ADDRESSING

The Macrotech SS256 is designed to operate in any S-100 system, 8-bit or 16-bit. While addressing the new generation of 16-bit processors is straightforward using the direct (24-bit) addressing mode of the SS256, most 8-bit systems provide only sixteen address lines, limiting the total system addressing capability to a 65,536 byte range. The SS256 occupies an address space of 262,144 (256k) bytes which requires four times the addressing capability of the typical 8-bit processor. Macrotech's unique solution to this problem is an advanced memory mapping architecture called M^3. M^3 is a powerful tool designed to provide on-board capability for the translation of 16-bit logical addresses from the S-100 bus into 20-bit physical addresses for the 256 kilobyte array.

16-BIT ADDRESSING: MACROTECH MEMORY MAPPING

When the M^3 addressing option is selected, each 4k block of the 16-bit (64k) logical address is dynamically translated to any 4k block of the 256 kbyte on-board physical memory. With this scheme, 'global' memory can be configured to any size and located anywhere in the logical address space. All remaining bank switched memory is accessed by simply reloading the appropriate map registers each time a bank switch is to occur. No memory or address space is wasted with this scheme and no constraints are imposed in the partitioning or sizing of either global or bank switched memory.

M^3 is designed to operate in two functional modes:

a. pass mode. Used to load mapping registers in the initial power-up interval. The SS256 functions as a 64k memory board in this mode, with the sixteen bit address passed directly to the array. This mode is activated when the system is initially powered up, and remains active until the last mapping register is loaded.

b. mapped addressing mode. When the mapping registers are completely loaded with the initial address map in the power-up interval, address control is automatically transferred to the mapped addressing mode. When this mode is invoked, addressing into the array is generated through the mapping registers. The mapped addressing mode remains in effect until the next power-up interval.

The mapping registers consist of two high speed bipolar rams located at board coordinates D5 and E5. Each ram has sixteen 4-bit words, and the two rams are organized as sixteen 8-bit registers to provide an address into the memory array. Each of the sixteen mapping registers can address any 4k block of the total 256k memory array by loading them with the desired mapping information.

M^3 TECHNICAL DESCRIPTION

register loading. The sixteen mapping registers occupy a contiguous I/O address block in the range n0 to nF hex. 'n' represents the upper nibble of the base I/O address selection set to any unused block of sixteen I/O addresses by the user. Multiple boards should all be set to the same 'bank select' address range chosen for this purpose (with some exceptions - see Vdisk section). The lower nibble of of each of the sixteen I/O addresses represents the 4k block to be addressed; i.e. n6H = 6000H-6FFFH. The eight data bits written to each register are formatted as follows:

bits 7-6. Board select bits. These two bits must match the two lsb's of the board address selection. Otherwise, no access can occur. Setting these bits to a non-existent 256k range can be useful at times, such as when ram memory must be masked for prom overlays in systems not incorporating the phantom* line. In most cases these two bits in each register will be written to match the appropriate board address select bits.

bits 5-4. 64k bank select bits. These two bits select which of the four 64k banks in the 256k memory array is to be addressed.

bits 3-0. 4k segment select bits. These four bits contain the desired segment information to delineate which of the 16 4k physical segments in the selected 64k block will be addressed.

BOARD LEVEL ADDRESSING

The Macrotech SS256 is addressable on any 256k boundary up to sixteen megabytes when the direct 24-bit addressing mode is selected. However, when the M^3 mode is selected for 16-bit addressing, an upper limit of one megabyte is imposed. For the most memory intensive requirements, this limitation should not be significant, inasmuch as a single SS256 can support five 48k user banks and 16k of global memory in a typical MP/M* operating system. When the M^3 addressing mode is selected, the four most-significant board select bits corresponding to A23-A20 should be set to zero (plugged to the right - see Board Parameterization section). The remaining two bits corresponding to A19-A18 can be plugged to any one of the four possible combinations to select the desired 256k address space in which the SS256 will reside. An example of coding for a typical power-up sequence to initialize the mapping registers follows:

*MP/M is a registered trade mark of Digital Research Corp.

```
init:   lxi  h,mmap     ;pointer to memory map
        mvi  c,40h      ;I/O base address
ini:    mov  b,m
        cout b          ;write initial map
        inx  h          ;next index
        inr  c          ;next register
        mvi  a,50h      ;done?
        cmp  c
        jrnz ini        ;no, continue
        ret

;  memory map table:
;  the 12 locations from mmap-mmap2 (0000-bfff)
;  will be changed as required by the bank select routine.
;  the 4 locations from 'globl' to end of table (c000-ffff)
;  are registers for fixed system (global) memory and will
;  remain unchanged.

;          written to   logical      physical
;          register     address      address

mmap:   db   0      ; 0000-0fff     00000-00fff
        db   1      ; 1000-1fff     01000-01fff
        db   2      ; 2000-2fff     02000-02fff
        db   3      ; 3000-3fff     03000-03fff
        db   4      ; 4000-4fff     04000-04fff
        db   5      ; 5000-5fff     05000-05fff
        db   6      ; 6000-6fff     06000-06fff
        db   7      ; 7000-7fff     07000-07fff
        db   8      ; 8000-8fff     08000-08fff
        db   9      ; 9000-9fff     09000-09fff
        db   0ah    ; a000-afff     0a000-0afff
        db   0bh    ; b000-bfff     0b000-0bfff
globl:  db   0ch    ; c000-cfff     0c000-0cfff
        db   0dh    ; d000-dfff     0d000-0dfff
        db   0eh    ; e000-efff     0e000-0efff
        db   0fh    ; f000-ffff     0f000-0ffff
```

Subsequent bank switching in this example would involve loading the sequences 10h-1bh, 1ch-27h, 28h-33h, and 34h-3fh into registers 40h-4bh to switch banks as required. The bank switch routine might be similar to the init routine above except that the line: mvi a,50h would be changed to: mvi a,4ch so that global memory would not be overwritten.

The example code on the following page demonstrates a typical bank switch sequence. Note that the code is intended to be descriptive rather than optimal.

```
; this routine demonstrates a typical bank switch sequence
; to switch 48k from logical 0000-bfff to the 4th bank
; (28000-3ffff)


switch:   lxi  h,mmap
          mvi  c,40h
sw1:      mov  b,m
          cout b
          inx  h
          inr  c
          mvi  a,4ch
          cmp  c
          jrnz sw1
          ret

;  user3 bank map:

          written to    logical       physical
          register      address       address

mmap:     db   28h   ; 0000-0fff      28000-28fff
          db   29h   ; 1000-1fff      29000-29fff
          db   2ah   ; 2000-2fff      2a000-2afff
          db   2bh   ; 3000-3fff      2b000-2bfff
          db   2ch   ; 4000-4fff      2c000-2cfff
          db   2dh   ; 5000-5fff      2d000-2dfff
          db   2eh   ; 6000-6fff      2e000-2efff
          db   2fh   ; 7000-7fff      2f000-2ffff
          db   30h   ; 8000-8fff      30000-30fff
          db   31h   ; 9000-9fff      31000-31fff
          db   32h   ; a000-afff      32000-32fff
          db   33h   ; b000-bfff      33000-33fff
globl:               ; c000-cfff      0c000-0cfff
                     ; d000-dfff      0d000-0dfff
                     ; e000-efff      0e000-0efff
                     ; f000-ffff      0f000-0ffff
```

INITIAL PAGE ADDRESS SELECTION

In some applications it may be found advantageous to deselect the SS256 when power is first applied. This is necessary for instance in **VIRTUAL DISK** applications where Vdisk memory is located in the second 256k address space (40000-7ffff). In the pass mode, at initial power-up, addresses A19-A16 are supplied through a driver (1/2 of an LS244 at location E11) to the values set by berg jumpers at J2 (see Board Parameterization section). In most applications, these four bits will be set to zero so that the SS256 will be selected upon power up.

DIRECT 24-BIT ADDRESSING

Direct 24-bit addressing involves configuring the six bits at jumper J1 that correspond to address bits A23-A18, and setting the addressing mode jumper to '24 bit' (see Board Parameterization section). The following chart shows the proper jumper configuration to set the address range for each of the first sixteen 256k blocks:

Board Address range	A23	A22	A21	A20	A19	A18
000000 - 03ffff	R	R	R	R	R	R
040000 - 07ffff	R	R	R	R	R	L
080000 - 0bffff	R	R	R	R	L	R
0c0000 - 0fffff	R	R	R	R	L	L
100000 - 13ffff	R	R	R	L	R	R
140000 - 17ffff	R	R	R	L	R	L
180000 - 1bffff	R	R	R	L	L	R
1c0000 - 1fffff	R	R	R	L	L	L
200000 - 23ffff	R	R	L	R	R	R
240000 - 27ffff	R	R	L	R	R	L
280000 - 2bffff	R	R	L	R	L	R
2c0000 - 2fffff	R	R	L	R	L	L
300000 - 33ffff	R	R	L	L	R	R
340000 - 37ffff	R	R	L	L	R	L
380000 - 3bffff	R	R	L	L	L	R
3c0000 - 3fffff	R	R	L	L	L	L

The above table represents only one fourth of the possible address range selections for the SS256. If it is necessary to generate more table entries, continue in binary until the maximum (16 megabytes) limit is reached. The maximum 24-bit address block will be fc0000-ffffff (16,515,072 - 15,777,215 decimal).

In the above example, references to 'R' indicate installation of berg jumpers to the right to assert a low level. references to 'L' indicate installation to the left to assert a high level (see Board Parameterization section).

OPTIONS

This section deals with all of the optional features on board the SS256 that must be set by the user according to the particular environment in which the memory will be installed. The following options are covered in this section:

a. refresh mode options

b. phantom line options

c. pstval* timing options

d. sm1 timing options

refresh cycles. Refresh cycles are generated on-board the SS256 from two sources:

1. Transparent refresh cycles are asserted by the cycle control module immediately following a specified type of memory read cycle. Depending on the CPU being used, the specified type of memory read cycle can be either an m1 (op-code fetch cycle) only or any memory read cycle. As a general rule, if the system implements an 'sm1' signal on pin 44 of the bus the SS256 should be parameterized to operate with transparent refresh on m1 cycles only. If the system does not implement the 'sm1' signal the SS256 should be set to assert refresh on all memory read cycles.

