RAM, ROM, EPROM BOXES, RAM BOXES, and the HP-41C

With all these new add-on memory options and the growing interest in assembly language programming for the HP-41 there are a lot of questions popping up about how these devices are used and how they are useful. I hope to cover some of the questions you may have about this exciting area of programming.

First some basics, a brief description of the types of memory 'chips' involved:

RAM

'RAM' stands for 'Random Access Memory'. Ram is charactized as memory into which data can be written and read from at random, without any particular special proceedure. Each bit of information is stored in a sort of 'flip/flop' consisting of two or more transistors each. As a result, ram must always have power supplied in order to 'remember' the data written to it. The instant power is gone, so is the memory.

ROM

'ROM' stands for 'Read Only Memory'. Rom is memory that has had its data (or programs) built into it during the manufacturing process, and now can only have it's data read. It has, however, the advantage of not needing power to maintain the data, so it is very useful in applications where the data (or programs) never need to change, and/or complete power removal is necessary. The data is impressed on the chip with a layer called a mask, and this mask must be custom engineered by the manufacturer for each data pattern desired. The charge for this service varies by size and manufacturer, but is generally several thousand dollars, and additionally, some minimum quanitity must be purchased, or a higher mask charge applies.

EPROM

EPROM stands for "Erasable, Progammable, Read-Only Memory". Read-only memory we covered above.

Programmable says that the ROM is to be programmed after the manufactoring process. Erasable means that the data can be removed for re-programming by means of short-wave U.V. light. Eproms have a window in their lids to allow the U.V. light to enter. This window must be covered at all times, except during erasure, as prolonged exposure even to room light will erase it. The data is stored as a charge on an internal capacitor, one for each bit. The leakage on this capacitor is very low to get the 5-10 year program storage life, so in order to program it, voltages higher than the normal operating voltage of 5 volts must be applied in order to 'break-down' this insulation and get a charge into this capacitor. Of the current eproms, that voltage Z^R*^B2=1QM +/- 1

volt for the 2716 and 2732; and 21 volts +/- .5 volts for the 2732A and above. The 2732A and newer eproms use a much denser silicone layout, and, as a result, the voltages are lower, and more critcal. The eprom is one of the handiest types of memory around, being permanent, like rom; but field reprogramable. This allows applications where 'permanent' programming can be generated quickly and, in lower volume, much more economically than rom with it's mask charge and volume requirments.

RAM, ROM AND THE HP-41

The HP-41, unlike most modern micro-computers, keeps it's data storage seperate from it's program storage. By program storage, I am refering to the machine code or 'M-Code' that the micro-processor in the HP-41 executes directly, not the user entered programs, which are stored in the data storage area (registers). Program storage is in 10 bit words with a 16 bit address for a maximum addressability of 65536 words (64k or 64*1024). I will refer to this area of memory as the 'rom address area'. The 41C operating system takes the first 12k, 4k is skipped, and 4k is reserved for the diagonstic rom, leaving 44k for expansion. Of this, the top 32k is divided into four 'ports' of 2*4k each, while the remaining 12k is used for the timer module, the printer M-Code and the HP-IL interface M-Code.

HP-41 data storage, on the other hand, is in 56 bit words or 'Registers' with a 10 bit address for a maximum addressability of 1024 registers. I will refer to this area as the 'ram address area'. Instead of being addressed by the HP-41 micro-processor internal program counter, the ram address area is treated almost as an I/O device. The address is computed and presented in the micro-processor 'C' register. As this register is 56 bits long, the HP-41 had a possible ram address space of 2^56 or 7.2057940379 E+16 registers, each containing 7 bytes for a grand total of 504,403,158,265 MEGA-bytes of DIRECT ram address. This is greater than any computer I have ever heard of, and totally riduclous in the context of a hand held computer. H.P., through the evolution from the HP-35, had settled on a 10 bit address, and this was the intended addressability of the HP-41. However, operating system bugs and compromises caused a 9 bit address to be the maximum usable as registers. The famous 'BUG 2' machines were capable of the full 10 bit address range for registers. H.P. has since come back with 'Extended Memory' which has recovered the 'lost' 512 registers. In light of the uses to which the HP-41 is now being put, the 10 bit address is too small, but, as the existing H.P. ram chips would not respond properly to an increase of address bits, we are stuck with the current 1024 maximum. Of the ram address area, the first 16 registers are used by the operating system as the stack, alpha register, and misc. system storage.

The registers from 193 to 512 are the user registers particitioned into user program and data. The remaining registers (17 - 192, 513 - 1024) are used by the Extended Function / Extended Memory modules with several gaps of 1 to 16 registers because of operating system anomlies.

Because the HP-41 internal micro-processor program counter is used to address the rom address space, and the internal 'C' register is used to address the ram address space, it is impossible for the HP-41 micro-processor to execute M-Code from it's ram area. Therefore, M-Code programming cannot be done and executed in the HP-41 without the addition of some external device (ram or eprom box).

User code programs can be written and executed on the HP-41, because the operating system reads the codes one at a time out of ram and looks them up in a table in order to interpet them. This is call 'Interpetive' execution, and the HP-41 operating system allows this type of program to exist in both it's rom and ram address space, although they can be entered only in the ram address space. In the rom address space, additional restrictions and 'overhead' are necessary in order for the operating system to locate and interpet it, instead of trying to execute the code directly (CRASH!).

