WANG LABORATORIES, INC.

MEMORANDUM

TO: Dr. Wang, R. Kolk, S. Gagliano, E. Demeo, P. Ricker, C. Glueck, B. Potticha, R. Soucy, M. Saulenas, J. Zidek, R. Bozeman, P. Barthakur, L. Ellis, P. Talley

FROM: B. Patterson

DATE: March 17, 1978

SUBJECT: MVP BASIC-2 RELEASE 1.1

The following documentation describes Release 1.1 of MVP BASIC-2. A Release 1.1 system platter and the enclosed description of Release 1.1 should be distributed to all existing MVP customers; distribution is controlled by Sam Gagliano. New customers need not receive the description of the new release since pertinent information will be provided with the reference manual (refer any questions to Marty Saulenas, Technical Writing Department). Vendor, analyst, marketing, and SWAP literature should mention the availability of the new release.

MVP BASIC-2 RELEASE 1.1

Release 1.1 of MVP BASIC-2 replaces all previous BASIC-2 releases of MVP BASIC-2. Release 1.1 is required for the proper operation of Wang MVP telecommunication packages and the 2209A mag tape drive. Listed below are the system changes since Release 1.0. Only 2 files on the system platter differ from Release 1.0:

"@@" — MVP OS & BASIC-2 interpreter
"@GENPART" — Partition generation utility

I. System Enhancements

A. KEYIN and $IF ON when used with terminal keyboard and CRT have been defined to operate as described below:

1. KEYIN to keyboard (/001).

KEYIN has 2 forms:

(1) Check for a character. Continue execution at the statement following KEYIN if there is no character to be received.
(2) Wait for a character (i.e., suspend execution until a character is received.

Form (1) is specified by including 2 line numbers in the KEYIN statement. If form (1) is executed while the terminal is detached from the partition executing KEYIN, execution continues at the statement following KEYIN (i.e., character not received). Form (2) will wait until the terminal is attached and a keystroke is received.

2. $IF ON to CRT (/005)

$IF ON to /005 senses the following status:

READY if the terminal is attached to the partition executing $IF ON.

BUSY if the terminal is detached.

3. $IF ON to keyboard (/001)

$IF ON to /001 senses the following status:

READY if the terminal is attached to the partition executing $IF ON and a key has been pressed.

BUSY if the terminal is detached or the terminal is attached but no key has been pressed.

$IF ON to a device opened by another partition.

B. A $GIO status request strobe issued to the terminal CRT (/005) has been defined to return the CRT size and the system type. Bit 10 of the return code is 1 if and only if the CRT size is 24 x 80. Bit 20 is 1 if and only if the system is an MVP. The remaining bits are as yet undefined.

Example: $GIO /005 (7601, G$): G$ = G$ AND HEX(30)

Returns status to the first byte of G$; HEX(30) is received if the system is an MVP and the CRT size is 24 x 80.

C. The execution scheduling logic has been improved as follows:

1. Disk operations have been given higher priority making multi-sector operations, such as LOAD, SAVE, COPY AND MOVE much faster. Previously, multi-sector disk operations slowed down if other partitions were being processed.

2. The MVP gives I/O activity priority over non-I/O program execution. In order to prevent I/O lockout of program execution, I/O level checking is periodically skipped, enabling program execution to be performed.

3. Execution of long statements is segmented by the MVP OS, allowing other partitions to be serviced between execution segments. This prevents a partition from monopolizing the CPU by executing long statements. Release 1.1 breaks up certain long statements not previously segmented.

D. The device contention handling logic has been improved.

1. Each BASIC I/O statement hogs the referenced device for the duration of the statement. Thus, for example, while one partition is listing a program to the printer, other partitions are inhibited from printing until the LIST is complete. Previously, an addressed device was hogged only if $OPEN had been executed.

2. ERR #48 results if a program attempts to use a device which has been designated as exclusive for another partition. Previously, the program would have hung waiting for the device, even though the device could never be assigned to the waiting partition.

3. $IF ON always senses device busy while the specified device is opened for another partition.

4. If a partition is waiting for an I/O device to become available when the current I/O operation for that device completes, the device will be assigned to the waiting partition (unless the device has been hogged with $OPEN). Previously, it was possible for a partition to lockout a waiting partition by starting another I/O operation soon after the last I/O completion.

E. If all partitions assigned to a particular terminal release the terminal, the operator loses control of the programs being executed until at least one of the partitions addresses the terminal. In particular, the keyboard is active only when attached to a partition. In order to prevent this lockout situation the RESET and HALT keys will always remain active. If no partition is attached to the terminal when HALT or RESET is pressed, the lowest numbered partition assigned to the terminal will be halted or reset. Therefore, it is good practice to setup a small control partition as the lowest partition if no foreground job is to be run.

II. Corrected Anomalies

A. CONTINUE generated ERR A08 in "disabled programming" mode.

B. Immediate mode statement or command errors occuring after global text has been halted, cause a '@' to be displayed before the line in error.

C. The LOAD DA statement (overlay) requires the variable that receives the next available sector address to be a common variable. ERR 54 (common variable required) was occuring even if the receiver variable was common.

D. During "partition generation" a terminal is attached to the first partition that attempts to use it. Initially, the terminal should be attached to the lowest numbered partition assigned to the terminal so that the system starts up in a predictable state. This is particularly important when programs that are automatically loaded into more than one partition assigned to a terminal must be coordinated.

WANG LABORATORIES, INC.

~~WORD PROCESSING SYSTEMS~~ Computer

MARKETING RELEASE

TO North American and International Sales Organizations. District Analysts	**PUBLICATION #** SPR 3
FROM J. G. Cocks	**DATE** June 1978
SUBJECT MVP Operating System Version 1.3	**DISTRIBUTION**
SRM FILE # SECT SUBJ	**REORDER FROM:**
THIS RELEASE SUPERSEDES:	**DESTROY SUPERSEDED INFORMATION** ☐ YES ☐ NO

Wang announces a new version of the MVP operating system software, version 1.3. Whilst the major thrust for this release was the incorporation of $GIO support to the disk to allow incorporation of the new 2260C disk software format utility, the opportunity was taken to respond to outside input and requests, and also to incorporate other enhancements in the MVP control logic. We regard these enhancements as extremely significant and of considerable assistance in marketing and implementing the MVP. Attached you will find explicit details of the far reaching changes produced by our MVP Development Group. The major changes are itemized below:

Features

- A $BREAK! instruction has been included to permanently put a partition to "sleep", avoiding the request for the controlling terminal as in STOP.
- The new $PSTAT function provides extensive information for the calling partition, including user messages, OS type(MVP or VP) and release #, size, etc.
- Users may now reconfigure their MVP without having to power their CPU down.
- The $GIO instructions have been extended to the disk allowing hog and release to disk multiplexer systems and formatting for the 2260C.
- A clean up of known anomalies.

Restrictions

- All MVP systems should be upgraded to OS 1.3.

7/7/78

XC - Peter - Bill - Randy - Fred - Kevin - Pat

This is the kind of Marketing release I'm looking for to accompany our VS Software release - including the attached technical documentation.

13-427
9-74-1C

WANG LABORATORIES, INC.

Computers

MARKETING RELEASE

TO North American and International Sales Organizations. District Analysts	PUBLICATION # SPR 3
FROM J. G. Cocks	DATE June 1978
SUBJECT MVP Operating System Version 1.3	DISTRIBUTION
SRM FILE # SECT SUBJ	REORDER FROM:
THIS RELEASE SUPERSEDES:	DESTROY SUPERSEDED INFORMATION ☐ YES ☐ NO

Order Information

Order through Software Distribution Center (Paula Welch) Package number 195-0049-3 consisting of one Shugart disk number 701-2294-C.

Availability

In stock, allow one week for delivery.

Price

No charge to MVP owners.

Support

This is a Type 1, Wang Supported product. Any suspected bugs found in this package should be documented and sent to Wang Laboratories via the local district analyst.

N.B. Version 1.2 was used internally only and is superceded by version 1.3.

MVP BASIC-2 RELEASE 1.3

Release 1.3 of 2200MVP BASIC-2 replaces all previous MVP BASIC-2 releases. Release 1.3 is required for formatting disk platters on the 2260C. Listed below are the system changes since Release 1.1. The following files on the system platter differ from Release 1.1:

"@@" --MVP OS & BASIC-2 interpreter
"@GENPART" --Partition generation utility
"@PSTAT" --Partition status display utility (new)
"@MOVE" --Move system files utility (new). This program moves the specified system files to a user specified platter.