2. Timed refresh cycles are generated from the on-board refresh timer approximately every 14 microseconds. The timer is reset each time a refresh cycle occurs. When the SS256 is set up to run in transparent refresh mode, under normal operating conditions the refresh timer will never time out. Transparent refresh activity will keep the refresh timer inactive. Timed refresh is selected by default as the principal refresh mode when no transparent refresh mode is selected. The refresh timer can be thought of as a 'watchdog' for failsafe refresh operation. When transparent refresh activity is suspended during wait, reset, or dma cycles, the refresh timer assumes the responsibility of keeping the memory array refreshed.

refresh mode selection. The SS256 is designed to provide the optimal refresh operation for any existing CPU type. The text below is intended to serve as a guide to refresh mode selection although some experimentation may be necessary to determine the most efficient refresh mode to be used.

For all Z80 and 8080 processors the transparent m1 refresh mode should be used. This mode provides the greatest access time for these processor types due to the extended m1 cycle. Both processors employ four clock cycles in an op-code fetch compared to the normal three cycles in a memory read or write cycle. The extra cycle time after the op-code is fetched is used internally by the processor to decode the instruction and the bus is idle during this time. See the section on Board Parameterization for procedure to set the refresh mode.

As a rule, the m1 transparent refresh mode should always be used with processors that provide the 'sm1' signal on the bus. This rule usually holds true due to the respective difference of the frequency of occurance of m1 cycles to memory read or write cycles even if the m1 bus cycle provides no extra clock cycles. A slight advantage in power consumption will result from minimizing refresh activity.

If the processor operates at a very low clock frequency (1 - 2 megahertz), transparent refresh on all memory read cycles should be selected. This choice assures that the ram refresh requirements will be met. When this refresh mode is selected wait states will never be invoked as a result of cycle conflict since refresh requests will be generated by the cycle control module.

Some processors such as the 6809 or 8086 do not offer a particular time slot into which transparent refresh cycles can be inserted. A different approach is required with these types of processors in order to properly refresh the SS256. An alternate mode is provided to satisfy the refresh requirements of processors of this category. The timed refresh mode asynchronously requests a refresh cycle every 14 microseconds. When a refresh request occurs while a memory cycle is in progress, the request is queued until the memory cycle is completed. If a memory cycle is requested while a refresh is in progress, the ready line is asserted and held until the refresh cycle is completed and the memory cycle has progressed to the point that data is sure to be stable. This mode will operate with any processor, although it is not the preferred mode in systems where a transparent mode can be implemented. To set the SS256 to operate in the timed refresh mode, see the section on Board Parameterization.

phantom line options. Three options are provided on the SS256 for the phantom line:

1. phantom disable all memory cycles
2. phantom disable memory read cycles only
3. ignore phantom line

See the section on Board Parameterization for information on setting the jumpers for desired phantom operation.

BOARD PARAMETERIZATION

This section is intended to serve as a guide to assist in the assignment of the various options required to personalize the SS256 to the user's system. All user-selectable options have been implemented with Berg jumpers in preference to dip switches to enhance the reliability of the programmed settings once the board is parameterized. Six jumper areas provided on the SS256 must be configured to fit the user's system before the board can be placed into operation.

memory address parameters: The SS256 can be addressed on any 256 kbyte boundary from 0 to 16 mbytes (16,777,216) in the direct 24-bit addressing mode. In the 16-bit addressing mode, the upper 4 bits (A23 - A20) must be set to zero, allowing addressing up to one mbyte (1,048,576) when four SS256 memory boards are installed in a system. Selection of phantom operation is included in this logical grouping.

pass mode (power-up page select) address: When power is initially applied to the system, the SS256 assumes the status of a 64k memory board by passing the logical address from the bus directly to the lower 64k block of the memory array. This allows the operating system to use the SS256 for system initialization before the mapping registers are written. Mapped mode addressing does not begin until immediately after the highest map register (xF) is loaded through i/o. If an application requires that the SS256 is not enabled on power-up, the power-up page address should be set to non-zero.

Note: This mode is not invoked when the 24-bit direct addressing mode is selected.

i/o address selection: When the SS256 is used in the 16 bit addressing mode, it is essential to load all sixteen mapping registers during the initial loading sequence. The registers can be loaded in any order, but the highest register (xF) must be loaded last in the sequence. Taking care to organize the initialization sequence in this manner will assure a smooth transition from the power-up mode to the mapped addressing mode after register xF is written.

Selection of map register i/o port addressing requires a sequential block of sixteen i/o addresses starting on an even 16 byte boundary (x0 - xF).

The following three tables list the jumper locations for the memory address, power-up page, and i/o assignments.

Memory addressing selections:

POSITION	FUNCTION	INSTALLED LEFT	INSTALLED RIGHT
J1-A	memory addr mode	16 bit mode	24 bit mode
J1-B	memory address bit	A19 = 1	A19 = 0
J1-C	memory address bit	A23 = 1	A23 = 0
J1-D	memory address bit	A18 = 1	A18 = 0
J1-E	memory address bit	A21 = 1	A21 = 0
J1-F	memory address bit	A22 = 1	A22 = 0
J1-G	memory address bit	A20 = 1	A20 = 0
J1-H	phantom	inactive	active

Power-up page select:

POSITION	FUNCTION	INSTALLED LEFT	INSTALLED RIGHT
J2-H	power-up addr bit	A19 = 1	A19 = 0
J2-G	power-up addr bit	A18 = 1	A18 = 0
J2-A	power-up addr bit	A17 = 1	A17 = 0
J2-F	power-up addr bit	A16 = 1	A16 = 0

I/O port address select:

POSITION	FUNCTION	INSTALLED LEFT	INSTALLED RIGHT
J2-E	i/o port addr bit	A7 = 1	A7 = 0
J2-C	i/o port addr bit	A6 = 1	A6 = 0
J2-D	i/o port addr bit	A5 = 1	A5 = 0
J2-B	i/o port addr bit	A4 = 1	A4 = 0

The status valid strobe - pstval*: The newer CPU cards provide a properly timed pstval* signal on pin 25 of the bus. If any doubt exists, a method to determine the validity is to check the CPU manual for the mnemonic assigned to this pin. If it is called out by any name other than pstval* then it probably was not designed to meet the IEEE/696 specifications. The SS256 provides a signal called "alt stval" which can be used in most pre-IEEE systems as a substitute for the pstval* signal if necessary. The implementation of this option involves cutting the etch at J3 between the center and left side pad, and installing a wire jumper between the center and right side pad. Jumper J3 is located down near the bus just above pin 25.

POSITION	FUNCTION	INSTALLED LEFT	INSTALLED RIGHT
J3	status valid strobe	alt stval*	pstval*(pin 25)

SETTING THE REFRESH MODE

refresh mode selection: The method of refreshing the SS256 is selected at jumper strip J4, pins A - E. A single shunt across two of these pins will set the desired mode. Read the section entitled "options" to determine the optimal refresh mode selection.

Jumper J4 is the only vertical jumper strip on the SS256, located along the left side of the 74F64 at coordinate location E20. Positions A through E are provided for refresh mode selection (position A is oriented at the top). When the optimal mode for your system has been determined (after reading the section entitled 'Options'), the following chart can be used to determine the proper shunt position for the selected refresh mode:

refresh mode	shunt position
1. Transparent refresh on all memory read cycles:	J4-B to J4-C
2. Transparent refresh on m1 cycles only:	J4-A to J4-B
3. Timed (asynchronous) refresh:	J4-D to J4-E

phantom line parameterization: The phantom line can be set to function in one of three different modes of operation:

1. Phantom disable all memory cycles: In this mode when the phantom line is active in either a memory read or a memory write cycle, the SS256 will be deselected.

2. Phantom disable memory read cycles only: This mode is provided for shadow prom applications where code is read from rom memory and written to overlayed ram in the power-up sequence prior to disabling (shadowing) the prom.

3. Ignore phantom line. Some applications do not require the use of the phantom line. This jumper setting assures that spurious phantom cycles will never occur regardless of the level of the phantom line.

Phantom mode selection:

POSITION	FUNCTION	INSTALLED LEFT	INSTALLED RIGHT
J1-H	phantom	ignored	enabled
J5	shadow	disable reads only when J1-H enabled	disable all mem cycles when J1-H enabled

INTRODUCTION

The installation of Macrotech's Virtual Disk software in CPM 2.2, in conjunction with the SS256 memory board provides an additional disk drive to the system's configuration. The virtual disk has two advantages. It adds another single density/single sided drive to the system and the solid state nature of the drive results in much higher access and data transfer rates.

Implementation of the virtual disk consists of installing the VBIOS routine in the CPM 2.2 BIOS module and adding the VDISK program to the system disk as an available utility program. The SS256 memory board is added to the systems hardware configuration and a 4K byte block of RAM memory in the existing address space is freed for the bank switching address window.

After installation, the operator enters **vdisk** commands as needed to initialize, open, close, activate, or deactivate the virtual disk.

The following sections assume the reader has familiarized himself with the source listings in Appendices A and B, with particular attention to the comments preceeding each of the source code listings.

RESIDENT MEMORY CONFIGURATION

A 4K byte window of memory must be provided in the systems' 64K address space that resides above the highest address used by CPM. This window must reside on a 4K byte boundary address; for example, 0C000H, 0D000H, 0E000H or 0F000H.

For periods of non-cpm use, an initialize routine can be coded that will switch Bank 0 of the SS256 into memory which when added to the existing RAM will provide a total of 64K address space. Bank 0 of the memory board is not used by virtual disk for this reason.

MEMORY BOARD CONFIGURATION

The SS256 memory board that is to be used as the virtual disk must be jumper configured as follows:

1. I/O port address matching PORTID equate in the VBIOS routine.

2. Board Select Bits matching BDSLCT equate in the VBIOS routine. These may not be 00, but may be 01, 10 or 11.

3. Bank Select Mode.

4. Board Select Address A20 to A23 of 0000.

5. Power up page select address A16 to A19 of 0000.

6. Other options as required for the user's system.

BIOS DISK ASSIGNMENTS

Virtual Disk may be assigned only as an equivalent of an 8 inch single density/single sided drive with 128 byte sectors, 26 sectors per track and 77 tracks. An entry must exist in the BIOS Disk Parameter Table that matches a true physical disk drive with this configuration. It is recommended that an additional entry in the Disk Parameter Table be provided that does not necessarily have a corresponding physical drive. This will allow the use of the virtual disk as an added drive instead of only as a temporary replacement for an existing physical drive.

Vdisk will accept disk assignments from A: thru P:. The actual assignment of a drive that does not have a corresponding entry in the BIOS Disk Parameter Table to vdisk will result in erratic operation and failure of the CPM operating system.

