ISSN # 0748-9331

THE COMPUTER JOURNAL®

For Those Who Interface, Build, and Apply Micros

Issue Number 18

$2.50 US

THE LS-100 DISK SIMULATOR KIT

BY James O'Connor

It seems like only yesterday that upgrading a system from 4K of memory to 8K was a giant leap forward in processing power, not to mention a giant step backwards for our equipment budget. Perhaps more than any other item we purchase for our computer systems, the price of memory has declined steadily and sharply. The prime factors that have led to this are the emergence of dynamic memory technology as a reliable product, plus the fierce competition among American and Japanese manufacturers of dynamic memory ICs.

Dynamic memory, reduced to its simplest explanation, is a collection of capacitors which are charged or not charged so as to represent data. The problem with capacitors is that as soon as they are charged up they immediately start to discharge and, if nothing were done about this, they would very quickly forget everything stored in them. To solve this problem, dynamic memory circuits provide a refresh signal which serves to recharge the capacitors without altering what is stored in them. In the early days this refresh signal was provided by separate logic circuits but today many microprocessors (such as the Z-80) inherently provide this signal. Another early problem for dynamic memories was excessive susceptibility to noise, much of which was generated by all those charges being stored and read out. Proper printed circuit board design was essential for correct operation. Today these design principles are well understood and almost all modern dynamic memory boards are just as reliable as static memory systems. A major plus for dynamic memories is that they have always occupied less space, consumed less power and thus produced less heat than static memories.

It wasn't long before it occurred to some clever people that dynamic memory was now practically and economically capable of taking the place of, or at least complementing, some of the basic peripherals of a computer system such as the disk drives. Now why would you want to simulate a disk drive with memory chips? Because memory chips are electronic devices, they operate at electronic speed—we all learned in Elementary Physics that electrons travel at the speed of light. Disk drives, on the other hand, are electro-mechanical and from Elementary Logic we learned that the top speed of any device is the speed of its slowest part (the mechanical aspect of disk drives.) The other reason is that almost all software, both operating systems and applications programs, understand tracks and sectors as opposed to memory locations for storing large amounts of data. Thus a memory that simulates a disk drive should work with almost all current software, while providing a most impressive boost in speed.

Disk simulators are an unquestioned success but most of the early ones cost over a thousand dollars for about 256K of memory, which is roughly equivalent to the capacity of a single sided, single density 8 inch diskette (240K bytes). For those of us on a limited budget, and who only use our systems occasionally, that can be too costly to justify a disk simulator.

Thus I was intrigued when I noticed an ad from Digital Research Computers (DRC) of Garland, Texas 75046 (not to be confused with the software company of Pacific Grove, California) for the Light Speed 100 (LS-100). This product is available either as a 256K kit for $229.00 or as a bare board with software for $69.95. I quickly scraped together the necessary funds and ordered the full kit, which arrived about 6 weeks later. I suspect that demand has been brisk since this kit represents a substantial value to price ratio.

The Kit

Upon opening the package I found that the parts were all present but were not separated into little envelopes as is the case with some other kit makers. You must be able to correctly identify electronic parts with this kit. There is a neatly typed assembly/usage manual of nineteen pages. DRC has wisely produced the schematic on an 11 × 17 inch page so that it is fully legible. It is also surprisingly simple because of the use of large scale ICs in the design of this board. A single sided, single density 8" diskette containing the software completes the package. I don't know if any other disk size or format is available, so you should contact DRC first if you cannot read this format or find someone who can transfer it to your format.

The circuit board for this kit has double sided traces with plated-through holes and a solder mask to avoid solder bridges between the closely spaced traces. The board appears to have been laid out by a computerized drawing system as the traces are very neat and machine-like in design. Part locations are outlined and identified in white on the top of the board. The parts in the kit were all of good quality. This board uses 64K × 1 bit dynamic memories (4164s), 32 of them, and these were very tightly piggybacked on top of each other in tiny little packages and then wrapped in aluminum foil to protect them from static damage. There are also 51 tiny little ceramic bypass capacitors and 39 16-pin DIP sockets—as with any memory type board there are a lot of parts.