I. System Enhancements

1. $PSTAT

The new partition status function, $PSTAT, returns the current status of the specified partition. The status information includes a user defined status message, operating system type (VP or MVP), operating system release number, partition size, terminal number, global name, ERR function value, and I/O device currently in use. See the description of $PSTAT on the following pages for a complete description of the status parameters.

$PSTAT should be used if a program need distinguish between the VP or MVP operating system. A $GIO status request strobe to the terminal CRT (/005) in order to determine the system type (as described in Release 1.1) is no longer supported; this method does not work properly with VP's containing MXD's.

2. LIST DT

LIST DT has been extended to display the Master Device Table following the partition's current device selections. See the following pages for a description of the display format.

3. $BREAK!

A new parameter, '!', has been added to the $BREAK statement. $BREAK! puts the partition "to sleep" permanently; the CPU skips processing this partition. $BREAK! can be used to end the execution of a global partition without requesting the terminal. Thus subsequent execution of 'RELEASE TERMINAL' will not attach the terminal to the global partition.

If it becomes necessary to attach the terminal to a permanently put-to-sleep partition, 'RELEASE TERMINAL TO' the partition can be executed to attach the terminal. RESET must be pressed in order to 'wake up' the partition.

4. System Reconfiguration

$INIT has been modified to allow system reconfiguration without powering the system off and then back on. See the description of $INIT on the following pages.

5. LOAD RUN

The default platter specification for LOAD RUN has been changed from 'F' to 'T'; that is, the platter to load from is specified by the device type in the #0 slot of the device table. For example,

```
:SELECT DISK B10
:LOAD RUN
```

loads and executes the program "START" from the removable platter of the 10 Mb drive.

6. DEFFN @PART " "

Executing a DEFFN @PART statement with the name equal to all spaces declares that the partition is NOT global. Other partitions cannot select (via SELECT @PART) this partition for global operations.

7. SELECT @PART " "

Executing a SELECT @PART statement with the name equal to all spaces selects the originating partition (i.e., #PART) for global operations. The originating partition need not be defined to be global. SELECT @PART " " is useful when a partition needs to refer to its own global variables or when global text needs to call subroutines in the calling partition.

8. $GIO to a Disk

$GIO is now legal for certain disk operations. In particular, $GIO can be used to hog or unhog the disk on those systems with disk multiplexers:

```
$GIO /310 (4580) -- hog disk /310
$GIO /310 (4500) -- release disk /310
```

$GIO hog and unhog have been added for 2200T and 2200VP compatibility. They are equivalent to $OPEN /310 and $CLOSE /310. $GIO is also legal for formatting 2260C disk platters.

9. New $GIO Microcommands

The $GIO command repertoire has been expanded to include:

C3h_3h_4 — same as C6h_3h_4 except WR before CPB
81h_3h_4 — same as 80h_3h_4 except WR before CPB
83h_3h_4 — same as 82h_3h_4 except WR before CPB
870 0 — same as 860 0 except WR before CPB
870 h_4 — same as 860 h_4 except WR before CPB
89h_3h_4 — same as 88h_3h_4 except WR before CPB
8Bh_3h_4 — same as 8Ah_3h_4 except WR before CPB

10. ERR P48

Error #48 (Illegal Device Address) is now recoverable using the ERROR statement.

11. RESET Disks

When RESET is pressed on any terminal, all disks not currently in use are reset. In particular, RESET now extinguishes the error light on a Diablo disk after power on. RESET can also be used to reset a floppy disk after a formatting error.

II. Corrected Anomalies

1. The error message indicating that the 30 ms. timer, used by the operating system for determining when to switch execution to another partition, is malfunctioning has been changed from 'VECM 0000' to 'TICK '.

2. ERR S13 resulted from program resolution if a global program contained a MAT statement, occurring after the DEFFN @PART statement, explicitly redimensioning a non-global one dimensional array. The array should have been ignored since it occurred after the DEFFN @PART statement. The error could occur in any of the following statements: MAT REDIM, MAT READ, MAT INPUT, MAT CON, MAT ZER, or MAT IDN.

3. LIST S' listed only the first page of output.

4. Selecting a non-existent MXD for print operations could cause the system to hang up when print was executed. Power off and on was required to restore system operation. Now ERR 48 is returned if the MXD does not exist.

5. If a memory overflow error (ERR A01) occurred during program loading, the disk might not be released for use by other partitions.

6. On MVP systems with disk multiplexers, disks can be hogged in 1 of 2 modes: by 80-bit of disk address set to 1 or by $OPEN. If a disk unit is hogged in one mode, hogging another disk in the other mode could change the hog mode of the first unit. This conflict has been resolved.

7. LIST T occasionally found erroneous matches following HEX literals.

General Form (as a statement)

$PSTAT = alpha-expression

where: alpha-expression is the same as for LET

General Form (as a function within an alpha-expression):

$PSTAT (expression)

where: 1<= value of expression <= number of partitions

Purpose:

The $PSTAT function returns an alphanumeric string describing the current status of the partition specified by the value of the expression. The $PSTAT function can only be used within alpha-expression (on the righthand side of assignment statements). The following information is returned by the $PSTAT function:

bytes 1-8 = user specified status

The user status is set by the $PSTAT statement.

byte 9 = OS type('M' if MVP, 'V' if VP)

byte 10 = OS release number

The release number is stored in packed decimal; thus, HEX(1.3) is release 1.3.

byte 11 = unused

bytes 12-13 = partition size (XX,YY K).

A packed decimal value indicating the partition size. Byte 1 is XX; byte 2 is YY.

byte 14 = programmability

A 'P' is returned if the partition is programmable; if programming has been disabled a 'space' is returned.

byte 15 = terminal number

A packed decimal value indicating which terminal is assigned to the partition.

byte 16 = terminal status

'A' if terminal attached.
'D' if terminal detached (via $RELEASE TERMINAL) and the partition has not requested the terminal.
'W' if the partition is waiting for the terminal to be attached.

bytes 17-24 = global name

If the partition has not declared itself to be global (by means of DEFFN @PART), the global name is all spaces.

byte 25 = ERR function value

byte 26 = text pointer

A packed decimal value indicating the number of the partition containing the program text currently being executed. This is the same as #PART except when global text is being executed.

byte 27 = global pointer

A packed decimal value indicating the number of the partition selected for global operations (via SELECT @PART).

byte 28 = DATA pointer

A packed decimal value indicating the partition containing the DATA statements that READ currently points at. This is the same as #PART unless a RESTORE statement has been executed in global text.

byte 29 = device address

Address of the device the partition is currently communicating with or waiting for.

If the partition specified in the $PSTAT function does not exist in the current configuration ERR X77 results.

The $PSTAT statement sets the user specified portion of the partition status to the value of the alpha-expression. The first 8 bytes of the partition status are user settable.

Examples of valid syntax:

```
10 A$ = $PSTAT (3)
20 B$ = $PSTAT (#PART)
30 C$() = $PSTAT (1) & $PSTAT (2) & $PSTAT (3)
40 $PSTAT = Q$
50 $PSTAT = "OP#12345"
```

The following utility displays the status of each partition in the current configuration. Non-printable characters (i.e., HEX(00) - HEX(0F)) in the user status and global name fields are displayed as periods (.).

```
@PSTAT

0010 REM %

@PSTAT -- DISLAY STATUS OF EACH PARTITION IN CURRENT CONFIGURATION


   : REM 6/6/78
0020 REM %

VARIABLES.....................................................

   : DIM P$29,P1$2,P2$80
   : P2$=ALL("-")
0030 REM %

DISPLAY TITLE.................................................

   : PRINT HEX(0D0306);TAB(18);"***** CURRENT STATUS OF EACH PARTI
     TION *****"
   : PRINT
   : PRINT "PART#  USER-MSG   OS  BANK  SPACEK PRGM  TERM#   @NAME
         ERR  TEXT @P DATA  I/O"
   : PRINT P2$
0040 REM %

GET & DISPLAY STATUS OF EACH PARTITION........................

   : PRINT AT(4,0);
   : FOR P=1 TO 16
   : P$=$PSTAT(P)
   : ERROR P=ERR
   : P=16
   : GOTO 90

0050 CONVERT P TO P1$,(##)
0060 REM ELIMINATE UNPRINTABLE CHARACTERS
   : $TRAN(STR(P$,,8),"................")
   : $TRAN(STR(P$,17,8),"................")
0070 PRINT "-";P1$;"    ";STR(P$,,8);"   ";STR(P$,9,1);HEXOF(STR(P
     $,10,1));"   ";HEXOF(STR(P$,11,1));"   ";HEXOF(STR(P$,12,1));
     ".";HEXOF(STR(P$,13,1));"   ";
0080 PRINT STR(P$,14,1);"   ";HEXOF(STR(P$,15,1));"-";STR(P$,16,1)
     ;"   ";STR(P$,17,8);"   ";HEXOF(STR(P$,25,1));"   ";HEXOF(STR
     (P$,26,1));"  ";HEXOF(STR(P$,27,1));"  ";HEXOF(STR(P$,28,1));
     "   ";HEXOF(STR(P$,29,1));HEX(0D);" "
0090 $BREAK
   : NEXT P
   : PRINT P2$
0100 REM %

DISPLAY AGAIN? ...............................................

   : PRINT AT(22,1);"PRESS ANY KEY TO STOP DISPLAY...    ";
   : KEYIN P3$,110,110
   : GOTO 40

0110 PRINT AT(22,1);"PRESS ANY KEY TO CONTINUE DISPLAY...";
   : KEYIN P3$
   : GOTO 40
```

WANG LABORATORIES, INC.

