Rod Hallen
Road Runner Ranch
PO Box 73
Tombstone AZ 85638

The PAIA 8700

For under $200, you can have the computer that's thoroughly examined in this article.

Although personal computing is still in its infancy, the multitude of products that have been produced, or will soon be introduced, is enormous. From the Altair and its contemporaries to the PET, TRS-80 and KIM-1, the hobbyist and small businessman have a large selection of computer equipment to choose from.

You can spend thousands of dollars or you can spend a few hundred dollars. Since you get what you pay for, there must be different levels or capabilities of equipment to explain this large price range. Let's look at the bottom end of the scale and investigate one computer that is available for well under $200 and see what it will do.

The PAIA 8700 is made by PAIA Electronics, Inc. (see Table 1 for address and prices). If you are into synthesized music, you are probably familiar with PAIA's extensive line of music synthesizer modules and systems. Write to ask them for their catalog. It contains many items that can be used in computer music systems.

The 8700 is the most compact computer I've seen (see Photo 1 to get an idea of what I mean). It is 6 x 10 x 1 inches overall and weighs less than a pound. These dimensions and weight do not include the power supply, which is an option and is external to the main unit as shown. The price of both is very reasonable.

Before discussing assembly and operation, let's examine its capabilities and limitations. First and foremost, this is not a general-purpose computer. It is intended for controller applications, and I will explore some of these after a while. The main limiter is the microprocessor: the MCS 6503 by MOS Technology, which is a little brother to the familiar 6502 used in the KIM-1, PET and Apple.

I call it a little brother because it is packaged in a 28-pin DIP (dual in-line package) and was designed for applications in which a lot of memory is not required. In order to put a 40-pin microprocessor into a 28-pin IC, something had to go. Part of