The assembly instructions occupy three pages of step by step procedures. There is a parts layout diagram in the back of the manual, but no other drawings or pictures of the completed card are included.

Construction

Whenever I assemble a kit I find that they fall into one of two categories. The first type seems to struggle against its own construction—nothing goes easily or smoothly, but with perseverance it eventually gets built. With the second category it almost seems that if you could shake the box long enough, everything would just naturally fall together. The LS-100 fits into the "shake the box" category; even though there are a lot of pins to be soldered it all went smoothly, and inexorably resulted in the finished board. It took me about two evenings to complete, but I was rushing it in anticipation of the improvement a disk simulator would make in my system.

Setup consists of setting two 8 position DIP switches, one to address the board (which uses four consecutive I/O ports), and the second to select the board. You can install up to eight of these boards in a system (2 MegaBytes!!). Each additional board would use the same four I/O port addresses, so the second DIP switch numbers each board to permit individual selection.

There are three jumpers to set. One jumper selects whether or not you plan to provide standby power to the circuit when your computer is off. The next jumper selects Advanced or Normal Ready; there is a half page explanation of this in the manual which didn't really explain it to me. I would have had to research this somewhere else except that the instructions say to select the Advanced option and run the diagnostic program – if there are no errors, then your system can handle that option. Otherwise you must use the Normal Ready. The last set of jumpers allows you to indicate whether your processor provides a refresh signal (Z-80) or if it doesn't (8080), and what bus line carries the signal at what polarity. This explanation requires you to know something about your own system. The documentation contains a transposition error in this section. If the refresh signal occurs on bus pin 65 you should jumper J9, not J8 as instructed, and if refresh is on pin 66, jumper J8 not J9. There is one sample page of jumper settings for an 8080 system and one for a Z-80 system, and on these, the jumper options appear to be correctly labeled (further proof of the value of examples in documentation).

Once I had chosen the options for my system I installed the board, ran the diagnostics, and it worked right away. I haven't had to touch it since.

Software

There are three separate software programs for this board; a diagnostic program, a format program, and an install program. They are all supplied on the disk in BOTH source and object form. They are designed for use with CP/M-80. The full documentation is also supplied on the disk so you can print out extra copies of the manual if you wish (don't neglect to fix the transposition error I mentioned). This is a really neat feature that I wish all manufacturers would routinely adopt whenever possible. I often lose or misplace manuals but I can almost always find a disk.

The diagnostic program does exactly what its name implies and is used once to verify the operation of the board and the jumper option for Advanced or Normal Ready, then afterwards only if you suspect some problem with the board. The format option is designed to fill the memory on the board with a data pattern exactly like a freshly formatted disk. Each time you turn on power you must run the format program, as the memory will contain random data. You can avoid this by providing stand-by power to the board, but it takes less than a second to format a board, thanks to electronic speed. Both of these programs can be run as is, provided you use the default port addresses. If you need other addresses then you must change the equates in each program and reassemble with CP/M's ASM.COM or equivalent.

The third program is the driver, which serves to translate the sector and track addresses of a disk drive to memory locations within the LS-100. It also provides a checksum feature to verify the accuracy of the stored data. When the driver is installed your system gains a new disk drive. The default is drive "E:" which worked fine on my system, but you can change this if you already have a drive E:.

It is unlikely that you can use the pre-assembled version of the installer/driver, as it must be tailored to your system size. It is really two separate programs which are merged together into one COM file; one part installs the other part (the LS-100 driver) into your operating system. You may need to edit these files so as to properly set the memory size of your operating system as well as the I/O addresses if you can't use the defaults. Then you use DDT or ZSID to merge the two programs together. The reason for this is that the driver is assembled to operate at an address just below your operating system in high memory. It is the job of the install portion to move it there and then modify the operating system jump table to reference it. If you tried to load the driver at its true address it would result in a very large COM file indeed. By combining it with the installer you get a much more compact program, plus it can't just load, it must alter some addresses in the operating system itself.

