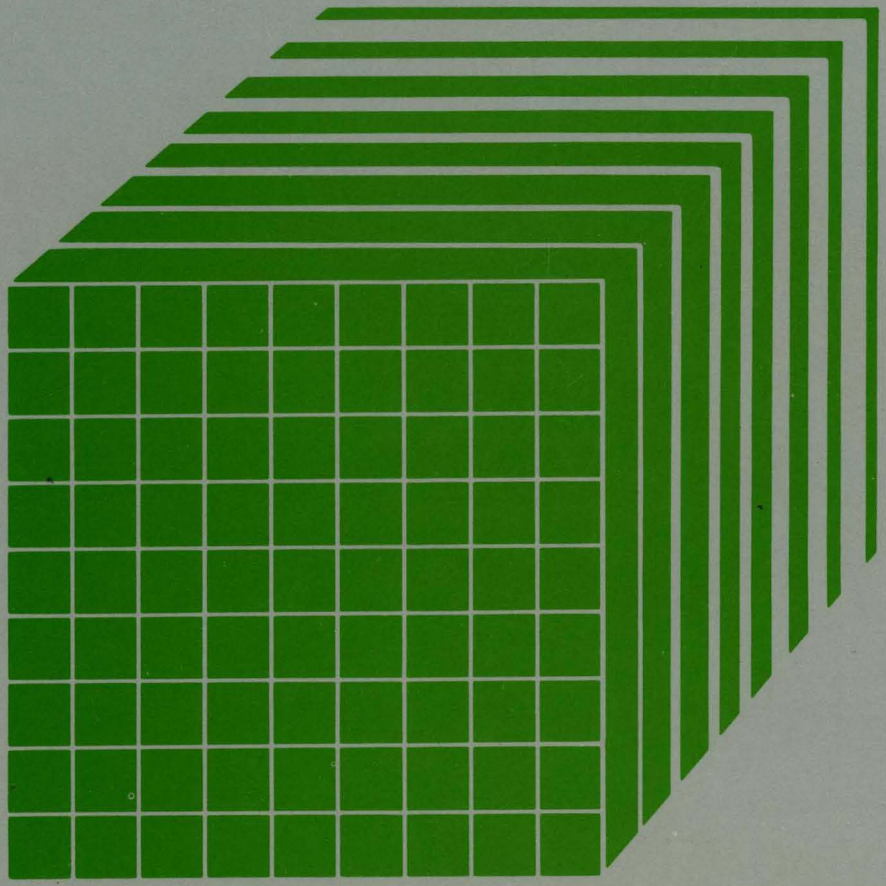
IBM

Virtual Machine Running Guest Operating Systems

VM/SP Release 5
VM/SP HPO Release 5

GC19-6212-5

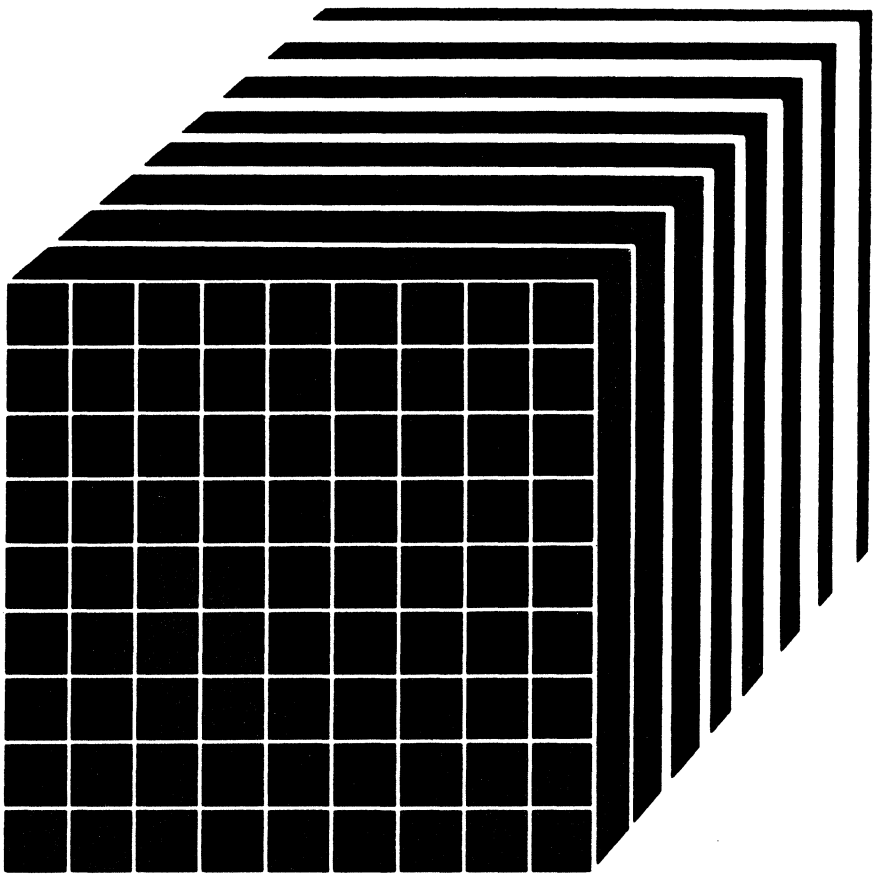


IBM

Virtual Machine Running Guest Operating Systems

VM/SP Release 5
VM/SP HPO Release 5

GC19-6212-5



| **Sixth Edition (August 1987)**

| This edition, GC19-6212-5, is a major revision of GC19-6212-4, and applies to Release 5 of the IBM
| Virtual Machine/System Product (5664-167), Release 5 of the Virtual Machine/System Product High Performance Option (5664-173), and to all subsequent releases and modifications until otherwise indicated in new editions or Technical Newsletters. Changes are periodically made to the information contained herein; before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/370, 30xx, and 4300 Processors Bibliography*, GC20-0001, for the editions that are applicable and current.

References in this publication to IBM products, programs, or services does not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM licensed program in this publication is not intended to state or imply that only IBM's licensed program may be used. Any functionally equivalent program may be used instead.

Publications are not stocked at the address given below; requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM, Dept. 52Q, Neighborhood Road, Kingston, New York, U.S.A., 12401. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Summary of Changes

Summary of Changes for
GC19-6212-5
As Updated August 1987 for
VM/SP Release 5 and
VM/SP HPO Release 5.0

4381 Processor Complex Models 11, 12, 13, and 14

Changed: Hardware Support

The 4381 Processor Complex Models 1, 2, and 3 are replaced and extended by the Models 11, 12, 13, and 14.

New Models of the 3090 Processor Complex

Changed: Hardware Support

In addition to supporting the 3090 Processor Complex Model 200, VM/SP HPO now supports the 3090 Processor Complex Models 150, 150E, 180, 180E, 200E, 400, and 400E (the 400 and 400E are supported in partitioned processing mode only). VM/SP HPO does **not** support the 300E and 600E.

3422 Magnetic Tape Subsystem

Changed: Programming Support

VM/SP HPO provides programming support for the 3422 Magnetic Tape Subsystem.

Documentation Changes

Minor editorial and technical changes have been made throughout this publication.

Summary of Changes for
GC19-6212-4 for
VM/SP Release 4 and
VM/SP HPO Release 4.2

VSE/SP 2.1

The VSE section of this manual contains updates in support of VSE/SP 2.1. Of particular interest to VSE users running VM are:

- Use of the VM/VSE Interface
- How CMS users can use the Interactive Interface
- Virtual Addressability Extension (VAE)
- The VM Writer Task of VSE/POWER
- CMS/DOS for VSE.

Control Switch Assist Extension to Preferred Machine Assist

New: Programming Support

This support allows an MVS/SP V=R virtual machine guest (Release 1 Enhancement or later) to use IUCV, many DIAGNOSE instructions, and some Service Call instructions. It also reduces line timeout problems for such guests by letting CP reflect virtual I/O interruptions to the guest.

3090 Processor Complex Support

New: Hardware Support

VM/SP HPO now supports the 3090 Processor Complex Model 200 when operating in System/370 mode as a dyadic processor.

Summary of Changes for
GC19-6212-3 for
VM/SP Release 4 and
VM/SP HPO Release 3.4

New Book

This publication replaces *VM/SP Operating Systems in a Virtual Machine*, Order Number GC19-6212 and *VM/SP HPO Operating Systems in a Virtual Machine*, Order Number GC19-6228. The term VM is used generically to refer to both VM/SP and VM/SP HPO.

This manual describes the basic concepts of running guest operating systems in a virtual machine and will help you to plan and execute the necessary steps to bring up the guest system under VM.

This publication is intended for the system programmer or operator with previous experience in installing the guest system (VSE, MVS, or VM) stand-alone, who now requires training to bring up that system under the control of VM/SP or VM/SP HPO.

Preface

The Virtual Machine/System Product (VM/SP) and Virtual Machine/System Product High Performance Option (VM/SP HPO) both provide an easy, convenient way to use a single terminal to run guest operating systems, such as VSE, MVS, or VM itself. A virtual machine is a functional equivalent of an IBM System/370 computing system. Each virtual machine has the functional equivalent of a real processor, main and auxiliary storage, and I/O devices. Because VM only simulates these functions, this simulated machine is referred to as a virtual machine. VM manages the functions of a real IBM System/370 in such a way that virtual machines are available to many users at the same time.

Audience

This book is intended for the system programmer or operator who has installed the guest system (VM, VSE, or MVS) stand-alone, and now requires assistance to bring up the guest system under the control of VM.

This book is not a substitute for training or for having a good basic understanding of the VM system. Therefore, before using this book, you should:

- Be able to operate the guest system on a real machine
- Understand basic System/370 data processing techniques
- Be familiar with the VM IPL procedure
- Understand the concepts and facilities of VM
- Be able to operate a VM terminal.

Organization

This publication is organized into three parts, with each operating system discussed exclusively in its own part:

- Part I. VSE/SP Versions 2 and 3 under VM
- Part II. MVS under VM/SP HPO
- Part III. VM under VM
- A bibliography

Note: In Part II, the discussion deals entirely with how to bring up MVS under VM/SP HPO; it does not include VM/SP. MVS can operate under VM/SP, but the recommended system is VM/SP HPO.

Terminology

This book uses the following terms:

VM Refers to both VM/SP Release 5 and VM/SP HPO Release 5. When unique considerations occur to either systems, they are noted separately.

VSE Refers to Version 2 and later releases, and Version 3 and later releases, of IBM Virtual Storage Extended/System Package (VSE/SP).

MVS Refers to Version 1 of the following releases of MVS/SP:

- MVS/SP - JES2 Release 1 with Release 1 Enhancement (5740-XYS)
- MVS/SP - JES3 Release 1 with Release 1 Enhancement (5740-XYN)
- MVS/SP - JES2 Release 3 (5740XYS) and later releases
- MVS/SP - JES3 Release 3.1 (5740-167) and later releases.

Cylinder Describes space on Direct Access Storage Devices (count-key-data devices) supported by the VM/SP system control program.

Block Describes DASD space on FB-512 devices supported by VM/SP.

area Appears in the text when there is no need to differentiate between DASD space on count-key-data devices or FB-512 devices.

Records Describes a spool file generated to represent physical card decks.

Any information about display terminal usage also applies to the IBM 3138, 3148, and 3158 Display Consoles when used in display mode, unless otherwise noted.

Any information pertaining to the IBM 3284 or 3286 printer also pertains to the IBM 3287, 3288, and 3289 printers, unless otherwise noted.

Any information pertaining to the IBM 2741 terminal also applies to the IBM 3767 terminal, Model 1, operating as a 2741, unless otherwise specified.

For a glossary of VM terms, refer to *IBM Virtual Machine/System Product Library Guide, Glossary, and Master Index*, Order No. GC19-6207.

Contents

Figures

Part One. VSE/SP Version 2 or 3 under VM

The procedure that follows assumes that you have your VSE/SP and VM systems up and running; it does not help you bring up either system. If you are not sure of all the basic functions of VM, please review them before you proceed.

Chapter 1. Introduction to Running VSE/SP Version 2 or 3 under VM

This section covers both VM/SP and VM/SP HPO. The generic term "VM" refers to both operating systems. Any differences between these systems will be addressed as they occur.

A virtual machine provides an easy, convenient way to run guest operating systems. When you run VSE/SP under VM, you get the functional equivalent of a real processor, main and auxiliary storage, and I/O devices. Because VM is simulating these functions, the simulated system is referred to as a "virtual" machine. This virtual machine is equivalent to an IBM System/370 computing system. When you run a guest VSE system under VM, it is running under the control program (CP) of the VM system.

VM manages the resources of the real computing system so that multiple virtual machines can execute commands at the same time. These virtual machines run independently of each other, and each can use a different operating system or different releases of the same operating system. The operating systems themselves execute as though they were controlling real devices and storage.

VM provides the guest system with a number of capabilities. It can:

- **Isolate online and batch production** — one VSE/SP virtual machine can be running a CICS/VS system, while another VSE virtual machine runs batch work only. In this mode of operation, a failure in the batch production VSE system does not impact the critical online system.

- **Isolate testing and production** — one virtual machine can be running production, while a second is running testing. Here again, the test virtual machine can be reIPLed without affecting the production system.

- **Run multiple batch production systems** — this can extend the number of partitions available if extra partitions are required. Alternatively, fewer partitions can be run in each of the virtual machines, thereby spreading the message traffic across several VSE consoles.

VM also offers:

- An outstanding interactive capability
- Ease of use (CMS)
- A wide range of Information Center products

- A true time sharing environment
- Complete isolation characteristics of virtual machines
- An environment for enhancing productivity.

Performance Considerations When Operating a VSE/SP Virtual Machine

Installations running VSE/SP under VM as a production environment (as opposed to a test or conversion environment) are naturally concerned with performance. Performance translates into actual production run times and online response times. The following factors affect the performance of a virtual machine:

- The amount of real storage available to the guest
- The amount of contention with other guests and CMS for channels, control units, devices, etc.
- The frequency of real interruptions
- The frequency and type of privileged instructions executed
- Whether the virtual machine assist or VM/370 extended control-program support hardware is on the machine
- The frequency of START I/O (SIO) instructions
- The location of reference within virtual storage
- The amount of fixed-head paging space
- The location of the paging areas on DASD.

Reducing Paging Activity

When a virtual machine refers to virtual storage addresses that are not in real storage, a page fault (and paging activity) occurs. Routines that have widely scattered storage references tend to increase the paging load caused by this virtual machine.

When possible, modules dependent upon each other, as well as the related reference tables, constants, and literals, should be located in the same 4K page. Infrequently used routines such as those that handle unusual error conditions should not be placed near main routines. To minimize paging, reentrant coding techniques should be used whenever possible.

The performance of both the VM system and the individual virtual machines running under it can be measured and evaluated. How well the system responds is of prime importance to the general user. How efficiently an individual virtual machine makes use of the storage, processor, and I/O facilities allotted to it is of prime importance to the system analyst.

However, performance characteristics are difficult to predict when VSE is running under VM, due to several complex factors. These factors can be broadly classified into three groups, they are:

- Configuration factors
- Operating system workload factors
- VM performance factors.

Although a specific virtual machine's performance may not equal that of a real system running stand-alone, in some situations the total throughput obtained in the virtual machine environment will be equal to or better than the throughput obtained on a real system.

Configuration Factors Influencing Performance

The following hardware configuration factors influence the performance of an operating system in a virtual machine :

- The amount of real storage available
- The speed, capacity, and number of paging devices
- The amount of channel and control unit competition and the arm rivalry affecting each paging device
- Whether virtual machine assist or VM extended control-program support is installed on the hardware and enabled
- Interference between system paging devices and devices for processing a user's I/O requests.

When you run VSE in a virtual machine instead of running VSE stand-alone, there is an increased need for real storage, DASD space, and processor speed. VM's need for increased dispatching, scheduling, and paging is relatively small in comparison with the overhead incurred in simulating privileged instructions.

When VSE operates stand-alone, it runs directly on its own hardware and manages its resources through the use of privileged instructions such as SIOF and LPSW. When executing in a virtual machine, VM dispatches VSE in problem state, and any privileged instruction issued by the virtual machine causes a real privileged-instruction exception interruption. This interruption either causes machine control to be transferred to VM microcode or it causes CP to simulate the instruction. The amount of work done by VM in analyzing and handling a virtual machine initiated interruption depends upon the type and complexity of the interruption. Therefore, reducing the number of privileged instructions issued by the virtual machine reduces the amount of extra work VM must do to support the VSE guest.

Virtual machine assist support has been specifically designed to reduce the VM overhead associated with simulating privileged instructions. It is the most effective method for reducing privileged instruction simulation time.

The virtual machine assist feature is described in *CP for System Programming*.

VM/370 extended-control program support (ECPS: VM/370) is a hardware assist function that provides support over and above that provided by virtual machine assist. It improves VM performance by reducing VM's real supervisor state time, which is needed to support virtual machines. *CP for System Programming* lists the specific functions of ECPS: VM/370 that certain System/370 models support.

Workload Factors Influencing Performance

The following workload factors influence the performance of VSE running within a virtual machine:

- The total number of virtual machines running under VM
- The type of work each virtual machine is doing, especially the amount of I/O processing required.

By measuring and evaluating the effects of these workload factors on a specific configuration, you can anticipate their effect on performance.

To measure workload performance in a specific configuration, you can use the licensed program called VM Performance/Monitor Analysis Program (VMMAP). This program plots a number of important system variables (such as processor usage, various contention measurements, and paging rates) against workload measurements for both the CMS and operating systems under VM. For a specific configuration, it allows you to relate processor usage, storage usage, and resource contention to the total system workload in both interactive and batch production environments. By using this analysis program, you can eventually determine the optimum processor model, storage size, and I/O configuration for a specific workload.

VM Performance Factors

After measuring the performance of both VM and the virtual machines it supports, the system analyst and the general user can use certain VM performance options. These options create a special performance environment for one or more virtual machines.

The options available to the system analyst are:

- Virtual=Real option [1]
- Locked pages
- Reserved page frames

---

1 This option cannot be specified in a command. To obtain it, a general user asks the VM system administrator to specify it on the OPTION control statement (VIRT=REAL option) for the user's virtual machine directory entry. The CP nucleus must also be generated with the V=R option.

- Virtual machine priority
- Favored execution
- QDROP OFF.

The options available to the general user are:

- Virtual machine assist
- VM/370 extended control-program support
- STBYPASS command for a virtual machine.

The following options are available to only **one** virtual machine at a time:

- Reserved page frames option [2]
- Virtual=Real option.

The following options are available to **as many** virtual machines as desired:

- Favored execution with a specified percentage
- Basic favored execution (without a specified percentage)
- User priority
- Virtual machine assist
- VM/370 extended control-program support (ECPS: VM/370)
- Locked pages
- QDROP.

For information about the effect of these options, refer to *CP for System Programming*. For specific details about how to use these options, refer to the *CP Command Reference*.

VM provides certain CP commands (INDICATE and MONITOR) to allow both VM's and the virtual machine's performance to be tracked and measured. Other commands allow the setting of certain options to improve performance. To reduce and help analyze the data produced by the MONITOR command, the licensed program called VM Performance/Monitor Analysis Program (VMMAP) is available. By using this program, an installation can eventually determine its optimum processor model, storage size, and I/O configuration for a specific workload. For a complete description of the INDICATE and MONITOR commands, refer to *CP for System Programming*.

Date and Time Zones in the VSE/SP Virtual Machine

When IPLing the VSE/SP virtual machine, the date and clock fields of the SET DATE CLOCK command are ignored. When VSE/SP tries to set the TOD clock to the values specified in the command, VM ignores the attempt.

If you do not use the VSE SET ZONE command, then VSE/SP uses ZONE=WEST/00/00 and assumes that the hardware TOD clock is set to local time.

---

[2] VM/SP HPO 3.4 and later releases allow multiple virtual machines to utilize reserved page frames.

You can set the zone value on a guest VSE/SP system by issuing the SET ZONE command any time before you enter the SVA command.

The SET CLOCK instruction cannot be simulated and is ignored if issued by a virtual machine.

Running Multiple VSE/SP Virtual Machines under VM

In a non-VM environment you might be running one online production VSE partition and one online test partition. These run in different VSE partitions and are dispatched based on the setting of the VSE PRTY command.

The situation may be different when you run VSE under VM. You can set up your present production system as one virtual machine and the test system as a second virtual machine, both running under the control of VM. VM schedules the requests that each virtual machine makes for I/O.

VM allocates time slices of the processor to virtual machines so that each virtual machine receives a comparable amount of processor time. It knows nothing about the programs that may be running within a virtual machine.

When a VSE virtual machine gains control of the processor, it schedules requests from the partitions based on their priority. If your test system runs in a low-priority partition, it may not get any of its requests serviced if the processor is kept busy servicing higher priority partitions running at the same time.

When the VSE virtual machine's time slice ends, VM does the following:

- Stops the VSE virtual machine,
- Schedules another time slice for it at a future time, and
- Passes control to the next virtual machine waiting in line.

This process is repeated for every time slice, so that all VSE partitions compete for resources during every time slice given to the VSE machine. In a heavily used system, low priority test partitions may have slow response time.

One approach is to run your VSE test system in its own virtual machine. The test system then receives its own share of the processor, according to the way you design your VM system. You may not have to add DASD to support the environment — the existing DASD can be shared. There are special considerations that should be reviewed (performance and data integrity) before sharing DASD. See "Chapter 4. VSE/SP Virtual Machines Sharing DASD" on page 75.

When running multiple VSE virtual machines under VM, you will want to make sure that you give the right resources to the VSE production virtual machine. You can have the SET FAVOR and SET PRIORITY options benefit the production VSE system rather than the test VSE system.