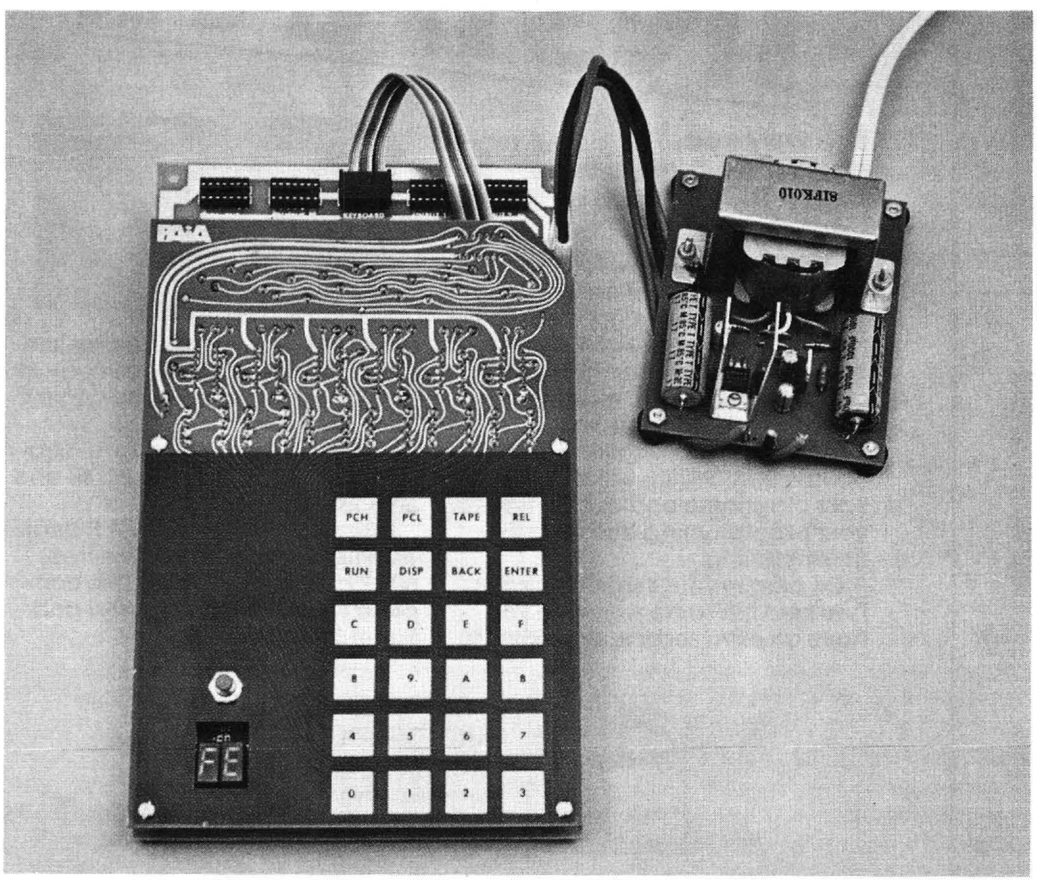


Photo 1. The PAIA 8700 and its optional power supply. The keyboard is mounted on top of the CPU board with 3/4 inch standoffs. The vacant IC sockets at the top are the input/output connectors. The expansion connectors are under the keyboard. Note the reset button and digital readout in the lower left-hand corner.

that something was the address lines A12 to A15, which means that the 6503 is limited to 4096 bytes of addressable memory ($2^{12} = 4096$).

This is why you wouldn't want to consider the 8700 as your general-purpose computer. Even if you could squeeze in a BASIC interpreter, you wouldn't have much room left for entering and running programs. You must program the 8700 in assembly language if you are going to use it as a controller.

ADDRESS	CODE	LABEL	INSTRUCTION	COMMENTS
0000	A9 00	BEGIN	LDA #0	CLEAR ACCUMULATOR
0002	8D 20 08	REPEAT	STA $0820	DISPLAY ACC
0005	A0 00		LDY #0	CLR Y
0007	A2 50		LDX #$50	SPEED SETTING IN HEX
0009	C8	LOOP	INY	DELAY LOOP
000A	D0		BNE LOOP	BRANCH UNTIL Y=0
000C	CA		DEX	CHECK SPEED
000D	D0 FA		BNE LOOP	BRANCH UNTIL X=0
000F	F8		SED	SET DECIMAL MODE
0010	18		CLC	CLR CARRY
0011	69 01		ADC #1	ADD 1 TO ACC
0013	4C 02 00		JMP REPEAT	DO IT ALL AGAIN

Program listing. Loading and running this listing in the 8700 will cause the display to count from 00 to 99 continuously. The number stored at 0008 (50) determines how fast the count goes. Entering a smaller number here will make the count go faster. Changing 000F to D8 will change the count to hex. Can you tell why? This is, of course, written in 6502 assembly language.

Assembly and Testing

The 8700 comes in kit form, and our first task is to put it together and test it. The 60-page manual supplied with the kit makes both of these chores easy to accomplish. All part identifications are screen printed on the boards, which also helps.

Parts are installed and checked off one at a time. Mounting and soldering are explained, and a layout diagram is provided. Two circuit boards are involved in construction—the main (CPU) board and the keyboard, which, after construction, are mechanically bolted together and electrically connected with a 14-conductor ribbon cable that ends in a 14-pin plug for easy disassembly. I am an experienced kit builder, and my total construction time was one hour and ten minutes. Even if you take twice that long, this is still an easy one-evening project.

Here are my few negative thoughts on the kit. I guess Heath has spoiled me because I was surprised that no solder was supplied . . . especially since the nearest solder store is 25 miles away. I also like tinned, soldermasked boards. The main board is tinned but not soldermasked, while the keyboard is bare copper and has started to tarnish. I like tinned boards because they are easier to solder to and soldermasking cuts down on the chances of inadvertent solder bridges between closely spaced components.

Neither board is densely populated (see Photo 2), and if you're careful, bridges shouldn't be a problem. One design procedure that helps in this regard is the lack of traces running between IC pads on the solder side of the board. All of the ICs on the main board are mounted in sockets, but those on the keyboard are not. I prefer sockets for all ICs, and since those on the keyboard are CMOS it is fairly easy to destroy one during installation and have a devil of a time removing it again. You can provide your own sockets, as I did.