This sort of procedure will be old hat to anyone who has ever modified their CBIOS. I have done this procedure many times but I found that even after reading the explanation of it in the LS-100 manual, I didn't realize that's what it was instructing me to do. The description of this process is somehow disjointed and disorganized. It mixes the process of creating the combined install/driver with an explanation of how they work. In essence, it is all explained but could use more clarity. There is a console listing example of the procedure, but be warned – you cannot copy it line for line if your system is a different total memory size – you must figure out where and how to alter it.

An alternative is to put the driver code directly into your CBIOS; the DRC manual describes the general theory of how to do this. This alternative would require that you know and understand your system's CBIOS. It is not possible for DRC to do more than provide a theoretical explanation (albeit a quite good one, as they use examples of the code needed to do this).

There is one problem that the manual makes no mention of and that could be disastrous on certain systems. The DRC software uses one byte at address 40 hex as a signal to indicate that the driver program has been installed. The locations from 40 to 4F hex are designated by CP/M as being available to the CBIOS for storing variables or switches as needed, and many do so. Typically some of these bytes will describe the nature and type of disks installed and status of certain operations. If your CBIOS already uses the byte at 40 hex, you will have two different parts of the operating system using the SAME location for different purposes, each thinking it is the only user of that byte. This could cause big trouble if the byte contains disk status, as your operating system might clobber a disk because of the data stored there by the LS-100. On my system it isn't as serious, as it is the operating system that clobbers the LS-100 byte value, so that every so often the LS-100 "disappears" (as far as the operating system is concerned) and I have to run the install program to make it reappear.

To solve this problem you can do as follows: If you have the source code for your CBIOS then check to see if it uses this location. Be careful though, because if it stores a word value (two bytes) at location 41 hex then it *does* use location 40 while not explicitly addressing that location. If you find this to be the case then you can either change your operating system (not recommended, as you would have to change every copy of it), or just change the location the LS-100 uses to one that your operating system doesn't use, between 40 and 4F hex. If you don't have the source code then you will have to contact the vendor who supplied your CBIOS. If that's not possible then you may want to change the LS-100 to use location 4F and run a lot of tests to insure that it works OK. What have I done? Well, nothing yet because my CBIOS uses ALL of the available locations and besides, I plan to integrate the driver into my CBIOS, eliminating the need for that byte (if the driver is always present you don't need a byte to tell it that).

One nice feature of the software is that if you add more LS-100 boards to your system, the programs will automatically detect this and adjust accordingly. However, if you have integrated the driver into your CBIOS then you will have to change the Disk Parameter Block to take advantage of the additional boards.

Using the LS-100

If you haven't integrated the driver into your operating system, you must first format the LS-100 and then install the driver each time you power up. I have made up a little SUBMIT file to do this. If you know how to do it, you can

(Graphic listing continued from page 11)

```
EXIT:    MOV      SI,OFFSET NSET
         CALL     PRINT
         RETF
PRINT:   MOV      DL,[SI]              ;Print message routine
         OR       DL,DL
         JNZ      PRINT1
         RET                           ;Zero terminates message
PRINT1:  PUSH     SI                   ;Save string pointer
         CALL     WRTLST               ;Write to list device
         POP      SI                   ;Restore pointer
         INC      SI
         JMPS     PRINT                ;Go to next character
LINE:    MOV      SI,OFFSET GSET
         CALL     PRINT                ;Set Epson for graphics
LINE1:   MOV      AL,BH                ;Set up video RAM address
         OUT      VIDHI,AL
         MOV      AL,BL
         OUT      VIDLO,AL
         IN       AL,VIDIO             ;Get the byte
         MOV      DL,AL
         CALL     WRTLST               ;Send to list device
         ADD      BX,8Ø                ;Go to next video line
         CMP      BH,96H               ;Test for end of screen
         JB       LINE1
         SUB      BH,96H
         MOV      DL,ØDH
WRTLST:  MOV      CL,5                 ;Write list function
BDOS:    PUSH     BX                   ;Save pointer
         INT      224                  ;Invoke BDOS
         POP      BX                   ;Restore pointer
         RET
;
         DSEG
;
         ORG      1ØØH
;
VSET     DB       ØDH,1BH,'A',8,Ø ;Return and set v.s. string
GSET     DB       1BH,'K',224,1,Ø ;Set graphics mode string
NSET     DB       1BH,'2',ØCH,Ø   ;Set normal and f.f. string
```