RAM BOXES

A typical Ram box consists of 16k of ram and usually some amount of rom (eprom) storage. The ram is configured to appear in the rom address space of the HP-41. The ability to disable the ram is essental in order to avoid crashing the HP-41 on first power-up, as it hardly expects to find the 'randomness' of un-initialized ram in its stable rom address area.

Once the ram has been properly loaded and initialized, then it can be placed 'on-line' and used for M-Code programming and execution. This permits execution of user writted M-Code by the HP-41. However, as you remember, there is no particular way for the HP-41 micro-processor to write to its rom address area so several techniques have been used to make the processor write to it's 'ROM'. Among the methods I have heard of are using one of the 16 'nop' class instructions to trigger a write in the ram box, and intercepting a user function (SIGN) to trigger the write; with the ten bit data and 16 bit address in the internal 'C' register in both cases.

In order for a RAM box to be useful, it must contain a back-up battery in order to maintain the programs when disconnected from the HP-41. 16k appears to be the best working size for the ram box, and additionally, some eprom storage is needed for most designs.

EPROM BOXES

This is my favorite subject, for some reason.

To begin with, we will go through a history of eprom boxes as I remember it. After deciphering and publishing the HP-41 bus interface specifics in April 1980, I got wrapped up in my career, and had no time to pursue it further. But, sometime in November-December of that year I received a call from Americian Micro-Products, Inc., referenced to me as someone knowlegdable on the HP-41 buss by Richard Nelson.

They were interested in building an Eprom storage device for the HP-41 and wanted my help getting started. I refered them to my article and walked them through the buss through several phone calls, and gave them Charles Close as someone who knew the rom structure. I thought little more of it until the PPC Rockville Conference in March, 1981. There I saw an H.P. Eprom box for the first time, running the final revision of the PPC Rom. It was huge (phone book size) and power hungery. Then I saw the AMI box. It was smaller and power hungery. I approached them about using some techniques I knew to increase battery life, but was met, for some reason, with a 'go away, kid, you bother me' reaction. I talked briefly with Richard about it, licking my wounds, when he suggested if I don't like theirs, build one of my own. The additional carrot of a clandestine loan of the PPC rom in eprom from the H.P. box for me to copy convinced me, for, like all PPCer's, I couldn't wait for my rom to arrive. I worked hard and fast in my spare time, and by the June Mini-Conference in Chicago (which I was unable to attend, but my box did), I had a working model in close to final size. The battery life was not too bad, and it seemed to work most of the time.

The one thing that still bothered me was that external battery. If it went while the HP-41 was running a program, there was no warning before the crash. I knew that in order to avoid crashes, the eprom box had to work from the HP-41 power supply. This was not easy, as the eprom set, when powered up, can draw as much as .5 amps, and by experimentation, I had found that the HP-41 will supply only 30-40 ma at maximum.

By using some crude eprom switching techniques, and translating the circuit to cmos logic, I got the power draw down to 20 ma, and did not need a power switch.

I was very excited until I talked to a friend in the H.P. lab who looked up the specs, and found that the power supply was rated at 20 ma. max, and the prosecessor took 10 ma of that. The bottom line was that in order to avoid damage to the HP-41, my box had to draw less than 2.5 ma. I almost gave up, until the idea hit me of pre-fetching the data from the eprom and storing it internally until the slow HP-41 processor wanted it. This gave me the break-through I needed (and possibly my first patent). Power draw on the first production units ran 1.6 ma average and 2.4 maximum. After futher investigation, it turns out that 1 ma of the total power was used in driving the signals up and down that cable, so the newest model eprom box in the card reader shell draws only .4 ma running with a 1.2 ma max (running M-Code). Reduced parts count and shorter traces account for the .2 ma additional savings.

An eprom box (for the HP-41) is a device in which eproms are interfaced to the HP-41 as part of it's rom address area. It needs no batteries, as it can run from the HP-41 power supply, and can maintain it programming when removed from that power.

Eprom boxes are most useful for small volume, time critical, or where frequent changes are needed to the programs. They are not nearly as handy as ram boxes for programming M-Code as the eproms must be removed and programmed externally, but they are far better for running finished M-code, and user code programs, since they are smaller, and cannot 'crash' thier programming the way ram sometimes can.

I hope this helps you understand this area better, as I must now call Richard and transmit this to him before I fall asleep (Deadlines!).

Jim De Arras
Vice-President, R & D.
Hand Held Products, Inc.
3928 Margate Drive
Richmond, Virginia 23235
(804) 272-9285