VDISK COMMANDS

The Vdisk program is the interface between the operator and the virtual disk. The operator executes the commands by entering VDISK X:COMMAND, where the X: is the disk assignment and the COMMAND is one of the below listed commands.

Vdisk tests only the first two characters of the command. The Open command; for example, might be entered as VDISK X:OP. The following describes each of the Vdisk commands.

vdisk x:open The entire 77 tracks of the physical disk on drive X: is copied to the virtual disk and the virtual disk is assigned as X:. All references to X: at this time will access the SS256 memory board. The physical drive that was X: has effectively been removed from the system.

vdisk x:close The entire 77 simulated tracks of the memory board are copied to the physical disk X:. The physical drive is assigned as X: and the virtual disk is effectively not present in the system.

vdisk x:activate The virtual disk is assigned as X: and any corresponding physical drive is effectively removed from the system. The data contents of the virtual disk are assumed correct. Operator procedure must assure this, prior to using this command.

vdisk x:deactivate The virtual disk is effectively removed from the system. Any corresponding physical drive for X: is now accessible. The virtual disk data contents are unchanged and may be later used via an Activate command unless a power down/power up cycle occurs. System resets will not affect the virtual disk data contents. If the active virtual disk is assigned as A: and a warm boot is requested, the physical drive will be accessed for the system tracks but following warm boot completion the virtual disk is again activated.

vdisk x:init Assigns the virtual disk as X: and fills all 77 simulated tracks with E5H. Each track is written with E5H and then is read back to verify the memory. It is not possible to use the command on a physical disk drive for initialization.

vdisk x:test If X: is a physical drive, the disk is filled with psuedo-random data and a read verify is performed. Note that the disk will be left with garbage data contents.

If X: is the active virtual drive, the same write, read verify process occurs and the data contents of the virtual disk is now garbage.

CAUTION: The operator should verify that the appropriate assignment is active prior to entering this command.

vdisk x:reset This command performs a BDOS Disk Reset function call. It is not a normally required command for virtual disk operation, but is included as a convenience to recover from systems that are left in read/only status.

vdisk status This command returns a message of VDISK INACTIVE or VDISK ACTIVE AS X:. This is an operator convenience command to report the current status of the virtual disk.

INTRODUCTION

The use of the Macrotech SS256 memory board in a system with an MPM II installation provides a flexible bank switch memory generation capability for multiple memory segments of various sizes to meet the user's needs. Installation of two SS256 memory boards provides sufficient memory segmentation to exceed the MPM II maximum capacity of 8 memory segments, each of up to 48K address space.

Appendix C contains the source listing of the two areas of code needed by the user for installation in their MPM BNKXIOS module. After this installation the operator uses the MPM GENSYS program to configure the desired memory segmentation by responding to the GENSYS prompts with the appropriate bank start addresses and sizes.

RESIDENT MEMORY CONFIGURATION

The SS256 memory board (or the first of two boards) must be configured with appropriate jumper positioning to provide the following board installation.

1. I/O Port Addressing to match the PORTID equate value.
2. Board Select Bits assignment of 00.
3. Bank Select Mode.
4. Board Select Address A20 thru A23 of 0000.
5. Power up Page Select Address A16 thru A19 of 0000.
6. Remaining jumpers should be set as system hardware requirements dictate.

If a second SS256 memory board is to be installed, it is configured the same as above except the Board Select Bits should be set to 01.

GENSYS REQUIREMENTS

The specification of memory assignments performed during SYSGEN requires the user to understand the capacity limitations of matching MPM memory to the available bank switch memory. The SS256 memory board will be used as a 64 bank board with each 4K byte bank assigned bank numbers of 0 to 63 decimal.

The MPM total size (to top of memory) will use banks 0 to X, where X is determined by the top of memory value. For example; if the top of memory is specified as 0EFH, then banks 0 thru 0EH will be assigned for common memory and segment 0. The implementation requires that this first segment be contiguous from address 0000H to the top of memory (in this example case, 0EFFFH). There are now 49 banks available for use in other memory segments since 15 banks are in use for segment 0 and common memory.

The response to the address for the common memory base page must be on a 4K boundary. For example, if the common memory size defines a possible base page address of 082H, then the response should be 80H. This will allow segment 0 bank switch to occur at the required 4K boundary. Segment 0 will be switched out of the system by removing banks 0 through 7 to allow other segments to enter the switched memory space. Banks 8 through 0EH will never be switched out of the memory address space.

The other memory segments (other than 0) may be specified as to base address and size, as long as the 4K byte boundary restriction is observed and no more than the maximum available banks are exceeded. In the example, 49 banks are available.

Below is an example of one of the many possible logical memory segmentations.

MPM II	Sys	8000H	7000H	Bank	00
Memseg	Usr	0000H	9000H	Bank	01
Memseg	Usr	0000H	9000H	Bank	02
Memseg	Usr	0000H	9000H	Bank	03
Memseg	Usr	0000H	9000H	Bank	04
Memseg	Usr	0000H	6000H	Bank	05
Memseg	Usr	4000H	4000H	Bank	06
Memseg	Usr	2000H	3000H	Bank	07
Memseg	Usr	0000H	8000H	Bank	00

Notice that all addresses are on 4K byte boundaries, the total number of banks required is 64, and that user segments other than segment 0 may be specified with base pages and lengths of varying 4K byte values. The installation of a second SS256 memory board increases the available banks from 64 to 128.