Listing 3

```
*
VIDLO    EQU      $FFFFØØ+1ØØ          *Low byte of address port
VIDHI    EQU      VIDLO+1              *High byte of address port
VIDIO    EQU      VIDLO+2              *Data I/O port
*
         TEXT
*
         MOVE.L   #VSET,AØ             *Vertical spacing string
         BSR      PRINT                *Set Epson for 8/72" spacing
         MOVE.W   #79,D2               *Start at upper r.h. corner
LOOP:    BSR      LINE                 *Print line
         MOVEQ    #$ØA,D1
         BSR      WRTLST               *Send linefeed here
         MOVEQ    #11,DØ
         BSR      BDOS                 *Look for key pressed
         TST.W    DØ
         BEQ      CONT                 *Keep going
         MOVEQ    #$FF,D1
         MOVEQ    #6,DØ
         BSR      BDOS                 *Input character without echo
         BRA      EXIT                 *Quit
CONT:    SUBQ.W   #1,D2                *Back up for next line
         BCC      LOOP                 *More to go
EXIT:    MOVE.L   #NSET,AØ
PRINT:   MOVE.B   (AØ)+,D1             *Print message routine
         BNE      PRINT1
         RTS                           *Zero terminates message
PRINT1:  MOVE.L   AØ,-(SP)             *Save string pointer
         BSR      WRTLST               *Write to list device
         MOVE.L   (SP)+,AØ             *Restore pointer
         BRA      PRINT                *Go to next character
LINE:    MOVE.L   #GSET,AØ
         BSR      PRINT                *Set Epson for graphics
LINE1:   ROL.W    #8,D2                ;Set up video RAM address
         MOVE.B   D2,VIDHI
```

(continued on page 27)

also cause this SUBMIT to execute automatically upon startup (this requires a small patch to the CP/M Operating System.) It takes the format and install programs combined less than two seconds to accomplish their tasks.

In operation, the LS-100 definitely improves the performance of any program that uses overlays, plus it is much quieter. Compilations and long assemblies are much faster. But there is a negative side—you must transfer files to the disk simulator prior to working on them. I have found that this step can take some time and effort as it frequently involves more than one or two files. Even more disconcerting is the fact that you can't always use wildcard names to process them all (the Public Domain SWEEP programs are really useful here). Time spent here can cancel out the time saved if you'll only be working for a brief period. The second, and perhaps more sinister problem, is that you must remember to transfer the files BACK to diskettes—otherwise all that work evaporates when you turn off the power. I have already had the experience of shutting down the system and, just as the little dot on the CRT was fading away, realizing that I hadn't transferred the files. Argghhh and double argghh!!! I have only done that once, so perhaps one must do it at least once in order to avoid it. These problems are generic to all disk simulators.

I think this board should really have stand-by power to realize its full potential, but that is easier said than done. The board draws about 600 milliamps which is a lot to expect from a battery. DRC recommends a filtered DC power supply of between 8 and 10 volts at 800 milliamps because the stand-by power goes through the on-board regulator and must be at least 8 volts for correct operation. A suitable supply could be made by following numerous examples of such circuits or by buying something similiar. However, such a unit will draw enough power that you should consider the impact on your electric bill, especially if you won't use your system for periods of several days. It would be cheaper to back up the files onto diskettes and then turn off the power, which you should do in any case for protection from power failures. The ideal would be a supply that feeds power from a battery that is constantly trickle charged from the AC line; this supply could ignore brief power interruptions. Perhaps some reader could design a suitable circuit for a future issue—it would be useful for many more applications.