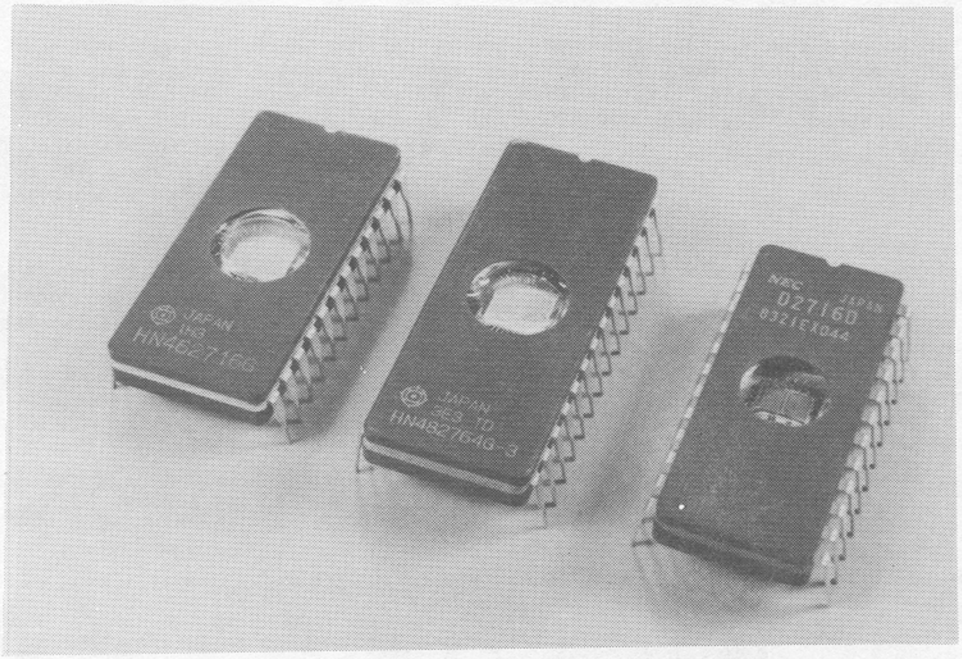

Figure 1 — Typical EPROMs. A 2764 is flanked by two 2716's of two different manufacturers.

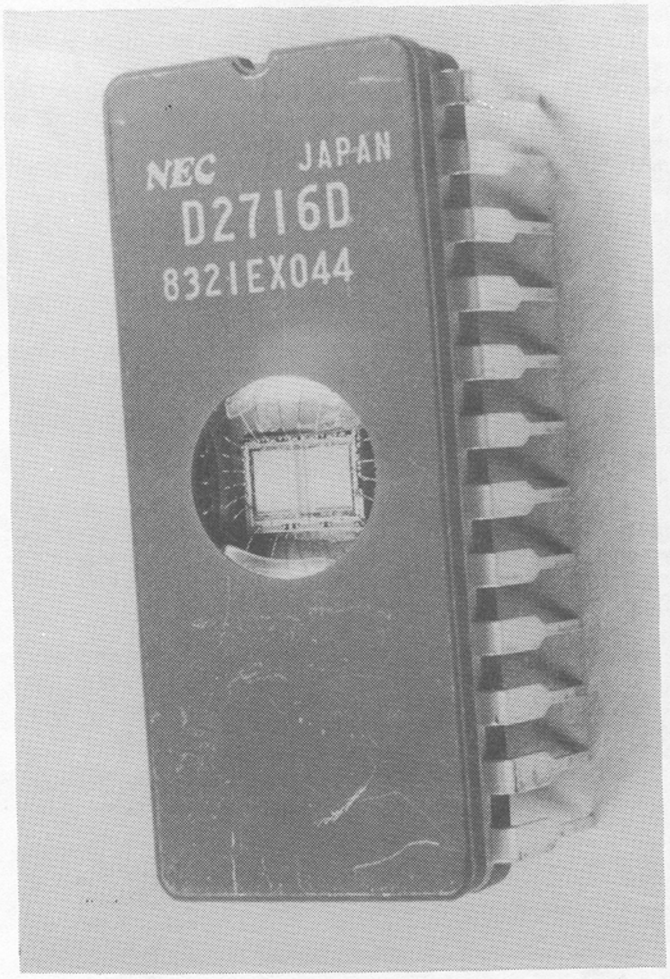

Figure 2 — 2716 EPROM close up. The "glass" window should be covered by a label.

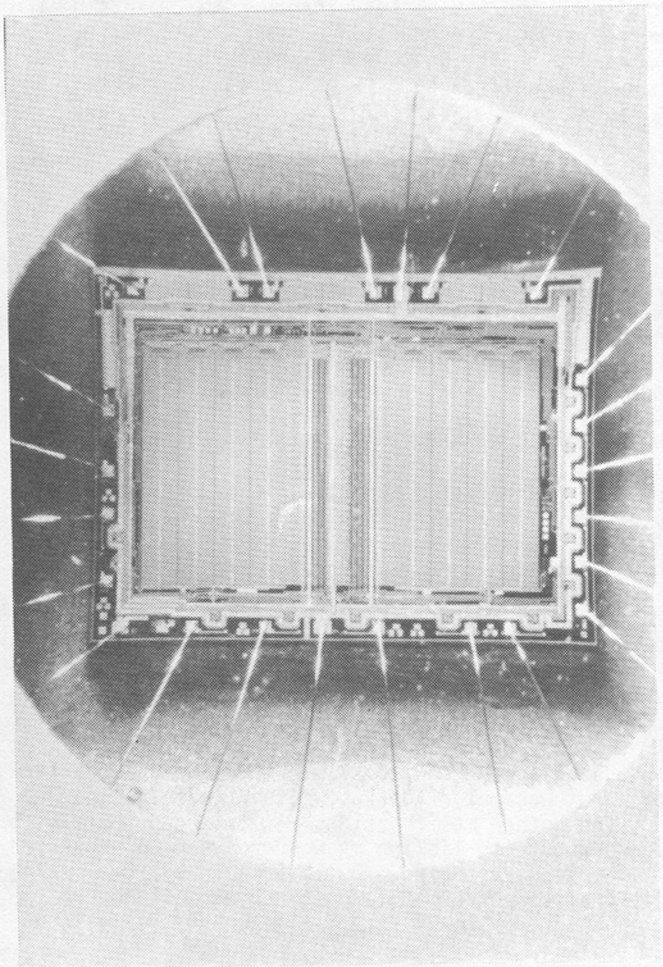

Figure 3 — The same 2716 shown at left, very close up. The bond wires are 0.001" in diameter.