The only other complaint I have is that the power connector is an unusual type, and unless you buy the optional power supply you don't get the matching plug. I soldered wires from my supply directly to the connector posts.

The 8700 monitor PROM (PIEBUG) contains a "Self-Test Micro-Diagnostic" routine that is designed to help you troubleshoot the unit when you are finished putting it together. It is intended to be used in conjunction with an oscilloscope, but I ran through it with a logic probe and satisfied myself that everything was OK. You can't look at pulse widths or duty cycles that way, but at least you know whether the logic level is steady or pulsing and whether it is high or low.

Large, simplified schematics and flowcharts also help with troubleshooting. A complete source and object listing in hex of the firmware monitor (PIEBUG) with comments is a welcome addition. A memory map and interrupt, break and reset vectors are provided to make programming easier. It is probably well to mention that assembly-language programming of the 8700 is done in hex.

I hope that I haven't scared anyone with this talk about troubleshooting. It probably won't be necessary. The whole layout is very simple and even the novice builder shouldn't have any problem. The manufacturer stands ready and willing to provide help by mail or phone.

One unusual feature of the 8700 is the keyboard. Look closely at Photo 1. If the keys look like they are painted on the board, that's because they are! The keyboard has no moving parts. The capacity of your finger touching the desired keypad triggers an IC gate for data entry. It takes a little getting used to but quickly seems like the natural way to do it. The lack of moving parts should make for a long, trouble-free life. If you buy the cassette option, the keyboard will beep along with each key entry.

In addition to the *PAIA 8700 Assembly and Using Manual,* the kit also comes with the *MOS Technology MCS6500 Programming Manual.* This big 240-page 8½ x 11 inch book is a complete course on assembly-language-programming the 6500 family of microprocessors. I say family because there are almost a dozen different versions of the 6502. All variants, including the 8700's 6503, use the same instruction set. It is, therefore, possible to write programs for the 8700 on a larger machine and then transfer them. This would be especially desirable if the larger machine were running a 6502 assembler.

Note the options in Table 1. The cassette interface allows the recording and playback of your programs. The optional music programs listed are on cassette tape. I didn't purchase the power supply because I already had the necessary voltages available. The current requirements aren't given in the manual, but I measured mine at +5 volts at 400 mA and −9 volts at 45 mA. These must, of course, be regulated, and you can build a suitable supply for a reasonable price if you already have most of the parts. Otherwise, the power supply kit is a good buy.

The PAIA 8700 Package

I've discussed the 6503 microprocessor, but let's explore it just a little bit more. As I've said it is completely software compatible with the 6502 but has certain limitations because of its intended use. Missing, in addition to the address lines (A12 to A15), are the Ready, Sync and Set Overflow leads. This lack of leads will probably not limit system use, except for the curtailed addressing capability.

The 8700 has room for 1K of RAM and 1K of ROM on the CPU board. Provided with the kit are the PIEBUG monitor ROM (a 256-byte 1702) and 512 bytes of RAM (four 2112s). Another 512

PAIA 8700 Computer Kit	$149.95
Options:	
Power Supply Kit	24.95
Cassette Interface Kit	22.50
Firmware:	
MUS-1 Music Generator PROM	22.00
Software:	
POLY—Synthesizer Program	
DRUMSYS—Drum Operating Program	
Seque 1.0—Ultimate Sequencer Program	

Software includes cassette tape, documentation and interfacing details for $4.95. Available without tape for $2.
Prices do not include postage.

PAIA Electronics, Inc.
PO Box 14359
Oklahoma City OK 73114

Table 1: The 8700 and its options, prices and manufacturer.

bytes of RAM can be added to the board for about $16 at current market prices. The MUS-1 PROM (see Table 1) contains a relatively complex music generator that can be installed on the CPU board. See the paragraphs about the expansion connectors for information concerning adding even more memory.