Chapter 2. Defining a Single VSE/SP Virtual Machine

This chapter discusses the necessary changes to both the host (VM) and guest (VSE) operating systems. We first address the changes to be made to the VM system; then we address changes to the guest VSE/SP virtual machine. Ultimately, we IPL the VSE/SP system under VM.

Preparing the Host VM System

A sample directory entry for the VSE/SP virtual machine is included in this chapter. (Before you follow these examples, you should evaluate their usefulness at your installation.) An explanation of each directory entry follows the example. An example of how to update the DMKRIO, DMKFCB, and DMKSYS system files is also included.

The VM Directory Entry for the VSE/SP Virtual Machine

Log on to your VM system and enter or change the directory entry for the VSE/SP virtual machine to include the following statements (if necessary):

- USER
- OPTION
- IPL
- CONSOLE
- SPECIAL
- SPOOL
- DEDICATE
- LINK
- MDISK.

A sample directory entry for the VSE/SP virtual machine is shown in Figure 1 on page 10. An explanation of the use of each statement follows.

```
VMUSER DIRECT    A1  F 80  TRUNC=72 SIZE=55 LINE=9 COLUMN
===== ********************************************************
===== *                    SYSTEM          USERIDS            *
===== ********************************************************
===== *
===== USER VSESP31 PASSWORD 16M 16M BG 64
===== OPTION ECMODE BMX REALTIMER CPUID 000001
===== ACCOUNT ### SYSPROG
===== IPL CMS
=====   CONSOLE 009 3215 T OPERATOR
=====   SPECIAL 080 3270
=====   SPECIAL 081 3270
=====   SPECIAL 082 3270
=====   SPECIAL 083 3270
=====   SPECIAL 084 3270
=====   DEDICATE 01F 01F
=====   SPOOL   00C 3505 A
=====   SPOOL   00D 3525 A
=====   SPOOL   00E 3211 A
=====   SPOOL   02E 3211 A
=====   SPOOL   05D 3525 A
=====   SPOOL   05E 1403 A
===== * Link to VM 5.0 Executable CMS Code
=====   LINK MAINT 190 190 RR
===== * Link to Program Products (Y-Disk)
=====   LINK MAINT 19E 19E RR
===== * To execute the VSEMAINT's profile
=====   LINK VSEMAINT 191 191 RR
===== * The following lines are sample MDISK statements based on the
===== * different DASD types available.  To use, take the "*" off the
===== * DASD type you'll be using.  Don't forget to change the virtual
===== * address to the virtual address your installation will be using.
===== * 3330 SYSTEM
===== * MDISK 150 3330 000 411 DOSRES MWV VSESP31 VSESP31
===== * MDISK 151 3330 000 411 SYSWK1 MWV VSESP31 VSESP31
===== * MDISK 152 3330 000 411 SYSWK2 MWV VSESP31 VSESP31
===== * MDISK 153 3330 000 411 SYSWK3 MWV VSESP31 VSESP31
===== * MDISK 154 3330 000 411 SYSWK4 MWV VSESP31 VSESP31
===== * 3340 SYSTEM
===== * MDISK 1C0 3340 000 698 DOSRES MWV VSESP31 VSESP31
===== * MDISK 1C1 3340 000 698 SYSWK1 MWV VSESP31 VSESP31
===== * MDISK 1C2 3340 000 698 SYSWK2 MWV VSESP31 VSESP31
===== * MDISK 1C3 3340 000 698 SYSWK3 MWV VSESP31 VSESP31
===== * MDISK 1C4 3340 000 698 SYSWK4 MWV VSESP31 VSESP31
===== * 3350 SYSTEM
===== * MDISK 350 3350 000 560 DOSRES MWV VSESP31 VSESP31
===== * MDISK 351 3350 000 560 SYSWK1 MWV VSESP31 VSESP31
===== * In our example we used the following:
===== * 3370 SYSTEM
===== * MDISK 530 FB-512 000000 712752 DOSRES MWV VSESP31 VSESP31
===== * MDISK 531 FB-512 000000 712752 SYSWK1 MWV VSESP31 VSESP31
===== * 3380 SYSTEM
===== * MDISK 240 3380 1 884 SPADOS MWV VSESP31 VSESP31
===== * MDISK 241 3380 1 884 SPASY1 MWV VSESP31 VSESP31
===== * MDISK 220 3380 0 885 SPADOS MW VSESP31 VSESP31
===== * MDISK 221 3380 0 885 SPASY1 MW VSESP31 VSESP31
```

Figure 1. VM Directory Entry for VSESP31 using a Dedicated Console

Note: At logon, as the directory control statements for the user are processed, CP checks the devices represented by each MDISK, CONSOLE, DEDICATE, LINK, SPECIAL and SPOOL statement for a possible conflict with the virtual control unit (VCU) interface. This conflict occurs because the VCU cannot support two different subchannel protocols (shared and nonshared) at the same time. For each directory control statement that violates the restriction, CP sends an error message to the user and does not create the virtual device. To avoid this problem, see the Planning Guide and Reference for a complete listing of the virtual device characteristics.

The USER Statement

```
USER VSESP31 password 16M 16M BG 64
```

USER VSESP31 defines the userid as VSESP31.

password password can be changed to the password of your choice.

16M 16M the first 16M defines the virtual machine's storage size. The second 16M entry defines the maximum virtual machine storage size that this user can define after logging on the system. The storage is usually set to 16M to allow maximum VSIZE for the VSE/SP virtual machine.

If you run the VSE/SP guest in MODE=370, it will do its own paging. In that case, you may want to define a virtual storage size of 6M or 8M, to limit the amount of double paging.

Note: The VSE stand-alone utilities (used to install the VSE/SP system) should not be IPLed in a 16M virtual machine because they cause extreme paging in the the VM environment. You may need to use the CP DEFINE storage command.

BG class B (resource) is assigned so that the VSE/SP virtual machine user can issue CP ATTACH and DETACH commands. Please refer to the *CP Command Reference* for a summary of the CP commands allowed by privilege classes.

class G (general) users control the functions associated with the execution of their virtual machine. The VSE/SP virtual machine is usually assigned with class G privileges; this prevents one VSE guest from interfering with another guest on the VM system.

Note: Generally, a guest user will need only class G authority. If a user has class B authority, unpredictable results can occur if he attaches real devices at the same virtual addresses as real addresses.

64 the priority setting depends on the use of your VSE/SP virtual machine. For example, a VSE virtual machine running teleprocessing will usually have a higher priority than a batch VSE virtual machine. The lower the priority value is numerically the higher is its relative priority. (The default is 64.)

Note: If you have a high-priority virtual machine, start its priority setting at about 30. If after running at this setting you wish to increase its priority, subtracting ten from the thirty setting will double its priority; adding ten will halve its priority.

The OPTION Statement

```
OPTION ECMODE BMX REALTIMER CPUID 000001
```

VM provides several optional services to virtual machines. You can specify these services with the OPTION control statement in the VM directory, or with the CP SET command.

ECMODE The ECMODE option allows the virtual machine to use the complete set of VM control registers and the dynamic address translation feature. Programming simulation and hardware features are combined to allow use of all the available features in the hardware.

You must specify the ECMODE option because the VSE/SP virtual machine is:

- Running in extended control mode
- Using dynamic address translation (DAT) (except in MODE=VM)
- Using extended control registers other than zero
- Addressing I/O channels 6 through 15.

If the ECMODE option is not specified in the directory, you can enter extended control mode by issuing the CP command SET ECMODE ON. For example:

```
#cp set ecmode on
```

Note: Setting the ECMODE option does not alter the ECMODE bit of the user's PSW.

BMX The BMX (virtual block multiplexer) option allows the VSE virtual machine to overlap multiple SIO(F) requests on a specified channel path. The selector channel mode is the normal (default) channel mode for virtual machines. When the BMX option is given control, it applies to all channels in the virtual machine, except to channel 0. This option can be specified regardless of whether block multiplexer channels are attached to the processor. The CP DEFINE command can be issued to redefine the channel mode for a virtual machine. For example:

```
#cp define channels bmx
```

REALTIMER The REALTIMER option causes the virtual interval timer to be updated during virtual wait state. In VSE/SP, only VSE/PT uses the interval timer. You should set the REALTIMER option off unless you are running VSE/PT. This option will have no effect on the CPU timer or the clock comparator.

With the REALTIMER option not in effect, a virtual interval timer reflects virtual processor time and virtual wait time, but not CP time used for services for that virtual machine (such as privileged instructions execution). The more services a virtual machine requires from CP, the greater the difference between the time represented by the interval timer and the actual time used by (and for) the virtual machine. The larger the number of active virtual machines contending for system resources, the greater the difference between virtual machine time and actual elapsed time.

If this option is not specified in the directory entry, you can obtain this timing facility by issuing the CP SET command with the TIMER operand. For example to turn on the timing facility, issue:

```
#cp set timer real
```

To turn off the option, issue:

```
#cp set timer off
```

CPUID When VSE guests are sharing resources like DASD, it is necessary to associate a unique CPU identification (CPUID) with each virtual machine to keep track of the resources the system is using. If you don't specify a unique CPUID, it will default to the real system CPUID with the first two characters replaced by "FF". For a complete discussion, refer to "Defining CPUIDs for the VSE/SP Virtual Machine" on page 42.

VIRT=REAL Specify the VIRT=REAL option if you use any MODE=370 guest (for example, VAE) in a V=R machine.

The ACCOUNT Statement

```
ACCOUNT ### SYSPROG
```

The ACCOUNT control statement defines an account number and a distribution identification (SYSPROG). The account statement is optional. If omitted, both the account number and the distribution code default to the userid. The ACCOUNT statement must follow the USER statement.

The IPL Statement

```
IPL CMS
```

The IPL statement automatically IPLs a system either by name (for saved systems) or by device address. You may want to IPL CMS (as we have done in our sample directory entry in Figure 1 on page 10) to execute the PROFILE EXEC that does your SET and ATTACH commands, thereby setting up the virtual environment.

The CONSOLE Statement

The CONSOLE control statement specifies the virtual machine console. In the VM/VSE environment, the way you define the VSE/SP console depends on the following four considerations:

- Will VM and VSE have separate consoles?
- Will the VSE console support the VM operations?
- Will VSE be autologged?
- Will VSE be logged on manually?

In our sample directory, we have dedicated the main processor console to the VSE virtual machine as the VSE operator's console. By doing so, the VSE virtual machine operator sees no changes in operation from when VSE was running stand-alone. You should always try to have a spare screen available in your installation and make it your CP console. Using this dedicated console approach, the CONSOLE statement for the VSE virtual machine could be defined in the VM directory as:

```
CONS 009 3215 T OPERATOR
```

where **OPERATOR** is the secondary userid receiving all CP messages for the VSE virtual machine when the primary userid is running disconnected. 009 is the virtual device address of the console in VSE's IPL procedure. The VSE console must be defined in 3215 mode.

The secondary userid can send CP commands to the disconnected VSE machine. For example, to send an external interrupt command from the secondary userid, issue:

```
send vsesp31 cp external 40
```

The SPECIAL Statement

```
SPECIAL 080 3270
```

The SPECIAL statement defines a virtual unit with device type and virtual address. Terminal addresses defined in this way don't really have to be available on the system because they are not real addresses. With the SPECIAL statement in the directory, the DIAL command can be issued to gain access to the guest machine. For an example of how to do this, refer to "Nondedicated Terminal Definitions" on page 35.

The DEDICATE Statement

The DEDICATE control statement specifies that a real device is to be dedicated to this userid. A real device can be dedicated to only one user at a time. Because of the way the CONSOLE control statement was set up in our sample directory, the DEDICATE control statement must be included in the directory.

```
DEDICATE 01F cuu
```

where **cuu** is the real device address of the terminal to be used as the VSE console and 01F is the virtual device address.

Following the above concept, the processor console 01F has to be disabled before the VSE virtual machine is logged on. (This can be done by having the VSE virtual machine automatically logged on through AUTOLOG1 or by the VM operator issuing a the CP AUTOLOG command.) When you have the console dedicated, the VM operator has the responsibility of handling all CP requests for the VSE virtual machine (as long as the VSE machine is in disconnected mode.) The disadvantage to this type of VSE console operation is that you have to have a second screen available in case VSE hangs.

Note: *In order to avoid a usage conflict caused by control unit I/O interface protocol, use caution in defining the virtual device address in the DEDICATE statements. Some devices use a shared subchannel protocol and others do not. Therefore, devices must be grouped by control unit within a given channel according to their subchannel usage. CP does not permit you to group devices that use the shared subchannel protocol together with devices that do not use the shared protocol. The following is an example of a virtual machine's DEDICATE statement that would be rejected at logon.*

```
DEDICATE 12E 30E (30E is a real 3211)
DEDICATE 12F 580 (580 is a real 3420 tape device)
```

The virtual addresses of both the 3211 and the tape device indicate the use of control unit (2). A real 3211 printer operates on a nonshared subchannel, and the real 3420 device is designed for shared subchannel operations. By definition the devices are virtual and therefore will share one virtual control unit (VCUBLOK) in CP which

has a range of eight devices. When the user logs on, the two dedicate statements results in the second virtual device (12F) not being created and an error message being sent to the user.

Therefore, when defining devices, make sure the devices are defined and separated within their own control unit range and not shared with other devices. This restriction also applies to the CONSOLE, MDISK, SPECIAL, SPOOL, and LINK statements. The effects of the DEDICATE, LINK and MDISK statements depend on the real device configuration at LOGON time. To avoid this problem refer to the Planning Guide and Reference for a complete listing of the virtual device characteristics.

For additional information on the various uses of the DEDICATE control statement refer to Chapter 3 under "Various Uses of the DEDICATE Statement" on page 32.

The SPOOL Statement

```
SPOOL   00C 3505 R A
SPOOL   00D 3525 A
SPOOL   00E 1403 A
SPOOL   02E 3211 A
```

The SPOOL control statement specifies the unit record device that is to be spooled. Multiple readers, punches, and printers may be specified, each on a separate SPOOL statement.

An entry in the directory is necessary for each unit record device that is not attached to the VSE system but will be used by VSE (except for the VSE dummy devices FEC, FED, FEE, FFC, FFA, FFD, and FFE). You should have matching device type definitions for VM and VSE in the ADD statement. If the definitions do not match, the VSE recorder file will soon fill up. The message:

```
RECORDER FILE FULL - RUN EREP
```

will be displayed indicating that repetitive error handling with nonmatching devices filled up the RECORDER FILE.

An example of matching definitions can be seen between Figure 1 on page 10 and Figure 3 on page 22.

Note: For some devices, like 3211s and 3262s, matching definitions are impossible.

READER

```
SPOOL 00C 3505 R A
```

Other virtual reader devices require that you issue **READY cuu** under some circumstances, and also require that you spool the reader continuous. (If you are using the VM/VSE Feature this entry should be included in the PROFILE EXEC for VSEMAINT.)

The VSE/POWER reader task should be PSTARTed to the lowest spooled card reader of the desired class, so that the attention interrupt will be processed correctly.

PRINTER

```
SPOOL 00E 1403
```

You should start your POWER print writers with the VM parameter unless you are using a dedicated printer.

For any print writer started with the VM parameter, POWER always sends the FCB as the first part of every print file. Therefore, it does not matter in which sequence the output is printed; the correct FCB will always be used.

The MDISK Statement

The MDISK control statement describes the DASD extent to be owned by the user on a direct access device. The DASD area assigned with this statement becomes the user's minidisk. The following MDISK statement defines a full FBA device with VOLID = DOSRES.

```
MDISK 540 FB-512 000000 712752 DOSRES MWV VSESP31 VSESP31
```

VM does not check for overlapping extents in the MDISK statement. Therefore, you must ensure that minidisk extents defined in the VM directory do not overlap each other, or, in the case of 3330, 3340, and 3350 disks, do not overlap the alternate track cylinders.

DASD can be assigned to a virtual machine as a whole volume or as part of a volume. If a whole volume is to be assigned, you can use either the DEDICATE or the MDISK statement. In deciding which statement to use, be aware that the DEDICATE statement allows only one user to access the disk drive through that cuu address, whereas the MDISK statement allows the disk to be shared between virtual machines. If you want to allocate part of a volume, use the MDISK statement. It is also possible to allocate part of a VM volume to VSE and part of a VSE volume to VM. To allocate part of a volume, you will need to set up an MDISK statement for the part of the volume you want VSE to own, and an MDISK for the VM part. Then you will need to initialize the VSE minidisk using the Device Support Facility. You can use the IPL DSF file on Maint's 'S' disk to initialize the volume.

In order to back up VSE-related data (for example, for Fast Copy), use addresses 240 and 241 from Figure 1 on page 10 and shown in the following example.

In order to back up VM-related data, uses addresses 220 and 221. This backs up the entire disk.

```
MDISK 240 3380 1 884 SPADOS MWV VSESP31 VSESP31
MDISK 241 3380 1 884 SPASY1 MWV VSESP31 VSESP31
MDISK 220 3380 0 885 SPADOS MW VSESP31 VSESP31
MDISK 221 3380 0 885 SPASY1 MW VSESP31 VSESP31
```

The LINK Statement

The LINK control statement makes a device that belongs to another user (userid) available to this virtual machine at logon. If you want to make one volume available to several virtual machines:

- Define the volume for one of the virtual machines with an MDISK statement.
- Define a link to that volume, using the LINK statement for all other virtual machines that use the volume.

Later, if you must move or change that volume, you need only update the one MDISK statement; the LINK statements need not be updated. In the directory example, you are linking to the VSEMAINT disk with read-only access authorization.

```
LINK VSEMAINT 191 191 RR
```

Upon Completion of Directory Changes: The directory entry is complete. If you made additions or changes you must file the new directory and issue the CMS DIRECT command. The DIRECT command processes the directory file to see if it follows the required format. To actually change or swap the current active VM directory, you must have write access to the system-owned (system residence or IPL device) volume that contains the current directory up to and including the directory cylinders, or to the volume that is to contain the new directory.

Issue:

```
direct filename
```

Note: Make sure that your VM virtual devices match the devices of the VSE system. In other words, make sure that the devices defined in the VM directory entry for the VSE userid match those in the Automated System Initialization (ASI) procedure you use to IPL VSE.

CP Nucleus Considerations

If your VSE/SP system is using devices unsupported by VM, you will have to make changes to the DMKRIO file. If you make any changes to the DMKRIO, DMKFCB, or DMKSNT files, you will have to generate a new CP nucleus. For a complete discussion on the system-dependent files, refer to the *Planning Guide and Reference.*

If you don't need to make changes to the system dependent files, skip this reference and continue with the section "Preparing the Guest VSE/SP Virtual Machine" on page 20.

Updating DMKRIO

In the DMKRIO file you define all the real devices that are attached to the system. If your VSE/SP system is using devices not supported by VM, or if you want the VSE/SP console defined at real address 01F instead of the VM console, you will have to make changes to DMKRIO.

When you have unsupported devices, you have to specify them as unsupported in DMKRIO and dedicate them to the VSE/SP system in its DIRECTORY entry. In the DMKRIO file you might have:

```
RDEVICE ADDRESS=cuu,DEVTYPE=type,CLASS=URI
```

------ and ------

```
RCTLUNIT ADDRESS=cuu,CUTYPE=UNSUPPORTED
```

where **cuu** is the real address of the unsupported device. These devices must have matching entries in the ASIPROC. For unsupported device types you must specify a device subclass in the CLASS operand. For a complete listing of the available subclasses refer to the *Planning Guide and Reference.*

Note: When preparing the RDEVICE and RCTLUNIT entries, refer to "Appendix B. Configuration Aid" in the Planning Guide and Reference.

Along with the changes in DMKRIO, the unsupported device should have a matching entry in the directory. In the sample directory, the entry for an unsupported device would be:

```
DEDICATE vaddr raddr
```

where **raddr** is the real address and **vaddr** is the virtual address. In this case the VSE/SP virtual machine is responsible for error recovery and error recording procedures.

The changed RIOGEN macro instruction would be:

```
RIOGEN CONS=010,ALTCONS=(01F,009)
```

where **010** is now the address of the VM primary console and **01F, 009** are the alternate consoles. These addresses must have been specified in the RDEVICE macro instruction.

Updating DMKSNT

The only change you need to make in the DMKSNT file is if you intend to save the VSE/SP virtual machine as a SAVE SYSTEM. This is generally not done in this environment because there is no good starting point from where to save the system.

Building a new CP nucleus

If changes were made to the system dependent files you will have to assemble them using the GENERATE EXEC or the VMFASM command. GENERATE is a multipurpose EXEC used to generate VM, perform updating maintenance of CP, CMS, and VM service programs, and it can also be used to regenerate the VM system after updating:

- The directory
- The real I/O configuration (DMKRIO)
- The system control file (DMKSYS)
- The system name table (DMKSNT).

For a complete discussion and examples on how to build a new CP nucleus, refer to the *Installation Guide.*

Preparing the Guest VSE/SP Virtual Machine

VSE/SP contains three pregenerated supervisors. You can use two of these when running VSE/SP under VM:

- $$A$SUPV for MODE=VM
- $$A$SUP3 for MODE=370

You do not need to change or regenerate these supervisors to use them.

When to Use MODE=VM

Use **MODE=VM** when you want to take full advantage of the VM/VSE handshaking facilities. These include:

- SET PAGEX ON for pseudo page fault handling
- Paging done only once (by VM)
- CCW translation done only once (by VM)
- PAGE release (DIAGNOSE 10)
- BTAM Autopoll assist
- Disconnected console feature
- CPCOM macro.

When to Use MODE=370

Use **MODE=370** when you want to use Virtual Addressability Extension (VAE) or run V=R. This will give you limited VM/VSE handshaking support:

- SET PAGEX ON
- BTAM Autopoll Assist
- CPCOM macro.