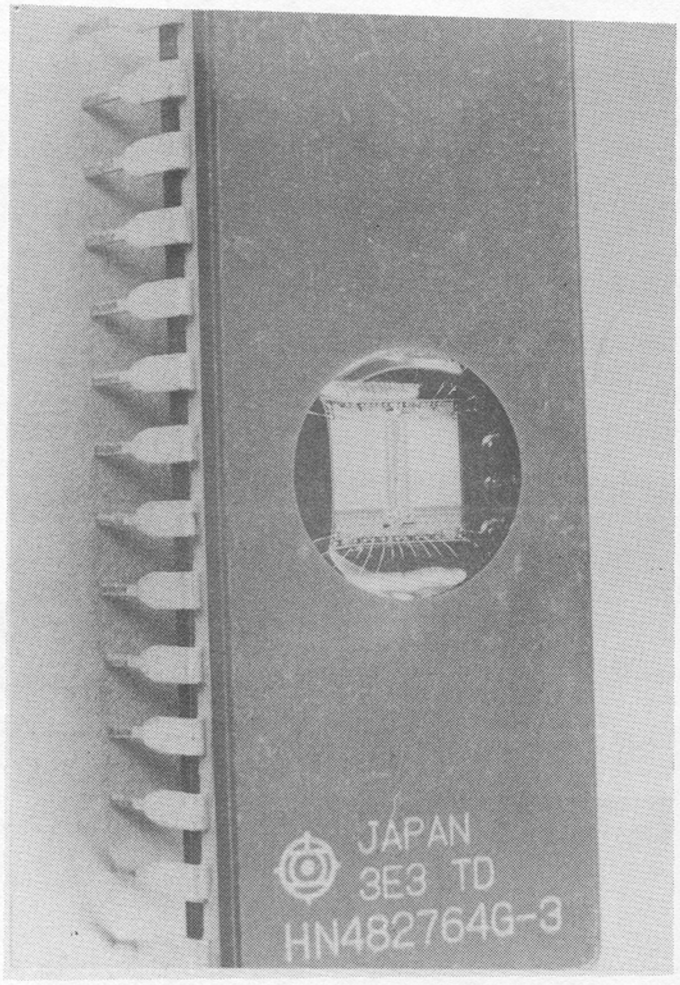

Figure 4 — 2764 EPROM, four times the capacity as the 2716.

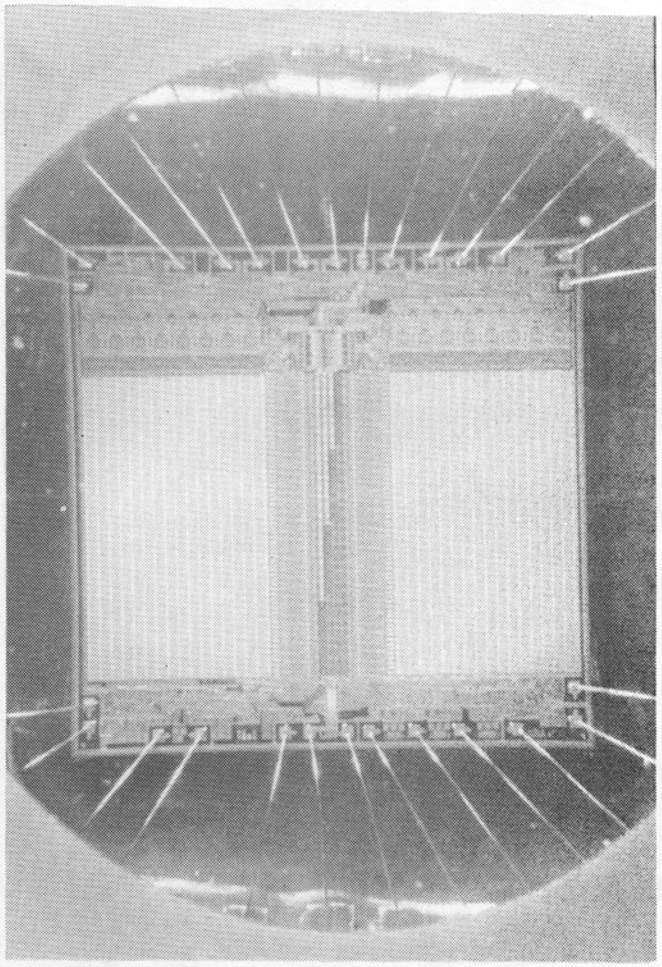

Figure 5 — 2764 EPROM, very close up.