Who Should Buy This Kit?

First, you should have an S-100 bus system with at least one available slot. You will need to know how to solder, and must be able to identify electronic parts. If you are a novice builder and know someone who can help you with this, you should be able to assemble this kit if you take your time; there are no tricky or difficult assembly steps.

You must also be able to adapt the software to your system. This requires that you know or be able to find out what port addresses are already used in your system and where to locate the signal byte mentioned earlier in this review. You will need to be able to edit the source files and reassemble them, and use DDT to combine the install/driver programs. None of this is very complex, unless you've never done it before. Again, try to find someone to guide you if necessary. One reason I like to build kits is that they force me to extend my knowledge into areas I would otherwise never enter. This kit is just right for anyone who has yet to modify some software; it is not too difficult, there are lots of people (especially in User Groups) who can help, and the results can be most worthwhile.

Should you want to integrate the driver into your CBIOS then you will need all the above skills plus the SOURCE code for your CBIOS. This may be a problem, as not all companies supply the code even if you ask them for it. And if you've never fiddled with a CBIOS you will need skilled guidance or a good book or two to help with the process. Don't rule out the LS-100 if you can't do this, as this step is only a matter of convenience, and it works just fine without it.

The bare board is available for those who want to obtain their own parts. One IC, the Intel 8203-1 is rare and costly so DRC also sells it. Obviously, bare board building is for the more experienced kit basher, but if you want to try it this is a good unit to start with.

The board costs $69.95 and includes the software disk. The 8203 IC is $29.95.

Summary

This is a good kit; it is well designed and includes quality parts. The documentation could use some improvement, but it is certainly adequate in that the needed information is there, even if you must sometimes study the text. I didn't need to contact DRC but I understand that they have been in business for some time, which usually indicates that a firm supports their products.

The resulting disk simulator can really improve the performance of a system (I am using it to prepare this review). And if you want more than 256K of electronic disk space you can easily add additional LS-100's (up to eight depending on the number of available slots in your system). Since I purchased the kit for $399, DRC has reduced the kit price to $229, probably due to reduced cost of memory ICs. Perhaps, too, the six week delivery has improved—I ordered just after the kit was advertised, so it may have taken DRC time to ramp up production.

Most of the problems you're likely to encounter with this product are really caused by the wide variance between S-100 bus CP/M systems, and the fact that any disk simulator needs to be melded into the operating system. This kit does as good a job of mitigating those difficulties as any that I have encountered. ■

ALL SALES ARE MADE SUBJECT TO THE TERMS OF OUR 90 DAY LIMITED WARRANTY. A COPY OF THIS WARRANTY IS AVAILABLE FREE, ON REQUEST.

DIGITAL RESEARCH COMPUTERS

(214) 225-2309

DISK DRIVE SPECIAL!

5¼" Double Sided-Double Density

Brand New, Unused. Mfg. in Japan by Canon. 2/3 Height - Direct Drive! Industry Standard Pin Out. 6MS Access. 40 Tracks.

\$49.95 ea

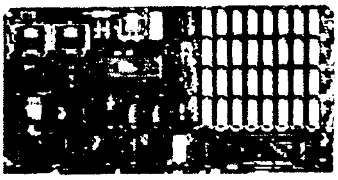
2 FOR \$85

Sold on a First Come Basis
Add \$2 Each U.P.S.

256K S-100 SOLID STATE DISK SIMULATOR!

WE CALL THIS BOARD THE "LIGHT-SPEED-100" BECAUSE IT OFFERS AN ASTOUNDING INCREASE IN YOUR COMPUTER'S PERFORMANCE WHEN COMPARED TO A MECHANICAL FLOPPY DISK DRIVE.