The system initialization routine in Appendix C does NOT perform verification of the address assignments or total bank requirements. The user must provide this logical checking at the time of GENSYS memory specification.

```
;
;   VBIOS ROUTINE TO BE INSTALLED IN USERS BIOS
;
;   VBIOS REV 1.2  1/12/82
;
;   COPYRIGHT 1982, MACROTECH INTERNATIONAL CORP.
;   ALL RIGHTS RESERVED.
;
;   THESE ROUTINES ARE DISTRIBUTED TO THE OWNERS OF THE
;   MACROTECH SS256 MEMORY BOARD FOR USE ONLY IN THOSE
;   SYSTEMS THAT ARE USING THE MEMORY BOARD.  ANY OTHER
;   USE CONSTITUTES A BREACH OF THE COPYRIGHT LICENSE TO
;   THE PURCHASER.
;
;   VBIOS DEFINES THE ONLINE ROUTINES TO BE INSTALLED
;   IN THE USER'S CPM2.2* BIOS.  THESE ROUTINES
;   PROVIDE AN INITIALIZATION AND INTERCEPT THE
;   SELECT, READ AND WRITE CALLS TO THE BIOS JUMP
;   TABLE.  IF THE CALL DOES NOT REFERENCE THE
;   CURRENTLY ACTIVE VDISK, EXECUTION IS PASSED TO
;   THE BIOS STANDARD READ/WRITE/SELECT ENTRY POINTS.
;
; NOTE* CP/M IS A REGISTERED TRADE MARK OF
;       DIGITAL RESEARCH CORPORATION
;
;
;   RESIDENT MEMORY - THE PROCESSOR'S ADDRESS SPACE
;                     (UP TO 60K) THAT IS PROVIDED
;                     BY OTHER THAN THE VDISK MEMORY
;                     BOARD.
;   WINDOW          - THE 4K BYTE AREA, NOT PROVIDED
;                     BY RESIDENT MEMORY THAT IS BANK
;                     SWITCHED IN/OUT FOR VDISK TRANSFERS.
;   COMMON BANK     - A 4K BYTE BANK THAT IS ACTIVE
;                     AFTER A COLD/WARM START AND IS NOT
;                     USED BY VDISK.
;   LOGICAL RECORD  - THE 63 BANKS CONTAIN 2002 LOGICAL
;                     RECORDS OF 128 BYTES EACH.  THE
;                     LOGICAL RECORD NUMBER IS DEFINED
;                     BY THE SUM OF TRACK #(0 TO 76)
;                     TIMES 26 AND THE SECTOR #(1 TO 26)
;                     MINUS 1.
;                     THE REQUIRED BANK #(1 TO 63) IS
;                     OBTAINED FROM BITS 5 THRU 10 OF
;                     THE LOGICAL RECORD #.
;                     THE RELATIVE RECORD WITHIN A BANK
;                     IS OBTAINED FROM BITS 0 THRU 4 OF
;                     THE LOGICAL RECORD #.
;
;   VBIOS CONVERTS THE DESIRED PHYSICAL DISK ADDRESS INTO
;   THE LOGICAL RECORD NUMBER, AND THEN OBTAINS THE BANK
;   NUMBER AND 128 BYTE RECORD CORE ADDRESS TO INTERFACE
;   TO THE VDISK MEMORY.  THE NEEDED BANK IS THEN SWITCHED
```

```
                        ;    INTO MEMORY AND THE APPROPRIATE DATA TRANSFER DONE.
                        ;
                        ;
                        ;    THE INITIALIZE ROUTINE SETS THE VDISK MEMORY BOARD
                        ;    MAPPING REGISTERS TO THE APPROPRIATE 4K BYTE BANKS.
                        ;
                        ;
                        ;    THE BANK SWITCHING CODE FORMS THE PORT # FROM THE
                        ;    BASE PORT VALUE(PORTID) AND THE 4 MOST SIGNIFICANT
                        ;    BITS OF THE WINDOW ADDRESS.  AN 8-BIT DATA VALUE
                        ;    IS OUTPUT TO PERFORM THE BANK SWITCH.  THIS VALUE
                        ;    IS OBTAINED BY COMBINING THE BOARD SELECT BITS
                        ;    (1, 2, or 3) WITH A 6-BIT VALUE (1 TO 63) THAT
                        ;    IS THE DESIRED BANK NUMBER.
                        ;
                        ;
                        ;    MEMORY BOARD CONFIGURATION EQUATES
                        ;         MUST BE CHANGED TO AGREE WITH USER'S SYSTEM
                        ;
00C0 =                  WDWADR: EQU     0C0H  ;WINDOW ADDR, UPPER BYTE ONLY
0040 =                  BSLCT:  EQU     40H   ;BOARD SELECT BITS, UPPER 2 BITS ONLY
0060 =                  PORTID: EQU     60H   ;BOARD I/O PORT ADDRESS
0030 =                  SELBITS:EQU     30H     ;DISK SELECT BITS
                        ;
                        ;    LINKS TO USER'S BIOS THAT MUST BE SATISFIED.
                        ;    USER MUST REMOVE THESE EQUATES PRIOR TO ASSEMBLY
                        ;    INTO USER'S BIOS.
                        ;
0000 =                  READ:   EQU     0       ;BIOS READ ENTRY
0000 =                  WRITE:  EQU     0       ;BIOS WRITE ENTRY
0000 =                  SELDSK: EQU     0       ;BIOS SELECT DISK ENTRY
0000 =                  SEKTRK: EQU     0       ;CURRENT TRACK #
0000 =                  SEKSEC: EQU     0       ;CURRENT SECTOR #
0000 =                  SEKDSK: EQU     0       ;CURRENT DESIRED DISK
0000 =                  SEKSEL: EQU     0       ;CURRENT DISK SELECT BITS
0000 =                  DPBASE: EQU     0       ;BIOS DISK PARAMETER HEADER TABLE
0000 =                  DMAAD:  EQU     0       ;CURRENT BUFFER ADDRESS
                        ;
                        ;    CHANGES REQUIRED TO USER'S BIOS.
                        ;
                        ;JMP READ:IN BIOS JUMP TABLE MUST BE CHANGED TO JMP VREAD.
                        ;
                        ;JMP WRITE:IN BIOS JUMP TABLE MUST BE CHANGED TO JMP VWRITE.
                        ;
                        ;JMP SELDSK:IN BIOS JUMP TABLE MUST BE CHANGED TO JMP VSLCT.
                        ;
                        ;CALL VINIT:MUST BE ADDED NEAR END OF WARM BOOT ROUTINE.
                        ;           NOTE THAT VINIT DESTROYS A, BC AND HL CONTENTS.
                        ;
                        ;
                        ;
                        ;    VBIOS LOCAL DEFS AND EQUATES
                        ;
0000 0000               REQCADR: DW 0       ;CORE ADDRESS
0002 00                 RWCMND:  DB 0       ;READ/WRITE COMMAND SAVE
```

```
0001 =          RFLAG:   EQU 1        ;READ FLAG
0000 =          WFLAG:   EQU 0        ;WRITE FLAG
0003 F0         INTVNO:  DB 0F0H      ;INITIAL VDRIVE #
0004 F0         CURVNO:  DB 0F0H      ;CURRENT VDRIVE #
                ;                       ;F0H  =  COLD START
                ;                       ;10H  =  WARM START, VDISK INACTIVE
                ;                       ;0 TO 0FH  =  ACTIVE VDISK #
                ;
                ;CAUTION: INTVNO/CURVNO WORD MUST/MUST BE
                ;         JUST BEFORE VREAD.
                ;
                ; NOTE: USER SHOULD SET INTVNO TO A VALUE
                ;       CORRESPONDING TO AN EXISTING 8" DRIVE:
                ;       A:=0  B:=1,ETC.
                ;
0005 CDAD00     VREAD:   CALL    VREQST   ;IS REQ FOR VDRIVE?
0008 C20000              JNZ     READ     ;NO, GOTO BIOS READ
000B 3C                  INR     A        ;SET RFLAG
000C C31500              JMP     VCOMON
000F CDAD00     VWRITE:  CALL    VREQST   ;IS REQ FOR VDRIVE?
0012 C20000              JNZ     WRITE    ;NO, GOTO BIOS WRITE
0015 320200     VCOMON:  STA     RWCMND
0018 2A0000              LHLD    SEKTRK
001B 2600                MVI     H,0      ;HL= TRACK
001D 54                  MOV     D,H
001E 5D                  MOV     E,L      ;DE= TRACK
001F 29                  DAD     H        ;HL= TRACK *2
0020 44                  MOV     B,H
0021 4D                  MOV     C,L      ;BC= TRACK * 2, SAVE
0022 19                  DAD     D        ;HL= TRACK * 3
0023 29                  DAD     H        ;HL= TRACK * 6
0024 29                  DAD     H
0025 29                  DAD     H        ;HL= TRACK * 24
0026 09                  DAD     B        ;HL= TRACK * 26
0027 1600                MVI     D,0
0029 3A0000              LDA     SEKSEC
002C 5F                  MOV     E,A      ;DE= SECTOR - 1
002D 19                  DAD     D        ;HL= REQUESTED LOGICAL RECORD
002E 7D                  MOV     A,L
002F E61F                ANI     01FH
0031 47                  MOV     B,A      ;B= RELATIVE RECORD WITHIN BANK
0032 7D                  MOV     A,L
0033 07                  RLC
0034 07                  RLC
0035 07                  RLC
0036 E607                ANI     07
0038 6F                  MOV     L,A
0039 7C                  MOV     A,H
003A 07                  RLC
003B 07                  RLC
003C 07                  RLC
003D 85                  ADD     L        ;A= REQUESTED BANK #
003E 3C                  INR     A        ;REQUESTED BANK + 1
003F 4F                  MOV     C,A
0040 78                  MOV     A,B
```

```
0041 0F                 RRC
0042 E60F               ANI     0FH
0044 67                 MOV     H,A
0045 78                 MOV     A,B
0046 0F                 RRC
0047 E680               ANI     80H
0049 6F                 MOV     L,A     ;HL= RELATIVE ADDR WITHIN BANK
004A 1100C0             LXI     D,WDWADR * 256
004D 19                 DAD     D
004E 220000             SHLD    REQCADR ;CORE ADDRESS OF 128 BYTE RECORD
0051 79                 MOV     A,C     ;A= REQUESTED BANK +1
0052 F640               ORI     BSLCT   ;BOARD SELECT BITS
0054 D36C               OUT     PORTID + WDWADR / 16
0056 0680               MVI     B,128
0058 2A0000             LHLD    DMAAD
005B EB                 XCHG
005C 2A0000             LHLD    REQCADR ;DE=DEST, HL=SOURCE
005F 3A0200             LDA     RWCMND
0062 A7                 ANA     A       ;TEST FOR READ OR WRITE
0063 C26700             JNZ     DOMOVE
0066 EB                 XCHG
0067 7E         DOMOVE: MOV     A,M     ;MOVE 128 BYTES
0068 23                 INX     H
0069 EB                 XCHG
006A 77                 MOV     M,A
006B 23                 INX     H
006C EB                 XCHG
006D 05                 DCR     B
006E C26700             JNZ     DOMOVE
0071 97                 SUB     A       ;GOOD RESPONSE
0072 C9                 RET
0073 3A0400     VSLCT:  LDA     CURVNO
0076 B9                 CMP     C
0077 C20000             JNZ     SELDSK  ;NOT VDRIVE
007A 320000             STA     SEKDSK
007D 3C                 INR     A
007E F630               ORI     SELBITS ;FORM SELECT BITS/DRIVE
0080 320000             STA     SEKSEL  ;SELECT BITS
0083 3A0300             LDA     INTVNO
0086 CDA100             CALL    GETDP   ;GET INIT VDRIVE DPH ADDR
0089 E5                 PUSH    H
008A 79                 MOV     A,C
008B CDA100             CALL    GETDP   ;GET DESIRED VDRIVE DPH ADDR
008E 060C               MVI     B,12
0090 D1                 POP     D
0091 EB         VSLOOP: XCHG
0092 7E                 MOV     A,M
0093 23                 INX     H
0094 EB                 XCHG
0095 77                 MOV     M,A
0096 23                 INX     H
0097 05                 DCR     B
0098 C29100             JNZ     VSLOOP  ;MOVE 12 BYTES TO DPH ADDR
009B 79                 MOV     A,C
009C CDA100             CALL    GETDP   ;RETURN DPH ADDR
```

```
009F 79                 MOV     A,C     ;RETURN DRIVE #
00A0 C9                 RET
00A1 210000     GETDP:  LXI     H,DPBASE  ;GET/RETURN HEADER TABLE ADR
00A4 111000             LXI     D,16
00A7 3D         GDPLOOP:        DCR     A
00A8 F8                 RM              ;HL= DPH ADDR
00A9 19                 DAD     D
00AA C3A700             JMP     GDPLOOP
00AD 3A0400     VREQST: LDA     CURVNO
00B0 67                 MOV     H,A
00B1 3A0000             LDA     SEKDSK
00B4 94                 SUB     H
00B5 C9                 RET             ;ZERO FLAG = ACTIVE VDRIVE
00B6 3A0400     VINIT:  LDA     CURVNO
00B9 A7                 ANA     A
00BA F0                 RP              ;ITS NOT COLD START
00BB 3E10               MVI     A,010H
00BD 320400             STA     CURVNO  ;FLAG WARM START OCCURRENCE
00C0 3E40               MVI     A,BSLCT
00C2 D36C               OUT     PORTID + WDWADR / 16 ;BANK 0 AT WDWADR
00C4 3E0F               MVI     A,0FH
00C6 D36F               OUT     PORTID+0FH      ;ENABLE BOARD
00C8 97                 SUB     A
00C9 47                 MOV     B,A     ;MEM FILL CHAR = 0
00CA 0E11               MVI     C,17    ;RUN WRLOP2  16 TIMES
00CC 2100C0             LXI     H,WDWADR * 256  ;START ADDRESS