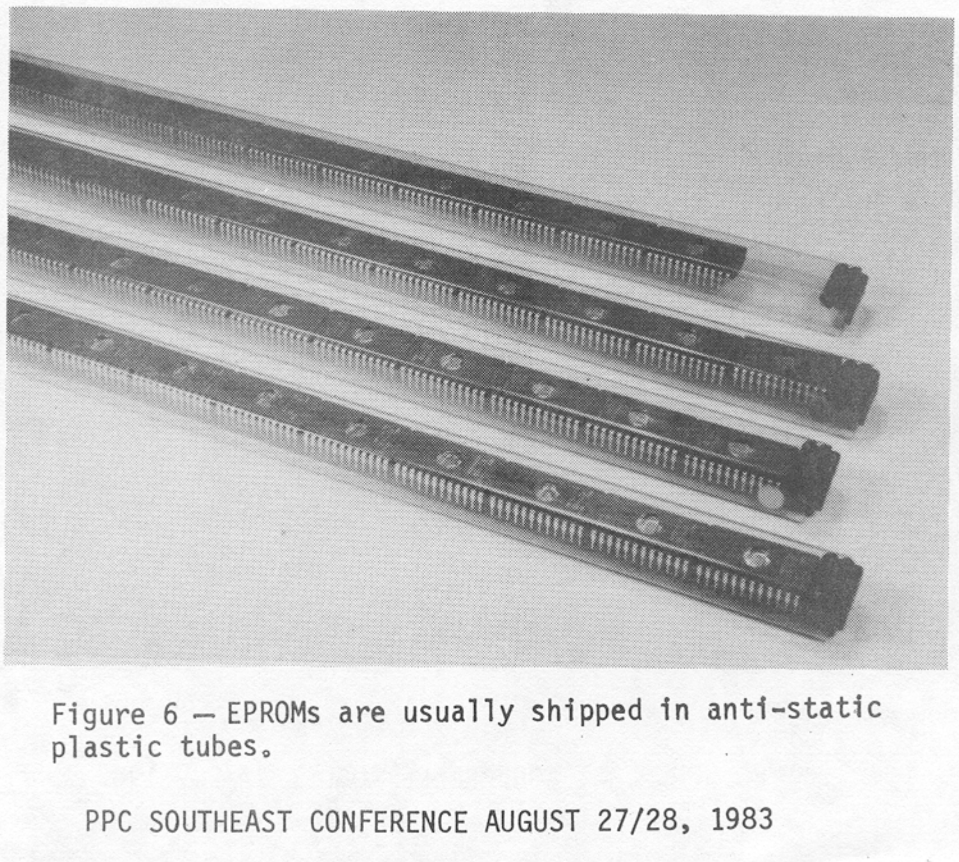

Figure 6 — EPROMs are usually shipped in anti-static plastic tubes.

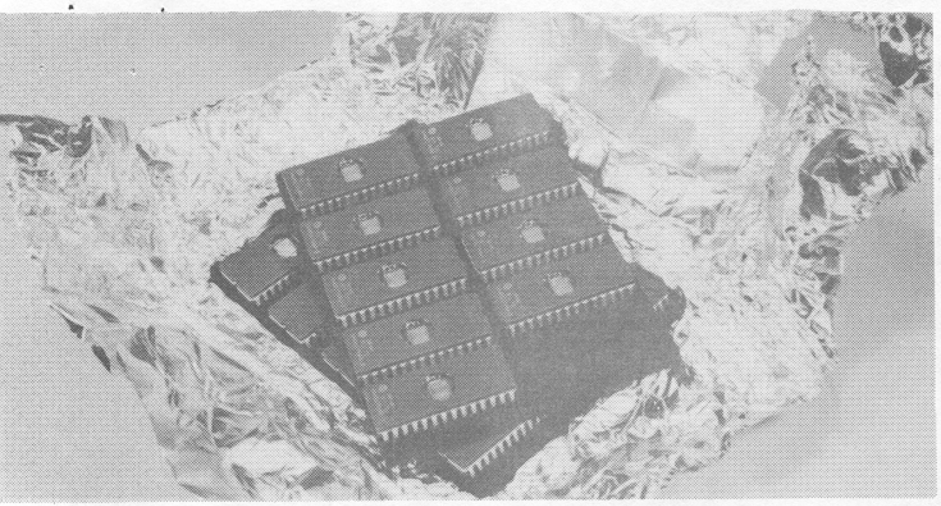

Figure 7 — EPROMs are usually kept "stuck" into a black conductive foam sheet. Here aluminum foil is wrapped around the EPROMs for static protection.

