12 and 13 (*A3, A2, A0* lit).

Before we can run this program, we have to load the two numbers we want added into addresses 128 and 129. For example, if we wanted to add 12 to 8, the procedure would be as follows:

Set address switches *A0* through *A15* for 0000000010000000 (128) and operate the EXAMINE switch (*A7* lit). Set *D0* through *D7* for binary 12 (00001100) and operate the DEPOSIT switch (*A7* still lit). Set *D0* through *D7* for binary 8 (00001000) and operate the DEPOSIT NEXT switch. The binary numbers for 12 and 8 are now stored at address locations 128 and 129, respectively (*A7, A0* lit).

Set address switches *A0* through *A15* to 0 and operate the EXAMINE switch (all A LED's are off). Operate the RUN switch, and the program will execute at a rate of about 30,000 times per second. Operate the STOP switch. Set the address switches to address 130 (10000010) and operate the EXAMINE switch. LED's *D0* through *D7* will display the sum of the two numbers added, which is 20, in binary format (00010100).

Basics of Programming. If you have never done any programming, it may seem a little mysterious at first, but the basic ideas of programming

GLOSSARY OF COMPUTER JARGON

Access time — Time interval between the instant at which information is called for storage and the instant at which delivery is complete.

Accumulator — Part of the logical-arithmetic unit of a computer used for intermediate storage, to form algebraic sums, or other intermediate operations.

Address — Label, name, or number identifying a register, location, or unit where information is stored.

Assembler — Translates input symbolic codes into machine instructions.

Bit — Abbreviation of binary digit; a single character in a binary number.

Buffer — Isolating circuit used to avoid reaction of a driven circuit upon its driving circuit.

Byte — Group of binary digits usually operated upon as a unit. Usually shorter than a word.

Clock — Time-keeping device used to synchronize the computer.

Data — Basic elements of information which can be processed or produced by a computer.

Hold — Function of retaining information in one storage device after transferring it to another device, in contrast to clear.

Instruction — Coded program step that tells the computer what to do for a single operation in a program.

Interrupt — Break in the normal flow of a system or routine such that the flow can be resumed from that point at a later time.

Jump — Depart from the normal sequence of executing instruction in a computer (synonymous with branch).

Memory — Storage. A device that holds information that can be extracted at a later time.

Processor — Device capable of receiving data, manipulating it, supplying results usually of an internally stored program.

Programming — Art of reducing the plan for the solution of a problem to machine-sensible instructions.

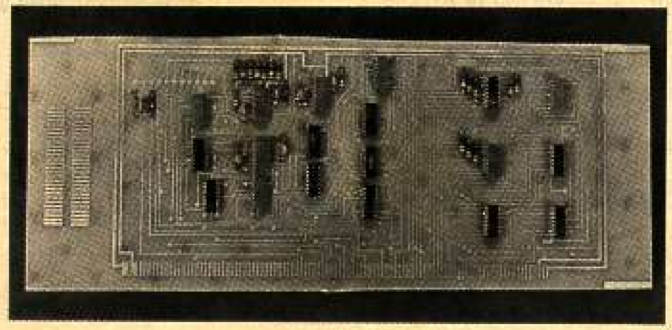
Register — Device for the temporary storage of one or more words to facilitate arithmetical, logical, or transferral operations.

Stack — Portion of a computer memory and/or registers used to temporarily hold information.

Subroutine — Set of instructions in machine code to direct the computer to carry out a well-defined mathematical or logical operation; a part of a routine.

Word — Set of characters that occupies one storage location and is treated by the computer as a unit and is transported as such. Word lengths are fixed or variable, depending on the particular computer being used.

Definitions were extracted from "Computer Dictionary" by Charles J. Sippl and Charles P. Sippl, published by Howard W. Sams & Co., Inc., The Bobbs-Merrill Co. Inc., Number 20943, 484 pages, $8.95 (in Canada $11.95).



Shown at far left is the display board atop the control board, with cables that connect to other boards. The central processor unit is shown in the center, and the control board at near left. Not shown is memory board, which holds 17 IC's.

MACHINE INSTRUCTIONS

Instruction	Binary Code (for instruction)	Octal	Comment
IN 6	11011011 (IN)	333,006	Bring data from input 6 and store in register A (accumulator).
MOV B,A	01 (MOVE) 000 (B) 111 (A)	107	Take A and move its contents to B.
IN 30	11011011 (IN) 00011110 (30)	323,036	Bring input 30 into accumulator
ADD B	10000 (ADD) 000 (B)	200	Add contents of A to B. Put results in A.
OUT 128	11010011 (OUT) 10000000 (128)	323,200	Transmit contents of accumulator to output 128.

are really very straightforward and easy to master. The procedures that are always used consist of the following:

Defining the Problem. This is by far the hardest part of the programming. Don't worry about the computer or the computer language when doing this part of the preparation. Simply decide what is required to do the job you want to accomplish.

Establishing an Approach. The computer and computer language have nothing to do with this step, either. It involves outlining a step-by-step procedure to achieve the desired results and getting it down on paper.

Writing the Program. Once you are familiar with programming, you will find that this step is the simplest. It is merely a matter of translating step 2 into the appropriate language.

There are many books available on programming. Some of them are quite good and are particularly useful for learning techniques such as flow programming, looping, etc. However, in essence, they can all be boiled down to the three steps above.

Software Example. To get a feel for what programming the Altair 8800 is like, let's go through a sample program, which is similar to the test program that we first went through to check out the computer operation. Assume that we want to take the data available from input channel 6 and input channel 30 and add them, placing the result in output channel 128. The machine instructions are shown in the box.