00CF 0D         WRLOOP: DCR     C
00D0 C8                 RZ
00D1 70         WRLOP2: MOV     M,B
00D2 23                 INX     H
00D3 3D                 DCR     A
00D4 C2D100             JNZ     WRLOP2
00D7 C3CF00             JMP     WRLOOP
00DA C9                 RET
                ;CAUTION - CHECK RESULTING BIOS OVERALL LENGTH FOR
                ;POSSIBLE SIZE PROBLEM.
```

```
                        ;
                        ;   VDISK - UTILITY PROGRAM FOR CPM* VIRTUAL DISK
                        ;
                        ;   VDISK REV 1.2  1/9/82
                        ;
                        ;   COPYRIGHT 1982, MACROTECH INTERNATIONAL CORP.
                        ;   ALL RIGHTS RESERVED.
                        ;
                        ;
                        ;   THESE ROUTINES ARE DISTRIBUTED TO THE OWNERS OF THE
                        ;   MACROTECH SS256 MEMORY BOARD FOR USE ONLY IN THOSE
                        ;   SYSTEMS THAT ARE USING THE MEMORY BOARD.  ANY OTHER
                        ;   USE CONSTITUTES A BREACH OF THE COPYRIGHT LICENSE TO
                        ;   THE PURCHASER.
                        ;
                        ; NOTE:* CP/M IS A REGISTERED TRADE MARK OF
                        ;        DIGITAL RESEARCH CORPORATION
                        ;
                        ;  VDISK IS CALLED BY USER TO INITIALIZE THE VDRIVE
                        ;  PARAMETERS, COPY THE PHYSICAL DISK TO VDRIVE,
                        ;  AND WHEN FINISHED WITH VDRIVE TO COPY BACK TO
                        ;  PHYSICAL DISK.
                        ;
                        ;  VDISK CALLS:
                        ;        X:OPEN     ASSIGN VDRIVE AS 'X' AND COPY
                        ;                   FROM PHYSICAL DISK
                        ;        X:DEACT    CLOSE VDRIVE, NO COPY
                        ;        X:ACT      OPEN 'X' AS VDRIVE BUT NO COPY
                        ;        X:CLOSE    COPY TO PHYSICAL DISK AND CLOSE VDRIVE
                        ;        X:INIT     OPEN 'X' AS VDRIVE AND FILL WITH E5'S
                        ;        RESET      RESET ALL DRIVES - RECOVER FROM R/O
                        ;        STAT       REPORT VDRIVE STATUS
                        ;        X:TEST     TEST 'X' WITH RANDOM PATTERN WRITE/READ
                        ;                   IF 'X' IS ACTIVE VDRIVE, THEN TEST WILL
                        ;                   WRITE RANDOM PATTERN TO MEMORY BANKS.
                        ;                   IF 'X' IS 8 INCH PHYSICAL DISK THEN TEST
                        ;                   WILL WRITE TO DISK.
                        ;
                        ;  WARNING - 'X:' IN THE OPEN AND CLOSE
                        ;                CALLS MUST BE AN 8 INCH SINGLE SIDE/SINGLE
                        ;                DENSITY DRIVE.
                        ;
                        ;
0100 =          TPA:    EQU     0100H   ;TRANSIENT PROG AREA
005C =          SFCB:   EQU     005CH   ;SOURCE FCB
0010 =          NOVDSK: EQU     010H    ;CURVNO VALUE FOR INACTIVE VDRIVE
0018 =          JMPOFF: EQU     8*3     ;OFFSET FROM WBOOT TO SELECT DISK
0009 =          PRINTF: EQU     9       ;PRINT BUFFER FUNCTION CALL
000E =          SLCTF:  EQU     0EH     ;SELECT DISK FUNCTION
0001 =          CINF:   EQU     01      ;CHAR IN FUNCTION
0002 =          COUTF:  EQU     02H     ;CHAR OUT FUNCTION
000B =          STATF:  EQU     0BH     ;GET CONSOLE STATUS
0003 =          CNTLC:  EQU     03H     ;CONTROL-C CHAR
000D =          RESTF:  EQU     0DH     ;RESET ALL DRIVES CALL
0005 =          BDOS:   EQU     05H     ;BDOS FUNCTION CALL ADDR
```

```
0002 =          WRUAL:  EQU     02      ;WRITE UNALLOCATED
0000 =          WRALL:  EQU     0       ;WRITE ALLOCATED
0001 =          WRDIR:  EQU     1       ;WRITE DIRECTORY
                ;
                ;
0100                    ORG     TPA
                ;
0100 210000     START:  LXI     H,0
0103 39                 DAD     SP      ;GET OLD STACK
0104 22CD05             SHLD    SAVSP   ;SAVE IT
0107 316513             LXI     SP,SPADR        ;MY OWN STACK
010A 3A5D00             LDA     SFCB+1
010D FE52               CPI     'R'     ;RESET?
010F CAF001             JZ      TRYRST  ;MAYBE
0112 FE53               CPI     'S'     ;STAT ?
0114 CA2901             JZ      BYPASS  ;MAYBE
0117 3A5C00             LDA     SFCB    ;DEFAULT FCB
011A A7                 ANA     A       ;VDISK DEFAULT?
011B CA4004             JZ      EXIT    ;A=0, VDRIVE MUST BE A THRU P
011E 3D                 DCR     A
011F 32CA05             STA     REQVNO
0122 FE0F               CPI     15      ;VDRIVE OUT OF RANGE?
0124 3E00               MVI     A,0
0126 D24004             JNC     EXIT    ;REPORT ERROR
0129 111800     BYPASS: LXI     D,JMPOFF        ;BUILD JUMP TABLE TO BIOS
012C 2A0100             LHLD    1
012F 19                 DAD     D
0130 22D405             SHLD    SELDA+1
0133 110300             LXI     D,3
0136 19                 DAD     D
0137 22D705             SHLD    STRKA+1
013A 19                 DAD     D
013B 22DA05             SHLD    SSECA+1
013E 19                 DAD     D
013F 22DD05             SHLD    SDMAA+1
0142 19                 DAD     D
0143 22E005             SHLD    DOREADA+1
0146 19                 DAD     D
0147 22E305             SHLD    DOWRITA+1
014A 2AE005             LHLD    DOREADA+1
014D 23                 INX     H
014E 5E                 MOV     E,M
014F 23                 INX     H
0150 56                 MOV     D,M
0151 EB                 XCHG
0152 2B                 DCX     H
0153 22CB05             SHLD    VNOADR  ;ADDR OF CURVNO IN ONLINE CODE
0156 3A5D00             LDA     SFCB+1
0159 FE4F               CPI     'O'     ;OPEN
015B CA8101             JZ      TRYOPN
015E FE43               CPI     'C'     ;CLOSE
0160 CA0403             JZ      TRYCLS
0163 FE49               CPI     'I'     ;INIT
0165 CA6603             JZ      TRYINT
0168 FE44               CPI     'D'     ;DEACTIVATE
```

```
016A CAE502             JZ      TRYDAC
016D FE41               CPI     'A'     ;ACTIVATE
016F CAF302             JZ      TRYACT
0172 FE53               CPI     'S'     ;STAT
0174 CAC702             JZ      TRYST
0177 FE54               CPI     'T'     ;TEST
0179 CA0002             JZ      TRYTST
017C 3E01       UCEXIT: MVI     A,1
017E C34004             JMP     EXIT    ;REPORT UNDEFINED COMMAND
0181 3A5E00     TRYOPN: LDA     SFCB+2
0184 FE50               CPI     'P'
0186 C27C01             JNZ     UCEXIT
0189 CD7E04             CALL    SETVOFF ;ASSURE VDRIVE NOT ACTIVE
018C CD6304             CALL    DOSLCT  ;SELECT DRIVE
018F 3E1A               MVI     A,26
0191 328003             STA     COMEND
0194 CDEF03             CALL    SETRPT  ;SETUP TRACK MSGE
0197 3EFF               MVI     A,0FFH
0199 327C03             STA     COMTRK
019C 3A7C03     OLOOP:  LDA     COMTRK
019F 3C                 INR     A
01A0 327C03             STA     COMTRK
01A3 FE4D               CPI     77
01A5 CADD01             JZ      OPNDONE
01A8 CDF603             CALL    TRPT    ;OUTPUT TRACK MSGE
01AB 3E01               MVI     A,1
01AD 327D03             STA     COMSEC
01B0 21E505             LXI     H,BUFFER
01B3 227E03             SHLD    COMBUF
01B6 97                 SUB     A
01B7 328103             STA     COMCMD  ;READ
01BA CD7E04             CALL    SETVOFF
01BD CD8203             CALL    IOCOM   ;DO I/O
01C0 C2E801             JNZ     IOEXIT  ;ERROR
01C3 3E01               MVI     A,1
01C5 327D03             STA     COMSEC
01C8 328103             STA     COMCMD  ;WRITE
01CB 21E505             LXI     H,BUFFER
01CE 227E03             SHLD    COMBUF
01D1 CD7604             CALL    SETVON
01D4 CD8203             CALL    IOCOM   ;DO I/O
01D7 C2E801             JNZ     IOEXIT  ;ERROR
01DA C39C01             JMP     OLOOP   ;NEXT TRACK
01DD CD7604     OPNDONE:        CALL    SETVON
01E0 2B                 DCX     H
01E1 77                 MOV     M,A     ;SET INIT DRIVE IN VBIOS ROUTINES
01E2 CD6D04             CALL    DOSLCTV ;SELECT VDRIVE
01E5 C3F801             JMP     DORST
01E8 CD7E04     IOEXIT: CALL    SETVOFF
01EB 3E03               MVI     A,3
01ED C34004             JMP     EXIT
01F0 3A5E00     TRYRST: LDA     SFCB+2
01F3 FE45               CPI     'E'
01F5 C27C01             JNZ     UCEXIT
01F8 0E0D       DORST:  MVI     C,RESTF
```

```
01FA CD0500            CALL    BDOS
01FD C33E04            JMP     GOODXIT
0200 3A5E00    TRYTST: LDA     SFCB+2
0203 FE45              CPI     'E'
0205 C27C01            JNZ     UCEXIT  ;UNDEFINED
0208 116405            LXI     D,WARNM
020B 0E09              MVI     C,PRINTF
020D CD0500            CALL    BDOS    ;WARNING MSGE
0210 0E01              MVI     C,CINF
0212 CD0500            CALL    BDOS    ;GET RESPONSE
0215 FE59              CPI     'Y'
0217 C23E04            JNZ     GOODXIT ;USER CANCELLED
021A 11C705    TRYTS2: LXI     D,CRLF
021D 0E09              MVI     C,PRINTF
021F CD0500            CALL    BDOS    ;BLANK LINE
0222 CD6304            CALL    DOSLCT
0225 3E1A              MVI     A,26
0227 328003            STA     COMEND  ;LAST SECTOR TO WRITE/READ
022A 3EFF              MVI     A,0FFH
022C 327C03            STA     COMTRK  ;SETUP FIRST TRK #
022F CDEF03            CALL    SETRPT  ;SETUP REPORT MSGE
0232 3A7C03    XLOOP:  LDA     COMTRK
0235 3C                INR     A
0236 327C03            STA     COMTRK
0239 FE4D              CPI     77
023B CAF801            JZ      DORST
023E CDF603            CALL    TRPT    ;SHOW ACTIVE TRK
0241 3E01              MVI     A,1
0243 327D03            STA     COMSEC
0246 328103            STA     COMCMD  ;WRITE
0249 21E505            LXI     H,BUFFER
024C 227E03            SHLD    COMBUF
024F CD7802            CALL    FILLRND ;FILL WITH RANDOM PATTERN
0252 CD8203            CALL    IOCOM   ;DO I/O
0255 C2E801            JNZ     IOEXIT  ;ERROR
0258 3E01              MVI     A,1
025A 327D03            STA     COMSEC
025D 97                SUB     A
025E 328103            STA     COMCMD  ;READ
0261 21E505            LXI     H,BUFFER
0264 227E03            SHLD    COMBUF
0267 CD8203            CALL    IOCOM   ;DO I/O
026A C2E801            JNZ     IOEXIT  ;ERROR
026D CD8F02            CALL    CHKRND  ;CHECK PATTERN
0270 CA3202            JZ      XLOOP   ;CONTINUE TEST
0273 3E05              MVI     A,5     ;BAD TEST MSGE
0275 C34004            JMP     EXIT
0278 97        FILLRND:        SUB     A
0279 32CF05            STA     FILORCHK        ;SET FILL FLAG
027C 3A7C03    RNDCOM: LDA     COMTRK  ;USE TRK # AS SEED
027F 3C                INR     A
0280 32D005            STA     SEED
0283 21E505            LXI     H,BUFFER
0286 010E00            LXI     B,14    ;LOOP FOR 13 TIMES, 2 SECTORS EACH
0289 C5                PUSH    B
```

```
028A C1         FILL13: POP     B
028B 0D                 DCR     C
028C C8                 RZ              ;DONE
028D C5                 PUSH    B
028E CDA902     FILL256:        CALL    RAND    ;GET NEXT BYTE
0291 3ACF05             LDA     FILORCHK
0294 A7                 ANA     A
0295 78                 MOV     A,B
0296 CA9E02             JZ      ITSFILL
0299 BE                 CMP     M
029A C0                 RNZ             ;BAD TEST
029B C39F02             JMP     CONTFIL
029E 77         ITSFILL:        MOV     M,A
029F 23         CONTFIL:        INX     H
02A0 C1                 POP     B
02A1 05                 DCR     B
02A2 C5                 PUSH    B
02A3 C28E02             JNZ     FILL256
02A6 C38A02             JMP     FILL13
02A9 3AD205     RAND:   LDA     INITFLG
02AC A7                 ANA     A
02AD 06E5               MVI     B,0E5H
02AF C0                 RNZ             ;ITS INIT, FILL WITH E5'S
02B0 3AD005             LDA     SEED
