# INSTALLING A RAM-DISK IN A CP/M 2.2 SYSTEM

Leonard C. Schwab

The installation of a RAM-disk is a quick, simple, and cost-effective way to upgrade the overall performance of an older S-100 computer. Modern systems will benefit as well. However, unless the RAM-disk board comes with the proper software for your system, the installation process may be difficult. And even if you have factory-supplied drivers available for your RAM-disk, you may want to add features and capabilities not included with the stock drivers.

I have written this article to share my experiences installing and using a CompuPro M-Drive/H 512K RAM-disk board in an IMSAI VDP-80 computer. The information should also be useful to owners of other S-100 systems. The article describes the software necessary to drive the M-Drive/H. It includes listings of a skeletal CP/M BIOS, a RAM-disk formatting program, and a RAM-disk loading batch procedure.

## RAM-DRIVE CONCEPTS

A RAM-disk, or RAM-drive, is a block of read-and-write memory that is made to appear to the system as if it were a floppy disk drive. RAM-disks are also called 'disk emulators.' To the user, this storage device acts precisely like a mechanical disk drive but at RAM speed.

There are three fundamental approaches to creating a RAM-drive. We could simply set aside a block of the system memory. This is the method most often used by RAM-disk software in the MS-DOS environment. However, in a non-banked eight-bit system there is simply not enough memory available to make this approach practical.

The second method of implementing a RAM-drive involves bank-switched memory or extended-address memory beyond the first 64K available under CP/M 2.2. This approach is possible only if your main processor board already has the bank-switching or extended-address capability. Furthermore, it is less than satisfactory because the burden of managing the RAM-drive memory is placed on your main processor.

The third approach, used by the M-DRIVE/H board, SemiDisk, and others, is to make the RAM-drive memory accessible to the main processor by way of one or more I/O ports. This method allows the RAM-drive board to handle most of the chores involved in managing the RAM-drive.

## OBJECTIVES OF RAM-DRIVE INSTALLATION

The objective of installing RAM-disk is of course to increase disk-access speed. In my particular case, this is what I wanted to accomplish and, ultimately, was able to achieve:

**1.** Speed up the execution of application programs that make frequent accesses to disk. This includes programs that store and read data and those that use overlays, like compilers, WordStar and Supercalc.

**2.** Accelerate the operation of SUBMITted procedures. I use SUBMIT extensively to chain related programs, especially when coding and testing new programs (see reference 1).

**3.** Reduce warm-boot time. Under the Fischer-Freitas version of CP/M 2.2, which I use, warm-boots are very slow. Also, since a warm-boot occurs frequently during SUBMIT procedures, this alone would improve the performance of SUBMIT.

**4.** Operate the system, after an initial loading sequence, with **no** floppy disks mounted, thus reducing the wear and tear on my valued, venerable, and expensive PerSci drives.

**5.** Learn more about the computer and its operating system.

## THE IMSAI VDP-80

My VDP-80 has been in constant use since 1978. It is an integrated system, weighing almost 100 pounds, and includes a 10″ CRT display, keyboard, dual 8″ PerSci floppy drives, and a 7-slot S-100 card cage (pre IEEE-696 specification). The main processor is an 8085, running at 3 MHz. Originally equipped with 32K of dynamic memory, the machine has been upgraded to 64K of CompuPro static memory.

I found a substantially discounted M-Drive/H board at a West Coast Computer Faire and purchased it. I have never regretted that purchase. Today, the same board is readily available for about half of what I paid, making the RAM-drive even more desirable.

## COMPUPRO'S M-DRIVE/H BOARD AND SOFTWARE

The M-Drive/H board consists of 512K bytes of dynamic RAM, a dynamic RAM controller circuit, and electronics for addressing the data in the RAM array. Multiple boards may be installed in a system.

The M-Drive appears to the CPU as a pair of I/O ports. It creates no bus-assignment problems and uses no memory space. The M-Drive/H is well designed, well made, and well tested.