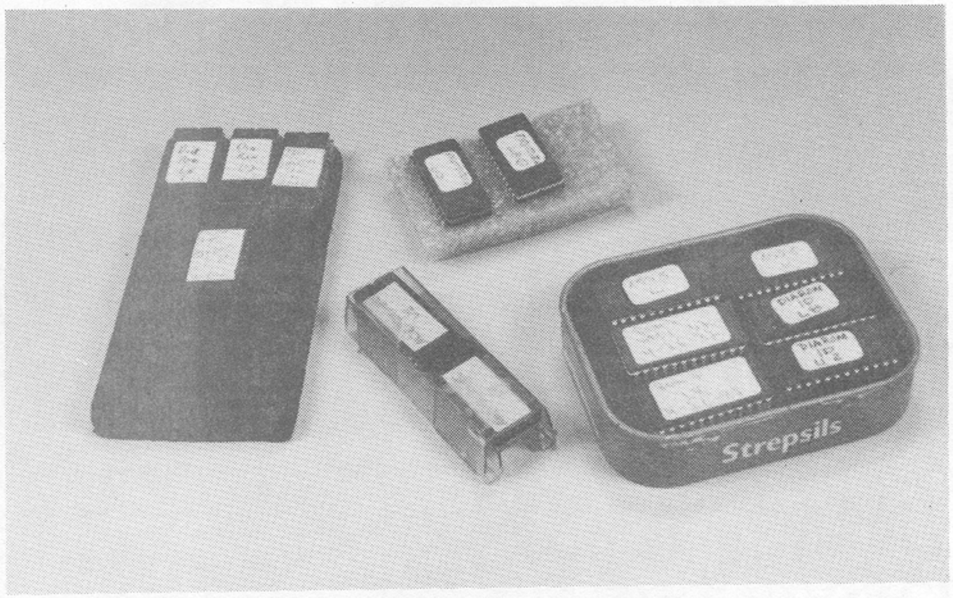

Figure 8 — Users will ship and store EPROMs in any way that works. NEVER USE STYROFOAM!

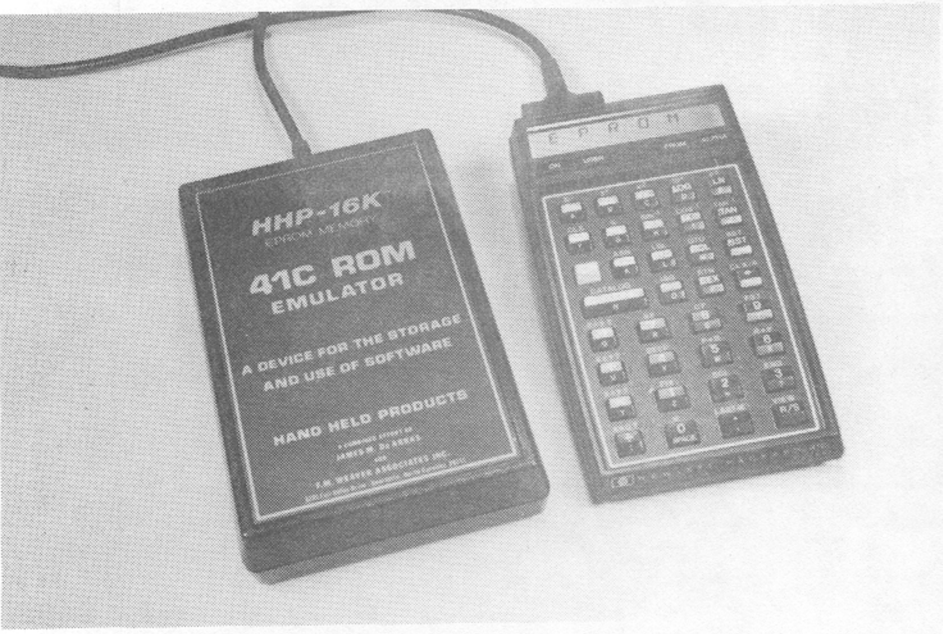

Figure 9 — EPROMs are used in an EPROM box.

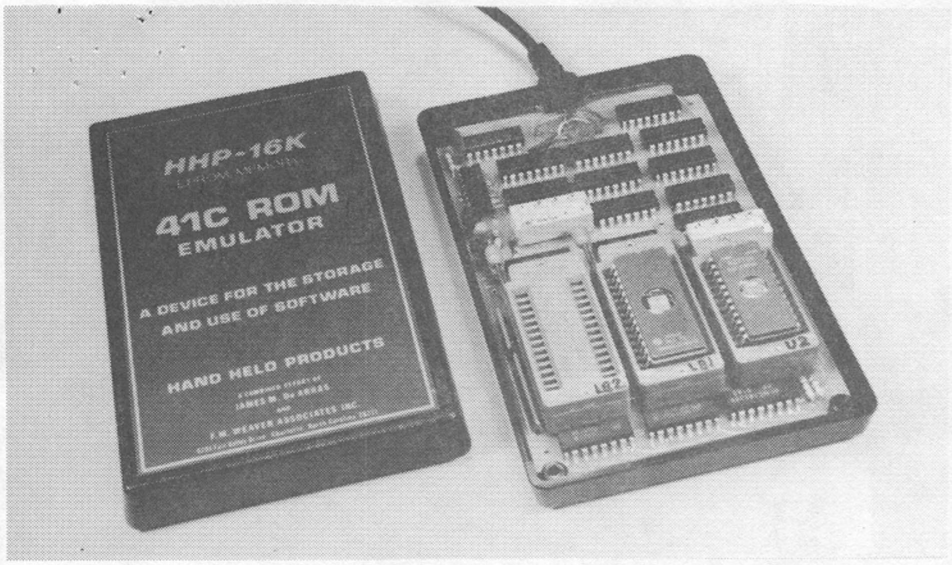

Figure 10 — Hand Held Product's HHP 16k EPROM box with cover removed showing EPROM sockets.