02B3 47                 MOV     B,A
02B4 1F                 RAR
02B5 1F                 RAR
02B6 1F                 RAR
02B7 A8                 XRA     B
02B8 1F                 RAR
02B9 78                 MOV     A,B
02BA 1F                 RAR
02BB 32D005             STA     SEED    ;AND RETURN RANDOM IN B
02BE C9                 RET
02BF 3EFF       CHKRND: MVI     A,0FFH
02C1 32CF05             STA     FILORCHK        ;FLAG CHECK OP
02C4 C37C02             JMP     RNDCOM  ;GOTO TO COMMON FILL/CHECK CODE
02C7 3A5E00     TRYST:  LDA     SFCB+2
02CA FE54               CPI     'T'
02CC C27C01             JNZ     UCEXIT
02CF 2ACB05             LHLD    VNOADR
02D2 7E                 MOV     A,M     ;GET VDRIVE VALUE FROM BIOS
02D3 C641               ADI     41H     ;FORM ASCII
02D5 323C05             STA     ONMSG+23        ;PUT IN ON MESSAGE
02D8 7E                 MOV     A,M
02D9 E6F0               ANI     0F0H
02DB 3E06               MVI     A,6     ;PRESET FOR OFFMSG
02DD C24004             JNZ     EXIT    ;REPORT OFF
02E0 3E07               MVI     A,7     ;ITS ACTIVE
02E2 C34004             JMP     EXIT
02E5 3A5E00     TRYDAC: LDA     SFCB+2
02E8 FE45               CPI     'E'
02EA C27C01             JNZ     UCEXIT
02ED CD7E04             CALL    SETVOFF ;DEACTIVATE VDRIVE
02F0 C3F801             JMP     DORST   ;RESET DRIVES
```

```
02F3 3A5E00     TRYACT: LDA     SFCB+2
02F6 FE43               CPI     'C'
02F8 C27C01             JNZ     UCEXIT
02FB CD7604             CALL    SETVON  ;SET ACTIVE,ASSUME DATA ALREAD`  JOD
02FE CD6D04     TRYAC2: CALL    DOSLCTV
0301 C3F801             JMP     DORST
0304 3A5E00     TRYCLS: LDA     SFCB+2
0307 FE4C               CPI     'L'
0309 C27C01             JNZ     UCEXIT
030C CD7E04             CALL    SETVOFF
030F CD6304     TRYCL2: CALL    DOSLCT
0312 3E1A               MVI     A,26
0314 328003             STA     COMEND
0317 CDEF03             CALL    SETRPT  ;SETUP TRACK MSGE
031A 3EFF               MVI     A,0FFH
031C 327C03             STA     COMTRK
031F 3A7C03     CLOOP:  LDA     COMTRK
0322 3C                 INR     A
0323 327C03             STA     COMTRK
0326 FE4D               CPI     77
0328 CA6003             JZ      CLSDONE
032B CDF603             CALL    TRPT    ;OUTPUT TRACK MSGE
032E 3E01               MVI     A,1
0330 327D03             STA     COMSEC
0333 21E505             LXI     H,BUFFER
0336 227E03             SHLD    COMBUF
0339 97                 SUB     A
033A 328103             STA     COMCMD  ;READ
033D CD7604             CALL    SETVON
0340 CD8203             CALL    IOCOM   ;DO I/O
0343 C2E801             JNZ     IOEXIT  ;ERROR
0346 3E01               MVI     A,1
0348 327D03             STA     COMSEC
034B 328103             STA     COMCMD  ;WRITE
034E 21E505             LXI     H,BUFFER
0351 227E03             SHLD    COMBUF
0354 CD7E04             CALL    SETVOFF
0357 CD8203             CALL    IOCOM   ;DO I/O
035A C2E801             JNZ     IOEXIT  ;ERROR
035D C31F03             JMP     CLOOP
0360 CD7E04     CLSDONE:        CALL    SETVOFF
0363 C3F801             JMP     DORST
0366 3A5E00     TRYINT: LDA     SFCB+2
0369 FE4E               CPI     'N'
036B C27C01             JNZ     UCEXIT
036E 3EFF               MVI     A,0FFH
0370 32D205             STA     INITFLG
0373 CD7604             CALL    SETVON
0376 CD6D04             CALL    DOSLCTV ;SELECT VDRIVE AND MARK INIT DRIVE
0379 C31A02             JMP     TRYTS2  ;USE TEST CODE TO FILL VDRIVE
                ;
037C 00         COMTRK: DB      0       ;SET TO DEGIRED TRACK
037D 00         COMSEC: DB      0       ;START SECTOR
037E 0000       COMBUF: DW      0       ;START BUFFER
0380 00         COMEND: DB      0       ;LAST SECTOR
```

```
0381 00          COMCMD: DB      0       ;0=READ, NON 0 = WRITE
                 ;
0382 CDD803      IOCOM:  CALL    SETIO   ;SETUP TRK, SEC, AND BUF
0385 060A                MVI     B,10    ;RETRIES
0387 C5          IOTRY:  PUSH    B
0388 3A8103              LDA     COMCMD
038B A7                  ANA     A
038C CA9703              JZ      READIT
038F 0E02                MVI     C,WRUAL ;WRITE UNALLOCATED
0391 CDE205              CALL    DOWRITA
0394 C39A03              JMP     IOCM4
0397 CDDF05      READIT: CALL    DOREADA
039A C1          IOCM4:  POP     B
039B A7                  ANA     A
039C CAA503              JZ      IOCM6   ;GOOD I/O
039F 05                  DCR     B
03A0 C28703              JNZ     IOTRY   ;RETRY
03A3 A7                  ANA     A
03A4 C9                  RET             ;A NON ZERO = ERROR
03A5 3A7D03      IOCM6:  LDA     COMSEC
03A8 47                  MOV     B,A
03A9 3A8003              LDA     COMEND
03AC 90                  SUB     B
03AD C8                  RZ              ;A=0, GOOD I/O
03AE 78                  MOV     A,B
03AF 3C                  INR     A
03B0 327D03              STA     COMSEC  ;NEXT SECTOR
03B3 2A7E03              LHLD    COMBUF
03B6 118000              LXI     D,128
03B9 19                  DAD     D
03BA 227E03              SHLD    COMBUF
03BD 0E08                MVI     C,STATF
03BF CD0500              CALL    BDOS    ;OP ABORT?
03C2 A7                  ANA     A
03C3 CA8203              JZ      IOCOM   ;NO
03C6 0E01                MVI     C,CINF
03C8 CD0500              CALL    BDOS    ;GET CHAR
03CB FE03                CPI     CNTLC
03CD C28203              JNZ     IOCOM   ;NOT ABEND
03D0 CD7E04              CALL    SETVOFF
03D3 3E08                MVI     A,8
03D5 C34004              JMP     EXIT    ;END MSGE = 8
                 ;
03D8 3A7C03      SETIO:  LDA     COMTRK  ;SET TRACK, SECTOR AND BUFFER
03DB 4F                  MOV     C,A
03DC CDD605              CALL    STRKA
03DF 3A7D03              LDA     COMSEC
03E2 4F                  MOV     C,A
03E3 CDD905              CALL    SSECA
03E6 2A7E03              LHLD    COMBUF
03E9 44                  MOV     B,H
03EA 4D                  MOV     C,L
03EB CDDC05              CALL    SDMAA
03EE C9                  RET
                 ;
```

```
03EF 21302F     SETRPT: LXI     H,02F30H
03F2 223B04             SHLD    RPTM+21
03F5 C9                 RET             ;TRACK NUMBERS ARE SETUP
03F6 3A3C04     TRPT:   LDA     RPTM+22
03F9 3C                 INR     A
03FA 323C04             STA     RPTM+22 ;BUMP LOW DIGIT
03FD FE3A               CPI     3AH
03FF C20E04             JNZ     TRPT4   ;NOT PAST 9
0402 3E30               MVI     A,30H
0404 323C04             STA     RPTM+22 ;RESET TO 0
0407 3A3B04             LDA     RPTM+21
040A 3C                 INR     A
040B 323B04             STA     RPTM+21
040E 21FFFF     TRPT4:  LXI     H,0FFFFH
0411 E5                 PUSH    H
0412 E1         TRPT5:  POP     H
0413 23                 INX     H
0414 7D                 MOV     A,L
0415 FE18               CPI     24
0417 C8                 RZ              ;ALL MSGE OUT
0418 E5                 PUSH    H
0419 112604             LXI     D,RPTM
041C 19                 DAD     D
041D 5E                 MOV     E,M
041E 0E02               MVI     C,COUTF
0420 CD0500             CALL    BDOS    ;OUTPUT A CHAR
0423 C31204             JMP     TRPT5
                ;
0426 0D         RPTM:   DB      0DH     ;CARRIAGE RETURN
0427 492F4F2041         DB      'I/O ACTIVE ON TRACK    '
043E 3E02       GOODXIT:        MVI     A,2
0440 32D105     EXIT:   STA     SAVERR
0443 11C705             LXI     D,CRLF
0446 0E09               MVI     C,PRINTF
0448 CD0500             CALL    BDOS    ;BLANK LINE
044B 3AD105             LDA     SAVERR
044E 07                 RLC
044F 5F                 MOV     E,A
0450 1600               MVI     D,0
0452 218304             LXI     H,MSGADRS
0455 19                 DAD     D
0456 5E                 MOV     E,M
0457 23                 INX     H
0458 56                 MOV     D,M
0459 0E09               MVI     C,PRINTF
045B CD0500             CALL    BDOS
045E 2ACD05             LHLD    SAVSP
0461 F9                 SPHL            ;RETURN THRU OLD STACK
0462 C9                 RET
0463 3ACA05     DOSLCT: LDA     REQVNO
0466 5F                 MOV     E,A
0467 0E0E               MVI     C,SLCTF
0469 CD0500             CALL    BDOS    ;SELECT PHYSICAL DRIVE
046C C9                 RET
046D 3ACA05     DOSLCTV:        LDA     REQVNO
```

```
0470 5F                 MOV     E,A
0471 4F                 MOV     C,A
0472 CDD305             CALL    SELDA   ;SELECT VDISK
0475 C9                 RET
0476 3ACA05     SETVON: LDA     REQVNO
0479 2ACB05     SETVONC:        LHLD    VNOADR
047C 77                 MOV     M,A
047D C9                 RET
047E 3E10       SETVOFF:        MVI     A,NOVDSK
0480 C37904             JMP     SETVONC
0483 9504       MSGADRS:        DW      VBAD
0485 AD04               DW      UNDEF
0487 C504               DW      GOODM
0489 D904               DW      IOERR
048B E904               DW      SCRERR
048D FE04               DW      TERRM
048F 0C05               DW      OFFMSG
0491 2505               DW      ONMSG
0493 4005               DW      ABORTM
                ;
0495 564449534BVBAD:    DB      'VDISK MUST BE A THRU P $'
04AD 434F4D4D41UNDEF:   DB      'COMMAND NOT RECOGNIZED $'
04C5 46554E4354GOODM:   DB      'FUNCTION COMPLETED $'
04D9 4449534B20IOERR:   DB      'DISK I/O ERROR $'
04E9 4E4F205644SCRERR:  DB      'NO VDRIVE YET SETUP $'
04FE 5445535420TERRM:   DB      'TEST FAILURE $'
050C 5649525455OFFMSG:  DB      'VIRTUAL DISK NOT ACTIVE $'
0525 5649525455ONMSG:   DB      'VIRTUAL DISK ACTIVE AS  : $'
0540 4F5045524IABORTM:  DB      'OPERATOR CANCELLED, VDISK INACTIVE $'
                ;
0564 5741524E49WARNM:   DB      'WARNING - DISK OR VDRIVE MEMORY '
0584 0D0A               DB      0DH,0AH
0586 57494C4C20         DB      'WILL BE OVERWRITTEN WITH RANDOM DATA '
05AB 0D0A               DB      0DH,0AH
05AD 434F4E5449         DB      'CONTINUE TEST - - Y OR N $'
                ;
05C7 0D0A24     CRLF:   DB      0DH,0AH,'$'
                ;
05CA 00         REQVNO: DB      0
05CB 0000       VNOADR: DW      0       ;SAVE CURVND ADDR IN ONLINE CODE
05CD 0000       SAVSP:  DW      0       ;SAVE FOR CPM STACK PTR
05CF 00         FILORCHK:       DB      0       ;FLAG FOR FILL/CHECK ROUTINE
05D0 00         SEED:   DB      0       ;SAVE FOR RANDOM SEED
05D1 00         SAVERR: DB      0       ;SAVE FOR ERR MSGE INDEX
05D2 00         INITFLG:        DB      0       ;IF NON-ZERO, INIT IN PROGRESS
                ;
                ;  FOLLOWING IS DYNAMICALLY BUILT JUMP TABLE
05D3 C3         SELDA:  DB      0C3H
05D4 0000               DW      0
05D6 C3         STRKA:  DB      0C3H
05D7 0000               DW      0
05D9 C3         SSECA:  DB      0C3H
05DA 0000               DW      0
05DC C3         SDMAA:  DB      0C3H
05DD 0000               DW      0
```

```
05DF C3          DOREADA:        DB      0C3H
05E0 0000                DW      0
05E2 C3          DOWRITA:        DB      0C3H
05E3 0000                DW      0
                 ;  END OF JUMP TABLE
05E5             BUFFER: DS      3328    ;I/O BUFFER FOR COPIES
12E5             SPAREA: DS      128     ;STACK AREA
1365 0000        SPADR:  DW      0       ;START OF STACK
```

```
                        ;
                        ;    MP/M II* ROUTINES FOR MACROTECH SS256 MEMORY BOARDS.
                        ;
                        ;       MPMBIOS  REV 1.1  2/1/8
                        ;
                        ;       COPYRIGHT 1982, MACROTECH INTERNATIONAL CORP.
                        ;       THESE ROUTINES ARE DISTRIBUTED TO THE OWNERS OF THE
                        ;       MACROTECH SS256 MEMORY BOARD FOR USE ONLY IN THOSE
                        ;       SYSTEMS THAT ARE USING THE MEMORY BOARD.  ANY OTHER
                        ;       USE CONSTITUTES A BREACH OF THE COPYRIGHT LICENSE TO