Changes in $ASIPROC

There are three ways to initialize VSE/SP :

1. $ASIPROC (ASI Master Procedure)

   The search for $ASIPROC.PROC in IJSYSRS.SYSLIB is always the first test performed by the IPL routine. The IPL routine searches for $ASIPROC and, if found, looks for an entry that matches the CPUID. If the CPUID matches, the procedures named are executed.

2. $IPL370 and $$JCL370 (Default IPL and JCL Procedure Names)

   If the $ASIPROC procedure is not found, the IPL routine executes procedures with these names (if available on the system).

3. Prompts (Interactive IPL)

   If neither 370 procedures nor $ASIPROC are found, the IPL routines prompt the operator for the appropriate IPL/JCL procedures.

To allow an IPL of VSE/SP both native and under VM, you can catalog an $ASIPROC with two entries in it. The first entry would be for VSE/SP running under VM and the second entry for VSE/SP running stand-alone. To allow an IPL of VSE when it is running under VM or stand-alone, duplicate the entry for running VSE native and change the following:

- Change the real CPUID prefix 00 or 02 to FF

  ------ or -----

  match the CPUID specified in the VM directory entry — prefix it with FF and suffix it with the processor type. (For example, compare the CPUID specified in Figure 1 on page 10 and Figure 2 on page 22.)

- Change or insert MODE=370.
- Change the IPL procedure name to the name for the VM/VSE IPL.

The following figure shows what your $ASIPROC can look like. The first entry is for VSE under VM and the second for VSE stand-alone. When you specify a unique CPUID in the $ASIPROC the same CPUID must be specified in the VSE/SP virtual machine directory entry or in a SET CPUID command before IPLing VSE/SP.

```
 CATALOG $ASIPROC.PROC     REPLACE=YES
CPU=FF0000014361,IPL,=$IPLVMG,JCL=$$JCLVMG,MODE=370
CPU=020600094361,IPL,=$IPLE,JCL=$$JCLE,MODE=E
/+
```

Figure 2. Master $ASIPROC with VSE/SP Virtual Machine

Changes in the $IPLxxx Procedure

In Figure 3 you can see the following changes were made to allow the IPL of the VSE/SP virtual machine:

- $$A$SUPV is the name of the supervisor generated for MODE=VM
- The DPD command was dropped because VM does the paging
- The following parameter was dropped from the SYS command:
  - BUFSIZE (VM does all CCW translations.).

```
  .
  .
01F,$$A$SUPV,VPOOL=64K,P,LOG     * SUPERVISOR FOR MODE=VM
ADD 00C,3505                     * VM SPOOLED DEVICE
ADD 00D,3525P                    * VM SPOOLED DEVICE
ADD 00E,PRT1                     * VM SPOOLED DEVICE (3211)
ADD 01F,3277                     * CONSOLE
ADD 02E,PRT1                     * VM SPOOLED DEVICE (3211)
ADD 05D,3525                     * VM SPOOLED DEVICE
ADD 05E,1403                     * VM SPOOLED DEVICE
ADD 080:084,3277                 * TERMINALS DEFINED AS SPECIAL
ADD 181,3420T9                   * ATTACHED TAPE
ADD 530:531,FBA                  * DEDICATED DISKS
ADD FEC, 3505                    * ADDED FOR POWER
ADD FED, 3525P                   * ADDED FOR POWER
ADD FEE, PRT1                    * ADDED FOR POWER
ADD FEF, PRT1                    * ADDED FOR POWER
ADD FFA,3505                     * ADDED FOR ICCF
ADD FFC,3505                     * ADDED FOR ICCF
ADD FFD,3525P                    * ADDED FOR ICCF
ADD FFE,PRT1                     * ADDED FOR ICCF
DEF SYSCAT=DOSRES,SYSREC=SYSWK1
SYS JA=YES,CHANQ=254,DASDFP=NO,SEC=NO
DLA VOLID=DOSRES,BLK=55118,NBLK=744,DSF=N,NAME=AREA1
SVA PSIZE=534K,SDL=250,GETVIS=64K
  .
  .
  .
  .
```

Figure 3. Sample IPL Procedure for VSE/SP Virtual Machine

The following PROFILE EXEC is shared by VSEMAINT and VSESP31; it resides on VSEMAINT's 191 disk. VSESP31 has a read only link to VSEMAINT's 191. Sharing the PROFILE EXEC, allows you to update the profile for VSESP31 from the VSEMAINT userid even while VSE/SP is up and running. We have made some minor changes to the profile for our own environment; you may want to add other commands and/or users as required for your installation.

```
&CONTROL OFF
*
* All IDS execute the following section of the profile.
*
IDENTIFY (STACK
&READ VARS &USERID &ACCOUNT
&IF .&USERID EQ .VSESP31 &GOTO  -VSESP31
*
* The following section is executed by all users except VSESP31.
*
&HI = Y
&LO = -
CP SET RUN ON
SET RDYMSG SMSG
CP LINK MAINT 319 319 RR ALL
&IF &RC = 0 &GOTO -ACCESS
&TYPE The 319 disk of MAINT can not be linked.
&TYPE Remember the read password of MAINT 319 must be &HI ALL &LO
&EXIT 4
-ACCESS
ACC 319 P
EXEC VSEIPF NOPAN
&EXIT
*
* Only VSESP31 executes the following section.
*
*
-VSESP31
CP SPOOL 00D SYSTEM
CP SPOOL 00E SYSTEM
CP SPOOL 05D SYSTEM
CP SPOOL 05E SYSTEM
&BEGSTACK
*
* At this point you can enter other commands to be executed on behalf
* of VSESP31.
*
&END
CP IPL 540
*
&EXIT
```

Figure 4. VSEMAINT's Profile When the Console is Dedicated to VSESP31

IPLing the VSE/SP Virtual Machine

The following is an overview on the logical flow of events when IPLing the VSE/SP virtual machine shown in Figure 1 on page 10.

The processor console 01F is disabled by the AUTOLOG1 virtual machine. CMS is IPLed and the profile of VSESP31 is executed. As the last statement of the PROFILE EXEC is executed; 540 is IPLed.

After address 540 is IPLed, the IPL routine finds $ASIPROC and selects the procedures specified by the CPUID. The VSE console for VSE operations is dedicated with 01F. The ASIPROC continues without intervention, as if it were running in native VSE.

Following the procedure outlined above, the VSE/SP virtual machine console and the VSE/SP console are on separate devices. Only with the aid of the * CP command (from the dedicated VSE/SP console) and/or commands prefixed with "# CP" (from the VSE/SP virtual machine console) is it possible to communicate with CP. The secondary userid (OPERATOR) specified in the VM directory for VSESP31, will be responsible for handling CP requests for the VSE/SP virtual machine; or you can log on to VSESP31 and issue the CP commands on behalf of the VSE/SP virtual machine.

Stacking CP Commands in the PROFILE EXEC

If you want to automate the IPL of the VSE/SP virtual machine, enter the following line at the end of the PROFILE EXEC. It will be the last line to execute.

```
CP TERM CON 3270 SCRN BRE ON GUEST|DEF STORAGE 16M|IPL cuu
```

From the XEDIT command line (after entering the line above in the exec) type:

```
set hex on
```

and press ENTER. From the XEDIT command line type

```
ch/|/X'15'/* *
```

and press ENTER to change the bar (|) to the equivalent hex code.

From the XEDIT command line type

```
file
```

and press ENTER.

Notes:

1. *The bar (|) could be any other character.*
2. *Anytime the last line is altered, you should perform the last three steps again.*

Chapter 3. More Topics on Operating a VSE/SP Virtual Machine

Using Virtual Addressability Extension (VAE)

VAE stands for Virtual Addressability Extension. It increases the virtual storage size of previous VSE releases from 16 Mb to 40 Mb. It does this by supporting up to three address spaces instead of one. Any address space can have a size up to 16 Mb as long as the total virtual storage size of the VSE/SP system is not more than 40 Mb. The extended storage layout includes shared system areas, shared subsystem partitions, and private partitions.

Conceptually, VAE is up to three copies of the total virtual storage (called spaces) and the supervisor selects which space to use. A non-VAE system is like a VAE system with only one space.

When a space is being used, it must have the supervisor and SVA in order to run. The concept is that each space has its own copy of the supervisor and SVA. However, there is only one SVA and one supervisor, which are *shared* between the spaces.

What Supervisor Should I Use with VAE?

When using the VAE option with VSE/SP, you must use the $$A$SUP3 supervisor for MODE=370. This means only a subset of the VM/VSE Handshaking facilities are available. They include:

- SET PAGEX ON
- CPCOM
- BTAM Autopoll Assist
- MODE=370 also implies double paging by VM/SP and VSE/SP. and double CCW translation unless you run in a V=R machine.

Should I Run VAE as a V=R Guest or a V=V Guest?

Run VAE as a V=R guest for a production system or whenever system performance is your main concern. As a V=R guest, VSE/SP does all paging and CCW translation itself without incurring overhead from VM. Another performance gain for a V=R guest is available by turning off shadow table support.

Run VAE as a V=V guest when you are testing a VAE system while other VSE production systems are running.

Should I Run One MODE=370 VAE Guest or Multiple MODE=VM Guests?

There are advantages and disadvantages with each mode. Consult the following table for an overview.

Option	Advantages	Disadvantages
Single MODE=370 VAE Guests	• A single VSE system to manage (one console) • Reduced Lock Manager overhead and a smaller working set size because many VSE systems are combined into one • If V=R, paging and CCW translation done only by VSE • Full VSAM Share Option 4 • No VSE Lock File (no DASD sharing).	• If V=R, loss of storage for paging can have negative effect on other users • If V=V, double paging and CCW translation • Usually have to re-IPL the VM system to alter it • Limited VM/VSE handshaking.
Multiple MODE=VM Guests	• Full advantage of VM/VSE handshaking support • VMCF available • VCNA can be used (IUCV supported).	• Extra overhead and DASD sharing requirements • VSE Lock manager overhead • Because of nonshared code, much duplicate code is needed (supervisor, SVA, POWER, CICS, etc.) • More VSE consoles are needed.

Figure 5. One MODE=370 VAE Guest or Multiple MODE=VM Guests?

4K Paging Support for VSE/SP Guests under VM

VSE will always use a 4K page size when running under VM, whether in MODE=370 or MODE=VM. This permits operation of VSE in MODE=370 on processors that support only 4K paging, like the 3084.

Operating the VSE/SP Virtual Machine

Autologging the VSE/SP Virtual Machine

AUTOLOG is a convenient way to initiate large production VSE systems with many I/O devices running under VM. The I/O devices needed by the VSE system require considerable contiguous free storage space for the I/O control blocks established by VM. If smaller users have logged onto VM before the large operating VSE system is started, there may be insufficient contiguous free storage space available for the required I/O control blocks. (The logon of the virtual machine will still be completed even if the I/O control blocks cannot be established.) As a result, there may be an insufficient number of I/O devices to run the guest VSE system and its application programs.

To ensure sufficient contiguous free storage space for a large production VSE system, the virtual machine should be logged on immediately after VM is loaded. This can be done by:

- Having the VM system operator issue the CP AUTOLOG command before enabling user terminals, or by
- Defining the AUTOLOG1 virtual machine in the VM directory. The AUTOLOG1 virtual machine is automatically logged on immediately after VM is loaded and can be used to log on and load virtual machines that require substantial contiguous storage.

Operator Issuing CP AUTOLOG Command

Before enabling user terminals, the VM system operator can issue the CP AUTOLOG command for each production guest virtual machine that requires substantial contiguous free storage. The virtual machine being logged on with the AUTOLOG command must have an automatic IPL defined in its directory and is allowed to issue one read to its virtual console. The virtual machine logged on operates in disconnected mode. The same restraints that apply to any disconnected machine also apply to virtual machines logged on with the AUTOLOG command. To invoke the command, the operator would issue:

```
AUTOLOG userid password
```

For more information about the format of the CP AUTOLOG command, refer to the *CP Command Reference*.

Defining AUTOLOG1 in the Directory

To use AUTOLOG1 to initiate several virtual machines, the VM directory statement loads CMS into the AUTOLOG1 virtual machine. In the PROFILE EXEC, each CP AUTOLOG command initiates one virtual machine containing a VSE guest operating system. When using the CP AUTOLOG command, the directory entries for the virtual machine referred by the CP AUTOLOG command must contain an IPL statement.

As a result of the CP AUTOLOG command in the PROFILE EXEC, the VSE virtual machine is loaded and runs in disconnected mode. The VSE virtual machine user gains access to the virtual machine by doing one of the following:

- Logging on with the userid specified in the CP AUTOLOG command
- Issuing the CP SEND command from the secondary user's console
- Issuing the CP DIAL command and specifying the guest userid.

When you log off, the contiguous storage space is relinquished and the virtual machine relinquishes operation. If you want to keep the virtual machine's I/O blocks in contiguous storage, and keep the virtual machine running while you temporarily give up use of the virtual machine console, issue the CP DISCONN command. To reestablish usage, issue the CP LOGON command to reconnect to the virtual machine.

Figure 6 shows the AUTOLOG1 entry in the VM directory. The IPL statement initializes CMS, causing the execution of the PROFILE EXEC. The VSE virtual machines are automatically logged on in disconnect mode.

```
USER AUTOLOG1 PASSWORD 512K 1M ABG
     ACCOUNT ACCTNO BIN1
     IPL CMS
          CONSOLE 009 3215
          SPOOL 00C 2540 R
          SPOOL 00D 2540 P
          SPOOL 00E 1403
          LINK MAINT 190 190 RR
          LINK MAINT 19E 19E RR
          LINK MAINT 19D 19D RR
          MDISK 191 3330 1 1 UDISKA WR RPASS WPASS
```

Figure 6. AUTOLOG1 Entry in VM Directory for VSE/SP Virtual Machines

Figure 7 shows the PROFILE EXEC containing several CP AUTOLOG commands, one for each virtual machine to be loaded. The last CP command in the PROFILE EXEC logs off AUTOLOG1.

```
/*PROFILE EXEC for AUTOLOGing virtual machine */
    TRACE E; ADDRESS COMMAND;
    CP SPOOL CONSOLE START; CP SET EMSG ON;
/*The following message will inform the operator that the guest */
/*operating systems are being autologged.                       */
    CP MSG OP The guest VSE virtual machines are being autologged.
    CP AUTOLOG VSESP31 VSESP31;
    CP AUTOLOG VSEUSER PASSVSE2;
    CP ENABLE ALL
    CP DISABLE 01F
    CP LOGOFF
    EXIT
```

Figure 7. PROFILE EXEC Containing Several CP AUTOLOG Commands

You can now access these virtual machines through the secondary console (if there is one), or by logging on with userid VSESP31 or VSEUSER2, along with the appropriate password.

Automated System Initialization (ASI) Procedures

The ASI procedure allows you to place the required control information into a procedure cataloged in IJSYSRS.SYSLIB and lets the system execute the commands without operator intervention each time an IPL or a partition start-up occurs.

When VSE/SP is running under VM, a few changes have to be made to the $ASIPROC (if you have one). To indicate that a virtual machine is running under VM the CPUID must have a prefix of FF. For example:

```
CPU=FF0200064331
```

Where:

The first two digits are the prefix
The next six digits are the CPUID
The last four digits are the processor type.

If the CPUID of a virtual machine is equal (except for the first two digits) to another CPUID in the master procedure, the entry for the virtual machine must be the first entry within the master procedure as shown below:

```
CPU=FF0020004331,IPL=$IPLV,JCL=$$JCLH,MODE=370
CPU=020020004331,IPL=$IPLN,JCL=$$JCLH,MODE=E
```

To allow an IPL of both native VSE and VSE under VM, catalog the $ASIPROC with two entries in it. One entry is for the IPL of the native VSE and the other for the IPL of VSE under VM. To do this, duplicate the entry for running VSE native and change:

- Prefix 02 to FF
- The mode option to MODE=370 (unless you are already running MODE=370)
- The IPL procedure name to the name referred by the CPUID with prefix FF

Next, if you are running MODE=VM, drop DPD. (VM does the paging.) Then drop the following parameter from the SYS command:

- BUFSIZE (VM will do all CCW translations.)

The ASI procedures can be reused as long as your system environment remains unchanged. The steps to bringing up a total system is reduced to IPling VM. In exceptional situations, you may have to bypass ASI and perform an interactive system initialization (non-automated).

Note: If you wish to interrupt the execution of the VSE ASIPROC to specify a different supervisor you need to log onto the VM console and issue the CP EXTERNAL command. Issuing this command simulates pressing the interrupt key on the real computer console, or other functions that cause an external interrupt. Control is given to the virtual machine immediately. The format of the EXTERNAL command is:

```
#cp ext
```

The external interrupt must be entered before the ASIPROC has been initialized.

Using EXEC Procedures

An EXEC procedure is a sequence of commands and other statements that can be processed by CMS. Although certain format and syntax rules apply, the process is the same as when you create any other type of CMS file.

Once an EXEC procedure is stored on direct access storage, the procedure may be executed simply by issuing the name of the EXEC file. The only exception to this rule is the PROFILE EXEC file. The PROFILE EXEC file is executed automatically when you IPL CMS. For detailed information on the use and format of EXEC files refer to the *CMS User's Guide.*

For the VSE/SP virtual machine, a PROFILE EXEC could be used to establish the proper links and SET commands. For example, you can issue SET FAVOR, SET QUEUE DROP, and SET PRIORITY from a PROFILE EXEC. (To avoid reserving routines in storage that are used only once, such as the initialization routine, do not issue the CP SET RESERVE command from a PROFILE EXEC.) Because the PROFILE EXEC runs automatically, you are relieved of the chore of entering these commands. The following is an example of a PROFILE EXEC that can be used to set up the CP commands mentioned above:

```
PROFILE EXEC     A1  F 80  TRUNC=132 SIZE=5  LINE=5 COL=1 ALT=0

===== &TRACE OFF
===== CP SET FAVOR
===== CP SET FAVOR 80
===== CP SET QUEUE DROP OFF
===== CP SET PRIORITY 5
```

Issuing CP Commands from the VSE/SP Virtual Machine

While operating the VSE virtual machine, use CP commands to:

- Communicate with the VM system operator or other virtual machine users.
- Query the status of virtual machine devices or spool files.
- Attach or detach devices from the virtual machine configuration.

You can communicate with CP by:

- Prefacing the CP command with "# CP" where "#" is the terminal linend character.

Note: At times, CP commands cannot be entered with the #CP function. For example, during the IPL procedure when VSE is processing IPL commands, the CCWs used for these reads expect only three bytes of data. Any additional information on CP command lines are truncated.

Notes:

1. *You can define the terminal break key by issuing the CP BRKKEY command. PA1 is the default. You must be sure that the PF key associated with this function is available on the VSE virtual machine console. Otherwise, choosing a PF key (to activate the breakin function) that does not exist on the VSE virtual machine console hardware will lock out CP mode. Once the break key has been pressed the virtual machine stops. If the SET RUN ON command* **is not in effect** *and you wish to return to the VSE virtual machine console, you must issue the CP BEGIN COMMAND. If the CP SET RUN ON command* **is in effect** *it allows you to activate the attention key (causing a read of a CP command) without stopping your virtual machine. When the CP command is entered, it is immediately executed and the virtual machine resumes execution.*

2. *If you have a local 3270 information Display device and your VSE console is in display mode, it is recommended that you issue the CP TERMINAL SCRNSAVE ON and CP TERMINAL CONMODE 3270 commands for your machine. This allows you to choose whether the guest screen is saved when your virtual machine goes into CP mode. You can enter the two commands together, for example:*

```
cp terminal conmode 3270 scrnsave on
```

Interrupting the ASIPROC under VM

While VSE is IPLing in the virtual machine, you can interrupt its execution by issuing the CP EXT command at any point prior to seeing the following message:

```
0J10I IPL RESTART POINT BYPASSED
```

Once the above message appears the ASIPROC cannot be interrupted.

To resume IPL after a successful interruption, enter a null line. Wait until the attention has been processed before signaling another one.

Various Uses of the DEDICATE Statement

The DEDICATE control statement specifies that a real device is to be dedicated to this userid. A real device can be dedicated to only one user at a time. In the sample directory entry in Figure 1 on page 10, we gave an example of how to dedicate the main processor console to the VSE virtual machine as the VSE operator's console. We will now discuss other unique uses of the DEDICATE control statement. •

Note: You can also use the CP ATTACH command to perform the following functions.

Magnetic Tapes

A device such as a magnetic tape drive can be used by only one virtual machine at a time. You can dedicate the tape drive if only one virtual machine will be using it (for example, for CICS logging), or different users can use it in turns by issuing the CP ATTACH command.

To dedicate the tape drive to one user, specify it in that user's directory. For example:

```
DEDICATE 181 281
```

This statement allows the operating system to access the device at real address 281 via a virtual address of 181.

Unit Record Devices

In many cases, spooling represents the most efficient way of handling the unit record input and output of many virtual machines. However, special cases may justify the dedication of a real unit record device to a single virtual machine.

One special case is when the virtual machine's operating system does its own spooling, such as VSE/POWER under VSE. To eliminate double spooling of printer output, include a DEDICATE statement in the virtual machine's directory entry, such as:

```
DEDICATE 00E 002
```

This statement causes VM to pass all virtual printer 00E output directly to the real printer at 002.

Note: If you dedicate unit record devices to a guest, then you don't need a spool entry for that same device.