Computers

MARKETING RELEASE

TO North American and International Sales Organizations. District Analysts.	PUBLICATION # SPR 9
FROM J. G. Cocks	DATE August 28, 1978
SUBJECT MVP Operating System 1.4	REORDER FROM:
THIS RELEASE SUPERSEDES:	DESTROY SUPERSEDED INFORMATION ☐ YES ☐ NO

We have discovered a bug in the current release of the MVP operating system, version 1.3. This bug causes problems when a 20MB daisy chained disk system, a 2260B-2 or a 2260C-2, is attached to the MVP. If two partitions issue access commands to the two disk drives simultaneously, a possible misdirection of data to the wrong partition can occur. Error I 93 is frequently reported.

The bug has been corrected in the new release, OS 1.4, however, as we will be bringing out version 1.5 in the near future, extending the operating system to cover up to 256K bytes of memory, we propose to give the 1.4 release a limited circulation. Copies will be sent to all analysts and new releases from manufacturing will include OS 1.4. Any sites with the MVP/2260B-2 or C-2 combination should be upgraded immediately and further copies may be obtained from Paula Welch, Software Distribution.

The package number for release 1.4 is 195-0049-3 consisting of one disk number 701-2294D.

MVP BASIC-2 RELEASE 1.5
(Platter 701-2294E)

Release 1.5 of 2200MVP BASIC-2 replaces all previous MVP BASIC-2 releases. Release 1.5 is required for any MVP with more than 64K of data memory. This release also provides all MVP systems with several new features and corrects all known system anomalies. Listed below are the system changes since Release 1.4. Please note that Release 1.4 had a limited distribution; it corrected a problem concerning the 2260B-2 and 2260C-2 disk drives, but otherwise was identical to Release 1.3. The following files on the system platter differ from Release 1.3 and 1.4:

"@@"	-	MVP OS & BASIC-2 interpreter.
"@GENPART"	-	Partition generation utility.
"@HELP"	-	Partition generation description (@GENPART overlay).
"@A"	-	CPU diagnostic.
"@B"	-	Control memory diagnostic.
"@C"	-	Data memory diagnostic.
"@MOVE"	-	Move system file utility.

A. System Enhancements

1. Multiple Memory Banks

Release 1.5 supports MVP's with up to 256K of data memory. The new "@GENPART" is required for configuring systems with more than 64K of data memory. Memory is divided into banks with bank 1 providing up to 61K for user partitions and banks 2, 3, and 4 providing up to 56K each for user partitions. The following definitions have been made for multi-bank MVP support:

a. SPACEK. Before partition generation, SPACEK equals the total amount of memory available for partitioning. After partition generation, SPACEK equals the partition size.

b. Global partitions. A partition in one bank may have the same global name as a partition in another bank. SELECT @PART refers only to partitions in the same memory bank or in the universal global area in bank 1 (i.e., the 1st 5K of partition space). However, $RELEASE TERMINAL TO "global-name" can refer to a partition in any bank. The system looks for the specified partition first in bank 1, then bank 2, ...; the terminal is released to the 1st partition found with the specified name that is available to this terminal.

2. Background Printing to the Terminal

The OS now supports background printing to a printer attached to the terminal. One partition can be outputting to the printer while a second partition has control of the CRT and keyboard. $RELEASE TERMINAL releases the terminal CRT and keyboard but not the terminal printer. $OPEN /204 can be used for hogging the printer.

Background printing is not recommended to terminals using communication rates lower than 1200 baud since keystroke echoing may be noticeably delayed when the printer is in operation.

In addition to Release 1.5 of the OS, Release 1.4 or later of the MXD firmware and Release 2.0 or later of the 2236D firmware are required for proper background printing operation. Background printing will appear to work with the old firmware, but data will be lost when the foreground partition is RESET.

The user can distinguish the various sets of terminal and MXD firmware by simply testing another new feature, the underline dead key, which allows underlined text to be entered: Hold down the shift key and touch the minus sign key next to zero on the typewriter pad. Next type the letter A. The screen display will be:

A	-	Release 1.4 MXD, Release 2.0 terminal.
-A	-	Release 1.0 terminal, ? MXD.
PRINT	-	Release 1.3 MXD, Release 2.0 terminal.

This test should be repeated for all terminals on the system.

3. System Diagnostics

The system diagnostics, accessible immediately after power on, have been rewritten to support the larger memory configurations and to provide better memory diagnosing with more consistent error messages and improved displays.

4. $RELEASE PART

A partition can be released from a terminal, by using the $RELEASE PART statement, so that it can be reassigned to another terminal (see the following pages for a complete description of $RELEASE PART).

5. Minimum Partition Size

The minimum partition size has been reduced from 2K to 1.25K.

6. MOVE/COPY

The MOVE and COPY disk operations now make more efficient use of memory for buffering; these operations are now up to 15% faster.

B. Corrected Anomalies

1. The $GIO microcommands CAhh, CBhh, CEhh and CFhh are illegal on the MVP. Attempting to execute these commands now generates ERR P46.

2. For certain errors the ERR function was not set until the terminal was attached to the partition with the error. Thus, $PSTAT could not always be used to check for errors in background jobs.

3. When a LIST S V or LIST S T command was executed and the listing stopped at the bottom of the screen, pressing a SF key defined for text entry or EDIT/RECALL caused the keys to be ignored, an erroneous error message to be displayed, or a dumps of parts of memory to appear on the CRT.

4. If the ERROR statement was used to recover from an error within a subroutine called from a special function key, the system lost the subroutine return information. This would result in ERR P41 when RETURN was executed.

5. The text pointer, data pointer and global partition pointer were returned in hexadecimal rather than packed decimal by $PSTAT.

6. LIST' did not work properly when executed after global text had been halted. LIST' should have listed DEFFN' references in the calling partition, but sometimes listed references in the global partition.

7. Unhogging an exclusive disk (i.e., one assigned to a particular partition by @GENPART) by CBS or address caused the disk to be made available to any partition.

8. The system did not always detect the illegal occurrence of alpha array elements where numeric variables were expected.

9. If variables were used to specify array dimensions (e.g., DIM X(R,C)) during program overlaying, erroneous memory overflow errors (ERR A02) might be reported by the system. This could only happen if the program text being overlayed was larger than that of the overlay loaded, more variables were defined in the overlay than existed in the original program, and most of memory was used.

10. A PACK statement with an exponential image could modify the values of the variables containing the data to be packed. If an exponential image in a PACK statement did not have exactly 1 integer digit specified and the data to be packed was specified by a numeric array designator (e.g., N()), the result of the pack would be correct but the exponents of the values in the numeric array would be changed.

11. Errors in program lines entered after global text was halted, caused an '@' to be displayed before the line in error. The '@' should not be displayed since the line was being entered into the calling partition.

$RELEASE PART

General Forms

$RELEASE PART

Purpose:

Each partition normally has a terminal assigned to it for program control and operator interaction. If more than one partition is assigned to a single terminal, the terminal is attached to only one partition at any given time. (See $RELEASE TERMINAL). The terminal assignment to a partition can be eliminated by executing a $RELEASE PART statement; the partition is, then, available for reassignment to a different terminal. In addition, the terminal will be attached to any partition waiting for it (in effect a $RELEASE TERMINAL is performed). Note that $RELEASE PART does not clear the partition nor terminate program execution; however, program execution would be suspended if the program attempted to communicate with a terminal.

A terminal can assign itself to a released partition by executing a $RELEASE TERMINAL TO statement to the released partition. The terminal will be released from its current partition and assigned and attached to the released partition. If more than one partition is available (i.e., released), the characteristics of those partitions (e.g., size and memory bank) can be examined with $PSTAT in order to determine which partition the terminal should be assigned to. Released partitions are identified by the terminal number (byte 15) of $PSTAT being zero.