Technical information supplied with the board is skimpy. On-board addressing is described in a general fashion in a couple of sentences, but the precise software interface is never explicitly explained. There is nothing in the manual explaining how to format the board at power-up, except for an implication that formatting will be necessary.

Software programs are available from the vendor which will patch RAM-disk driver routines into a non-CompuPro version of CP/M 2.2 and format the RAM-disk on power-up. With this software, the RAM-disk appears as the M: logical disk.

However, use of this software would not satisfy one of my important objectives - running SUBMIT procedures from the RAM-disk. CP/M 2.2 will run SUBMIT from the A: logical disk only (see reference 2, pp 127-128). I needed the ability to make my RAM-disk appear as the A: drive in my system. Furthermore, the CompuPro drivers eat up 1K of working memory, which I could ill afford. This meant that I had to write my own drivers.

My first task was to write a stand-alone program, in assembly-language, to be a test-bed for routines. Fortunately, the manual includes a set of 'sample read/write routines' without which it would have been impossible to proceed.

After learning how to access the M-Drive, how to manage the addressing function (track and sector), and verifying that I could retrieve what I had written, I set about designing the BIOS modifications that would provide the functions that I wanted.

**INDEX OF LISTINGS DESCRIBED IN THIS ARTICLE**

Listing 1 — Page 37
Listing 2 — Page 59
Listing 3 — Page 58

## RAM-DISK SUPPORT ROUTINES

In CP/M, procedures that are related to the specific hardware installed in a system are contained in the BIOS (Basic Input-Output System). General procedures that manage the filing system are contained in the BDOS (Basic Disk Operating System). In most cases, it is unnecessary and undesirable to change the BDOS.

Listing One shows those parts of the BIOS affected by the presence of the M-Drive/H. I omitted the details of procedures that are dependent upon other hardware devices (console, printer, floppy disks, etc.).

Salient features of the RAM-disk software design follow:

### Warm Boots

In order to be able to warm-boot from the RAM-disk, a copy of the CP/M 2.2 system image (CCP and BDOS, 5632 bytes total) must be stored on the RAM-disk. The first six tracks (6K bytes) of the RAM-disk have been reserved for that purpose.

When the RAM-disk is initially formatted, a copy of the system image is read from the floppy into the RAM-disk (see Listing One, Cold-Boot section). Subsequent warm-boots read this image into memory virtually instantaneously, shaving several seconds off the warm-boot time. (see Listing One, Warm-Boot section).

### Access to Tracks, Sectors, and Memory (DMA)

When data on a disk is to be accessed, the BDOS first asks the BIOS to set the disk number, track number, sector number, and memory buffer address. The BIOS is responsible for telling the physical device, usually a floppy disk controller, how to set up for the access. The device is not actually accessed until BDOS sends a read or write request to BIOS. The disk number, track number, and memory buffer addresses are reset for subsequent accesses only when necessary.

To set up a track for access, the BDOS sends a two-byte physical track number to the SETTRK (set track number) routine in the BIOS. I found that the BIOS in my system only used the low-order byte of the track number, assuming that no drive would have more than 256 tracks. The RAM-disk, however, is organized as 512 tracks. Each track contains 1K bytes, divided into eight 128-byte sectors (see Listing One, RAM-Disk-Data section). Obviously, SETTRK had to be modified to use all 16 bits in the track number, in order to be able to deal with the 512 tracks (track numbers 0000H through 01FFH) on the RAM-drive.

No modification was required for the BIOS routines named SETSEC (set sector number) or SETDMA (set direct memory access address). (See Disk-Address-Pointers section).

Other disk-related functions — SELDSK (select disk), READ (read from disk), and WRITE (write to disk) — must include a test to determine whether a floppy disk or RAM-disk is being accessed and must include appropriate branching instructions (see Select-Disk and Disk-I/O Sections).

## Sector Translation

In order to maximize speed when accessing floppy disks, the logical sectors are not stored on the disk in sequence. A technique known as interleaving is used which requires the BIOS to convert the logical sector numbers sent by the BDOS into physical sector numbers appropriate to the interleaving scheme in use by the system. This conversion is known as sector translation. It is the responsibility of the SECTRN routine in the BIOS.