Remote Devices

You can use the DEDICATE statement to attach remote 3270 Information Display Printers (such as 3262, 3268, 3284, 3286, 3287, and 3289) to a virtual machine. For example, a directory entry can include the statement:

```
DEDICATE NETwork 120 0102
```

120 is the virtual address of the device in the virtual machine and 0102 is the resource ID as specified in the DMKRIO. Remote 3270 Information Display System Printers can also be attached by the NETWORK ATTACH command. For more details, see the *Planning Guide and Reference.*

Unsupported Devices

You can use the DEDICATE statement to place a device that VM does not support into a virtual machine configuration. To dedicate a device, the device must:

- Be physically connected to the VM system
- Be supported by the VSE operating system
- Not violate any of the restrictions contained in the VM restriction section of the *Planning Guide and Reference.*

For example, a directory entry can include the statement:

```
DEDICATE 007 012
```

where real address 012 could represent a real device that is unsupported by VM but is attached to the processor.

Dedicated Terminal Definitions

When the end user is going to use CICS, the terminal can be dedicated to the VSE system. To ensure that the end user gains access to only the VSE system, the DEDICATE control statement must be in the directory or the terminal should be attached to the VSE virtual machine by an authorized user. The terminal for this end user must be disabled before DEDICATED control statements and CP ATTACH commands are executed. By having an entry for the end user in the AUTOLOG1 PROFILE EXEC, the terminal for the end user can be disabled before the IPL of the VSE virtual machine.

The following is an example of a DEDICATE statement entry that can be used in the VM directory for the VSEUSER virtual machine.

```
VMUSERS DIRECT    A1  F 80  TRUNC=72 SIZE=55 LINE=9 COLUMN

===== ************************************************************
===== *                      SYSTEM          USERIDS
===== ************************************************************
===== USER VSEUSER VSEUSER 16M 16M ABDG
===== ACCOUNT 206 7030/85
===== IPL CMS
===== OPTION ECMODE BMX 370E REALTIMER CPUID 000001
===== CONSOLE 009 3215 T OPERATOR
===== SPECIAL 012 3270
===== SPECIAL 013 3270
===== *
===== * The following statement dedicates the console to VSEUSER after
===== * the execution of AUTOLOG1 disables it.
===== *
===== *
===== DEDICATE 01F 01F
===== *
===== * In this VM directory example we are defining device type 2540
===== * to be the virtual reader.
===== *
===== SPOOL 00C 2540 READER A
===== SPOOL 00D 3525 A
===== SPOOL 00E 1403 A
===== SPOOL 02C 2540 R A
===== SPOOL 02D 3525 A
===== SPOOL 02E 3203 A
===== LINK MAINT 190 190 RR
===== LINK MAINT 19D 19D RR
===== LINK MAINT 19E 19E RR
===== *
===== *
===== MDISK 240 FB-512 000000 558000 DOSRES MWV VSESP31 VSESP31
===== MDISK 241 FB-512 000000 558000 SYSWK1 MWV VSESP31 VSESP31
```

Figure 8. VM Directory for VSEUSER with Dedicated Console

The next example is the PROFILE EXEC of AUTOLOG1 where 01F is being disabled before IPLing VSESP31.

```
===== &CONTROL OFF
===== CP SLEEP 10 SEC
===== CP AUTOLOG CMSBATCH CMSBATCH IPL CMS PARM BATCH
  .
  .
  .
  .
===== CP DISABLE 01F
===== CP AUTOLOG VSESP31 VSESP31
===== *
===== CP LOGOFF
```

Figure 9. PROFILE EXEC of AUTOLOG1 for Use with a Dedicated VSE Console

Nondedicated Terminal Definitions

If the end user needs the facilities of both CMS and CICS, a SPECIAL control statement must be included in the directory of the VSE virtual machine. The SPECIAL statement must be included for each terminal needing access to CMS and CICS. With the SPECIAL statement in the directory, the CP DIAL command can be used to gain access to the VSE system.

For example, if the VM directory entry for the VSE virtual machine had the following SPECIAL statement:

```
SPECIAL 012 3270
```

You could issue:

```
dial vseuser 012
```

----- or -----

```
dial vseuser
```

If you issue the CP DIAL command without a specified address, VM connects the terminal to the first available line as defined in the SPECIAL control statement; the line belongs to the specified userid. If no lines are available or if all lines are busy, VM issues an error message and does not make the connection.

The end user will remain connected until one of the following happens:

- The virtual machine logs off using standard logoff procedure
- The virtual machine is forcibly logged off
- The terminal is powered off/on
- The Normal/Test switch is used.

The end user will also get disconnected if the CP RESET command is issued from the VSE virtual console or from a user authorized with the CP RESET command.

Once disconnected, the end user is free to use the DIAL command to connect to another userid.

Spooling Options

Most multiprogramming operating systems have their own spooling subsystem. In VSE/SP this subsystem is VSE/POWER. Because VM also provides its own spooling, double spooling will occur. This raises the questions of whether an installation should:

- Use only the operating system's spooling subsystem
- Use only VM's spooling
- Use double spooling.

Spooling Recommendations

If an installation has a significant amount of printing or punching to do, it might appear that one of the spooling subsystems should be eliminated. This is not necessarily true.

Use double spooling if:

- You have large quantities of printed output on standard forms, or
- DASD space is not a limiting factor.

Many VSE virtual machines spool data and must use a common pool of unit record devices. In this case double spooling will reduce the privilege operations VM must simulate.

For VM to do the spooling for the VSE virtual machine, the spool statement must exist in the VM directory. This statement must match the ADD statement in the VSE ASI procedure.

Note: If possible, generate a scaled-down version of the virtual machine's spooling subsystem, eliminating those functions not used by that virtual machine. Make the I/O buffer sizes as large as possible to cut down on SIO instructions.

Let the VSE virtual machine do the spooling if:

- An installation has only enough DASD spooling space for one spooling subsystem, and
- If only one VSE virtual machine generates significant amounts of spooled output.

Note: If VSE controls the spooling by either attaching or dedicating unit record devices, you do not need a duplicate SPOOL statement in the VM system directory to define the device.

VSE/POWER under VM

The VSE/POWER operation remains the same in the VM/VSE environment. VSE/POWER makes no distinction between real or virtual devices and it executes input and output regardless of the devices used. It is advisable that special forms be printed on a dedicated printer to VSE.

Controlling Printed Output

Most of the VSE supported printers use a forms control buffer (FCB) to control the length of forms skips. In addition, some printers can be equipped with the universal character set feature that is controlled by the Universal Character Set Buffer (UCB). The following two discussions will describe the effects of loading FCB and UCB while VSE is running under VM.

Loading Universal Character Set Buffer

You can load UCB from either VM or VSE. If you are going to load the UCB under VSE the printer needs to be attached to the VSE machine. The UCB can be automatically loaded at IPL time or the LUCB command can be used to load the buffer after VSE/SP has been IPLed. For example, the command:

```
lucb 00e,$$bucb00,fold
```

will load UCB $$BUCB00 on the printer at 00e, and FOLD will translate lower case to upper case. There are several default UCB's provided by VSE/SP and are documented in *VSE/SP Installation.*

If you want to load the UCB under VM, use the LOADBUF command:

```
loadbuf 00e ucs,p64,fold
```

This command loads the UCB P64 to the printer at address 00E and the FOLD option translates lower case to upper case when printing. The buffer can be loaded at IPL time by adding the above command to AUTOLOG1's PROFILE EXEC.

Note: The printer must be drained prior to loading any buffer under VM.

```
drain 00e
```

Loading Forms Control Buffer

There are two recommended ways to use a printer with VSE/SP:

- Dedicate the printer to the VSE guest and start a POWER print writer without the VM parameter. In this case, POWER will load the FCB.
- Share the printer between the VSE guest(s) and VM users. In this case, you should start the POWER print writer with the VM parameter. POWER will send the FCB as part of the print file. POWER will also pass to VM the forms ID and number of copies. You do not need to load the FCBs for any POWER output.

If you want to load the FCB under VM, you must use the LOADBUF command:

```
loadvfb 00e fcb,fcb1
```

The LOADVFB command can be used in installations that do not have a 3203, 3211, 3262, 3289E, or 4245 printer. The virtual machine's directory entry must indicate a 3203, 3211, 3262, 3289E, or 4245 even though both the program and operating system have a 1403 printer defined. Then the LOADVFCB command can be used to specify a virtual FCB image for 1403 printers so that programs that use printer overflow sensing can be spooled to disk.

This command loads the FCB FCB1 on the printer at address 00E. You will find a list of the FCBs available with VM in *CP for System Programming.*

If you want to use a special FCB, you must have two identical FCB with the same name - one created under VSE and the other loaded onto the VM nucleus. You can confirm that the FCB is working by dedicating your printer to VM.

Details on how to load the FCB and UCB in VSE/SP can be found in *VSE/SP Installation.*

Printer Considerations for the 3203-5

When you run VSE/SP under VM, you must define 3203-5 printers as 3211 printers. Catalog an FCB appropriate for a 3203-5, but use 3211 naming conventions. Otherwise, the virtual printer will be sensed as a 3203-1 and the wrong FCB will be loaded.

Starting the VSE/POWER Printer

When you run VSE/SP under VM, the address you use to start a VSE/POWER printer must be a *virtual device address.*

VSE/POWER will load the correct FCB.

Use the following command to start the printer:

```
s 1st,cuu,a,,vm
```

This starts a list-writer task to print spooled output to the virtual printer with address *cuu.* The *vm* operand tells VSE/POWER that the device is a virtual device owned by VM.

When *,,vm* is specified, VSE/POWER:

- Issues NO FORMS CHANGE messages
- Loads FCB
- Spools to the specified userid (DEST = (,userid)
- Prints/punches only one copy
- Tells VM about Class, Forms name, Copies
- Closes the device at end of entry.

For more information, refer to *VSE/SP Installation.*

Varying Devices Offline/Online

Once you IPL the virtual machine, the devices that were not accessible to that machine at IPL time are considered offline. However, the operator can attach more devices to this machine and have them placed online as required. The operator can issue the CP VARY and CP ATTACH commands to make the devices available for use by a particular virtual machine.

For example, if a graphic device is offline and VSEUSER needs the graphic device to be made available, notify the operator via a message:

```
#cp msg op Please attach 080 to VSEUSER as 080
```

The operator would issue:

```
vary online 080
```

SYSTEM RESPONSE:

```
080 VARIED ONLINE
```

Operator issues:

```
attach 080 to vseuser 080
```

SYSTEM RESPONSE:

```
080 ATTACHED
```

The operator informs VSEUSER that the graphic device is now attached and ready for use.

Switching Devices between Systems

It is possible to switch devices such as tape drives and printers between VM users. For example, you might have a VM system that has a VSE virtual machine user (VSEUSER) and a CMS user (CMSUSER). At different times, each system might have the need to use a tape drive and printer. Because a tape drive cannot be shared, the device needs to be attached to one user at a time. If CMSUSER has only class G privileges, a message must be sent to the operator to attach 181 to CMSUSER. For example:

```
#cp msg op Please attach 181 to CMSUSER as 181
```

When the tape drive is attached, the operator would notify CMSUSER with the message:

```
ATTACHED 181 TO CMSUSER AS 181
```

CMSUSER will have the tape drive for as long as needed. When CMSUSER is finished using the tape drive, issue the following command.

```
#cp detach 181
```

As soon as the tape drive is detached, it is available to other users via the ATTACH command.

The printer can be set up in two ways:

- VM does all the printing for its users, or
- The VSE system does its own printing.

For a VSE virtual machine to do its own printing, the printer has to be attached to the VSE userid via the CP ATTACH command. (A userid with class G privileges cannot issue the CP ATTACH command. A message must be sent to the operator as shown in the example above.)

Note: If VM was previously doing the printing, the printer must be drained prior to attaching it to the VSE virtual machine. The CP DRAIN command brings the spooling system to a controlled halt, or halts the activities on a device whose spooling status is to be changed. The operator would issue:

```
drain 00e
```

The CP ATTACH command can now be issued:

```
attach 00e vseuser 00e
```

The printer is now available to VSEUSER for as long as needed. When the VSE virtual machine no longer needs the printer, the printer must be stopped in the VSE machine before detaching it. Issue the POWER command:

```
pstop 00e
```

and issue the CP command:

```
detach 00e vseuser
```

Definition and Use of the Virtual Console Facility

To use a 3270 display terminal as the primary console in display operator console mode, either have a SPECIAL statement in the VSE virtual machine's VM directory entry or issue the CP DEFINE GRAF command after logon to VM.

- If the SPECIAL statement is used, it would appear in the directory as:

```
SPECIAL 01f 3270
```

- If the SPECIAL statement is not used, assume that a local 3270 line has been enabled by the VM operator. Next, issue the following CP DEFINE command:

```
define graf 01f 3270
```

In either case, after you log onto VM (by using the device specified in the CONSOLE statement) and load the operating system into the virtual machine (by using the IPL command), you must issue the CP DIAL command at the 3270 console that is to be used in display mode. This action logically connects that 3270 console to the operating system.

The CP TERMINAL CONMODE 3270 command can also be used to obtain a display mode console for the VSE virtual machine. The console of the VSE virtual machine is defined in the VM directory as:

```
CONS 01F 3215 T
```

The following command allows both VM and VSE operator actions from the same screen. Before IPLing the VSE machine you have to define the type of console operation by issuing the CP command:

```
term conmode 3270
```

Note: If CMS is running when you issue the CP TERM CONMODE command, CMS will abend.

In addition, specify:

```
terminal scrnsave on
```

This command saves the screen before going into CP mode. It is valid only if TERMINAL CONMODE 3270 has been specified. When you return from CP mode, the VSE screen is automatically displayed as it appeared before you entered CP mode.

If you issue:

```
terminal breakin guestctl
```

this command prevents CP messages from appearing on the VSE console and gives the operator the impression of running stand-alone. CP messages will be displayed only when:

- Priority messages are to be displayed.
- The CP function is requested.

Notes:

1. *The CP TERMINAL CONMODE 3270 command is not supported for 3270 terminals going through a VTAM service machine or for remote 3270's.*

2. *The DIAL function is not supported for terminals controlled by a VTAM service machine through VCNA.*

Special Considerations for VSE/SP Users Running Under VM

When you use the VSE/SP *Display System Activity* and the *Display Channel and Device Activity* dialogs to monitor system activity, remember that:

- The SIO/second rate from *Display System Activity* may seem unusually high. The dialog calculates the total I/O rate on the system, including unit record virtual I/O. To specifically monitor disk or tape device activity, use the *Display Channel and Device Activity* dialog.

- VSE accounting support is used. This implies that VM simulates some privileged instructions, but the data gives you a better overview of the VSE/SP guest virtual machine.
- All data is valid, *except* for data displayed as the number of events per second (for example, SIO/second). This type of data describes only VSE activity.
- When the VSE/SP guest is running MODE=VM, the dialog displays zeros for paging activity. This shows you that only VM handles paging.
- When the VSE/SP guest is running MODE=370, the paging activity data reflects VSE paging.

Defining CPUIDs for the VSE/SP Virtual Machine

If you are in a DASD sharing environment, you have the lock files that keep track of the resources the systems are using. Part of that resource lock is a unique CPUID. If you don't specify a unique CPUID, they will all default to the real system CPUID. If this happens, the resource lock will not have the correct ID for the system you want it to have.

By default, each virtual machine running under control of VM will have a processor identification as follows:

FFbbbbbbccccdddd

Where:

FF identifying it as a virtual machine running under control of VM

bbbbbb real/virtual CPUID

cccc processor type (e.g., 4361 or 4381)

dddd all zeroes

Because the six-digit processor identification number is stored in the VSE Lock File, and used to control DASD sharing and system startup, the CPUID must be unique for each of the VSE/SP machines. A maximum of 31 sharing CPUIDs are permitted. This allows a maximum of 31 VSE/SP virtual machines sharing the same Lock File.

The CPUID may be set by the CP SET CPUID command and queried using the CP QUERY CPUID command. Only the six hexadecimal digits containing the CPUID number may be set. Both CP SET CPUID and CP QUERY CPUID are class G commands available to the general user. For more information on these commands refer to the *CP Command Reference for General Users.*

The CPUID may also be specified in the VM directory OPTION statement. For example,

```
===== OPTION .... CPUID 000001
```

The above entry in the VM directory for the VSE/SP virtual machine will ensure that the CPUID for this virtual machine will always be set correctly. Also, the CPUID may then be used in the UNLOCK command and the master ASI procedure.

The data set name constructed by VSE/AF for the Label Area includes the first 12 digits of the CPUID. In the case where the same VSE system is run both natively and under control of VM, the data set name will differ in the first two digits (FF instead of the real processor model number).

You should ensure that the JCL ASI procedure loads the label area every time, because you will have to delete the other label areas when switching between real and virtual VSE machines.

Submitting Jobs to the VSE/SP Virtual Machine

There are two ways of submitting jobs to the VSE/SP virtual machine for execution. You can prepare the JCL under CMS using XEDIT and submit it to the VSE virtual machine for execution, or you can use the SUBVSE EXEC supplied as part of the VM/VSE Interface of VSE/SP.

Submitting Jobs under CMS

If you are under CMS and you wish to send a job to the VSE virtual machine for execution, create the job stream (JCL) using XEDIT or the editor of your choice. Before using the virtual punch to punch jobs to the virtual machine, take the precaution of clearing any files that remain in it from previous jobs. The following command ensures that the virtual punch does not have any other punch files in it:

```
spool punch nocont purge
```

Now you can spool your punch to VSE virtual machine. In our example it will be VSESP31.

```
spool punch vsesp31 cl n
```

Where **n** is the spool class of the VSESP31 virtual reader. The class can be changed by issuing the CP SPOOL command on the VSESP31 virtual machine. The default class is A.

You can now issue the CMS PUNCH command.

```
punch filename filetype filemode (noh
```

This sends the job stream to the VSE virtual machine. A job stream spooled to the VSE virtual machine remains in the virtual reader until it is instructed to begin reading the job stream. From the VSE console you must issue the POWER reader task start to the virtual reader:

```
pstart rdr,cuu
```

The virtual device address must match the one specified in the directory entry for VSE/SP virtual machine.

Submitting Jobs Using SUBVSE EXEC

All jobs should be submitted to a VSE/SP virtual machine by using the SUBVSE EXEC.

The format of the SUBVSE EXEC is:

```
SUBVSE fn ft fm TO user1 FOR user2 ECHO option
```

where:

user1 is the virtual machine id the job is sent to
user2 is the virtual machine id the echo is sent to
ECHO *option* is REPLY|YES|NO|JOB

Optionally, you can enter just the command SUBVSE and fill in the blanks on the panel that is displayed.

When you use SUBVSE, it is your responsibility to add the PDEST/LDEST parameter to the POWER JOB card or add the DEST parameter to the POWER LST/PUN cards.

After you use the SUBVSE EXEC, use the VSEREP module to respond to the outstanding requests you will receive. For example:

```
vserep vsesp31 0
```

Note: This command works only if you are using a MODE= VM supervisor.

Transferring Output with the VM Writer Task of VSE/POWER

The VM writer task transfers output created in VSE/SP to the CMS user. It is a standard feature of VSE/POWER 2.3.

For the VM writer task to work, you must use either the DEST parameter on the POWER LST/PUN cards or the PDEST/LDEST parameters on the POWER JOB card. When you do this, you can transfer back to the originating CMS userid all print and punch output of jobs submitted through SUBVSE.

Notes:

1. *If an ICCF user has a VM userid with the same name as one of the DEST parameters, the VM user will receive the output from the VM writer task.*

2. *If you start a printer or punch task without the VM parameter, a POWER print writer will issue the CP CLOSE command. at the end of each file being spooled.*

Example of the VM Writer Task

The VM writer task returns to the CMS userid *user2* the output created provided that:

- The CMS userid *user2* exists on the VM system, and
- A virtual printer is started with the same class as specified on the POWER LST/PUN cards and with the VM parameter.
- The DEST (or LDEST or PDEST) for the output must be the same as the CMS userid.

You can issue the PSTART command from the VSE/SP console, or from the CMS userid using VMCF as follows:

```
vsecmd vsesp31 pstart lst,05e,v,,vm
```

Initializing Minidisks

VSE/SP always uses DOSRES and SYSWK1 as the volume IDs for its system disks. If you want to run several VSE/SP guest machines under VM that do not share DASDs, you will have multiple DOSRES and SYSWK1.

VM, however, does not accept duplicate volume IDs on real disks. To solve this problem, you must have two volume IDs on such disks:

- A unique volume ID on the real disk used by VM
- The label DOSRES or SYSWK1 on a minidisk.

To set this up:

1. Use the information in *CP for System Programming* to initialize and format the DASDs that will contain the minidisks. This creates the unique volume labels required by VM. For CKD devices, the volume label and allocation byte map are on cylinder 0. For FBA devices, the volume label and VTOC are part of the first 16 blocks.

2. In the directory entries for each VSE/SP guest machine, define minidisks on the DASDs you initialized and formatted. *Do not* define minidisks that start on block or cylinder 0. For CKD devices, minidisks can begin at cylinder 1. For FBA devices, they can begin at a MAX-CA boundary following the initial blocks reserved for use by VM.

3. Initialize the minidisks for each VSE/SP guest system. To do this,

   a. Log on to VM using the user ID and password for each VSE/SP system.

   b. Use DSF to initialize the minidisks as shown in the following two examples. Note that these definitions do not affect the VM information that you created in step 1.

Case 1: Using DSF for an FBA Device (3370-2)

```
INIT UNIT(cuu) NVFY NOMAP PURGE FBAVTOC(711946,99,1024)
VOLID(xxxxxx)
```

Case 2: Using DSF for a CKD Device (3350)

```
INIT UNIT(cuu) NVFY PURGE MIMIC(MINI(554) DEVTYPE(3350)
DVTOC(553,0,30) VOLID(xxxxxx)
```

Notes:

1. *If the above INIT commands are too long to fit into a single line on your screen, use a dash (-) as the continuation character. The system will then prompt you for additional information.*

2. *You cannot use these DASDs when running VSE/SP in native mode.*

VM/VSE Interface

The VM/VSE interface provided by VSE/SP Versions 2 and 3 is a set of VSE phases and CMS modules. The interface routines let CMS users operate VSE/SP systems. These routines will work only if you are using a **MODE = VM supervisor**.