If a terminal is assigned to only one partition and that partition is released, the terminal is no longer part of the active MVP system. However, if RESET is pressed and a partition is available (i.e., released and waiting for a terminal), the partition will be assigned and attached to that terminal.

At partition generation time (after power on), partitions can be specified to not be assigned to any terminal by entering terminal number 0 as a partition parameter.

Example of Valid Syntax:

```
:10 $RELEASE PART
```

User Memory Allocation

Memory on the 2200MVP is divided into two distinct kinds. The operating system, BASIC-2 interpreter, and other system data are contained in control memory. The 2200MVP has approximately 60K bytes of control memory, which is completely separate from memory available for user programs and data. User memory is available in increments from 16K bytes to 256K bytes. When referring to the memory size of a 2200MVP, only the user memory is considered.

User memory on the 2200MVP is divided into areas known as "banks", of which a maximum of four are allowed. A bank contains a maximum of 64K bytes of user memory; if a system containing from 16K bytes to 64K bytes of user memory is purchased, user memory is contained in only one bank. In this first bank, memory may be added in 16K byte increments. A system containing more than 64K bytes of user memory contains more than one bank, depending on the amount of memory purchased. Memory in bank two may be purchased in 32K byte increments only, while banks three and four may be purchased only as complete 64K byte banks.

General system overhead on the 2200MVP requires only 3K bytes of user memory in bank #1. In banks #2, #3, and #4, 8K bytes in each bank are unavailable for partitions (regardless of the size of that bank), leaving a maximum of 56K bytes available in each of those banks for partitions. A 256K byte MVP then provides a total of 229K bytes of user memory available for partitions. In addition to the general system overhead, each partition requires 1K bytes of housekeeping information for program control and buffering, leaving the remaining memory allocated for a partition free for programs and data. Partitions must be at least 1.25K bytes in size; they may be allocated space in 256 byte (0.25K byte) increments up to a maximum of the full extent of user memory in any one bank (exclusive of housekeeping and unavailable areas).

User memory in one bank is inaccessible to user memory in any other bank; thus, a partition may not extend from one bank to another. Figure 16-1 illustrates memory bank organization.

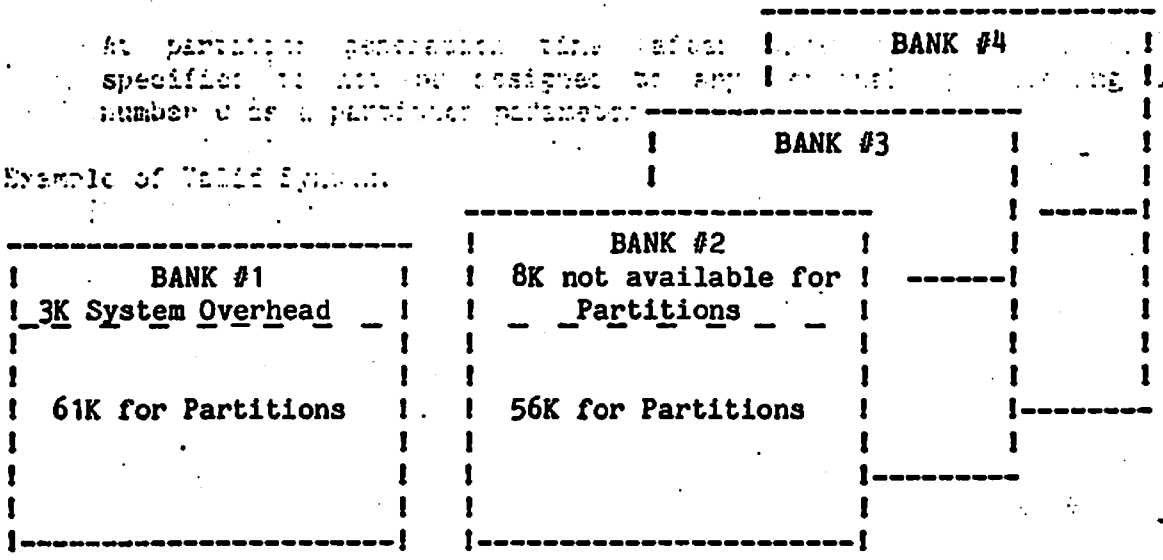


Figure 16-1.
Memory Bank Organization

A global partition is accessible only by other partitions in the bank where the global partition resides. However, the first available 5K bytes of user memory in bank #1 constitute a special area of memory known as the "universal global" area, which is illustrated in Figure 16-2 below. A global partition which is contained entirely in this area may be accessed by a partition in any bank. A universal global partition can be used to store programs and data which are to be shared by all users on the system.

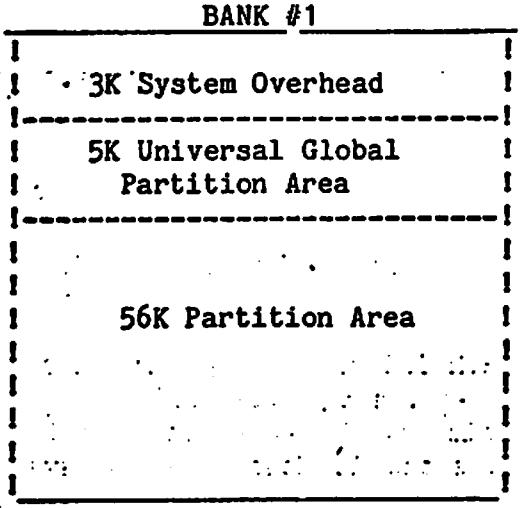
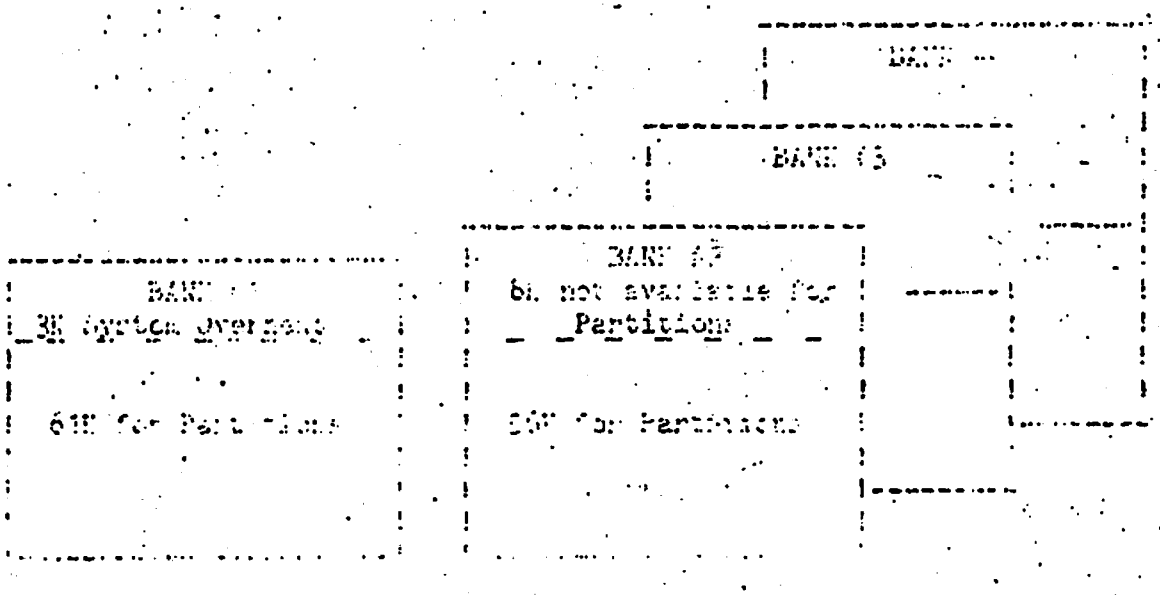


Figure 16-2
The Universal Global Area

Note that the entire universal global area need not be used for universal global partitions; the only restriction is that a universal global partition reside entirely within that area. For all other purposes, the entire area is treated exactly as all other memory in bank #1.