In order to make practical use of any computer, it is necessary to get information into and out of it. That is the purpose of I/O (input-output) ports, and the 8700 is well supplied with them. The on-board ports interface the keyboard and the digital display. The keyboard is operated by the capacity of your fingers and contains eight PIEBUG commands as well as the hex characters 0 1 2 3 4 5 6 7 8 9 A B C D E F.

The only precaution in the use of the keyboard has to do with the TAPE command. The TAPE key should never be touched if you don't have the cassette option. Otherwise, you'll destroy any programs that you have in memory. The manual emphasizes this by warning that this will load your memory with garbage and eat your lunch!

Just to the left of the digit 8 in Photo 1 is a push button. This is the reset button, which is used to gain control of the processor each time that the computer power is turned on or when a program has gone astray. This latter problem will usually be caused by an incorrectly written program step that forces a continuous loop or branches to nonexistent memory.

Below the Reset button are two digital readouts that serve as a visual output port . . . sort of like a video screen that can only display two characters at one time. The two characters displayed are the last two that you have entered from the keyboard or the contents of a memory location if you key DISPLAY, ENTER, PCH or PCL (more on this later).

A latched eight-bit parallel output port is used to control external devices or processes. A strobe is also provided to indicate when the port has been loaded and when valid data is available on the data lines. Two pages are devoted to describing the operation and use of the output port. A simple program that will cause the port data lines to count from 0 to 255 in binary as an exercise is written. Information is also given on how to use the port data lines to control CMOS or TTL gates and transistor relay driver circuits.

Two eight-bit parallel input ports are also provided. These can be read by the processor, and the data derived can then be used by the program. The MCS 6500 family of microprocessors addresses I/O ports as memory locations. This is similar to the way the 6800 handles I/O. If a memory read instruction addresses an input port, then the data read will be whatever is tied to that port. Various methods of inputting information are discussed. Input ports rate two pages of the manual.

A cassette port is available if the cassette option is purchased. The option consists of parts that are installed on the

✔ Reader Service—see page 179

CPU board. Recorder motor control is one feature that is provided. I have recently written for the cassette option but I haven't received it yet.

And finally, the data and address buses are available through expansion connectors. This will allow you to add more memory, peripherals, ports or whatever. While the data lines will handle five CMOS or one LS TTL load and the address bus will handle one TTL load, it is probably best if all lines extended off the board are buffered.

The NMI (non maskable interrupt), IRQ (interrupt request) and R/W (read/write) leads, as well as some other control lines, are also available at the expansion connectors. All of the I/O port and expansion connectors are 14-pin IC sockets. Connection to them is made with 14-pin plugs that have solder tabs on the top or ribbon cables attached to them. Many companies make both types of these plugs.

The monitor program contained in the PIEBUG ROM has some interesting features. I mentioned earlier the "Self-Test Micro-Diagnostic." In addition, each of the keyboard command functions is the result of a routine in PIEBUG.

Touching four hex keys and then DISPLAY will cause the contents of the location pointed to by the four-key entry to appear on the display. If you touch 001A and DISPLAY, then the display will hold the contents of that location. Touching ENTER will step the display counter by one and the display will hold the contents of the next location (001B). In this way, you can step through memory examining programs or data. To recap, the starting address is keyed in followed by DISPLAY and then the address count is incremented with the ENTER key.

Any time you want to change the contents of a memory location, you bring the location to the display (as described above), touch the new data and then ENTER. The new data will be entered and the address incremented by one to let you see the contents of the next location. The BACK key steps the address pointer backward by one with each touch.

After a program has been entered using the above keys, you touch in the starting address, DISPLAY and RUN. There is a short assembly-language program in the manual that is used as an example of how to enter a program (see program listing). This program causes the display to count from 0 to 99 and then start again. It will continue until you hit RESET. By making a few changes you can speed up or slow down the count. Make another change to place the count in hex instead of decimal. I believe that examples, even simple ones such as this, are the best way to teach a new subject.

The PCH and PCL keys display the current program counter. The manual calls these Pointer High and Pointer Low, but I prefer to call them Program Counter High and Program Counter Low, which is what I think PCH and PCL really stand for. The TAPE key has already been mentioned.