When you operate a VSE/SP system this way, you can:

- **Submit jobs** from a CMS terminal to a VSE/SP virtual machine and have messages from the job be echoed to a specific job owner (CMS userid).

- **Execute CP commands within JCL statements** and have the resulting CP messages routed to the CMS job owner.

- **Retrieve** up to twenty of the most recent messages from the console of a VSE/SP virtual machine.

- **Reply to messages** that result from the execution of a job. The job must have a unique job owner (CMS userid).

- **Issue VSE/SP commands** to a VSE virtual machine and have the resulting AR (Attention Routine) messages echoed to the CMS user.
- **Issue CP commands** for execution in the VSE/SP virtual machine and have the resulting CP messages routed to the CMS job owner.

The VM/VSE interface routines are distributed in IJSYSRS.SYSLIB. You must obtain these routines from the library and install them onto a CMS minidisk.

Installing the VM/VSE Interface

Before you can use the VM/VSE interface, you must distribute the following CMS modules and the related EXPLAIN files to all CMS users who are authorized to use the appropriate function. The installation process is described in *VSE/SP Installation.*

Modules and EXPLAIN Files for the VM/VSE Interface

CMS File Name (fn)	CMS File Type (ft)	VSE Library Book Name	Function
		$VMCF.PHASE	VM/VSE Interface processing routines.
		$VMCFOPN.PHASE	VM/VSE Interface initialization routines.
VSEREP	MODULE	VSEREP.Z	Reply to outstanding messages.
VSEMSG	MODULE	VSEMSG.Z	Retrieve messages from VSE/SP system.
VSECMD	MODULE	VSECMD.Z	Execute VSE commands on virtual VSE/SP system.
VSECP	MODULE	VSECP.Z	Execute CP commands on virtual VSE/SP system.
VSEREP	EXPLAIN	EXPREP.Z	VSEREP command HELP panel.
VSEMSG	EXPLAIN	EXPMSG.Z	VSEMSG command HELP panel.
VSECMD	EXPLAIN	EXPCMD.Z	VSECMD command HELP panel.
VSECP	EXPLAIN	EXPCP.Z	VSECP command HELP panel.
SUBVSE	EXEC	SUBVSE.Z	Submit a job for execution on a virtual VSE/SP system.

Notes:

1. *Be careful about your control of VSECMD and VSECP—they are intended mainly for the system administrator. The other modules and EXPLAIN files can reside on a disk to which all CMS users have access.*

2. *See VSE/SP Installation for information on the SKVMVSE skeleton in the ICCF library 59. You use this skeleton to punch modules, EXPLAINS, and EXECs from VSE to VM.*

   If the VM/VSE interface is activated during IPL and the console is defined as TERM CONMODE 3270, do not use the VSE command SET HC=CREATE.

Overview of VM/VSE Interface Routines

Function	Used In	Target Environment	Comments
VSECP	CMS	VSE/SP virtual machine	All CP commands allowed
CPCMD	VSE	VSE/SP virtual machine	Some CP commands allowed
* CP	VSE	VSE/SP virtual machine	All CP commands allowed
* CP DISCONNECT	VSE	VSE/SP virtual machine	
* CP RECONNECT	VSE	VSE/SP virtual machine	
CPCOM macro	VSE	VSE/SP virtual machine	All CP commands allowed
VSEMSG	CMS	VSE/SP system	Retrieve VSE SYSLOG
VSECMD	CMS	VSE/SP system	VSE/SP commands and replies
VSEREP	CMS	VSE/SP system	Answer outstanding VSE requests
SUBVSE	CMS	VSE/SP system	Submit jobs to VSE/SP virtual machines

Using CMS/DOS with VSE/SP

CMS/DOS enables the CMS user to use the interactive facilities of VM to develop programs and then execute them in a virtual machine.

CMS/DOS support in VM is based on the VSE/AF Version 1 licensed program. CMS/DOS contains terms for both CMS (in the form of commands) and VSE (in the form of control cards); it simulates many of the functions of the DOS VSE/AF operating system.

How the Library Structure of VSE/SP Restricts CMS Users

VSE/SP Version 2 and Version 3 have different library structures from previous releases of VSE. They also manipulate the library in different ways. Therefore, many CMS/DOS functions cannot be used with VSE/SP Version 2 or Version 3 unless you use alternative methods to ones prior to VSE/SP 2.1.

Using VSE Librarian Functions in CMS/DOS

You cannot use the following VSE librarian functions from CMS when you are using VSE/SP Version 2 or Version 3:

DSERV
ESERV
SSERV
PSERV
RSERV

Suggested Alternatives When Using VSE/SP Version 2 or Version 3

1. Use the librarian functions from the Interactive Interface of VSE/SP Version 2 and Version 3. You cannot use these functions from CMS, therefore, log off or disconnect from the CMS virtual machine and DIAL into the VSE system. (You can do this from terminals that have been defined with the SPECIAL statement in the VM directory.)

   To end this session, log off from the Interactive Interface and log on again to CMS.

   To save time in switching between CMS and the Interactive Interface, you can use VM/PASSTHRU. See "Using VM/PASSTHRU" on page 50 for more information.

2. A second alternative is to create a CMS file with JCL and librarian statements. You can submit this file to the VSE system and route it back with the VM Writer Task. See "Transferring Output with the VM Writer Task of VSE/POWER" on page 44 for more information.

Other CMS/DOS Restrictions When You Use VSE/SP Version 2 or 3

Command or EXEC	Restriction	Comments
DOSLKED	You cannot use this command with the libraries of VSE/SP.	You can still use this command when your input is a CMS TEXT file or a CMS DOSLNK file.
FCOBOL	You cannot use this EXEC with VSE/SP.	

Command or EXEC	Restriction	Comments
VMFDOS	You cannot use this command to load or scan modules from a VSE/SP distribution library tape.	VSE SYSIN tape format is still supported.
FETCH	You cannot use this command to fetch a phase from a VSE/SP library.	You can still use this command to fetch a phase from a CMS DOSLIB.
SET DOS ON mode	You will receive an error message if you use this command with the *mode* operand.	Use SET DOS ON without filemode to activate CMS/DOS.
ASSGN	You cannot assign the following system logical units to a VSE disk: SYSCLB, SYSRLB, SYSSLB with VSE/SP.	You can still use this command to assign all other logical units to input and output devices.
DLBL	IJSYSCL, IJSYSRL, and IJSYSSL are invalid filenames for VSE/SP libraries.	You can still use this command to identify CMS and VSAM files.

What Features of the Interactive Interface Can a CMS User Use?

A CMS user can use all, some, or none of the facilities of the Interactive Interface, depending on his Interactive Interface user profile. This profile is defined by the VSE system administrator. As Interactive Interface user profiles may be different, so will the selection panels each CMS user sees.

Using VM/PASSTHRU

VM/PASSTHRU is a VM/SP optional licensed program. It enables a virtual machine on one system to pass through to an operating system or application on:

- The same processor
- Any other processor defined to PASSTHRU

VM/PASSTHRU runs in a disconnected virtual machine under the control of VM. You can activate it or deactivate it at any time. You usually activate it using AUTOLOG1. The userid of the PASSTHRU virtual machine can be any name, but usually it is called PVM.

To use the PASSTHRU facility, you can either:

- Execute the PASSTHRU EXEC from the active CMS environment, or
- DIAL into the PASSTHRU virtual machine.

The PASSTHRU EXEC is supplied with the VM/Pass-Through licensed program (5748-RC1).

How to Switch Between CMS and the Interactive Interface

You can use VM/PASSTHRU to pass through to the Interactive Interface and pass back to CMS again. Use either a **PF key** or a **four-character string** (for example, %%%%) to switch back and forth between the Interactive Interface and CMS.

By using the Interactive Interface, a CMS user can do many things, such as:

- Transfer a copy of the VSE/SP system console to a CMS console
- Display VSE/SP system activity
- Interactively display VTOC information
- Execute CICS transactions
- Use the Online Problem Determination dialog
- Display VSE/POWER queue entries
- Use VSE/ICCF.

Time-out Limit of VM/PASSTHRU

The default time-out limit for PASSTHRU is 20 minutes. You can increase it to as much as 9999 seconds (2.7 hours). To do this, redefine the TDISC parameter in the file PVM CONFIG (a file on the PVM machine).

Example of the PASSTHRU EXEC

The following is an example of the PASSTHRU EXEC:

```
EXEC PASSTHRU NODEA * PVM 11 24 80 %%%% ####
```

The eight parameters of this EXEC are described in the following table.

Parameter	Example	Description
Name of node	**NODEA**	The VM system to where you want to pass through.
Port address	*	Use * for local use.
Name of PASSTHRU disconnected virtual machine	**PVM**	The CMS userid on which PASSTHRU is installed
PF key for capture facility	*11*	When you press this PF key (after passing through to the destination machine), a hard copy of the screen is saved in a CMS file called PASSTHRU DATA A. This file is stored on the system from where you started the PASSTHRU EXEC.
Number of lines to be captured	**24**	
Width of lines to be captured	**80**	
Temporary disconnect	**%%%%**	You can specify either a PF key or a four-character string. Use this to switch between CMS and the Interactive Interface. If you disconnect from a selection panel, you return to that panel. If you disconnect from within a dialog, you return to the initial panel "VSE/SP Function Selection".
Permanent disconnect	**####**	You can specify either a PF key or a four-character string. This will sign you off from the other system. With this, you can permanently disconnect from the Interactive Interface and return to CMS. If you want to use the PASSTHRU EXEC again, you must again sign on to the Interactive Interface. *Note: IBM recommends that you sign off from the Interactive Interface before you use the permanent disconnect.*

PF Key Overrides

If you use a PF key for the capture facility or for the temporary or permanent disconnect, this setting overrides the setting in the Interactive Interface. For example, if you use PF10 for the capture facility, you cannot use PF10 for the function it represents in the Interactive Interface.

IPLing the Device Support Facility under VM

Use the Device Support Facility service program to format minidisks for use by VSE. The *VSE/SP Installation* manual describes how you can do this when running VSE/SP under VM.

Problem Determination and the VSE/SP Virtual Machine

IBM provides a variety of support services for its hardware and software. In order to obtain prompt service for problems you encounter, you will need to know which resource to use for each major type of problem you experience. Answer the questions posed below to determine what type of problem you have, and what type of support is available to assist you in resolving it.

1. Does the problem involve IBM software/hardware or other software/hardware?

   - IBM - Proceed to Question 2.
   - Other - Refer problem to internal resources or appropriate vendors.

2. Is the problem related to hardware or software?

   - Hardware - Contact IBM Customer Service.
   - Software - Identify the problem source.

The procedure you would normally follow to perform problem source identification is basically the same for all problems.

Once a problem is detected, you would observe all the symptoms. Ordinarily these symptoms would be identified as error messages, an abnormal end of job, loops, wait state, or program check. Also note special conditions associated with the failure. Generally, these special conditions will be one or more of the following:

- A new job running?
- A recent sysgen?
- A change in system configuration?
- A new procedure is being used?
- Something different from when the last time the run was made?

These are the kind of conditions that may have a bearing on the failure. In all cases, document the symptoms and conditions you observed.

The best procedure in problem determination is to gather enough information so a meaningful search can be made against the problems that have already been reported to the IBM Support Center. Statistically, there is the probability that someone else has had the same problem. If this is the case, the symptoms you report will lead to the known solution quicker.

VSE/SP Virtual Machine DUMP Procedure

The way to get a stand-alone dump of a VM/VSE system is similar to the way it's done in native mode. The differences are:

1. You must issue the CP command CP STORE STATUS instead of doing a machine save. Then issue SET RUN OFF.

2. The tape unit must be attached to the VSE machine when you IPL the stand-alone dump program.

If you are using handshaking and the VSE storage is in the virtual machine, you can use the CP VMDUMP command but you will not be able to use VSE's IPCS to debug the VSE virtual machine.

You may want to use VSE's IPCS-E to debug VSE problems but it will require using VSE DUMP utilities. This allows you to interactively debug problems, create problem reports on the VSE machine, go into dump scan mode using VSE's IPCS-E.

Backup/Restore Procedure for the VSE/SP Virtual Machine

VM's DASD Dump/Restore (DDR), TAPE DUMP, MOVEFILE, VMFPLC2 DUMP, and VSE FASTCOPY should be used for backing up and restoring the VM and/or the VSE system. It is advisable to back up your system on a regular basis. The following chart shows the functions each utility has:

	VM DATA	VSE DATA
ALL	DDR	FASTCOPY or DDR
CMS	CMS FAST BACKUP PP, TAPE DUMP, MOVEFILE, VMFPLC2 DUMP, VMTAPE, VMBACKUP	
VSAM DATA SETS	Access Method Services	Access Method Services

Note: You cannot use DDR on a file that was backed up using VSE/FASTCOPY or use VSE/FASTCOPY on a file that was backed up using DDR.

Time should be spent in planning the back up of your total system. In deciding which backup method to use, ask yourself:

- What type of tape drive will be used in the backup procedure?

- How often will the system be backed up?

When backing up CMS minidisks, you might want to consider using the high speed CMS Minidisk Backup and Restore Utility, VMTAPE, or VMBACKUP.

Backing up and restoring VSAM DATA SETS will be handled with the normal access method service utilities in both the VM and the VSE environment.

Note: The VSE system should be shut down before a backup of the VSE system is taken using DDR.

To perform a stand-alone backup in a virtual machine (i.e., MAINT's userid), you must spool the punch to yourself:

```
spool punch *
```

Bring up the DDR.

```
punch ipl ddr s (noh
```

Load the DDR into the virtual machine.

```
ipl 00C
```

Using VM/VCNA in a VSE-under-VM Environment

VM/VCNA is an ACF/VTAM application that runs in a VSE partition. The VSE virtual machine controlling the SNA network is called the VTAM service machine.

The installation steps are shown in *Virtual Machine/VTAM Communications Network Application (VCNA) Installation, Operation, and Terminal Use.*

VM/VCNA allows the use of an SNA terminal as a console for a virtual machine such as CMS; it is the linkage between the SDLC protocol on the SNA network and VM.

Figure 10 on page 56 illustrates the data flow of VM/VCNA.

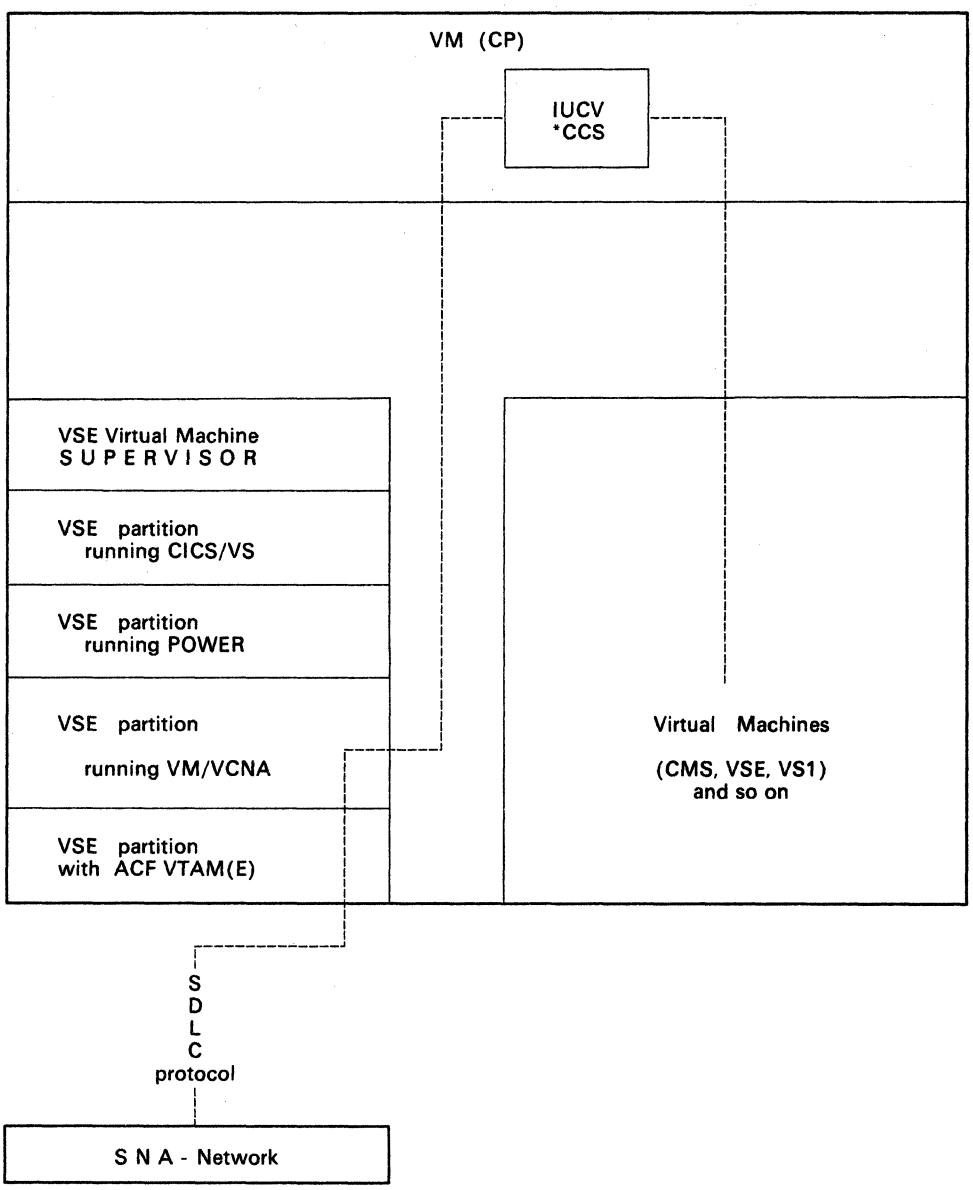


Figure 10. Data Flow of VM/VCNA

Using VSCS in a VSE-under-VM Environment

VSCS (VM/SP SNA Console Services) is a component of VM/VTAM. Together they comprise an alternative to VCNA.

The group control system (GCS) component of VM/SP coordinates a set of VM/VTAM virtual machines. A typical GCS "group" includes:

- A VTAM virtual machine
- An RSCS Version 2 virtual machine
- An NCCF virtual machine
- An NPDA virtual machine
- A GCS recovery virtual machine.

VSCS may run as a separate virtual machine in a GCS group, or it may run together with VTAM in the VTAM virtual machine. VSCS supports all the functions of VCNA and has the following advantages:

- A guest operating system is not needed
- VSCS installation is simpler than VCNA
- VSCS terminals display the VM/SP system identifier
- You can use the 3290 display station
- You can use the CP DIAL command on SNA 3270 and 3290 display terminals.

VTAM Configuration for a VSE Guest

The following example is a sample directory entry for a VTAM machine.

```
USER VTAM GCS8508 10M 16M ABCDEFG
 ACCOUNT 112 BOX04-19
 OPTION DIAG98 ECMODE MAXCONN 400
 IUCV *CCS P M 10
 IUCV ANY P M 0
 IPL GCS PARM AUTOLOG
 CONSOLE 01F 3215
 SPOOL 00C 2540 READER *
 SPOOL 00D 2540 PUNCH  A
 SPOOL 00E 3211 A
 LINK MAINT 595 595 RR
 LINK MAINT 298 191 RR
 LINK MAINT 29A 29A RR
```

Notes:

1. *DIAG98 allows the use of "real I/O" operations. CCWs are not translated by CP but are provided by VTAM in its virtual machine, thereby providing a fast path through CP for I/O processing. Because VTAM uses specialized code for this transaction, it requires less CPU time than CP generalized translation. Part of DIAG98 protocol requires that pages containing VTAM buffers be locked.*

2. *CCS is used by VSCS to communicate with CP.*

3. *ANY is used by VTAM to communicate with the members of the GCS group.*

Figure 11. A Sample VTAM Directory Entry

VTAM Restrictions

- VM/VTAM cannot support a TERM CONMODE 3270 console. This means that you should have a non-SNA control unit or a display/printer-adapter (DPA) or a console port in order to have a VSE DOC console.

- You cannot log on the VTAM virtual machine on an SNA terminal.

Migrating from VCNA to VSCS

If you decide to migrate from a VCNA environment such as in Figure 15 on page 61 to a full SNA environment with VM/VTAM such as in Figure 16 on page 62 then you can use the following checklist as a help.