Consider the multibank system configuration in Figure 16-3 below:

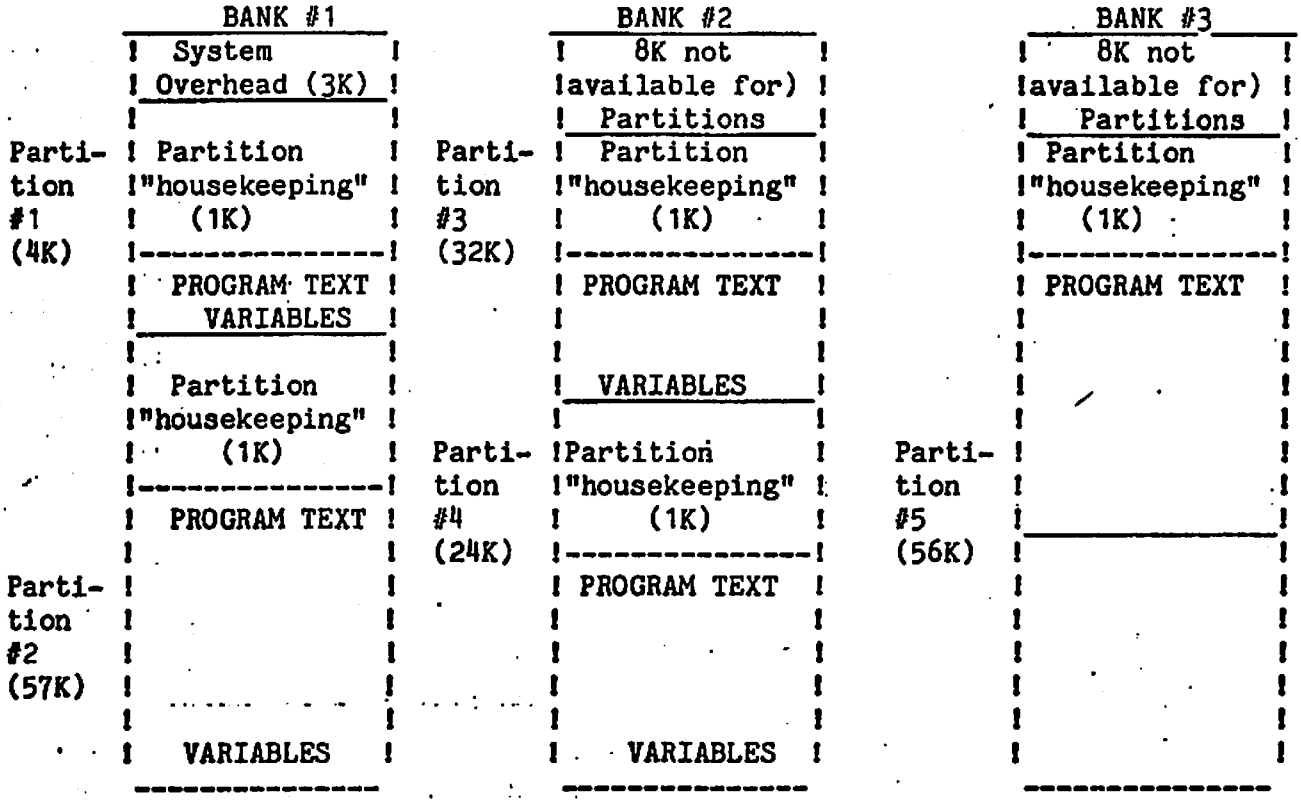


Figure 16-3
A Multibank System Configuration

If partition #3 were defined as global, it could be accessed only by partition #4 since that is the only other partition in bank #2. However, if partition #1 were defined to be global, it could be accessed by any of the other partitions since partition #1 resides entirely within the universal global area.

The END statement, SPACE and SPACEK functions apply to the current partition. For example, SPACEK returns the partition size (e.g., 32K for partition #3 above) rather than the total amount of memory in the system. However, before memory has been partitioned, SPACEK returns the total amount of memory available for partitioning in all banks.

The system configuration utility, "@GENPART", has been modified to configure MVP systems with more than 64K of user memory. When entering a new configuration, "@GENPART" displays the size of each memory bank as follows:

```
AVAILABLE USER MEMORY = 61.00K   56.00K   56.00K   56.00K
REMAINING USER MEMORY = 61.00K   56.00K   56.00K   56.00K
```

The sizes of banks 1, 2, 3 and 4 are displayed in that order. "@GENPART" requests the number of partitions in each bank and the number of terminals on the system, and then displays the partition attributes for editing. A dotted line is displayed to separate partitions in different memory banks. If a partition can be universally global (i.e., it entirely resides within the 1st 5K of partition space in bank 1), a 'u' is displayed next to the partition number.

WANG LABORATORIES INC.

MEMORANDUM

TO: Technical Information Center, G. Reynolds, J. Zidek, P. Talley
L. Ellis, D. Gowell, J. Proulx

FROM: B. Patterson

DATE: February 7, 1979

SUBJECT: ANOMALIES OF MVP RELEASE 1.5

The following error was introduced into the MVP BASIC-2 interpreter with Release 1.5

ONE INDUSTRIAL AVENUE, LOWELL, MASSACHUSETTS 01851 • TEL. (617) 851-4111 • TWX 710-343-6769 • TELEX 94-7421

MVP MCN 1.5/1

Date Found: 2/6/79
Release: 1.5
Found by: Vendor

Date Corrected: 2/6/79
Release: 1.6

Description: PACK/MULTI-USER

Execution of a PACK statement while another partition is executing a program, may cause the PACK statement to execute improperly. Usually, this will result in an error X71. This bug was introducted with MVP Release 1.5 when the "Exponential PACK" error was corrected.

How to Generate Error:

Execute the following program in 2 partitions:

```
:10 DIM X(4)
:20 PACK (-####)A$ FROM X()
:30 PRINT HEXOF(A$): GOTO 20
```

Error X71 may be produced.

Correction:

The breakpoint logic in the PACK routine was corrected to save AUX5 as well as other critical registers.

DEFFN' 15: $RELEASE TERMINAL TO PART #

$RELEASE TERMINAL TO PARTITION # - ENABLES YOU TO LOOK AT DIFFERENT PARTITIONS ATTACHED TO ONE TERMINAL

@ REFERS TO ERROR IN SECONDARY PARTITION, (GLOBAL OR BACKGROUND)

FOREGROUND

```
10 SELECT @ PART "START"
15 X=0
20 X=X+1
30 GOSUB '100
40 X=X+1
50 PRINT X
```

SAYS THERE IS A BACKGROUND PARTITION. WILL LOOK IN THIS PARTITION 1ST FOR '100 THEN IN BACKGROUND

BACKGROUND

```
10 DEFFN' @ PART "START"
20 STOP
30 DEFFN'100
40 X=X+1
50 RETURN
```

MUST RUN DEFFN' SO COMPUTER KNOWS IT IS IN THERE

IF WANT TO GO TO THIRD PARTITION MUST GIVE IT A NAME, SELECT @ PART "START1", & PUT IN PARTITION PREVIOUS TO JUMPING TO NEW PARTITION BUT AFTER JUMPING TO PARTITION ALREADY SELECTED BY PREVIOUS SELECT @ PART " "

$OPEN/215 HOG MOD FOR PERIPHERAL
$CLOSE/215 RELEASE HOG MODE

$OPEN WNG NUMBER,/215 IF PRINTER HOGGED GOTO LINE NUMBER

WANG LABORATORIES, INC.

Computers

Tina

MARKETING RELEASE

TO North American and International Sales Organizations; District Analysts	PUBLICATION # SPR 11
FROM Al Breveleri	DATE February 1979
SUBJECT MVP Operating System Release 1.6	REORDER FROM:
THIS RELEASE SUPERSEDES:	DESTROY SUPERSEDED INFORMATION ☐ YES ☐ NO

Wang announces version 1.6 of the 2200 MVP operating system. This version supports expanded memory, adds several new capabilities, and corrects all known firmware errors. Release 1.6 replaces all previous releases and is required for any MVP with more than 64k of user memory.

Features

- Expanded (multiple bank) memory
- Background output to terminal printer
- Release partition from terminal
- New minimum partition size

Restrictions

- All MVP systems should be upgraded to OS 1.6.

Ordering Information

Order through Software Distribution Center, package number 195-0049-3 consisting of one Shugart diskette number 701-2294-G.

Availability

In stock, allow one week for delivery.

Price

No charge to 2200 MVP owners.

Support

This is a category 1, Wang supported product. Any suspected errors or anomalies found in this package should be documented and sent to Wang Laboratories via the local district analyst.