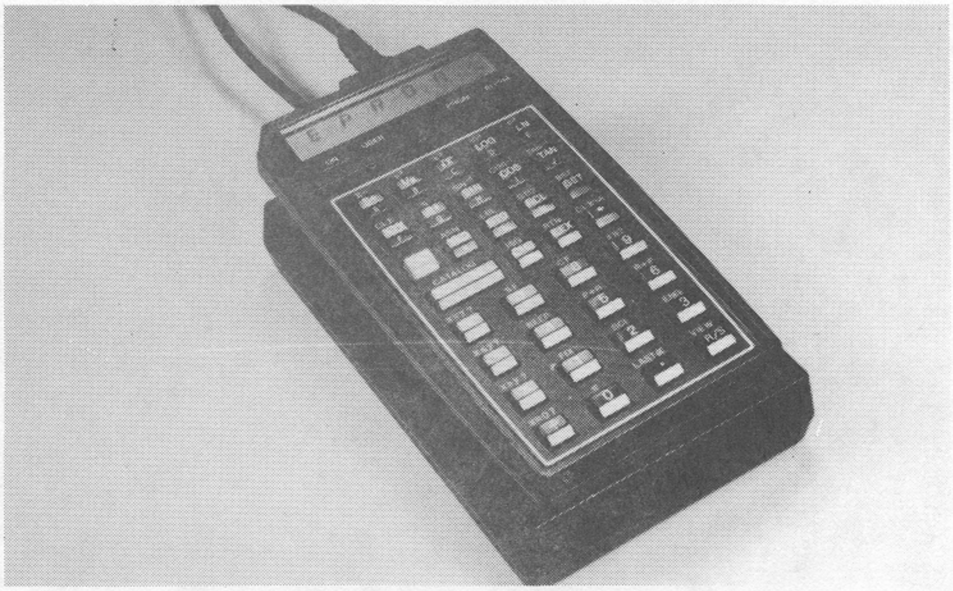

Figure 11 — HHP EPROM box fits nicely under the HP-41.

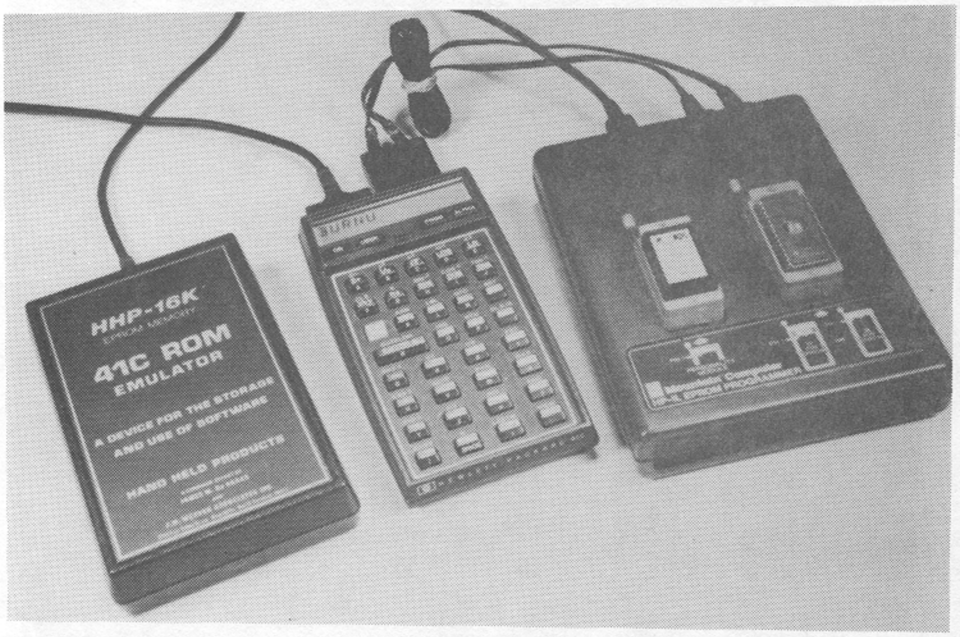

Figure 12 — EPROM burner, right from Mountain Computer. MCODE in the EPROM box burns EPROMs in a few minutes.

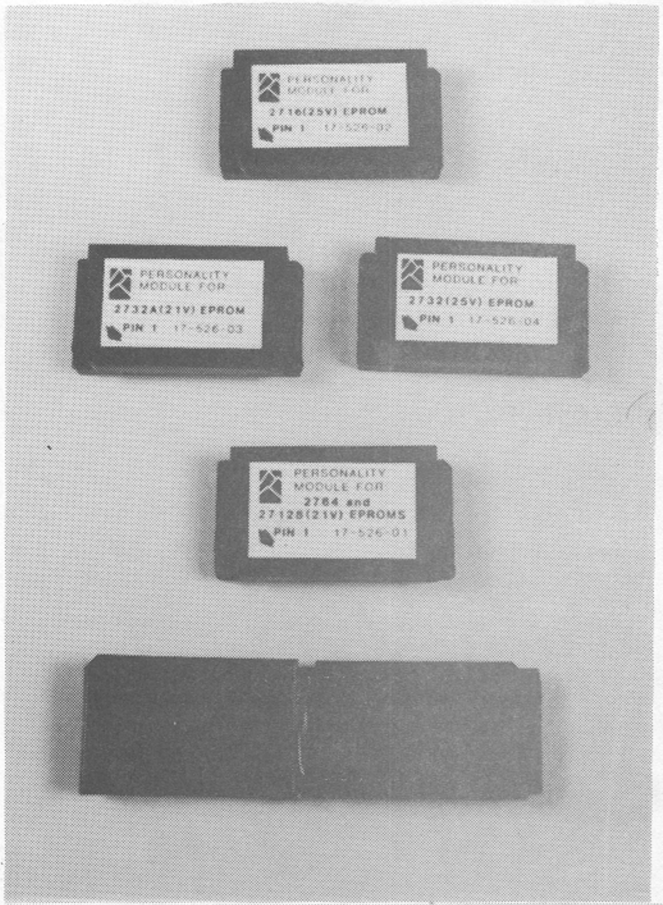
PERSONALITY MODULE FOR 2716(25V) EPROM PIN 1 17-526-02
PERSONALITY MODULE FOR 2732A(21V) EPROM PIN 1 17-526-03
PERSONALITY MODULE FOR 2732(25V) EPROM PIN 1 17-526-04
PERSONALITY MODULE FOR 2764 and 27128(21V) EPROMS PIN 1 17-526-01