1. Install VM/VTAM.
2. Punch your old VTAM definitions either from your ICCF Library or from your system library to the MAINT machine. See Figure 12 and Figure 13 on page 60 for examples.
3. Start an appropriate printer and punch to get the members sent to MAINT. For example,

   ```
   s pun,02d,r,,vm

       - or -

   s lst,02e,r,,vm
   ```

4. Give each B.Book a VM name such as CDRMROM VTMLST and file them on MAINT's 298 disk.
5. Check your Startup book and remove the PROMPT statement if there is one. Change or remove your buffer specifications, because VM/VTAM has different kinds of buffers. (See Figure 14 on page 60.)
6. Start VM/VTAM according to "Starting Up VM/VTAM" on page 70.

```
* $$ JOB JNM=DTS,CLASS=0,DISP=D
* $$ LST CLASS=R,DEST=(*,MAINT)
* $$ PUN CLASS=R,DEST=(*,MAINT)
// JOB DTS      PUNCH DTSMEMBERS TO VM
// ASSGN SYS010,DISK,VOL=SYSWK1,SHR
// EXEC DTSUTIL
PUN M(44 ATCSTRNB)
PUN M(44 ATCCONCD)
PUN M(44 VTMAPPL)
PUN M(44 SNAROM)
PUN M(44 NSNAROM)
PUN M(44 PATHROM)
PUN M(44 CAROM)
PUN M(44 CAROMKJ)
PUN M(44 CDRMROM)
PUN M(44 CDRSROM)
PUN M(44 VTMSW1)
END
/&
* $$ EOJ
```

Figure 12. Punching VTAM Definitions to the MAINT Machine (Sample 1)

```
* $$ JOB JNM=LIBR,CLASS=0,DISP=D
* $$ LST CLASS=R,DEST=(*,MAINT)
* $$ PUN CLASS=R,DEST=(*,MAINT)
// JOB LIBR      PUNCH DTSMEMBERS TO VM
// EXEC LIBR
A S=PRD2.CONFIG
PU ATCSTRNB.B
PU ATCCONCD.B
PU VTMAPPL.B
PU SNAROM.B
PU NSNAROM.B
PU PATHROM.B
PU CAROM.B
PU CAROMKJ.B
PU CDRMROM.B
PU CDRSROM.B
PU VTMSW1.B
/*
/&
* $$ EOJ
```

Figure 13. Punching VTAM Definitions to the MAINT Machine (Sample 2)

```
vtam d net,bfruse
Ready;
IST097I DISPLAY ACCEPTED
IST350I VTAM DISPLAY - DOMAIN TYPE = BUFFER POOL DATA
IST632I BUFF   BUFF  CURR   CURR   MAX    MAX  TIMES   EXP/CONT    EXP
IST633I  ID    SIZE TOTAL  AVAIL  TOTAL  USED   EXP   THRESHOLD   INCR
IST356I IO00  00311 00300  00300  00300 00020  00000 00050/-----  00012
IST356I LP00  01016 00012  00009  00012 00006  00000 00003/-----  00004
IST356I WP00  00160 00013  00011  00013 00003  00000 00001/-----  00024
IST356I LF00  00120 00007  00007  00007 00000  00000 00001/-----  00032
IST356I CRPL  00116 00200  00188  00200 00013  00000 00030/-----  00032
IST356I SF00  00072 00056  00052  00056 00004  00001 00001/00103  00051
IST356I SP00  00112 00002  00002  00002 00000  00000 00001/-----  00034
IST356I AP00  00488 00010  00002  00010 00008  00000 00001/-----  00008
IST449I CSALIMIT = NOLIMIT, CURRENT = 000228K, MAXIMUM = 000228K
IST595I IRNLIMIT = NOLIMIT, CURRENT = 000000K, MAXIMUM = 000000K
IST314I END
```

Figure 14. Buffers for VM/VTAM

Possible Networks When Using Virtual Addressability Extension

For installations running with virtual addressability extension (VAE), the following two figures illustrate possible network scenarios. With the installation of ACF/VTAM Version 3 for VSE, SNA physical units may be owned by a VSE virtual machine running VCNA, while some terminals may establish sessions with a CICS running in a VSE VAE system. (VCNA does not function in a VAE environment.) Or this VCNA virtual machine can be replaced by a VTAM virtual machine.

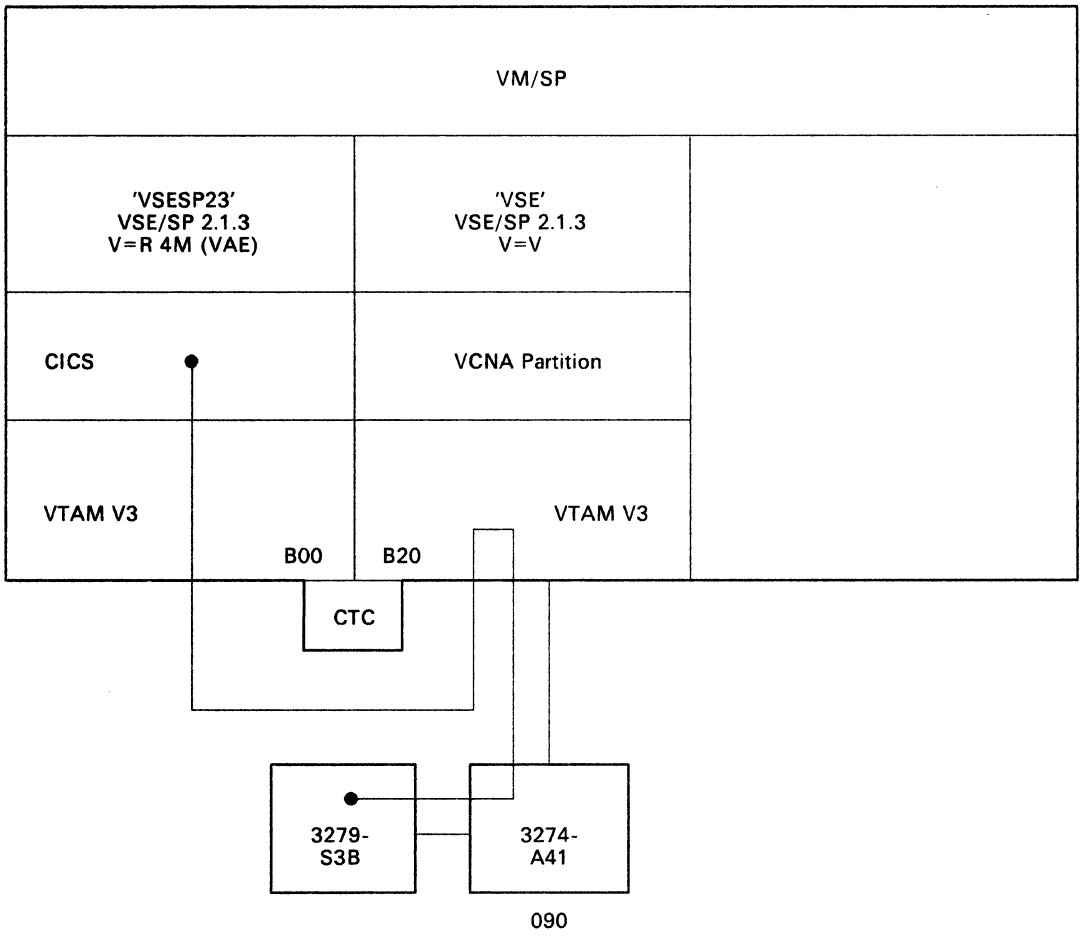


Figure 15. Terminals Owned by VSE V=V. With the possibilities of a cross-domain session to CICS via VCTC and a session to VM via VCNA.

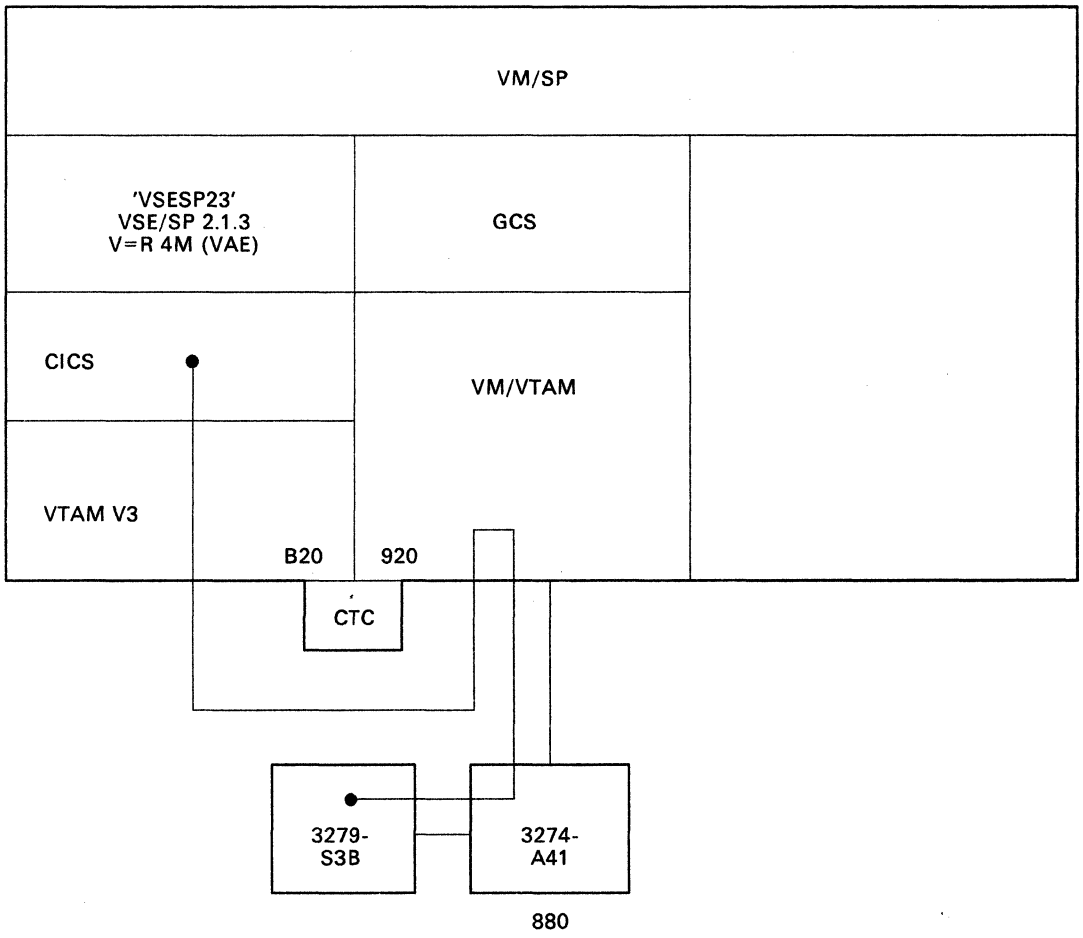


Figure 16. Terminals Owned by VM/VTAM. With the possibility of a cross-domain session to CICS via VCTC and a session to VM.

Generating a USSTAB for VM/VTAM

To make your work easier, generate a new USSTAB. The samples are provided in Figure 17 on page 64.

1. Copy the provided sample USSTAB 'ISTINCDT ASSEMBLE' which is on MAINT's 298 disk.

2. Change the USSTAB name to VTMV3USS and add your DBDCCICS application.

3. Assemble the new USSTAB:

```
global maclib vtamac

assemble vtmv3uss
```

4. Add the USSTAB to the LKEDCTRL file (as, for example, ISUSER LKEDCTRL in Figure 18 on page 65).

5. Link the ISCUSER:

```
vmflked iscuser
```

   As a result you will get a new ISCUSER LOADLIB.

VTMV3USS ASSEMBLE (USSTAB)

```
VTMV3USS USSTAB   TABLE=STDTRANS
         SPACE 4
LOGON    USSCMD   CMD=LOGON,FORMAT=PL1
         USSPARM  PARM=APPLID
         USSPARM  PARM=LOGMODE
         USSPARM  PARM=DATA
         EJECT
LOGOFF   USSCMD   CMD=LOGOFF,FORMAT=PL1
         USSPARM  PARM=APPLID
         USSPARM  PARM=TYPE,DEFAULT=UNCOND
         USSPARM  PARM=HOLD,DEFAULT=YES
         EJECT
UNDIAL   USSCMD   CMD=UNDIAL,FORMAT=PL1
         EJECT
VM       USSCMD   CMD=VM,REP=LOGON,FORMAT=BAL
         USSPARM  PARM=P1,REP=DATA
         USSPARM  PARM=LOGMODE
         USSPARM  PARM=APPLID,DEFAULT=VM
         EJECT
CICS     USSCMD  CMD=CICS,REP=LOGON,FORMAT=BAL
         USSPARM PARM=P1,REP=APPLID,DEFAULT=DBDCCICS
         USSPARM PARM=P2,REP=DATA
         EJECT
IBMTEST  USSCMD   CMD=IBMTEST,FORMAT=BAL
         USSPARM  PARM=P1,DEFAULT=10
         USSPARM  PARM=P2,DEFAULT=ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
         EJECT
MESSAGES USSMSG   MSG=1,TEXT='INVALID COMMAND SYNTAX'
         USSMSG   MSG=2,TEXT='% COMMAND UNRECOGNIZED'
         USSMSG   MSG=3,TEXT='% PARAMETER UNRECOGNIZED'
         USSMSG   MSG=4,TEXT='% PARAMETER INVALID'
         USSMSG   MSG=5,TEXT='UNSUPPORTED FUNCTION'
         USSMSG   MSG=6,TEXT='SEQUENCE ERROR'
         USSMSG   MSG=7,TEXT='SESSION NOT BOUND'
         USSMSG   MSG=8,TEXT='INSUFFICIENT STORAGE'
         USSMSG   MSG=9,TEXT='MAGNETIC CARD DATA ERROR'
         USSMSG   MSG=11,TEXT='% SESSIONS ENDED'
         USSMSG   MSG=12,TEXT='REQUIRED PARAMETER OMITTED'
         USSMSG   MSG=13,TEXT='IBMECHO % '
         EJECT
STDTRANS DC       X'000102030440060708090A0B0C0D0E0F'
         DC       X'101112131415161718191A1B1C1D1E1F'
         DC       X'202122232425262728292A2B2C2D2E2F'
         DC       X'303132333435363738393A3B3C3D3E3F'
         DC       X'404142434445464748494A4B4C4D4E4F'
         DC       X'505152535455565758595A5B5C5D5E5F'
         DC       X'606162636465666768696A6B6C6D6E6F'
         DC       X'707172737475767778797A7B7C7D7E7F'
         DC       X'80C1C2C3C4C5C6C7C8C98A8B8C8D8E8F'
         DC       X'90D1D2D3D4D5D6D7D8D99A9B9C9D9E9F'
         DC       X'A0A1E2E3E4E5E6E7E8E9AAABACADAEAF'
         DC       X'B0B1B2B3B4B5B6B7B8B9BABBBCBDBEBF'
         DC       X'C0C1C2C3C4C5C6C7C8C9CACBCCCDCECF'
         DC       X'D0D1D2D3D4D5D6D7D8D9DADBDCDDDEDF'
         DC       X'E0E1E2E3E4E5E6E7E8E9EAEBECEDEEEF'
         DC       X'F0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF'
END      USSEND
         END                    END OF ASSEMBLY
```

Figure 17. VTMV3USS ASSEMBLE (USSTAB)

ISCUSER LKEDCTRL

```
%ERASE
%MAXRC 8
%LEPARMS NCAL LIST XREF LET RENT
INCLUDE DTIUSER3
NAME DTIUSER3(R)
INCLUDE ISTINCDT
ENTRY ISTINCDT
NAME ISTINCDT(R)
INCLUDE VTMV3USS
ENTRY VTMV3USS
NAME VTMV3USS(R)
```

Figure 18. ISCUSER LKEDCTRL

DTIUSER3 ASSEMBLE

```
DTIUSER3 DTIGEN   DTIUSER=3,
                  PRTSHR=Y,
                  TIMECPY=30,
                  APPLID=VM,
                  TIMEREL=120
         END
```

Figure 19. DTIUSER3 ASSEMBLE

ACF/VTAM Version 3 for VSE/SP 2.1.3

The default buffer size of VFBUF and VPBUF is 4K. You cannot modify this size. You can, however, change the size of LFBUF.

Buffers for ACF/VTAM Version 3 for VSE/AF 2.1.1

```
F3 016 5D50I VTAM DISPLAY - DOMAIN TYPE = BUFFER POOL DATA
F3 016 5G32I BUFF   BUFF  CURR   CURR   MAX    MAX   TIMES   EXP/CONT    EXP
F3 016 5G33I  ID    SIZE TOTAL  AVAIL  TOTAL  USED   EXP    THRESHOLD   INCR
F3 016 5D56I VF     4096  00015P 00010P  N/A  00005P  N/A      N/A       N/A
F3 016 5D56I VP     4096  00064P 00026P  N/A  00044P  N/A      N/A       N/A
F3 016 5D56I SF     00392 00020  00017  00020 00004  00000 00003/-----  00010
F3 016 5D56I LF     00343 00050  00040  00050 00028  00000 00003/-----  00011
F3 016 5D56I SP     00112 00003  00003  00003 00000  00000 00001/-----  00032
F3 016 5D56I LP     01344 00012  00008  00012 00007  00001 00002/00008  00003
F3 016 5D56I WP     00184 00040  00037  00040 00005  00001 00019/00059  00020
F3 016 5F95I IRNLIMIT = NOLIMIT, CURRENT = 0000000, MAXIMUM = 0000000
F3 016 5D14I END
```

Figure 20. Buffers for ACF/VTAM for VSE/AF 2.1.1

Virtual Storage Requirements of ACF/VTAM V3

To bring up VTAM Version 3 with the same nodes and buffer definitions, you will need to increase the virtual and real storage sizes in the VTAM partition. If there is not enough storage available, one of the following messages will appear.

```
F3 018 5A48I VTAM START REJECTED - INSUFFICIENT STORAGE FOR BUFFERS
F3 018 5B33I VTAM TERMINATION IN PROGRESS
F3 018 5B02I VTAM IS NOW INACTIVE
F3 003 EOP $3JCLROM

F3 016 5B16I MEMBER CAROM NOT FOUND ON VTAM DEFINITION LIBRARY
F3 016 5A61I VARY ACT FOR ID = CAROM FAILED - NODE UNKNOWN TO VTAM
```

Figure 21. Possible Error Messages in Bringing Up VM/VTAM

Setup of a VCTC and Operational Considerations

Definitions for VM/VTAM

All VTAM definitions and profiles reside on MAINTs minidisk 298, which should be linked by the VTAM machine in read-only mode as 191. After maintaining your VTAMLSTs (B-Books), you have to reaccess this disk from the VTAM machine in order to have your changes activated. Therefore, put the following in the PROFILE GCS:

```
ACC 191 A
```

Defining the Virtual Channel

You can define the virtual CTC in the PROFILE GCS.

```
DEF CTCA 900
DEF CTCA 920
```

Defining the VTAMLST for CTC

```
S2CTC    VBUILD    TYPE=CA
S2CTCG   GROUP     LNCTL=CTCA,REPLYTO=25.5,MAXBFRU=(10,30)
S2CTCL1  LINE      ADDRESS=920
S2CTCP1  PU        PUTYPE=4
S2CTCL4  LINE      ADDRESS=900
S2CTCP4  PU        PUTYPE=4
```

Figure 22. CTC VTAMLST

The default value of REPLYTO= is three seconds. The maximum value is 25.5 seconds.

Other VTAMLSTs

```
SSCPID=2
```

Figure 23. ATCSTR00 VTAMLST

```
SSCPID=2,
HOSTSA=2,
MAXSUBA=31,
CONFIG=YK,
CRPLBUF=(200,,15,,01,30),
IOBUF=(300,256,40,,01,50),
NOTRACE,TYPE=VTAM
```

Figure 24. ATCSTRYK VTAMLST

```
        APPLROMV,
        SNAYK,
        NSNAROMV,
        CAYK,
        PATHYK,
        CDRMYK,
        CDRSYK
```

Figure 25. ATCCONYK VTAMLST

```
         VBUILD TYPE=APPL
VM       APPL   AUTH=(PASS,ACQ,BLOCK),AUTHEXIT=YES,ACBNAME=VM,
                PARSESS=YES,PRTCT=VM
```

Figure 26. APPLROMV VTAMLST

```
         VBUILD TYPE=LOCAL
D3274A41 PU     CUADDR=880,MAXBFRU=15,VPACING=1,
                PUTYPE=2,ISTATUS=ACTIVE
D3279P16 LU     LOCADDR=18,DLOGMOD=D4A32793,MODETAB=VTMV3USS
D3279P17 LU     LOCADDR=19,DLOGMOD=D4A32793,MODETAB=VTMV3USS
D3279P18 LU     LOCADDR=20,DLOGMOD=D4A32793,MODETAB=VTMV3USS
D3279P19 LU     LOCADDR=21,DLOGMOD=D4A32793,MODETAB=VTMV3USS
```

Figure 27. SNAYK VTAMLST

```
         LBUILD
V0B0     LOCAL CUADDR=0B0,TERM=3277,FEATUR2=(MODEL2),
               DLOGMOD=S3270,MODETAB=VTMV3USS
V0B1     LOCAL CUADDR=0B1,TERM=3277,FEATUR2=(MODEL2),
               DLOGMOD=S3270,MODETAB=VTMV3USS
V0B2     LOCAL CUADDR=0B2,TERM=3277,FEATUR2=(MODEL2),
               DLOGMOD=S3270,MODETAB=VTMV3USS
V0B3     LOCAL CUADDR=0B3,TERM=3277,FEATUR2=(MODEL2),
               DLOGMOD=S3270,MODETAB=VTMV3USS
```

Figure 28. NSNAROMV VTAMLST

```
S2CTC    VBUILD    TYPE=CA
S2CTCG   GROUP     LNCTL=CTCA,REPLYTO=25.5,MAXBFRU=(10,30)
S2CTCL1  LINE      ADDRESS=920
S2CTCP1  PU        PUTYPE=4
S2CTCL4  LINE      ADDRESS=900
S2CTCP4  PU        PUTYPE=4
```

Figure 29. CAYK VTAMLST

```
PATHYK   PATH DESTSA=1,
              ER5=(1,1),
              VR5=5
```

Figure 30. PATHYK VTAMLST

```
CDRMYK    VBUILD    TYPE=CDRM
VMVTAM    CDRM      SUBAREA=2,CDRSC=OPT,CDRDYN=YES,
                    ISTATUS=ACTIVE,VPACING=2
VSEVTAM   CDRM      SUBAREA=4,CDRSC=OPT,CDRDYN=YES,
                    ISTATUS=ACTIVE,VPACING=2
SP23VTAM  CDRM      SUBAREA=1,CDRSC=OPT,CDRDYN=YES,
                    ISTATUS=ACTIVE,VPACING=2
```

Figure 31. CDRMYK VTAMLST

```
CDRSYK    VBUILD    TYPE=CDRSC
ROMACICS  CDRSC     CDRM=VSEVTAM
VCNA      CDRSC     CDRM=VSEVTAM
RM80      CDRSC     CDRM=VSEVTAM
RM81      CDRSC     CDRM=VSEVTAM
RM82      CDRSC     CDRM=VSEVTAM
RM83      CDRSC     CDRM=VSEVTAM
LJ109001  CDRSC     CDRM=VSEVTAM
LK109001  CDRSC     CDRM=VSEVTAM
LL109001  CDRSC     CDRM=VSEVTAM
LM109001  CDRSC     CDRM=VSEVTAM
DBDCCICS  CDRSC     CDRM=SP23VTAM
D080      CDRSC     CDRM=SP23VTAM
D081      CDRSC     CDRM=SP23VTAM
D082      CDRSC     CDRM=SP23VTAM
D083      CDRSC     CDRM=SP23VTAM
LJ188001  CDRSC     CDRM=SP23VTAM
LK188001  CDRSC     CDRM=SP23VTAM
LL188001  CDRSC     CDRM=SP23VTAM
LM188001  CDRSC     CDRM=SP23VTAM
```

Figure 32. CDRSYK VTAMLST

Starting Up VM/VTAM

There are two ways to start up VM/VTAM. During your first tests with VM/VTAM you may want to bring up VTAM manually. The PROMPT parameter in the VTAM startup book is not supported.