The REL key is an innovative time-saver. In writing assembly-language programs for the 6500 family, it is often necessary to figure relative addresses. A relative address is the number of address locations forward or backward a program must jump or branch to satisfy certain conditions.

Normally the programmer has to figure this out by hand. PIEBUG has a routine that does it for you. Whenever you enter an instruction that requires a relative address, you enter the absolute (actual) address of the jump or branch location and touch REL. The relative address will appear on the display and will also be entered at the correct memory location when you touch ENTER to move to the next entry address.

One plus feature of PIEBUG is a debug breakpoint routine. Any time that you want to stop a program at a certain point to examine the condition of the various registers, you substitute 00 for the op code at that point. When PIEBUG encounters a 00 op code during the running of a program, it will store all of the 6503 registers in certain designated memory locations, cause the display to read bb and return control to the keyboard. By examining these designated storage locations you can determine how well the program is running at that point. You can also change any of the registers or memory locations and then start the program running again from there.

System Applications

That pretty well defines the hardware and firmware attributes of the PAIA 8700. There is certainly an awful lot crammed into such a small package. Your next question is, "OK, but what can I use it for?" Let's skim over some of the things you might want to do with the 8700.

The most obvious answer is, "Make music!" PAIA manufactures music synthesizers, and, strangely enough, some of their equipment incorporates a mounting space for the 8700. But your computer can make music without a synthesizer. For now I'd advise you to order their catalog or go back over your computer magazine back issues for music articles.

Another application that comes to mind is printer buffering for a bigger computer. Your processor moves along at a microsecond clip, but most printers are a thousand or more times slower than that. Instead of having the processor wait while each individual character is handled by the printer, why not let your big machine dump blocks of characters to the 8700 and then go back to computing while the 8700 feeds the printer?

This wouldn't gain you much in a situation where the majority of the program is concerned with printing, but if the program computes, prints, computes, etc., for a while, a lot of time could be saved. My biorhythm program is an excellent candidate for such treatment. While time is not money in most hobbyist situations, if you are impatient or a businessman trying to make money, it is.

I recently saw a program in one of the magazines to make a KIM-1 run like a clock. This same program with a little modification would probably run on the 8700. This is OK, but digital clock chips are inexpensive, and you wouldn't be able to use the computer for anything else since it would always be running the clock program. I imagine that many other programs that are written for the KIM will also run on the 8700.

Another possibility is multiprocessing. Multiprocessing involves two or more processors working together on one chore. Many tasks that would fit in this category come to mind. Suppose that you want to control an organ, music synthesizer or other instrument. Why not let your big machine do the programming, arranging, scoring and program execution. The 8700 could be the actual controller of the instrument, and it in turn would be under the control of the larger machine.

This same analogy fits many other situations. You might not want to tie up your personal computer controlling a solar energy system, your fire and burglar alarms and other chores that really call for a dedicated controller. That is what the MCS 6503 (and the PAIA 8700) was designed for. It could monitor various conditions through its input and output ports and interrupt the main computer when it decided that a situation was developing that needed to be taken care of.

Another possibility is a programming trainer. If each student had an 8700, he could write, debug and run sample programs in assembly language. The knowledge thus gained could subsequently be used on larger machines that use the 6502.

Conclusion

While doing some research for this article, I came across a review of the 8700 in the February 1978 issue of a magazine called *Electronics Australia*. The anonymous writer was generally complimentary, although he criticized the manufacturer for not providing enough information for the novice. He felt that the supplier's message seemed to be, "Here it is, you figure out how to use it." Actually, the same can be said about almost any personal computer maker.