MVP BASIC-2 RELEASE 1.6
(Platter 701-2294G)

Release 1.6 of 2200MVP BASIC-2 replaces all previous MVP BASIC-2 releases. Release 1.6 is required for any MVP with more than 64K of data memory. This release also provides all MVP systems with several new features and corrects all known system anomalies. Listed below are the system changes since Release 1.5. Please note that Release 1.4 and Release 1.5 both had limited distribution. Release 1.4 corrected a problem concerning the 2260B-2 and 2260C-2 disk drives, but otherwise was identical to Release 1.3. Release 1.5 contained an error in the execution of a PACK statement, and the "@GENPART" utility provided was too large to execute in a 16K MVP; Release 1.6 corrects these anomalies but is otherwise identical to 1.5. The following files on the system platter differ from Release 1.3 and 1.5:

"@@"	-	MVP OS & BASIC-2 interpreter.
"@GENPART"	-	Partition generation utility.
"@HELP"	-	Partition generation description (@GENPART overlay).
"@A"	-	CPU diagnostic.
"@B"	-	Control memory diagnostic.
"@C"	-	Data memory diagnostic.
"@MOVE"	-	Move system file utility.

A. System Enhancements

1. Multiple Memory Banks

Release 1.6 supports MVP's with up to 256K of data memory. The new "@GENPART" is required for configuring systems with more than 64K of data memory. Memory is divided into banks with bank 1 providing up to 61K for user partitions and banks 2, 3, and 4 providing up to 56K each for user partitions. The following definitions have been made for multi-bank MVP support:

a. SPACEK. Before partition generation, SPACEK equals the total amount of memory available for partitioning. After partition generation, SPACEK equals the partition size.

b. Global partitions. A partition in one bank may have the same global name as a partition in another bank. SELECT @PART refers only to partitions in the same memory bank or in the universal global area in bank 1 (i.e., the 1st 5K of partition space). However, $RELEASE TERMINAL TO "global name" can refer to a partition in any bank. The system looks for the specified partition first in bank 1, then bank 2, ...; the terminal is released to the 1st partition found with the specified name that is available to this terminal.

2. Background Printing to the Terminal

The OS now supports background printing to a printer attached to the terminal. One partition can be outputting to the printer while a second partition has control of the CRT and keyboard. $RELEASE TERMINAL releases the terminal CRT and keyboard but not the terminal printer. $OPEN /204 can be used for hogging the printer.

Background printing is not recommended to terminals using communication rates lower than 1200 baud since keystroke echoing may be noticeably delayed when the printer is in operation.

In addition to Release 1.6 of the OS, Release 1.4 or later of the MXD firmware and Release 2.0 or later of the 2236D firmware are required for proper background printing operation. Background printing will appear to work with the old firmware, but data will be lost when the foreground partition is RESET.

The user can distinguish the various sets of terminal and MXD firmware by simply testing another new feature, the underline dead key, which allows underlined text to be entered: Hold down the shift key and touch the minus sign key next to zero on the typewriter pad. Next type the letter A. The screen display will be:

why?

A	-	Release 1.4 MXD, Release 2.0 terminal.
-A	-	Release 1.0 terminal, ? MXD.
PRINT	-	Release 1.3 MXD, Release 2.0 terminal.

This test should be repeated for all terminals on the system.

3. System Diagnostics

The system diagnostics, accessible immediately after power on, have been rewritten to support the larger memory configurations and to provide better memory diagnosing with more consistent error messages and improved displays.

4. $RELEASE PART

A partition can be released from a terminal, by using the $RELEASE PART statement, so that it can be reassigned to another terminal (see the following pages for a complete description of $RELEASE PART).

5. Minimum Partition Size

The minimum partition size has been reduced from 2K to 1.25K.

6. MOVE/COPY

The MOVE and COPY disk operations now make more efficient use of memory for buffering; these operations are now up to 15% faster.

B. Corrected Anomalies

1. The $GIO microcommands CAhh, CBhh, CEhh and CFhh are illegal on the MVP. Attempting to execute these commands now generates ERR P46.

2. For certain errors the ERR function was not set until the terminal was attached to the partition with the error. Thus, $PSTAT could not always be used to check for errors in background jobs.

3. When a LIST S V or LIST S T command was executed and the listing stopped at the bottom of the screen, pressing a SF key defined for text entry or EDIT/RECALL caused the keys to be ignored, an erroneous error message to be displayed, or a dumps of parts of memory to appear on the CRT.

4. If the ERROR statement was used to recover from an error within a subroutine called from a special function key, the system lost the subroutine return information. This would result in ERR P41 when RETURN was executed.

5. The text pointer, data pointer and global partition pointer were returned in hexadecimal rather than packed decimal by $PSTAT.

6. LIST' did not work properly when executed after global text had been halted. LIST' should have listed DEFFN' references in the calling partition, but sometimes listed references in the global partition.

7. Unhogging an exclusive disk (i.e., one assigned to a particular partition by @GENPART) by CBS or address caused the disk to be made available to any partition.

8. The system did not always detect the illegal occurrence of alpha array elements where numeric variables were expected.

9. If variables were used to specify array dimensions (e.g., DIM X(R,C)) during program overlaying, erroneous memory overflow errors (ERR A02) might be reported by the system. This could only happen if the program text being overlayed was larger than that of the overlay loaded, more variables were defined in the overlay than existed in the original program, and most of memory was used.

10. A PACK statement with an exponential image could modify the values of the variables containing the data to be packed. If an exponential image in a PACK statement did not have exactly 1 integer digit specified and the data to be packed was specified by a numeric array designator (e.g., N()), the result of the pack would be correct but the exponents of the values in the numeric array would be changed.

11. Errors in program lines entered after global text was halted, caused an '@' to be displayed before the line in error. The '@' should not be displayed since the line was being entered into the calling partition.

$RELEASE PART

General Forms

$RELEASE PART

Purpose:

Each partition normally has a terminal assigned to it for program control and operator interaction. If more than one partition is assigned to a single terminal, the terminal is attached to only one partition at any given time. (See $RELEASE TERMINAL). The terminal assignment to a partition can be eliminated by executing a $RELEASE PART statement; the partition is, then, available for reassignment to a different terminal. In addition, the terminal will be attached to any partition waiting for it (in effect a $RELEASE TERMINAL is performed). Note that $RELEASE PART does not clear the partition nor terminate program execution; however, program execution would be suspended if the program attempted to communicate with a terminal.

A terminal can assign itself to a released partition by executing a $RELEASE TERMINAL TO statement to the released partition. The terminal will be released from its current partition and assigned and attached to the released partition. If more than one partition is available (i.e., released), the characteristics of those partitions (e.g., size and memory bank) can be examined with $PSTAT in order to determine which partition the terminal should be assigned to. Released partitions are identified by the terminal number (byte 15) of $PSTAT being zero.

If a terminal is assigned to only one partition and that partition is released, the terminal is no longer part of the active MVP system. However, if RESET is pressed and a partition is available (i.e., released and waiting for a terminal), the partition will be assigned and attached to that terminal.

At partition generation time (after power on), partitions can be specified to not be assigned to any terminal by entering terminal number 0 as a partition parameter.

Example of Valid Syntax:

```
:10 $RELEASE PART
```

User Memory Allocation

Memory on the 2200MVP is divided into two distinct kinds. The operating system, BASIC-2 interpreter, and other system data are contained in control memory. The 2200MVP has approximately 60K bytes of control memory, which is completely separate from memory available for user programs and data. User memory is available in increments from 16K bytes to 256K bytes. When referring to the memory size of a 2200MVP, only the user memory is considered.

User memory on the 2200MVP is divided into areas known as "banks", of which a maximum of four are allowed. A bank contains a maximum of 64K bytes of user memory; if a system containing from 16K bytes to 64K bytes of user memory is purchased, user memory is contained in only one bank. In this first bank, memory may be added in 16K byte increments. A system containing more than 64K bytes of user memory contains more than one bank, depending on the amount of memory purchased. Memory in bank two may be purchased in 32K byte increments only, while banks three and four may be purchased only as complete 64K byte banks.

General system overhead on the 2200MVP requires only 3K bytes of user memory in bank #1. In banks #2, #3, and #4, 8K bytes in each bank are unavailable for partitions (regardless of the size of that bank), leaving a maximum of 56K bytes available in each of those banks for partitions. A 256K byte MVP then provides a total of 229K bytes of user memory available for partitions. In addition to the general system overhead, each partition requires 1K bytes of housekeeping information for program control and buffering, leaving the remaining memory allocated for a partition free for programs and data. Partitions must be at least 1.25K bytes in size; they may be allocated space in 256 byte (0.25K byte) increments up to a maximum of the full extent of user memory in any one bank (exclusive of housekeeping and unavailable areas).

User memory in one bank is inaccessible to user memory in any other bank; thus, a partition may not extend from one bank to another. Figure 16-1 illustrates memory bank organization.