1. Make a comment line out of the 'EXEC VMVTAM' in the PROFILE GCS of the VTAM machine. The GCS recovery machine will be autologged by the AUTOLOG1 machine. VTAM will be autologged by the GCS recovery machine.
2. Log on to the disconnected VTAM machine.
3. Enter:

   ```
   VMVTAM xx
   ```

   where *xx* is the suffix of your VTAM startup list.

   For example:

   'VMVTAM YK' will start VTAM with ATCSTRYK VTAMLST after executing ATCSTR00 VTAMLST.

   Entering just 'VMVTAM' will bring up the default list ATCSTR00 VTAMLST.

Another way is **not to change the PROFILE GCS of the VTAM machine**, but to prevent the GCS recovery machine from autologging the VTAM machine. After the autolog of the GCS recovery machine, logon to the VTAM machine. After seeing that GCS has been loaded, you then give the command VMVTAM and answer in the "OS-way" with 'R replid suffix', where suffix is the 'xx' of the ATCSTRxx VTAMLST.

To start up VM/VTAM automatically, modify the 'EXEC VMVTAM' statement in the PROFILE GCS to 'EXEC VMVTAM xx', where xx is your startup list. An example is given in "PROFILE GCS."

PROFILE GCS

```
/**
*** Title-
***      PROFILE OF THE VTAM MACHINE
***
**/
/**  NOW FIND OUT WHICH VTAMLST WE WANT TO USE
**/
/** SAY 'WHICH VTAM LIST ARE WE USING TODAY?'
    SAY "ENTER IT'S TWO CHARACTER CODE 'XX'"
    PARSE UPPER PULL LIST_NO
**/
/**
*** Set CP options to improve performance of VTAM virtual machine
**/
 'CP SET QDROP VTAM OFF'          /* DON'T FLUSH PAGES WHEN IDLE   */
 'CP SET FAVORED VTAM'            /* VTAM ALWAYS DISPATCHABLE      */
 'CP SET PRIORITY VTAM 1'         /* GIVE PRIORITY TO VTAM         */
 'CP SET PF11 RETRIEVE'           /* LET'S REMEMBER A COMMAND      */
 'CP SET PF23 RETRIEVE'           /* LET'S REMEMBER A COMMAND      */
 'CP DEF GRAF 0B0'                /* TERMINAL ADDRESSES TO DIAL    */
 'CP DEF GRAF 0B1'                /* VTAM MACHINE                  */
 'CP DEF GRAF 0B2'
 'CP DEF GRAF 0B3'
 'CP DEF CTCA 900'                /* VIRTUAL CTC TO VSE            */
 'CP DEF CTCA 920'                /* VIRTUAL CTC TO VSESP23        */
 'CP LINK MAINT 59F 59F RR RVM4'
 'ACC 191 A'                      /* REACCESS MAINT 298            */


/**
*** VTAM initialization
**/
    'EXEC VMVTAM YK'
/* 'CP AUTOLOG RSCS password'  Starts RSCS Virtual Machine if any*/
    exit 0
```

Notes:

1. *The REXX PULL instruction in conjunction with GCS will give the possibility to answer in an OS-format; that is, 'R replid answer'.*
2. *If you activated the REXX PULL instruction, you should change the 'EXEC VMVTAM YK' statement to : 'EXEC VMVTAM list_no'*

Figure 33. PROFILE GCS

VMVTAM GCS

```
/**
*** Title-
***      VMVTAM
***
*** Function-
***      Initialize VM/VTAM and VSCS for use.
***
*** Parameters-
***      list_value
***
*** Returns-
***      00 (VTAM has been successfully activated)
***      ¬0 (VTAM activation failed)
***
**/
    parse source . . exec_name
    arg list_value . '(' options
    if list_value = '' then
      list_value='00'


/**
*** VTAM initialization
**/
    'ACC 29A F/F'
    'ACC 59F G/G'
    'GLOBAL LOADLIB ISCUSER VTAM VSCS RSCS'
    'LOADCMD VTAM ISTINV00'
    'LOADCMD VSCS DTISLCMD'
    'VTAM START LIST='list_value
    rcode=rc
    if rcode ¬=0 then                /* If VTAM start failure          */
      do                             /* Error, VTAM startup failed     */
        say '**ERROR** VTAM initialization failed'
        exit rcode
      end                            /* Error, VTAM startup failed     */

/**
*** VSCS initialization
**/
    'VSCS START PARM=3'              /* Initialize VSCS with user3     */
    rcode=rc                         /* Save startup return code       */
    if rcode ¬=0 then                /* If VTAM start failure          */
      do                             /* Error, VTAM startup failed     */
        say '**ERROR** VSCS initialization failed'
        exit rcode
      end                            /* Error, VTAM startup failed     */
    exit 0
```

Figure 34. VMVTAM GCS

Starting the CTC Majornode

Before you can start the CTC majornode you should couple the virtual channels to each other. Issue the following command:

```
couple 920 vsesp23 b20
```

This command can successfully complete only if the VSESP23 machine is logged on and the appropriate channel B20 is defined. Because VTAM is normally one of the first machines logged on, issuing the couple command from a VSE guest is recommended. Refer to "Starting the CTC Majornode for a VSE Guest" on page 74.

Definitions for VSE Systems

Defining the Virtual Channel

The CTC can be defined in the same manner as for VM/VTAM. The appropriate 'DEF" statements are put in the PROFILE EXEC of the VSEMAINT machine which shares its profile with all VSE guest machines.

Defining the IPLPROC Entry

The entry for a virtual CTC in the IPLPROC should look as follows:

```
ADD B20,CTCA,EML
```

The EML option prevents the system sensing the address at initial IPL. If you do not code this option, you will get a wrong PUB entry when the CTC is not ready. In this case the system sends a message to the console like:

```
0171I ACTUAL DEVICE TYPE X'FF' FOR B20 INSERTED IN PUB
```

The next table should clarify this.

Situation	q v B20	Pub Entry
CTC not defined	DEV B20 DOES NOT EXIST	Correct
CTC defined	CTC B20 NOT READY	Wrong
CTC defined and coupled	CTC B20 COUPLE to VTAM 920	Correct

Figure 35. PUB Entry for CTC

Defining the B-Book for CTC

```
CTC      VBUILD TYPE=CA
CTCG     GROUP  LNCTL=CTCA,REPLYTO=25.5,MAXBFRU=(10,30)
CTCL4    LINE   ADDRESS=B00
CTCP4    PU     PUTYPE=4
CTCL2    LINE   ADDRESS=B20
CTCP2    PU     PUTYPE=4
```

The definition of the CTC is the same as for VM/VTAM except for the addresses of the channels.

Starting the CTC Majornode for a VSE Guest

Before you activate your CTC majornode you should make sure that the virtual CTC is coupled to another machine. Put the couple command in the PROFILE EXEC of the VSEMAINT machine for every VSE guest machine that uses a VCTC.

Check the status of the channel by issuing:

```
q v B20
```

If the channels are still not coupled, from the VSE console you can issue:

```
* cp couple b20 vtam 920
```

It is possible that some intervention will be required on the VM/VTAM console, because VM/VTAM does not activate the CTC majornode if the channels have not been coupled at initialization.

Chapter 4. VSE/SP Virtual Machines Sharing DASD

DASD sharing is possible under VM using the VSE Lockfile mechanism. Minidisks that are defined with the multiple write feature can be used by different VM users as shared disks (fullpack or partial minidisks).

Resource sharing across systems will function properly only if each sharing virtual machine has a unique CPU identification (CPUID). Therefore, a different CPUID must be defined for every virtual machine. Before IPLing a VSE/SP virtual machine, its VM directory entry must have a unique CPUID defined in the OPTION control statement or you can define it with the CP SET CPUID command. Without this command, **catastrophic errors will occur** in the VSE Lock File.

In general, those benefits that would be realized in the VSE environment are also applicable when operating under the control of VM.

- LIBRARY DUPLICATION

  As the number of VSE virtual machines increases, so does the requirement for direct access storage space. In many cases, VSE libraries are duplicated for each of the virtual machines. DASD sharing allows you to save direct access storage space, by avoiding duplication of libraries.

- REDUCTION IN MAINTENANCE EFFORT

  As additional virtual machines are added, additional system programmer time is required to generate and maintain the VSE system. Often, there is a requirement to keep several virtual machines synchronized, so that the application of service to one virtual machine necessitates the same service to other virtual machines. With DASD sharing, updates and maintenance need only be applied in one place.

- SHARED SPOOLING

  When the number of virtual VSE machines exceeds the number of physical printers available, the printer(s) may be switched back and forth between the virtual VSE machines. This requires the operator to insure that a VSE/POWER printer task is not started for any virtual machine that does not have a real printer available (in cases where a real attached printer is required). Alternatively, some virtual machine output can be spooled to VM, and the printer switched between VSE/POWER and VM. With the VSE/POWER Shared Spooling Feature, one set of VSE/POWER files can be shared between a maximum of 9 VSE/SP virtual machines. This allows multiple VSE

virtual machines to send their output to a single VSE/POWER controlled real printer. This situation is less prone to operator error, because no switching of printers between virtual machines is required.

DASD Sharing Considerations for the VSE/SP Virtual Machine

Whatever the benefits of DASD sharing you'd like to bring to your installation, there will be the cost of additional complexity, performance impact, and data integrity exposures. There are clearly more elements to consider and plan for. These elements are:

- System Configuration — you must compromise between flexibility and complexity.
- System design in terms of DASD mapping of files.
- Resource and load balancing.
- Recovery and Restart.
- Performance implications.
- Operational control of the multi-processor environment.
- Programming considerations for user resource protection.
- Data Integrity.

Careful planning is necessary to use DASD sharing efficiently and successfully.

If possible, avoid sharing DASD. If not, aim for a shared DASD design that is as simple as possible.

When designing a shared DASD system there are many factors to be considered, many of which apply equally to a nonshared system. Considerations unique to the DASD sharing environment include:

Hardware Design:

- The internal speed of all processors involved must be well balanced. For instance, the faster machine will probably dominate the slower one. However, there might be a valid configuration if the slow machine is performing a specific task in support of the fast machine, rather than operating as an equal partner. Alternatively, a slow machine can be used to control one or more 3800 printing subsystems, and share SYSRES and spool files with a faster processor. In this case, the amount of interference between the two processors will be kept to a minimum; while giving the opportunity for an operator to be in complete control of the subsystem.

In cases where both processors are supporting similar applications and sharing the work load on a more equal basis, the power of the processors should be reasonably comparable.

- An adequate I/O configuration should be provided. To achieve appropriate performance and availability, alternate I/O paths must be provided by using a combination of channel and string switching.

Software Design:

There are numerous software design considerations that influence the overall performance of a system. They become even more complex, if multiple systems have to coexist.

In order to minimize the SEEK time of DASD devices, the placement of the following system data sets should be evaluated very carefully:

- Lock File (Communication Area) — This file controls the actual file sharing activity. It should be placed on the least active of the shared DASD.

- VSE/POWER QUEUE and DATA files.

- The Volume Table of Contents (VTOC) for each individual DASD.

To optimize the I/O traffic, the following file should be split up into multiple extents, if the DASD configuration permits this:

- VSE/POWER DATA file.

In general all frequently used files on the same spindle should be grouped together. Again, the purpose is to keep the SEEK times as small as possible.

Files that are not shared between CPUIDs should be placed on a separate DASD string to avoid unnecessary resource contention.

Special attention has to be given to the definition of library chains. The following aspects should be considered:

- Length of Library chain
- Search sequence within chain
- Permanent vs. temporary chains.

In order to improve overall system throughput, the most frequently used phase directory entries should be loaded into the System Directory List, and the SVA-eligible phases should be made core resident by loading them into the Shared Virtual Area. It should be noted that SVA-eligible phases can be loaded from all sublibraries active in the BG concatenation chains.

A careful evaluation of the VSE/SP Supervisor and the VSE/POWER generation parameters is strongly recommended. In particular, the DBLK and TRACKGP parameters in VSE/POWER can significantly influence I/O traffic and therefore system throughput.

Under VM, the DBLK parameter should be set in the range of 4K to 6K in order to reduce the number of VSE/POWER START I/O's.

If your installation consists of more than one computing system, you might consider sharing some or all DASD devices between the different VSE systems. Rather than assigning a fixed number of disk drives to the different systems, you can combine the total number of available drives into a disk pool that is shared by all VSE systems. DASD sharing between two or more VSE systems has several advantages:

1. Library maintenance is easier, if only one set of libraries has to be maintained.

2. The total system throughput may increase when the VSE systems running under VSE/POWER share the POWER work files.

3. Direct access storage space can be saved, since only one copy of the data is required instead of multiple copies.

Reserve/Release Support

The CP component of VM does not issue reserve/release Channel Command Word (CCW) for itself; neither does CMS. CP issues them only on behalf of the guest operating system that issues the CCWs. VM checks all CCWs passed by guest operating systems running in VM and bases reserve/release CCW processing on:

- The device type

- The presence or lack of alternate path support

- Whether the MDISK statement in the VM directory contains a "V" in the mode operand.

Depending upon the various combinations of the above items, VM either permits the reserve CCW to execute on the hardware or changes the reserve CCW to a sense CCW. To determine the conditions under which a reserve is changed to a sense CCW, refer to Figure 36 on page 79.

Device Type	Alternate Path Online?	Resv/Rel Hardware Present?	Virtual Resv/Rel Defined?	What is Sent to Hardware?	Error Condition From CCW?	Integrity Problems with Links?	Integrity Problems with Multiple Paths?
Ded [1]	No	yes/no	-	reserve	no/yes	-	no
Ded [2] [5]	yes	yes	-	sense	no	-	yes
Mdisk [1]	no	yes	no	reserve	no	yes	no
Mdisk [1]	no	yes	yes	reserve	no	no	no
Mdisk [3] [1]	no	no	no	reserve	yes	yes	no
Mdisk [4]	no	no	yes	sense	no	no	no
Mdisk [5]	yes	yes	no	sense	no	yes	yes
Mdisk [5]	yes	yes	yes	sense	no	no	yes

Figure 36. Summary of VM Reserve/Release CCW Support

Notes:

1. *Normal Operation. The command is passed unchanged to the hardware.*

2. *When the VM system has been generated with alternate path support for those devices, and these alternate paths are online, then CP does not allow the real reserve CCW to be sent to the hardware. This action causes VM to avoid a possible channel lockout. VM does not return any indication that the device was not physically reserved to the operating system issuing the CCW.*

3. *VM sends the reserve/release CCW unchanged to the hardware. However, without the two-channel switch special feature or string switch, the hardware rejects the command and does not reserve the device. For a complete discussion on the existence of the hardware reserve/release feature along the path to the DASD device, please refer to "Hardware for DASD Sharing" on page 82.*

4. *Before sending the command to the hardware, VM changes reserve CCWs to sense CCWs, and places a virtual reserve on the minidisk. The real device is not reserved. The virtual reserve prevents other operating systems running under the same VM system from accessing the minidisk. However, these same virtual operating systems can virtually reserve other minidisks located on the same real volume. Because the reserve/release hardware is not present along the path to the DASD devices, VM's virtual reserve/release processing modifies the reserve CCW to a sense CCW. If the reserve CCW had not been modified, it would have been rejected by the hardware.*

5. *When alternate paths to a device are online, VM changes the reserve/release CCW to a sense CCW to prevent a possible channel lockout. In an MP environment, a symmetric alternate path is automatically defined. If that symmetrical alternate path is online the reserve CCW is changed to a sense CCW* **in all cases.**

By examining the table, you can determine:

- The device type (Dedicated DASD/tape or Minidisk)
- Whether alternate paths are online
- Whether the reserve/release hardware feature is present
- Whether virtual reserve/release has been defined for the shared DASD
- What VM sends to the hardware
- Whether the guest virtual machine receives an error condition after issuing a reserve or a release CCW
- Any problems occurring with the use of the LINK statement or with the existence of multiple paths to the shared DASD from the same or different processor.

To better understand how to use the table, we have included two examples.

Example 1 on how to use "Summary of VM Reserve/Release CCW Support"

This example refers to row 1 in Figure 36 on page 79.

Column Number Explanation

Column 1 The DASD device is either dedicated or attached to a virtual machine.

Column 2 No alternate paths are defined or online to this device.

Columns 3, 6 This column must be interpreted with Column 6:

- When column 3 is Yes, column 6 is No. This means that if the reserve/release hardware exists somewhere along the path to the device, no error condition will be returned by the hardware to CP when a reserve or release CCW is issued by a guest virtual machine to this shared device.
- When column 3 is No, column 6 is Yes. This means that if the reserve/release hardware ***does not exist*** along the path to the device, a COMMAND REJECT will be returned by the hardware to CP that will reflect this error to the guest virtual machine which issued the reserve or release.

Column 4 Virtual reserve/release support is not relevant in this case.

Column 5 When a guest virtual machine issues a reserve CCW to the device, the command is sent ***unmodified*** to the hardware.

Column 7 In this case, there can be no links to a dedicated volume. This column is not applicable.

Column 8 Because the reserve CCW is always passed to the hardware, there are no problems with having multiple paths to this device online. For example, there can be more than one path to this device either from the same or from a different system as long as it is not defined as an alternate path.

Example 2 on how to use "Summary of VM Reserve/Release CCW Support"

This example refers to row 5 in Figure 36 on page 79.

Column 1 The DASD device is defined as a minidisk, either full-pack or not.

Column 2 No alternate paths are defined or online to this device.

Columns 3,6 The reserve/release hardware feature does not exist along the path to the DASD device. Consequently, a command reject will always be returned to the guest virtual machine when it issues either a reserve or a release CCW to this DASD device.

Column 4 Virtual reserve/release support is ***not specified*** for this minidisk.

Column 5 When a guest virtual machine issues a reserve CCW to the device, the command is sent ***unmodified*** to the hardware.

Column 7 ***A data integrity problem exists*** if another user links to this minidisk in read/write (R/W) mode expecting that RESERVE CCWs from the minidisk owner will prevent him from corrupting the shared data.

Column 8 Multiple paths to this minidisk cannot exist because the reserve/release hardware feature does not exist anywhere along the hardware path to this device. (For example, a two channel switch or string switch.)

Hardware for DASD Sharing

Knowledge of the DASD hardware supporting the reserve/release hardware feature is crucial to understanding the software DASD sharing mechanism. Therefore, read this topic in its entirety before continuing with "Using Real Reserve/Release under VM" on page 87.