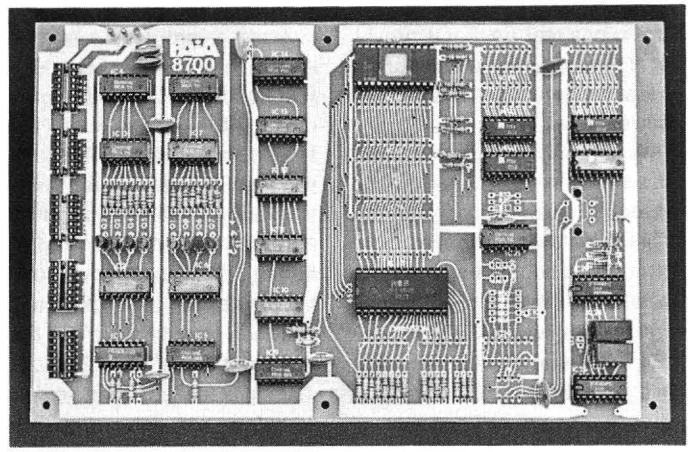


Photo 2. The PAIA 8700 CPU board. Notice the uncluttered layout. The 6503 CPU is just below the center of the board and the PIEBUG ROM is top center. The vacant socket positions from the center to the right-hand edge are for the cassette option, expansion connectors and additional RAM and ROM. The power connector is the three pins in the upper left corner. *(Photos courtesy of PAIA)*

At least PAIA attempts to explain the rudiments of programming and usage in their manual. The MOS Technology 6500 programming manual that they provide is also a great help for those of us who are not too familiar with the 6502 instruction set. I didn't get any of this type of information with my thousand dollar plus 8080 machine.

PAIA's foremost intention for this computer is, of course, the generation of music. Music is their business, and the fact that the 8700 fits into many of their synthesizers indicates their primary goal. But don't let that hide the many applications that are possible. I've touched on a few, but you should be able to think of many more.

I have always shied away from control applications with my SOL because I use it a lot for program writing, manuscript printing and general experimentation. With a separate dedicated controller I can investigate more areas of the world of personal computers at one time. I think that I'll work on the music and printer buffer ideas first.

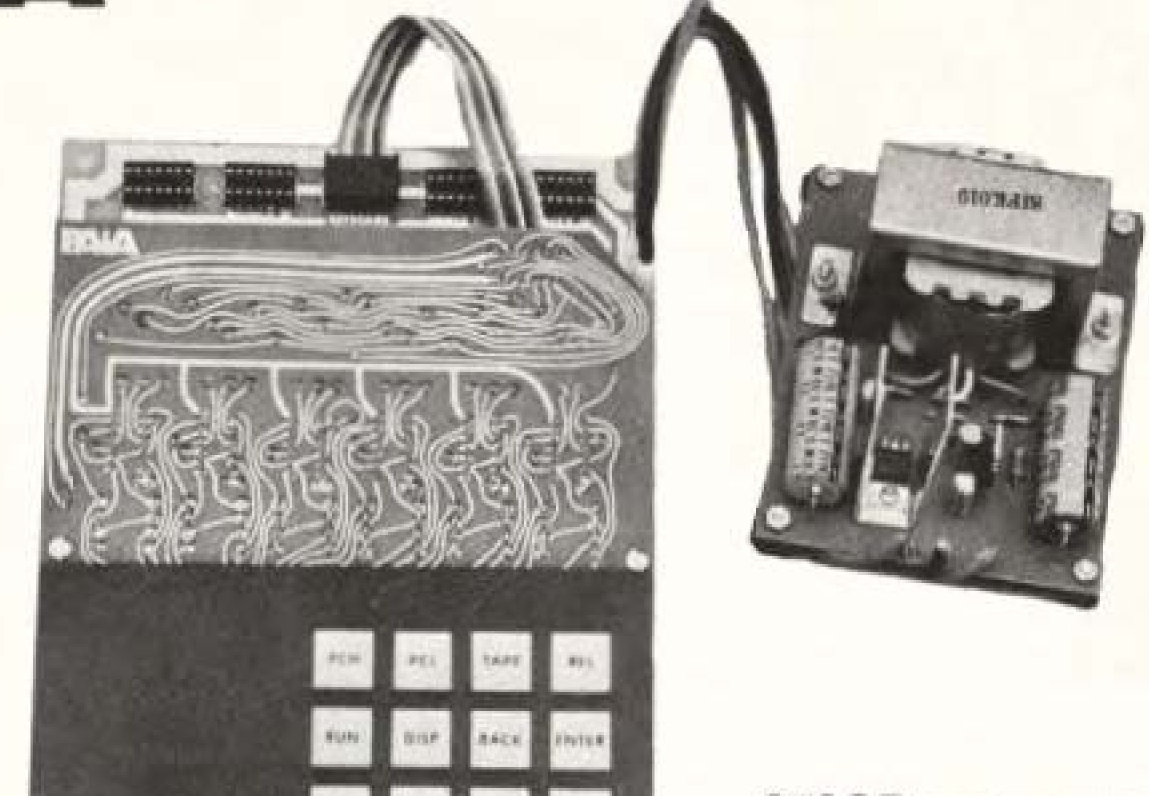
Whenever I consider buying something new, whether it is computer related or not, I try to decide if it is worth what I'm going to have to spend for it. This takes into account such things as original cost, probable lifetime, utility and personal desires. I do the same calculations when evaluating a product or service for a review article such as this.