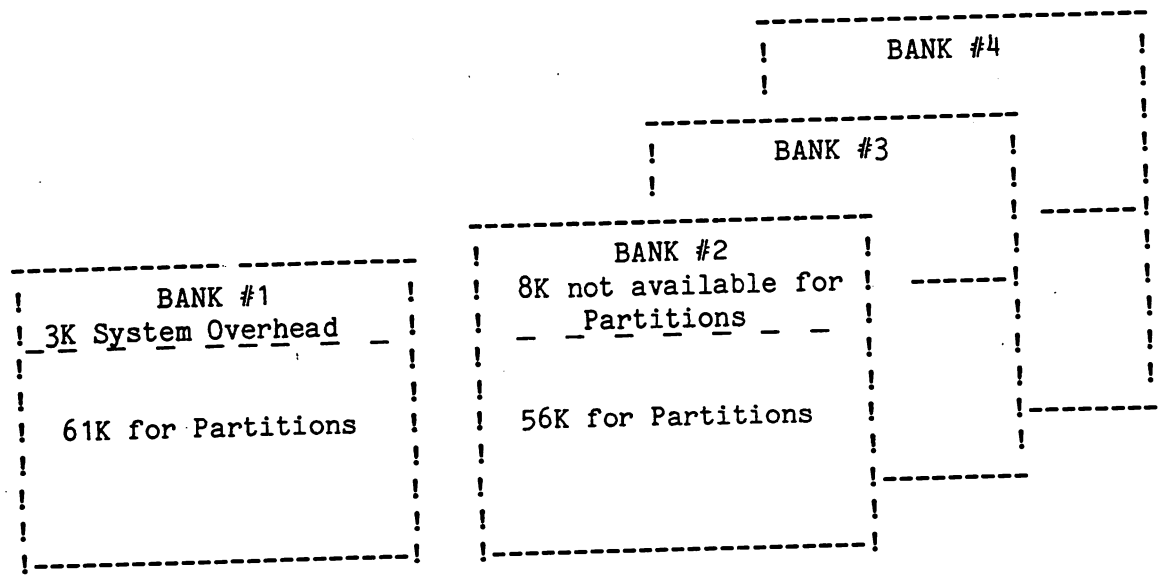


Figure 16-1.
Memory Bank Organization

A global partition is accessible only by other partitions in the bank where the global partition resides. However, the first available 5K bytes of user memory in bank #1 constitute a special area of memory known as the "universal global" area, which is illustrated in Figure 16-2 below. A global partition which is contained entirely in this area may be accessed by a partition in any bank. A universal global partition can be used to store programs and data which are to be shared by all users on the system.

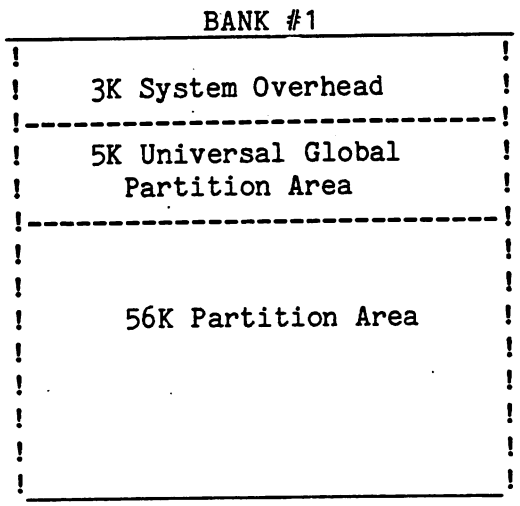


Figure 16-2
The Universal Global Area

Note that the entire universal global area need not be used for universal global partitions; the only restriction is that a universal global partition reside entirely within that area. For all other purposes, the entire area is treated exactly as all other memory in bank #1.

Consider the multibank system configuration in Figure 16-3 below:

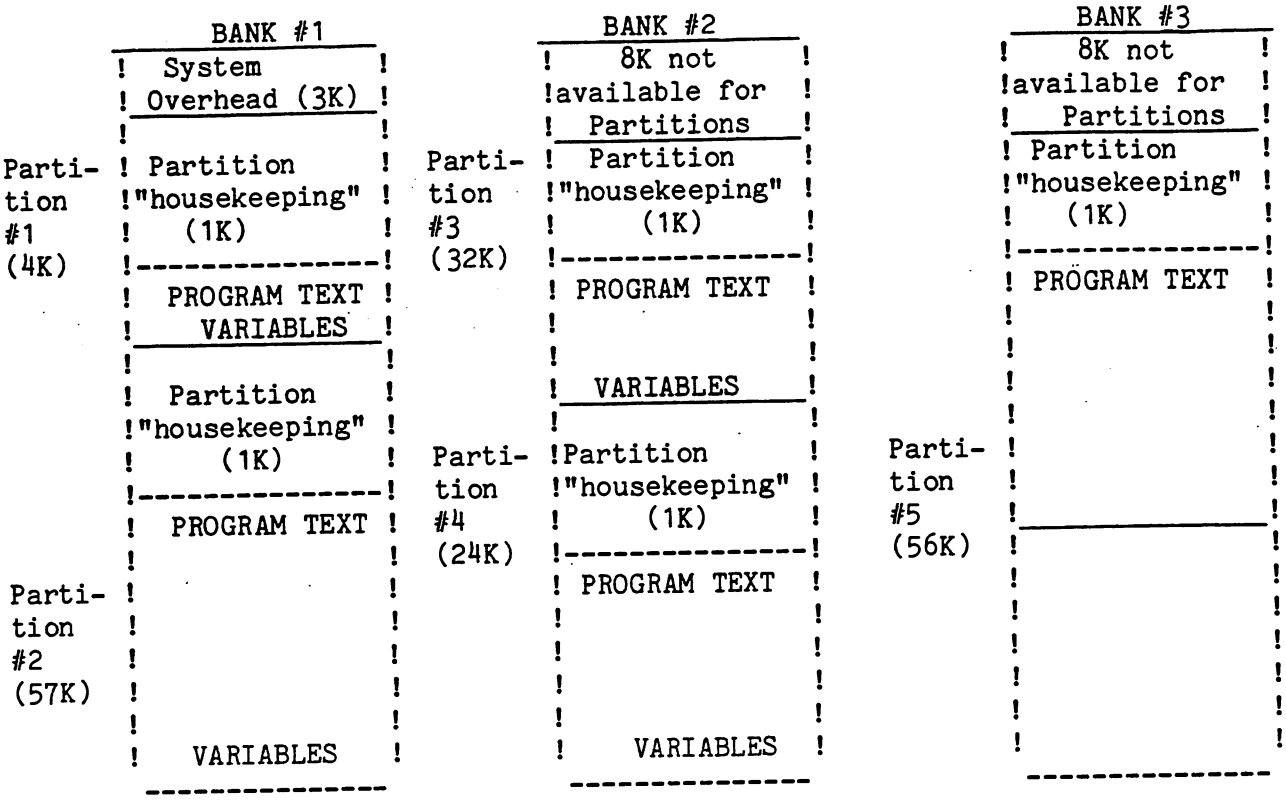


Figure 16-3
A Multibank System Configuration

If partition #3 were defined as global, it could be accessed only by partition #4 since that is the only other partition in bank #2. However, if partition #1 were defined to be global, it could be accessed by any of the other partitions since partition #1 resides entirely within the universal global area.

The END statement, SPACE and SPACEK functions apply to the current partition. For example, SPACEK returns the partition size (e.g., 32K for partition #3 above) rather than the total amount of memory in the system. However, before memory has been partitioned, SPACEK returns the total amount of memory available for partitioning in all banks.

The system configuration utility, "@GENPART", has been modified to configure MVP systems with more than 64K of user memory. When entering a new configuration, "@GENPART" displays the size of each memory bank as follows:

```
AVAILABLE USER MEMORY = 61.00K   56.00K   56.00K   56.00K
REMAINING USER MEMORY = 61.00K   56.00K   56.00K   56.00K
```

The sizes of banks 1, 2, 3 and 4 are displayed in that order. "@GENPART" requests the number of partitions in each bank and the number of terminals on the system, and then displays the partition attributes for editing. A dotted line is displayed to separate partitions in different memory banks. If a partition can be universally global (i.e., it entirely resides within the 1st 5K of partition space in bank 1), a 'u' is displayed next to the partition number.

WANG LABORATORIES, INC.

Computers

MARKETING RELEASE

TO North American and International Sales Organizations; District Analysts	PUBLICATION # SPR 13
FROM Al Breveleri	DATE May 1979
SUBJECT 2200MVP Operating System Release 1.7	REORDER FROM:
THIS RELEASE SUPERSEDES:	DESTROY SUPERSEDED INFORMATION ☐ YES ☐ NO

Wang announces version 1.7 of the 2200MVP operating system. This version supports the new model 2280 disk drives (26.8, 53.6, and 80.4 MB), adds new capabilities, increases system performance, and corrects all known firmware errors. Release 1.7 replaces all previous versions and is required for any MVP configured with a 2280 disk drive.