PRICE CUT!



BLANK PCB
(WITH CP/M* 2.2
PATCHES AND INSTALL
PROGRAM ON DISKETTE)
\$69.95
(8203-1 INTEL \$29.95)

FEATURES:
- 256K on board, using + 5V 64K DRAMS.
- Uses new Intel 8203-1 LSI Memory Controller.
- Requires only 4 Dip Switch Selectable I/O Ports.
- Runs on 8080 or Z80 S100 machines
- Up to 8 LS-100 boards can be run together for 2 Meg. of On Line Solid State Disk Storage.
- Provisions for Battery back-up.
- Software to mate the LS-100 to your CP/M* 2.2 DOS is supplied.
- The LS-100 provides an increase in speed of up to 7 to 10 times on Disk Intensive Software.
- Compare our price! You could pay up to 3 times as much for similar boards.

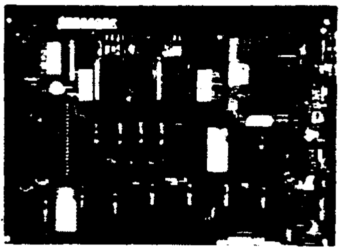
\$199.00

#LS-100 (FULL 256K KIT)

THE NEW ZRT-80
CRT TERMINAL BOARD!

A LOW COST Z-80 BASED SINGLE BOARD THAT ONLY NEEDS AN ASCII KEYBOARD, POWER SUPPLY, AND VIDEO MONITOR TO MAKE A COMPLETE CRT TERMINAL. USE AS A COMPUTER CONSOLE, OR WITH A MODEM FOR USE WITH ANY OF THE PHONE-LINE COMPUTER SERVICES.

FEATURES:
- Uses a Z80A and 6845 CRT Controller for powerful video capabilities.
- RS232 at 16 BAUD Rates from 75 to 19,200.
- 24 x 80 standard format (60 Hz).
- Optional formats from 24 x 80 (50 Hz) to 64 lines x 96 characters (60 Hz).
- Higher density formats require up to 3 additional 2K x 8 6116 RAMS.
- Uses N.S. INS 8250 BAUD Rate Gen. and USART combo IC.
- 3 Terminal Emulation Modes which are Dip Switch selectable. These include the LSI-ADM3A, the Heath H-19, and the Beehive.
- Composite or Split Video.
- Any polarity of video or sync.
- Inverse Video Capability.
- Small Size: 6.5 x 9 inches.
- Upper & lower case with descenders.
- 7 x 9 Character Matrix.
- Requires Par. ASCII keyboard.



BLANK PCB WITH 2716
CHAR. ROM, 2732 MON. ROM
\$49.95
SOURCE DISKETTE - ADD \$10
SET OF 2 CRYSTALS - ADD \$7.50

WITH 8 IN.
SOURCE DISK!
(CP/M COMPATIBLE)

\$99.95 # ZRT-80

(COMPLETE KIT,
2K VIDEO RAM)

64K S100 STATIC RAM

\$139.00 KIT

NEW!

LOW POWER!

150 NS ADD \$10

BLANK PC BOARD
WITH DOCUMENTATION
\$49.95

SUPPORT ICs + CAPS
\$17.50

FULL SOCKET SET
\$14.50

FULLY SUPPORTS THE
NEW IEEE 696 S100
STANDARD
(AS PROPOSED)

FOR 56K KIT \$125

ASSEMBLED AND
TESTED ADD \$50

PRICE CUT!