Because the RAM-disk is not a rotating device, like a floppy drive, no sector translation is necessary. Hence, SECTRN must be modified to include an escape branch when the RAM-drive is being accessed (see Sector-Translation section).

## Cold Boots

The code in Listing One will generate two separate versions of the BIOS. The version generated is determined by the setting of compilation-time switches RAM$A and RAM$C. In one version, which I call RAMCSYS, the floppy disks are designated A: and B:, the RAM-disk is C:. This version is generally operative only during the initial formatting and loading of the RAM-disk after a power-on. In the second version, RAMASYS, the RAM-disk is A:, and the floppies are B: and C: respectively.

The two separate versions are necessary in order to have a system that will boot from a floppy (RAMCSYS) and another system that will support batch processing with SUBMIT from the RAM-disk (RAMASYS).

The cold-boot routine in the BIOS must include a subroutine that moves a pristine copy of the system from high-memory into the reserved area of the RAM-disk. This copy will be pristine because it will be moved after it is read from the disk but before the CCP is entered (see Cold-Boot section).

## Formatting the RAM-disk

It is necessary to write a format program for the RAM-disk (see Listing Two). This program is run once, and only once, immediately after the system has been powered up and cold-booted from a floppy disk.

It is important that this program **not** be autoloaded on cold-boot. It must be manually started. The format program will destroy any data on the RAM-disk, and you want to retain the capability of doing a cold-boot, to recover from some error, without reformatting the RAM-disk.

The format program places

**BE-82 VIDEO BOARD FOR THE S-100 BUS**

FEATURES:
- Bit-mapped characters and graphics, 640 by 330 pixel display
- Full Greek set in EPROM, 7 by 9 character, 8 by 11 window
- Characters placed on any raster line, 30 char-line display
- Load a user program in BE-82 RAM for special commands
- Create graphics on one screen while viewing another
- 660 raster lines available for a single drawing
- Effective baud rate to 10,500 in terminal mode

S-100 BUS HACKERS:
- Full documentation, user can customize to special needs
- Purchase items singly, as a group, or choose additions later
- Bare board plus hardware/software documentation..............$135 ☐
- Programmed EPROMS (three 2716s)..................................30 ☐
- Chip set plus hardware components..................................125 ☐
- Assembly and testing....................................................100 ☐

HARDWARE:
- Amdek 310A Video Monitor and Key Tronic keyboard recommended
- S-100 bus system or ZENITH Z-100 Low Profile Computer
- BE-82 VIDEO BOARD acts as a slave device at all times on bus
- Board operation is independent of host processor clock speed

SCIENTISTS AND ENGINEERS:
- DB-WRITE Technical Word Processor.................................$100 ☐

Write your papers using a display that shows exactly what is printed. Operates under CP/M 80 by Digital Research and requires a 64k system. The BE-82 VIDEO BOARD is required to use DB-WRITE. Full documentation and all files provided. Use CP/M to customize keyboard layout by changing jump table, modify printer driver, etc. Printer driver included for the NEC SPINWRITER and the Tech Math/Times Roman thimble. Example of display screen and printout:

$$f(\lambda) = \int_{-\infty}^{\infty} e^{-\beta(\lambda^2+\xi^2)}\left[\frac{\partial^2 \rho(\xi)}{\partial \xi^2} + H_0(\xi)\sin^2(\lambda-\xi)\right]d\xi \qquad \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix}$$

Position char's $0\;^{2}\;_{4}\;^{6}\;_{8}\;^{10}\;_{8}\;^{6}\;_{4}\;^{2}\;0$ from line

α – β – γ – δ – ε – η – λ – μ – ξ – π – ρ – σ – τ – ω

Name ____________________

Address ____________________

City ____________________

State __________ ZIP __________

☐ Additional Information

**D BAGANOFF SYSTEMS, 3141 DAVID COURT, PALO ALTO, CA 94303**

'no-data' symbols (0E5H) into the directory area on the RAM-disk. It is immaterial whether or not the remainder of the RAM-disk is formatted, but it is important not to overwrite the reserved area in which the system image has been stored.