The first instruction simply stores the data from channel 6 in register A (the accumulator). The next instruction moves this data from register A to register B. This clears A for the next input. The third instruction brings the data from input channel 30 into the A register. The fourth instruction adds the contents of register A (data from channel 30) to register B (data from channel 6) and puts the results back into register A. The final instruction transmits the answer from A to output channel 128. Total computer time used to perform this operation with the Altair 8800 is 18 microseconds. To put it another way, the computer could perform 56,000 of these operations in one second.

The instructions could be entered into the processor in one of three ways. The first and easiest would be with the use of an assembler. This is essentially a piece of software that converts alphanumeric symbols to machine language (binary code). For example, the assembler would convert our first instruction (IN 6) to the correct binary code. The problem with using an assembler is that you need a computer terminal for an input device and the assembler itself requires about 6000 words of memory storage. If extensive program development is to take place, the assembler is a good tool to have.

The next easiest method of entering the instructions is with the use of the Very Low Cost Terminal featured in the December 1974 issue of POPULAR ELECTRONICS. With this terminal, the instructions could be entered by using the octal code. The procedure would be to write the program in assembly language and then enter the corresponding code for each instruction. This system, while not being as fast as the use of an assembler is less expensive.

The third method, using front panel entry, is of course inexpensive but time consuming.

This has been only a brief summary of the programming procedures for the computer. Complete programming information is provided with the Intel 8080 integrated circuit and with the Altair 8800 computer kit. ◈

EXPANDING THE COMPUTER

In describing the assembly of the Altair 8800 Minicomputer in last month's article, it was noted that the interior of the cabinet provides plenty of room for expansion. The room can be used to add many functions to the basic computer. For example, the present memory board in the Altair 8800 can be expanded with the addition of three 256-word memories (Kit 8802-MS available from the manufacturer, MITS at $34 per 256-word memory). Further additions require an expansion mother board having four connectors that can accommodate any four memory or input-output (I-O) cards. This expansion board (Kit 8800-EB) is available for $44, while a 4K dynamic memory card (Kit 8840-MC) costs $198. Various other kits—a vectored interrupt card and a real-time clock, among them—are also available.

MACHINE INSTRUCTIONS

Instruction	Binary Code (for instruction)	Octal	Comment
IN 6	11011011 (IN)	333,006	Bring data from input 6 and store in register A (accumulator).
MOV B,A	01 (MOVE) 000 (B) 111 (A)	107	Take A and move its contents to B.
IN 30	11011011 (IN) 00011110 (30)	323,036	Bring input 30 into accumulator
ADD B	10000 (ADD) 000 (B)	200	Add contents of A to B. Put results in A.
OUT 128	11010011 (OUT) 10000000 (128)	323,200	Transmit contents of accumulator to output 128.

are really very straightforward and easy to master. The procedures that are always used consist of the following:

Defining the Problem. This is by far the hardest part of the programming. Don't worry about the computer or the computer language when doing this part of the preparation. Simply decide what is required to do the job you want to accomplish.

Establishing an Approach. The computer and computer language have nothing to do with this step, either. It involves outlining a step-by-step procedure to achieve the desired results and getting it down on paper.

Writing the Program. Once you are familiar with programming, you will find that this step is the simplest. It is merely a matter of translating step 2 into the appropriate language.

There are many books available on programming. Some of them are quite good and are particularly useful for learning techniques such as flow programming, looping, etc. However, in essence, they can all be boiled down to the three steps above.

Software Example. To get a feel for what programming the Altair 8800 is like, let's go through a sample program, which is similar to the test program that we first went through to check out the computer operation. Assume that we want to take the data available from input channel 6 and input channel 30 and add them, placing the result in output channel 128. The machine instructions are shown in the box.

The first instruction simply stores the data from channel 6 in register A (the accumulator). The next instruction moves this data from register A to register B. This clears A for the next input. The third instruction brings the data from input channel 30 into the A register. The fourth instruction adds the contents of register A (data from channel 30) to register B (data from channel 6) and puts the results back into register A. The final instruction transmits the answer from A to output channel 128. Total computer time used to perform this operation with the Altair 8800 is 18 microseconds. To put it another way, the computer could perform 56,000 of these operations in one second.

The instructions could be entered into the processor in one of three ways. The first and easiest would be with the use of an assembler. This is essentially a piece of software that converts alphanumeric symbols to machine language (binary code). For example, the assembler would convert our first instruction (IN 6) to the correct binary code. The problem with using an assembler is that you need a computer terminal for an input device and the assembler itself requires about 6000 words of memory storage. If extensive program development is to take place, the assembler is a good tool to have.

The next easiest method of entering the instructions is with the use of the Very Low Cost Terminal featured in the December 1974 issue of POPULAR ELECTRONICS. With this terminal, the instructions could be entered by using the octal code. The procedure would be to write the program in assembly language and then enter the corresponding code for each instruction. This system, while not being as fast as the use of an assembler is less expensive.

The third method, using front panel entry, is of course inexpensive but time consuming.

This has been only a brief summary of the programming procedures for the computer. Complete programming information is provided with the Intel 8080 integrated circuit and with the Altair 8800 computer kit. ◈

EXPANDING THE COMPUTER

In describing the assembly of the Altair 8800 Minicomputer in last month's article, it was noted that the interior of the cabinet provides plenty of room for expansion. The room can be used to add many functions to the basic computer. For example, the present memory board in the Altair 8800 can be expanded with the addition of three 256-word memories (Kit 8802-MS available from the manufacturer, MITS at $34 per 256-word memory). Further additions require an expansion mother board having four connectors that can accommodate any four memory or input-output (I-O) cards. This expansion board (Kit 8800-EB) is available for $44, while a 4K dynamic memory card (Kit 8840-MC) costs $198. Various other kits—a vectored interrupt card and a real-time clock, among them—are also available.