FEATURES:
- Uses new 2K x 8 (TMM 2016 or HM 6116) RAMs.
- Fully supports IEEE 696 24 BIT Extended Addressing.
- 64K draws only approximately 500 MA.
- 200 NS RAMs are standard. (TOSHIBA makes TMM 2016s as fast as 100 NS. FOR YOUR HIGH SPEED APPLICATIONS.)
- SUPPORTS PHANTOM (BOTH LOWER 32K AND ENTIRE BOARD).
- 2716 EPROMs may be installed in any of top 48K.
- Any of the top 8K (E000 H AND ABOVE) may be disabled to provide windows to eliminate any possible conflicts with your system monitor, disk controller, etc.
- Perfect for small systems since BOTH RAM and EPROM may co-exist on the same board.
- BOARD may be partially populated as 56K.

64K SS-50 STATIC RAM

\$119.00 (48K KIT)

NEW!

LOW POWER!

RAM OR EPROM!

BLANK PC BOARD
WITH
DOCUMENTATION
\$52

SUPPORT ICs + CAPS
\$18.00

FULL SOCKET SET
\$15.00

56K Kit \$129

64K Kit \$139

ASSEMBLED AND
TESTED ADD \$50

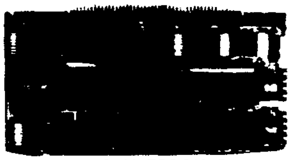
FEATURES:
- Uses new 2K x 8 (TMM 2016 or HM 6116) RAMs.
- Fully supports Extended Addressing.
- 64K draws only approximately 500 MA.
- 200 NS RAMs are standard. (TOSHIBA makes TMM 2016s as fast as 100 NS. FOR YOUR HIGH SPEED APPLICATIONS.)
- Board is configured as 3-16K blocks and 8-2K blocks (within any 64K block) for maximum flexibility.
- 2716 EPROMs may be installed anywhere on Board.
- Top 16K may be disabled in 2K blocks to avoid any I/O conflicts.
- One Board supports both RAM and EPROM.
- RAM supports 2MHZ operation at no extra charge!
- Board may be partially populated in 16K increments.

32K S100 EPROM/STATIC RAM

NEW! FOUR FUNCTION BOARD! **NEW!**

EPROM II
FULL
EPROM KIT
\$69.95

A&T EPROM
ADD \$35.00



BLANK
PC BOARD
WITH DATA
\$39.95

SUPPORT
IC'S
PLUS CAPS
\$16

FULL
SOCKET SET
\$15

We took our very popular 32K S100 EPROM Card and added additional logic to create a more versatile EPROM/RAM Board.

PRICES SLASHED!

FEATURES:
- This one board can be used in any one of four ways:
  - A. As a 32K 2716 EPROM Board
  - B. As a 32K 2732 EPROM Board (Using Every Other Socket)
  - C. As a mixed 32K 2716 EPROM/2K x 8 RAM Board
  - D. As a 32K Static RAM Board
- Uses New 2K x 8 (TMM2016 or HM6116, RAM's
- Fully Supports IEEE 696 Buss Standard (As Proposed)
- Supports 24 Bit Extended Adressing
- 200 NS (FAST!) RAM'S are standard on the RAM Kit
- Supports both Cromemco and North Star Bank Select
- Supports Phantom
- On Board wait State Generator
- Every 2K Block may be disabled
- Addressed as two separate 16K Blocks on any 64K Boundary
- Perfect for MP/M* Systems
- RAM Kit is very low power (300 MA typical)

32K STATIC RAM KIT — \$109.95

For RAM Kit A&T — Add \$40

Digital Research Computers

P.O. BOX 461565 • GARLAND, TEXAS 75046 • (214) 225-2309

TERMS: Add \$3.00 postage. We pay balance. Orders under \$15 add 75¢ handling. No C.O.D. We accept Visa and MasterCharge. Tex. Res. add 5-1/8% Tax. Foreign orders (except Canada) add 20% P & H. Orders over \$50. add 85¢ for for insurance.

*TM OF DIGITAL RESEARCH INC. (CALIF.)

WE ARE NOT ASSOCIATED WITH DIGITAL RESEARCH INC. (CALIF.) THE SUPPLIERS OF CPM SOFTWARE