## Parity checking

Although the CompuPro manual shows a method for parity-checking all RAM-disk I/O, I decided to omit that technique. In more than a year of constant use, I have observed no I/O errors during RAM-disk accesses. In any case, the presence of parity checking will only prevent access to what is probably substantially correct data.

## ASSEMBLING AND INSTALLING THE SYSTEM

In the skeletal BIOS code (Listing One, RAMSYS.ASM), elements which are in uppercase are specifically applicable to RAM-disk operations. Other code, in lowercase, is presented only for the purpose of indicating the context of the relevant code. The only relevant items in RAMSYS.ASM that are system-dependent are the MSIZE and NDISKS constants. The code is written for Digital Research's ASM.COM assembler and must be edited into the source code for your system's BIOS before assembly.

Some method for installing the modified BIOS into memory will be needed. This will vary from system to system.

The Fischer-Freitas version of CP/M 2.2 includes a transient program, LDBIOSxx.COM (where xx designates the size of the system). LDBIOS is autoloaded during cold-boots by special code in the CCP-BDOS image, but it may be executed by the user at any time to bring a new BIOS into memory.

In order to load the two versions of my RAM-drive BIOS, I created two separate versions of the LDBIOS program. One is named LDBIOS56.COM, so that it will be found and loaded by the cold-boot routine. This version loads the RAMCSYS system. The second version of the BIOS loader is named RAMASYS.COM. Obviously, it loads the RAMASYS system and makes the RAM-disk appear to be the A: drive to the system.

For other systems, you must find a way to install the RAMCSYS into your cold-boot loader. A method for doing this is described in the CP/M 2.2 manual, Section 6, 'CP/M 2 Alteration.'

If you want to follow the two-system approach described above, you will also have to write a simple loader to replace the RAMCSYS BIOS with a RAMASYS BIOS for work purposes. Alternatively, a program may be written which patches the BIOS code in memory, thus converting from one BIOS version to the other. The sections of code to be patched are those which are bracketed by IF and ENDIF in Listing One.

## USING THE RAM-DISK

I have adopted the following operating procedure when using the RAM-disk:

**1.** Power on.

**2.** Cold-boot. RAMCSYS BIOS is loaded. The RAM-disk is C:.

**3.** Run batch file RAM.SUB (Listing 3). RAM.SUB performs the following tasks:

a) Runs the RAM-disk format program, RAMFMT.COM.

b) Transfers general utilities from the boot-disk to the RAM-disk.

c) Runs the BIOS loader RAMASYS.COM which installs the final working BIOS, making the RAM-disk appear as A:. At this point, the two floppies, A: and B:, are converted to B: and C:, respectively.

In my system, the above procedure takes approximately 100 seconds. This includes transferring about 42K of utility software to the RAM-disk. The system may now be used without floppy-disk support, if desired.

VERY IMPORTANT. Before turning the power off, any new or edited files on the RAM-disk must be transferred back onto a floppy. Any data on the RAM-disk will be lost when the power is removed. Frequent transfers of modified data from RAM-disk to floppy are a prudent operating procedure.

Because the RAM-drive memory is physically separate from working memory, it is possible to cold-boot the system or to load and run versions of CP/M that don't support the RAM-disk. Obviously, if an operating system does not support the RAM-disk, it will be impossible to access it. But, as long as the RAMFMT program is not rerun and the power is not turned off, the RAM-disk data may be made to reappear merely by running the RAMASYS or RAMCSYS system loaders.

The installation of the M-Drive/H has given my old system many more years of very useful life. I hope that this case-study will help you achieve similar benefits. ■

**References:**

1. Schwab, Leonard. Minimizing the Inconvenience of Compiled Languages under CP/M. *Microsystems,* May/June 1982.
2. Cortesi, David E. *Inside CP/M.* Holt, Rinehart and Winston, 1982.
3. *CP/M Operating System Manual.* Digital Research, 1982.