Figure 13 — EPROM burners need personality modules for the EPROM being burned. These come with the Mountain Computer Burner. The bottom one is open and is supplied for future EPROMs. 27/28 and 27256's?

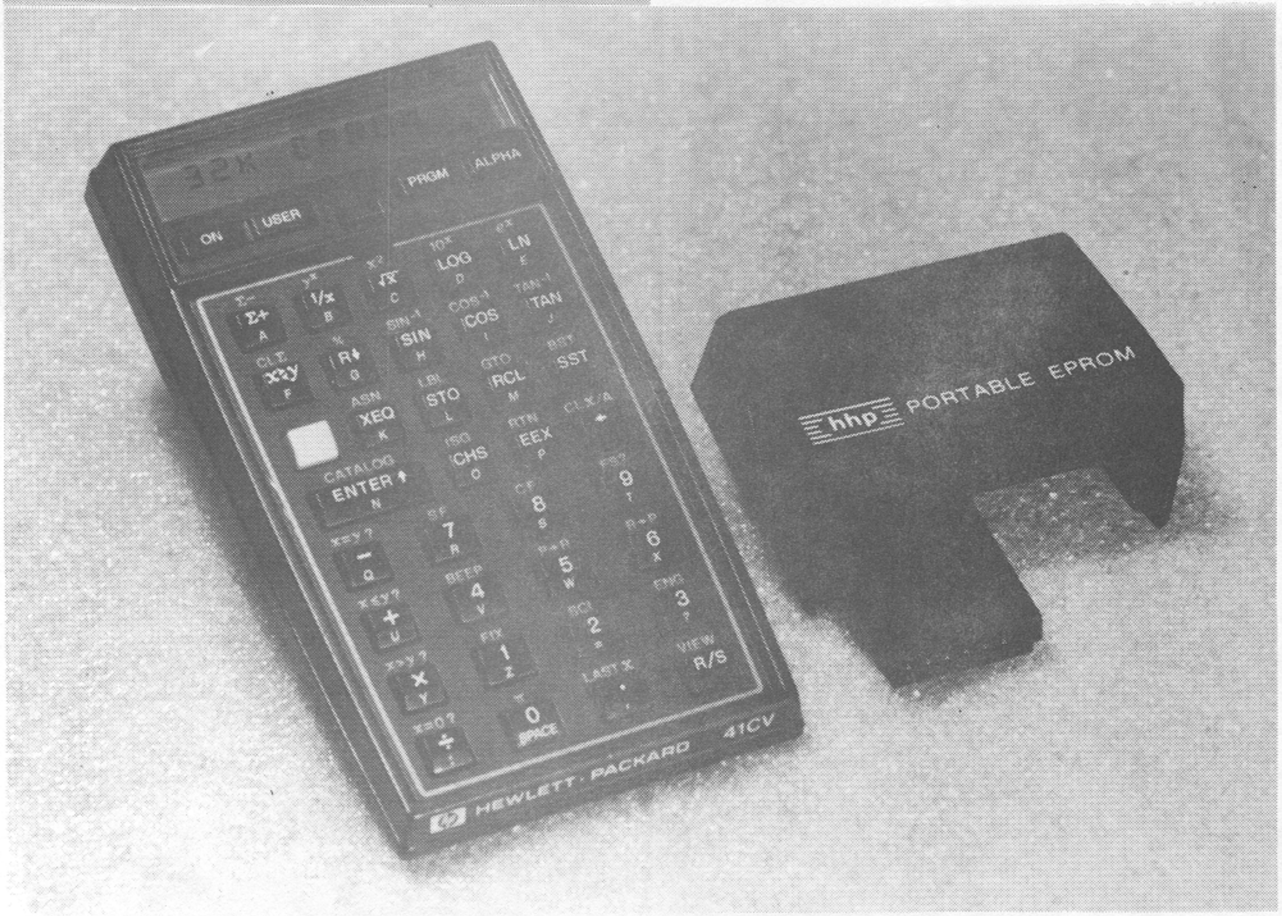

Figure 14 — State-of-the-art in small size large capacity EPROM memory. The Portable EPROM from Hand Held Products was announced August 15, 1983.