                        ;       THE PURCHASER.
                        ;
                        ; NOTE* MP/M IS A REGISTERED TRADE MARK OF
                        ;        DIGITAL RESEARCH CORPORATION
                        ;
                        ;    MEMORY BOARD CONFIGURATION EQUATES - MUST BE CHANGED TO
                        ;    AGREE WITH USER'S SYSTEM.
                        ;
0060 =                  PORTID: EQU     60H     ;BOARD (OR BOARDS) I/O ADDRESS
0000 =                  BD0SLCT:        EQU     00H  ;FIRST BOARD SELECT BITS
0040 =                  BD1SLCT:        EQU     40H  ;2ND BOARD(IF USED) SELECT BITS
0080 =                  NONSLCT:        EQU     80H  ;NONEXISTENT BOARD SELECT BITS
                        ;
                        ;    LINK TO USER'S BNKXIOS THAT MUST BE SATISFIED.
                        ;       USER MUST REMOVE THIS EQUATE PRIOR TO ASSEMBLY.
                        ;
0000 =                  SYSDAT: EQU     0       ;SYSTEM DATA PAGE ADDR
                        ;
                        ;    SELMEM LOCAL DEFS AND EQUATES
                        ;
0000 FF                 CURSEG: DB      0FFH  ;CURRENT SEGMENT # OF SELECTED BANKS
0001 0000               SAVJ0:  DW      0     ;JUMP ZERO LINK ADDR
0003 0000               SAVBHA: DW      0     ;BREAKPOINT HANDLER ADDR
0005 0000               SAVBVA: DW      0     ;BREAKPOINT VECTOR ADDR
0007 00                 SEG0:   DB      0     ;HIGHEST SEG 0 BANK #
0008 00                 CURTOP: DB      0     ;CURRENT HIGHEST USED BANK #
0009 00                 CURSZ:  DB      0     ;CURRENT BANK SIZE
000A 00                 NSEGS:  DB      0     ;NUMBER OF SEGMENTS COUNTER
000B 00                 BDSLCT: DB      0     ;CURRENT BOARD SELECT BITS
000C 0000               MDADR:  DW      0     ;CURRENT MEM DESCRIPTOR ADDR
                        ;
000E                    BKTAB:  DS      32  ;BANK TABLE FOR SEG # TO BANK # VALUES
                        ;
                        ;
                        ;    CAUTION - THE ABOVE DEFS AND THE FOLLOWING ROUTINE
                        ;    MUST BE INSTALLED IN THE USER'S BNKXIOS IN A LOCATION
                        ;    IN MPM COMMON MEMORY THAT WILL BE ACCESSIBLE DURING
                        ;    MPM ONLINE OPERATIONS.  THE ENTRY - SELMEM - MUST
                        ;    BE PLACED IN THE EXTENDED XIOS JUMP TABLE.
                        ;
                        ;
002E 210300             SELMEM: LXI     H,3
0031 09                         DAD     B       ;HL = ENTRY IN MEM DESCRIPTOR TABLE
0032 46                         MOV     B,M     ;GET SEG # FROM TABLE
0033 3A0000                     LDA     CURSEG
```

```
0036 B8                 CMP     B
0037 C8                 RZ                      ;SAME MEM SEG, NO BANK SWITCH
0038 78                 MOV     A,B
0039 320000             STA     CURSEG  ;NEW CURRENT SEG #
003C A7                 ANA     A
003D CA6400             JZ      DOSEG0  ;SPECIAL SEG 0 SWITCH
0040 07                 RLC
0041 07                 RLC
0042 5F                 MOV     E,A
0043 1600               MVI     D,0
0045 210E00             LXI     H,BKTAB
0048 19                 DAD     D       ;BANK TABLE ADDR FOR SELECTION
0049 46                 MOV     B,M     ;# OF 4K BANKS NEEDED
004A 23                 INX     H
004B 56                 MOV     D,M     ;PORTID + CORE ADDR
004C 23                 INX     H
004D 5E                 MOV     E,M     ;BDSLCT + BANK #
004E 05         SELOOP: DCR     B
004F F8                 RM              ;BANK SWITCH LOOP DONE
0050 7A                 MOV     A,D
0051 325600             STA     OUTID+1 ;SETUP OUT INSTRUCTION
0054 7B                 MOV     A,E
0055 D300       OUTID:  OUT     0       ;DO BANK SWITCH
0057 14                 INR     D
0058 1C                 INR     E
0059 7B                 MOV     A,E
005A E63F               ANI     03FH    ;ISOLATE BANK #
005C C24E00             JNZ     SELOOP  ;NOT PAST END OF FIRST BOARD
005F 1E40               MVI     E,BD1SLCT       ;NEXT BOARD, BANK 0
0061 C34E00             JMP     SELOOP  ;CONTINUE LOOP
0064 3A0700     DOSEG0: LDA     SEG0
0067 47                 MOV     B,A
0068 1660               MVI     D,PORTID
006A 1E00               MVI     E,BD0SLCT
006C 7A         SEGZ:   MOV     A,D
006D 327200             STA     ZOUT+1  ;PRESET OUT INSTRUCTION
0070 7B                 MOV     A,E
0071 D300       ZOUT:   OUT     0       ;ONE OF THE SEG 0 BANKS
0073 B8                 CMP     B
0074 C8                 RZ              ;DONE IF HIGHEST SEG 0 BANK
0075 14                 INR     D
0076 1C                 INR     E
0077 C36C00             JMP     SEGZ    ;CONTINUE SEG 0 BANK SWITCH
                ;
                ;
                ;   THE FOLLOWING INITIALIZE ROUTINE MAY BE INSTALLED
                ;   IN AN AREA OF THE BNKXIOS THAT WILL BE REUSED
                ;   AS BUFFERS ETC. DURING ONLINE OPS.  IT IS USED ONLY
                ;   DURING SYSTEM INIT.   THE ENTRY SYSINIT MUST BE
                ;   INSTALLED IN THE EXTENDED XIOS JUMP TABLE , OR THE
                ;   ROUTINE MAY BE INSTALLED IN THE EXISTING INIT CODE.
                ;
                ;
007A 220100     SYSINIT:        SHLD    SAVJ0
007D EB                 XCHG
```

```
007E 220300             SHLD    SAVBHA
0081 2600               MVI     H,0
0083 79                 MOV     A,C
0084 07                 RLC
0085 07                 RLC
0086 07                 RLC
0087 6F                 MOV     L,A
0088 220500             SHLD    SAVBVA  ;SAVE CALLER PARAMETERS
008B 210E00             LXI     H,BKTAB
008E 97                 SUB     A
008F 0620               MVI     B,32
0091 CD5401             CALL    FILLM   ;ZERO FILL THE BANK TABLE
0094 2A0000             LHLD    SYSDAT
0097 7E                 MOV     A,M     ;TOP OF MEM
0098 E6F0               ANI     0F0H    ;FORCE 4K BOUNDARY
009A 0F                 RRC
009B 0F                 RRC
009C 0F                 RRC
009D 0F                 RRC
009E 320800             STA     CURTOP  ;CURRENT HIGHEST USED BANK #
00A1 320E00             STA     BKTAB   ;# OF BANKS FOR SEG 0
00A4 F600               ORI     BD0SLCT
00A6 320700             STA     SEG0    ;PREPARE SEG 0 TOP BANK #
00A9 3E60               MVI     A,PORTID
00AB 320F00             STA     BKTAB+1 ;SEG 0 BOTTOM PAGE ADDR
00AE 3E00               MVI     A,BD0SLCT
00B0 321000             STA     BKTAB+2 ;SEG 0 BOTTOM BANK #
00B3 320B00             STA     BDSLCT  ;SETUP FOR BUILD LOOP
00B6 2A0000             LHLD    SYSDAT
00B9 110F00             LXI     D,15
00BC 19                 DAD     D
00BD 7E                 MOV     A,M
00BE 3D                 DCR     A
00BF 320A00             STA     NSEGS   ;# OF USER SEGMENTS
00C2 110800             LXI     D,1 + 4 + 3     ;SKIP PAST COMMON ENTRY
00C5 19                 DAD     D
00C6 220C00             SHLD    MDADR   ;ADDR OF CURRENT MEM DESCRIPTORS
00C9 111200             LXI     D,BKTAB+4       ;SETUP BANK TABLE ADDR
00CC 0600               MVI     B,0     ;PRESET BANK SETUP LOOP PTR
00CE 04         SEGLOOP:        INR     B       ;WORK ON NEXT SEG #
00CF 3A0A00             LDA     NSEGS
00D2 3D                 DCR     A
00D3 320A00             STA     NSEGS   ;UPDATE SEGLOOP COUNTER
00D6 CA7201             JZ      SETSEG0 ;DONE, NOW SETUP SEG 0
00D9 0E07               MVI     C,7     ;# OF ENTRIES FOR SEARCH
00DB 2A0C00             LHLD    MDADR
00DE 7E         SRCHLOOP:       MOV     A,M     ;GET THIS ENTRIES SEG #
00DF B8                 CMP     B
00E0 CAEE00             JZ      FOUND   ;FOUND MATCHING SEG #
00E3 0D         TRYNEXT:        DCR     C
00E4 FACE00             JM      SEGLOOP ;LOOK FOR NEXT SEG #
00E7 23                 INX     H
00E8 23                 INX     H
00E9 23                 INX     H
00EA 23                 INX     H
```

```
00EB C3DE00             JMP     SRCHLOOP        ;LOOK AT NEXT ENTRY
00EE 2B         FOUND:  DCX     H
00EF 2B                 DCX     H
00F0 7E                 MOV     A,M     ;SEG SIZE
00F1 A7                 ANA     A
00F2 C2FA00             JNZ     GOTONE
00F5 23                 INX     H
00F6 23                 INX     H
00F7 C3E300             JMP     TRYNEXT ;THIS ENTRY HAS SIZE = 0
00FA E6F0       GOTONE: ANI     0F0H    ;FORCE TO MULTIPLE OF 4K
00FC 0F                 RRC
00FD 0F                 RRC
00FE 0F                 RRC
00FF 0F                 RRC     ;# OF BANKS NEEDED
0100 EB                 XCHG
0101 77                 MOV     M,A     ;INTO BANK TABLE
0102 320300             STA     CURSZ   ;SAVE IT
0105 23                 INX     H
0106 EB                 XCHG
0107 2B                 DCX     H
0108 7E                 MOV     A,M     ;BASE ADDR
0109 E6F0               ANI     0F0H    ;FORCE TO 4K BOUNDARY
010B 0F                 RRC
010C 0F                 RRC
010D 0F                 RRC
010E 0F                 RRC
010F F660               ORI     PORTID
0111 EB                 XCHG
0112 77                 MOV     M,A     ;OUT INST FOR THIS SEG
0113 23                 INX     H
0114 0E01               MVI     C,1
0116 CD5B01             CALL    BUMPTP  ;SET CURTOP TO NEXT BANK
0119 4F                 MOV     C,A     ;NEW CURTOP
011A 3A0B00             LDA     BDSLCT
011D B1                 ORA     C
011E 77                 MOV     M,A     ;BANK # FOR THIS SEG
011F 2B                 DCX     H
0120 4F                 MOV     C,A
0121 7E                 MOV     A,M     ;PORTID
0122 23                 INX     H
0123 E6F0               ANI     0F0H    ;FORCE TO 0000H
0125 322A01             STA     OUTBSE+1
0128 79                 MOV     A,C
0129 D300       OUTBSE: OUT     0       ;SWITCH THIS BANK INTO 0000H
012B E5                 PUSH    H
012C 2A0100             LHLD    SAVJ0
012F 220100             SHLD    1       ;JUMP 0 VECTOR
0132 2A0500             LHLD    SAVBVA
0135 3EC3               MVI     A,0C3H  ;BUILD JUMP INSTRUCTION
0137 320000             STA     0
013A 77                 MOV     M,A     ;ALSO IN BREAKPOINT
013B 23                 INX     H
013C 3A0300             LDA     SAVBHA
013F 77                 MOV     M,A
0140 23                 INX     H
```

```
0141 3A0400             LDA     SAVBHA+1
0144 77                 MOV     M,A     ;BREAKPOINT VECTOR SETUP
0145 E1                 POP     H
0146 23                 INX     H
0147 23                 INX     H       ;SET TO NEXT ENTRY
0148 EB                 XCHG
0149 3A0900             LDA     CURSZ
014C 3D                 DCR     A
014D 4F                 MOV     C,A
014E CD5B01             CALL    BUMPTP  ;SET CURTOP TO NEXT USER
0151 C3CE00             JMP     SEGLOOP ;DO NEXT SEG
0154 77         FILLM:  MOV     M,A     ;PUT FILL CHAR
0155 23                 INX     H       ;NEXT
0156 05                 DCR     B       ;TEST FILL COUNT
0157 C25401             JNZ     FILLM   ;MORE TO DO
015A C9                 RET
015B 3A0800     BUMPTP: LDA     CURTOP  ;LAST BANK # USED
015E 3C                 INR     A
015F FE40               CPI     64
0161 C26A01             JNZ     NOTNXT  ;NOT OFF END OF BOARD
0164 3E40               MVI     A,BD1SLCT
0166 320B00             STA     BDSLCT  ;SET FOR NEXT BOARD
0169 97                 SUB     A       ;FIRST BANK
016A 320800     NOTNXT: STA     CURTOP
016D 0D                 DCR     C
016E C25B01             JNZ     BUMPTP  ;AGAIN
0171 C9                 RET
0172 CD6400     SETSEG0:        CALL    DOSEG0  ;SWITCH IN SEG 0
0175 3A0700             LDA     SEG0
0178 FE0F               CPI     BD0SLCT+0FH
017A CA8101             JZ      DONE    ;BOARD ALREADY ENABLED
017D 3E8F               MVI     A,NONSLCT+0FH
017F D36F               OUT     PORTID+0FH      ;ENABLE BOARD
0181 C9         DONE:   RET     ;ADDITIONAL USER'S INIT MAY GO HERE
```