Features

- 2280 Disk drive support
- $FORMAT DISK statement
- Variable device address specification

Restrictions

- All MVP systems should be upgraded to OS 1.7.

Ordering Information

Order through the Software Distribution Center, Package number 195-0049-3, diskette platter number 701-2294H.

Availability

In stock; allow one week for delivery.

Price

No charge to 2200MVP owners.

Support

This is a category 1, Wang supported product. Any suspected errors or anomalies found in this package should be reported to Wang Laboratories via the local district analyst.

Listed below are the systems changes since 1.6. The following files on the system platter differ from Release 1.6:

"@@" - MVP OS & BASIC-2 interpreter.
"@MOVE" - Move system file utility.

WANG LABORATORIES, INC.

Computers

MARKETING RELEASE

TO North American and International Sales Organizations; District Analysts	PUBLICATION # SPR 13
FROM Al Breveleri	DATE May 1979
SUBJECT 2200MVP Operating System Release 1.7	REORDER FROM:
THIS RELEASE SUPERSEDES:	DESTROY SUPERSEDED INFORMATION ☐ YES ☐ NO

"@FORMAT" - Format disk utility.

A. System Enhancements

The following enhancements are also available with VP BASIC-2 Release 2.0.

1. 2280 Disk Drives

Release 1.7 supports the 2280 disk drives; previous releases should not be used. For details concerning 2280 operations, see the "2280 Disk Drive User Manual".

2. Disk Platter Formatting

A format disk platter utility, "@FORMAT", for formatting 2260C, 2260BC, and 2280 disk platters resides on the system diskette. To format a disk platter, LOAD RUN "@FORMAT" and respond to the prompts as requested. The format utility makes use of the new $FORMAT DISK statement to format the specified disk platter (see following description of $FORMAT DISK).

3. Variable Device Address Specification

Device addresses used in SELECT statements can now be specified by the value of an alpha variable as well as explicitly by 3 hexdigits. Device addresses in SELECT statements have the following format:

device address = hexdigit hexdigit hexdigit / alpha-variable

where the value of the alpha variable must be 3 ASCII hexdigits representing the device type and address.

Examples:

A$ = "320": SELECT #3 A$ selects #3 to disk 320

TO North American and International Sales Organizations; District Analysts	PUBLICATION # SPR 13
FROM Al Breveleri	DATE May 1979
SUBJECT 2200MVP Operating System Release 1.7	REORDER FROM:
THIS RELEASE SUPERSEDES:	DESTROY SUPERSEDED INFORMATION ☐ YES ☐ NO

```
A$ = "215": SELECT PRINT  A$  (132)      selects print to
                                         device 215 with a
                                         line width of 132
                                         characters.
```

$FORMAT DISK

```
! General Form:                                                    !
!                                file #                            !
!      $FORMAT DISK platter                                        !
!                                disk-address                      !
!                                                                  !
```

Purpose:

The $FORMAT DISK statement issues a command to the disk processing unit to format the specified disk platter. This statement can only be used with disks that support formatting under software control (e.g., 2260C, 2260BC, 2280). Formatting on certain 2200 disks (e.g., 2270) is initiated by pressing the format button located on the disk unit.

Before a platter can be used for the storage and retrieval of data by the user, the platter must be formatted. Formatting involves recording a unique address for each sector on the disk platter, along with other control information used by the disk processing unit when accessing a sector. All data within the sector is zeroed.

```
+-------------------------------------------------+
! Caution:  Formatting a disk platter overwrites all !
! data that may previously have been stored on the   !
! platter.  It is recommended that the Wang supplied !
! format utility be used, to prevent the accidental  !
! formatting of the wrong disk platter.              !
+-------------------------------------------------+
```

Examples of valid syntax:

```
10 $FORMAT DISK T/310
20 $FORMAT DISK T/D11
30 $FORMAT DISK R#2
```

MVP BASIC-2 RELEASE 1.7
(Platter 701-2294H)

Release 1.7 of 2200MVP BASIC-2 replaces all previous MVP BASIC-2 releases. Release 1.7 is required for any MVP with 2280 disk drives. This release also provides all MVP systems with several new features and corrects all known system anomalies. Listed below are the system changes since Release 1.6. The following files on the system platter differ from Release 1.6:

"@@"	-	MVP OS & BASIC-2 interpreter.
"@MOVE"	-	Move system file utility.
"@FORMAT"	-	Format disk utility.

A. System Enhancements

The following enhancements are also available with VP BASIC-2 Release 1.9.

1. 2280 Disk Drives

Release 1.7 supports the 2280 disk drives; previous releases should not be used. For details concerning 2280 operations, see the "2280 Disk Drive User Manual".

2. Disk Platter Formatting

A format disk platter utility, "@FORMAT", for formatting 2260C, 2260BC, and 2280 disk platters resides on the system diskette. To format a disk platter, LOAD RUN "@FORMAT" and respond to the prompts as requested. The format utility makes use of the new $FORMAT DISK statement to format the specified disk platter (see following description of $FORMAT DISK).

3. Variable Device Address Specification

Device addresses used in SELECT statements can now be specified by the value of an alpha variable as well as explicitly by 3 hexdigits. Device addresses in SELECT statements have the following format:

device address = { hexdigit hexdigit hexdigit | <alpha-variable> }

where the value of the alpha variable must be 3 ASCII hexdigits representing the device type and address.

Examples:

A$ = "320": SELECT #3<A$> selects #3 to disk 320

A$ = "215": SELECT PRINT<A$>(132) selects print to device 215 with a line width of 132 characters.

WANG LABORATORIES, INC.

Computers

MARKETING RELEASE

TO North American and International Sales Organizations; District Analysts	PUBLICATION # SPR 15
FROM Tom Camp	DATE July 1979
SUBJECT MVP Operating System Release 1.8	REORDER FROM:
THIS RELEASE SUPERSEDES:	DESTROY SUPERSEDED INFORMATION ☐ YES ☐ NO

Wang proudly announces 1.8 of the 2200MVP Operating System. This new release expands the number of terminals allowed on the 2200MVP from 8 to 9 and provides features to utilize Wang's exciting new interactive terminal, the 2236DE.

Features

- Allows 9 terminals.
- New function PRINT BOX for 2236DE.
- Improved random number generation.

Restrictions

- All 2200MVP systems should be upgraded to Release 1.8.

Ordering Information

Order through Software Distribution:

Package Number: 195-0049-3
Diskette Number: 701-2294J

Availability

In stock, allow one week for delivery.

Price

No charge to 2200MVP owners.

Support

This is a category 1, Wang supported product. Any suspected errors or anomalies found in this package should be documented and sent to Wang Laboratories via the local district analyst.

MVP BASIC-2 RELEASE 1.8
(Platter 701-2294J)

Release 1.8 of 2200MVP BASIC-2 replaces all previous MVP BASIC-2 releases. This release provides all MVP systems with several new features and corrects all known system anomalies. Listed below are the system changes since Release 1.7. The following files on the system platter differ from Release 1.7:

"@@"	-	MVP OS & BASIC-2 interpreter.
"@GENPART"	-	System configuration utility.
"@PSTAT"	-	Partition status utility.

A. System Enhancements

1. NUMBER OF TERMINALS

The number of terminals allowed on the MVP has been increased from 8 to 9. This allows a 22C32 triple controller (single port MXD) to be used in conjunction with 2 4-port 2236MXD's.

2. PRINT BOX FUNCTION

A new function, BOX (X, Y), has been added to the PRINT statement for drawing or erasing lines and boxes on terminals having box graphics capability (such as the 2236DE). The BOX function has also been added to 2200VP BASIC-2 Release 2.1.

3. AUTOMATIC PROGRAM BOOTSTRAPPING

Before automatic program bootstrapping, the terminal (if attached to this partition) is initialized and the ready message is displayed. Note, the colon on the next line does not appear. Previously, the screen remained blank while a program was automatically loaded.

4. RANDOM NUMBERS

Random number generation has been improved to ensure less predictability for programs that are automatically loaded in and generate random numbers.

5. @GENPART

The system configuration utility has been updated to support 9 terminals.

6. @PSTAT

Capability for releasing a terminal or partition has been added to the partition status utility.

B. CORRECTED ANOMALIES

1. Immediate Mode load no longer allows the BEG parameter in its syntax since it has no function.

2. ON ERROR GOTO can now be used to trap D82 (file not found) errors occuring because of an attempt to overlay in a non-existent file.