Is the PAIA 8700 worth the price asked for it or is there something else available at or near the same cost that will do the job better? Taking into account the limitations of the 6503 and considering its possibilities as a dedicated controller and its low price, I believe that it is a viable purchase in three areas.

First, for the novice who is short on cash, it makes a nice learning machine. He can move up to something more sophisticated later and then use this as a controller. Second, this is a good starting point for the music buff who would like to try his hand at computer music. Using the modules and systems available from PAIA or building from scratch, I believe that complex, sophisticated results are possible. Third, the dedicated controller applications already mentioned enhance its appeal.

I really would like to find a 6502 assembler that will run on my 8080 machine and then programming would move along at a greater pace, but in the meantime I'm learning 6502 assembly-language programming little by little. Taking into consideration all that we've discussed, is there a place in your plans for a compact, inexpensive, useful personal computer product? If so, consider the PAIA 8700. ■

PAIA 8700 COMPUTER/CONTROLLER



8700 Processor: 6503 MPU, Wear free "Active Keyboard", Micro-Diagnostic®, Extensive documentation, Fully Socketed.

Piebug Monitor: User Subroutines, Relative address calculator, Pointer High-low, Back-step key.

Cassette Interface: Load & Dump by file #, Positive indication of operation, Tape motion control.

The Answer For... Student, Hobbyist, Manufacturer.

An exceptional price on an applications oriented 6503 based microprocessor system featuring: 1K bytes RAM locations (512 bytes supplied), 1K bytes ROM locations (256 byte monitor included), two 8 bit input ports, two 8 bit output ports, one latched and one buffered.

A 24 key touch operated keypad is used by the monitor to allow entry and execution of user programs as well as controlling features not normally found on low-cost single board computers; including a relative address calculator that completely eliminates this normally tedious hexadecimal calculation and back-space key that eases entry and editing of programs. Pointer High and Pointer Low keys allow the 8700's twin seven segment displays to serve the multiple functions of indicating both address location and data.

The 8700 fits in a space reserved in the 8782 encoded keyboard's case. PAIA software support available for Electronic Music Synthesizer interface.

8700 COMPUTER/CONTROLLER KIT $149.95 wt. 4 lbs.

CASSETTE INTERFACE OPTION

This is one of the most reliable and easy to use cassette systems that we've seen. A single LED indicates proper cassette volume control setting and provides a positive indication of data flow. We've even made software and hardware provision for tape motion control (relays must be added). The CS-87 option fits entirely on the 8700 circuit board and consists of the POT-SHOT PROM and a handfull of additional components.

CS-87 Cassette Interface Option $22.50 plus postage

POWER SUPPLY

The logical choice to power the 8700. Fully regulated 5v. at 1 amp, —9 volts @ 300 ma. Also provides a 60 Hz. output for real-time clock applications.

PS-87 Power Supply Kit $24.95 wt. 3 lbs.