WARRANTY

Macrotech warrants its products to be free from defects in materials and/or workmanship for a period of one (1) year. In the event of malfunction or other indication of failure attributable directly to faulty workmanship and/or material, then, upon return of the product (postage paid) to MACROTECH International Corporation at 22133 Cohasset Street, Canoga Park, Ca 91303 "Attention Warranty Claims Department", Macrotech will, at its option, repair or replace the defective part or parts to repair said product to proper operating condition. All such repairs and/or replacements shall be rendered by Macrotech without charge for parts or labor when the when the product is returned within the specified period of the date of purchase. This warranty applies only to the original purchaser.

This warranty will not cover the failure of Macrotech products which at the discretion of Macrotech, shall have resulted from accident, abuse, negligence, alteration, or misapplication of the product. While every effort has been made to provide clear and accurate technical information on the application of Macrotech products, Macrotech assumes no liability in any events which may arise from the use of said technical information.

This warranty is in lieu of all other warranties, expressed or implied, including warranties of mercantability and fitness for use. In no event will Macrotech be liable for incidental and consequential damages arising from or in any way connected with the use of its products. Some states do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

IMPORTANT: Please include name, address, telephone number, and description of failure with any returned product. Failure to do so may result in lost product. Proof of purchase necessary for products returned for repair under warranty. Before returning any product please call our customer service department for a return authorization number.

MACROTECH International Corporation
22133 Cohasset street
Canoga Park, California 91303
(213) 887-5737