In hardware for DASD sharing, two base configurations are possible. Either you have a common control unit that is equipped with a two Channel Switch (or four Channel Switch), or you have separate control units. In the latter case, the Head-of-String needs a String Switch feature.[3]

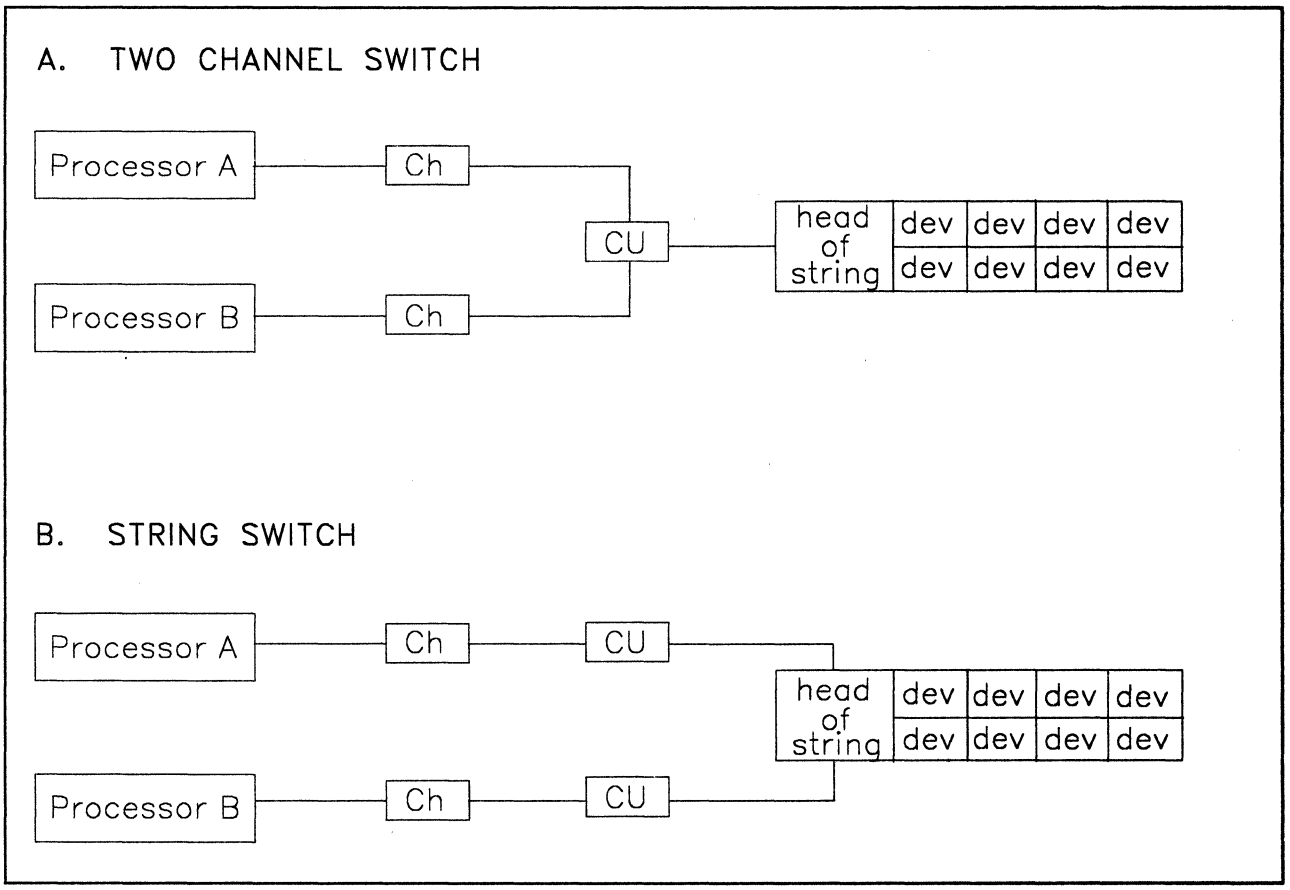


Figure 37. Two-Channel Switch and String Switch

From a DASD sharing standpoint, the two configurations are similar. The VM software supporting shared DASD doesn't see a difference. It is the path to the DASD that matters. Therefore, in Figure 38 on page 84 the channels and the control units were omitted.

---

[3] Technically, the 3380-AA4, -AD4, and -AE4s do not have a string switch feature. However, the equivalent function of a string switch is contained within the Dynamic Path Selection (DPS) feature. This is the only portion of DPS that VM supports and uses.

Sharing DASD devices is not restricted to just accessibility from different paths and different systems. It also means that it must be possible to share the DASD in write mode from different paths. To avoid concurrent update or simultaneous writing — and destroying the DASD contents — the hardware is equipped with reserve/release.

As soon as your hardware has more than one path (For example, either a Two Channel Switch or a String Switch is installed along the path from the channel to the DASD device.); then it has the reserve/release facility as well. If only one path exists for the DASD device and neither a two channel switch nor a string switch exist along that path, the DASD device type will determine whether the hardware will reject the reserve/release commands or not.

For example, IBM 3375s, 3380s, and 3350s will not reject a reserve or release command, whether or not the switching hardware is installed along the path to the device. On the other hand, IBM 3370s and 3330s will issue a COMMAND REJECT to a reserve or release CCW if either a two channel switch or a string switch do not exist on the path to the DASD device.

Optionally, SCP software can use the hardware reserve/release facility. However, complete protection and data integrity in a multiple processor environment is possible only if reserve/release CCWs are used along all paths.[4]

To get a better understanding of how reserve/release works, see Figure 38 on page 84.

---

[4] Using reserve/release CCWs is not the only way to ensure data integrity. Additionally, the software can have its own locking system. For example, VSE uses its own software locking mechanism to support shared DASD. The hardware reserve/release facility is used only to protect the VSE Lock File.

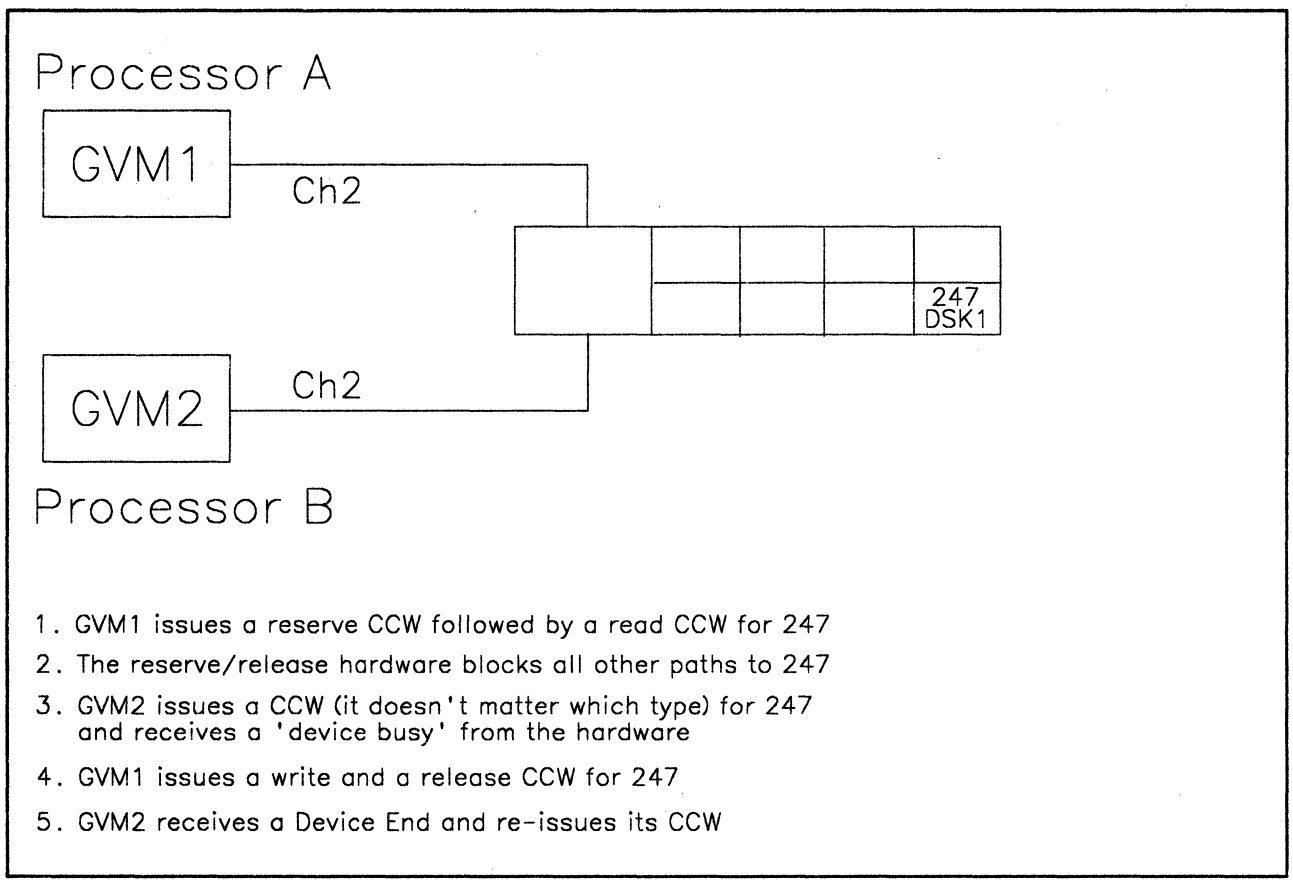


Figure 38. Reserve/Release Hardware

A reserve CCW is an I/O command that is sent from an operating system to a channel. It reserves a single DASD device to a particular channel on a specific processor for its own exclusive use. This is accomplished through the reserve/release hardware contained in either the DASD control unit or DASD head-of-string. Essentially, the reserve/release hardware restricts access to this particular DASD device to a specific channel/control unit/head-of-string path. A reserve CCW is treated as a path reservation.

Once a path is reserved, the device can only be accessed via this path. All access to the device from other systems using a different path results in a device busy condition.

The release CCW is sent via the reserved path only. It ends the path reservation for this user. All paths that received a device busy condition will receive a Device End (DE) indicating that the device is now available. The device can now accept I/O operations from all paths.

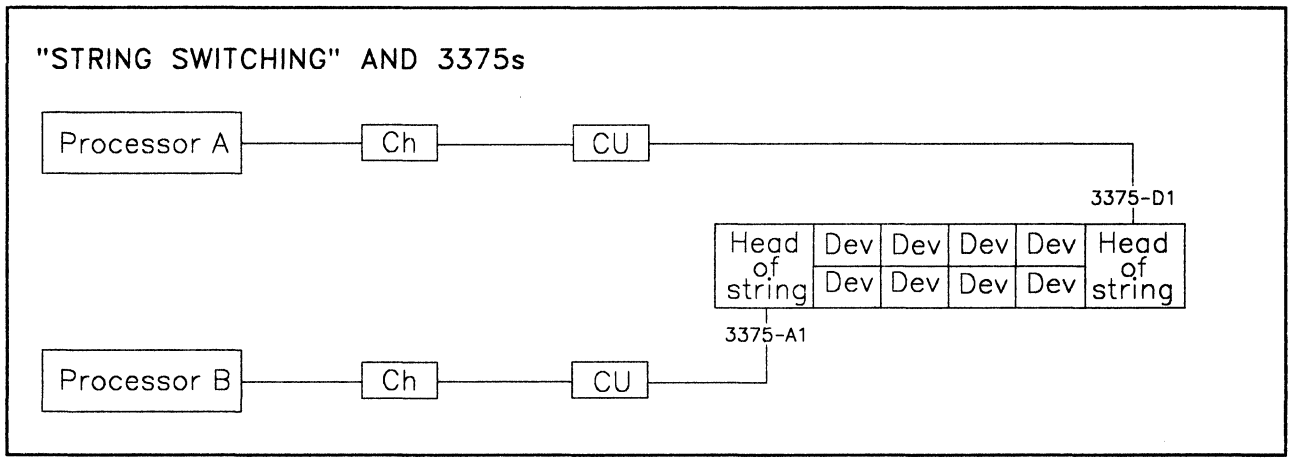


Figure 39. 3375 Configuration with Two Heads-of-String

With 3375 and 3380 - AA4 DASD devices the situation is slightly different. In Figure 39 we have a 3375 string connected to Processor-A through the 3375-D1 unit and to Processor-B through the 3375-A1 unit. Confusion arises when we try to understand how an I/O coming across the path from Processor-A through the 3375-D1 will detect a device reservation which was made from Processor-B through the 3375-A1.

The DASD device does not have reserve/release status information or logic of its own; all of the device reservation status and logic is handled through the 3375 head-of-string.[5]

When a reserve CCW is sent across the path from Processor-A through the D1-unit, the head-of-string updates its device status information and internally sets up the path reservation to that device so that no other path can access it. Also, the D1-unit will signal this path reservation status to its corresponding A1-unit so that the A1-unit can update its reservation status for that shared device. Therefore, both head-of-strings support the device reservation and are aware of the reservation status contained in each other.

The release CCW for this device must be made across the same path as the original reserve CCW in order to be accepted by the hardware. When the D1-unit receives the release CCW for this device, it will release the path reservation, update its own device status information, and inform the A1-unit of this status change. Once again, this device can be accessible from all paths.

---

5 This is one of the main reasons why VM does not have to be concerned with whether the DASD sharing is implemented through a string switch or a two channel switch. The actual reservation status has to be maintained in the head-of-string mechanism. This applies to all IBM DASD such as the 3350s, 3330s, 3370s, 3375s, and 3380s.

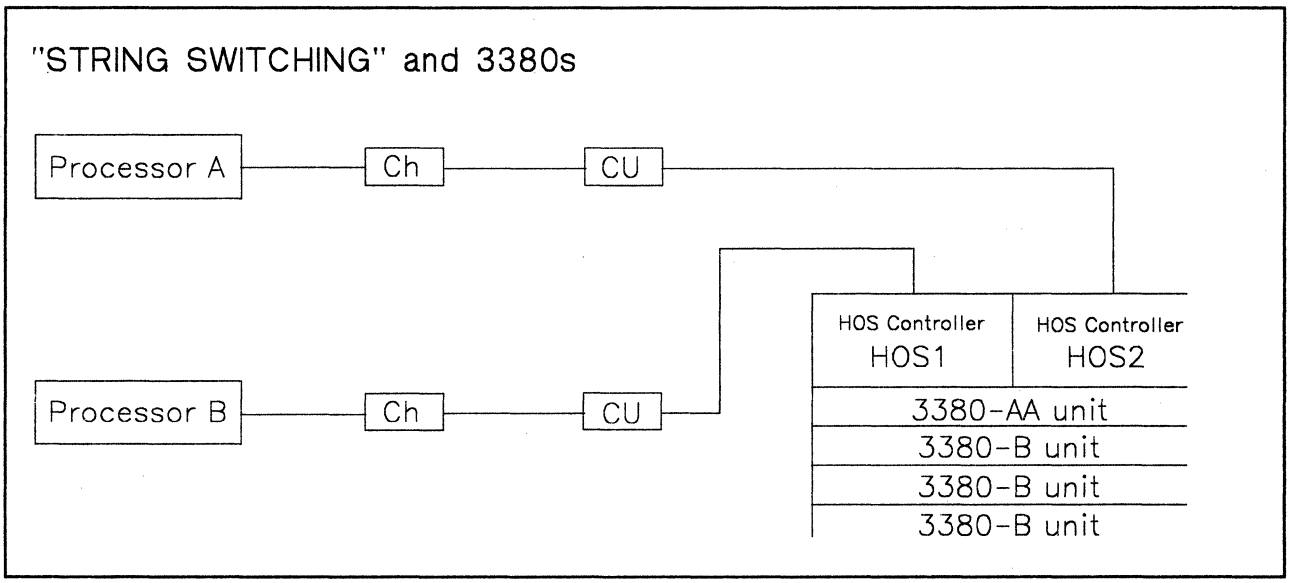


Figure 40. 3380 AA-4 Configuration (Two Heads-of-String)

From a conceptual point of view, the 3380-AA4 handles the hardware reserve/release logic in a VM/SP environment just like the 3375-A1/D1 combination. Comparing Figure 40 with the 3375-A1/D1 example, the 3380 Head-of-String Controller, HOS1, corresponds to the 3375-A1 and HOS2 corresponds to the 3375-D1.

Apart from the software, both HOS1 and HOS2 can have access to any device on that 3380 string just as the 3375-A1 and the 3375-D1 can have access to any device on their 3375 string. Functionally, the synchronization of the device reservation status is the same for 3375-A1/D1 combinations and 3380-AA4s.

Neither VM/SP nor VM/SP HPO support the full DPS feature. This precludes VM from using the "system-related reserve".

Using Real Reserve/Release under VM

Device reservation works for the path to the DASD device, whether it comes from different processors or not.

Consider the case in Figure 41. Two virtual machines use a dedicated path, each to the same string of DASD. To be able to do so, we must tell CP that we have each device (and control unit) twice.[6]

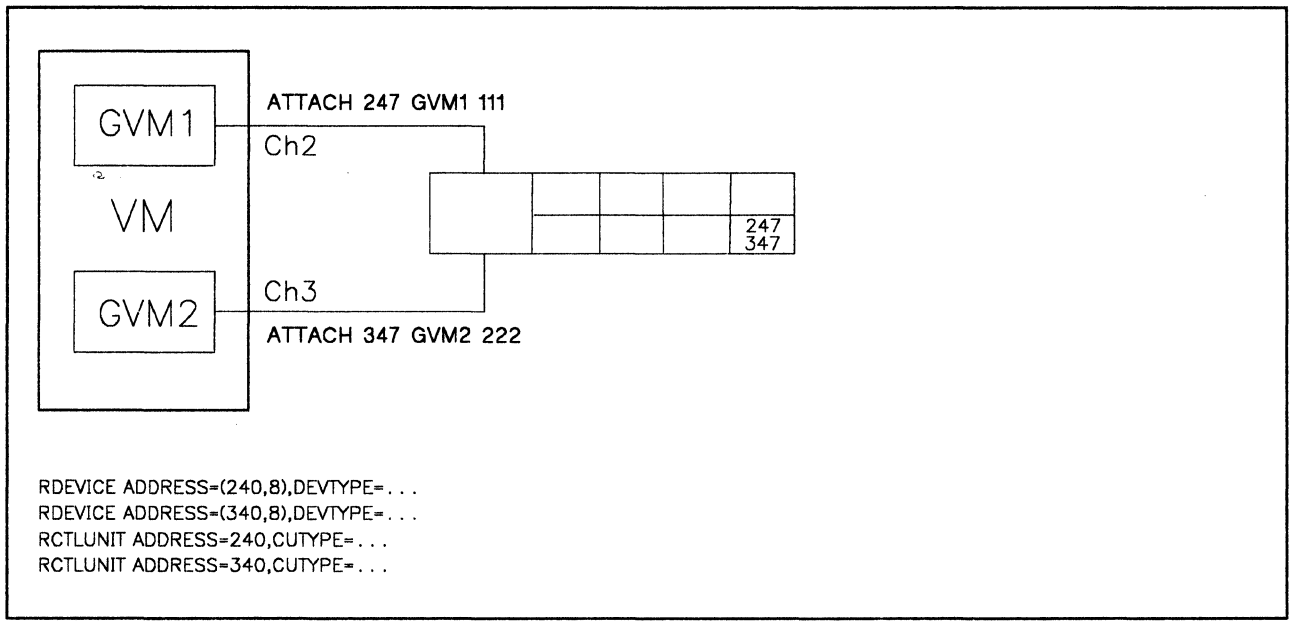


Figure 41. Reserve/Release on Dedicated Paths (Example 1)

---

6 At IPL time, CP sees two paths and two addresses for this string of DASD. Because the second path is not defined to VM as an alternate path, CP varies offline the devices with the higher addresses. (For example, 340-347.) However, you can vary the devices back online and attach those devices to the second guest operating system (GVM2).

VM/VSE Sharing DASD with a Stand-alone VSE System

VSE uses the Lock File to logically protect resources against parallel updates by different VSE systems. If separate access paths are available, hardware assistance is necessary to ensure that only one VSE system at a time can write to the Lock File.

This hardware assistance is obtained by using reserve/release CCWs. The release CCW will be accepted by the hardware only if it is issued along the same physical path that the reserve CCW was presented. VM itself never uses reserve CCWs. As mentioned earlier, the hardware would reject the release CCW not coming via the same path, and the DASD would never be available. Figure 41 on page 87 gives us our first rule for DASD sharing under VM.

RULE 1: FOR DEDICATED OR ATTACHED DASD

If the reserve/release hardware IS present and no alternate paths are online, CP sends the reserve/release CCWs unmodified to the hardware. [7]

If all paths to a DASD device are dedicated, and if the guest operating systems use reserve/release CCW, then there is no VM data integrity problem. In this respect, running natively or using a dedicated path under VM are functionally equivalent modes of operation.

In Figure 42 on page 89, we have only one guest operating system (GVM1) sharing DASD with an operating system (VSE) on another processor. The DASD is dedicated to the virtual machine. According to Rule 1, CP sends the reserve/release CCWs unmodified to the hardware. Therefore, data integrity is ensured. (Figure 42 on page 89 is still valid if (VSE) is running as a guest system on processor B.)

---

[7] For more information on reserve/release and alternate paths see "VM Alternate Path Support and Reserve/Release" on page 95.

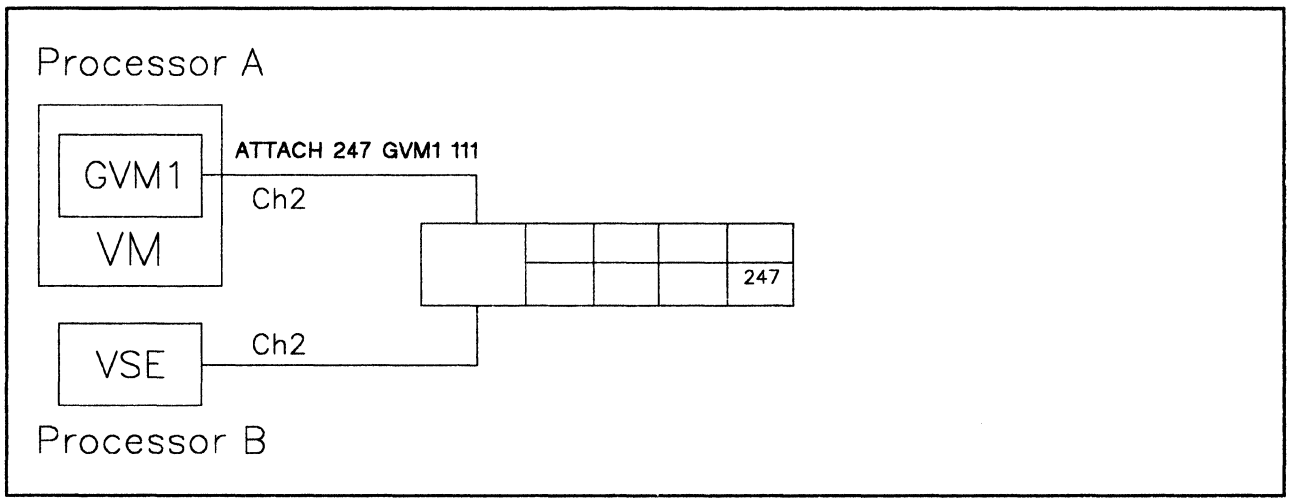


Figure 42. Reserve/Release on Dedicated Paths (Example 2)

Virtual Reserve/Release

Consider the case of two guest operating systems wanting to share DASD but **only one physical path exists** from your processor to the DASD device. Under VM, in order to share this device it must be defined as a minidisk for one of the guests. The other guest will have to link to this minidisk, as usual.

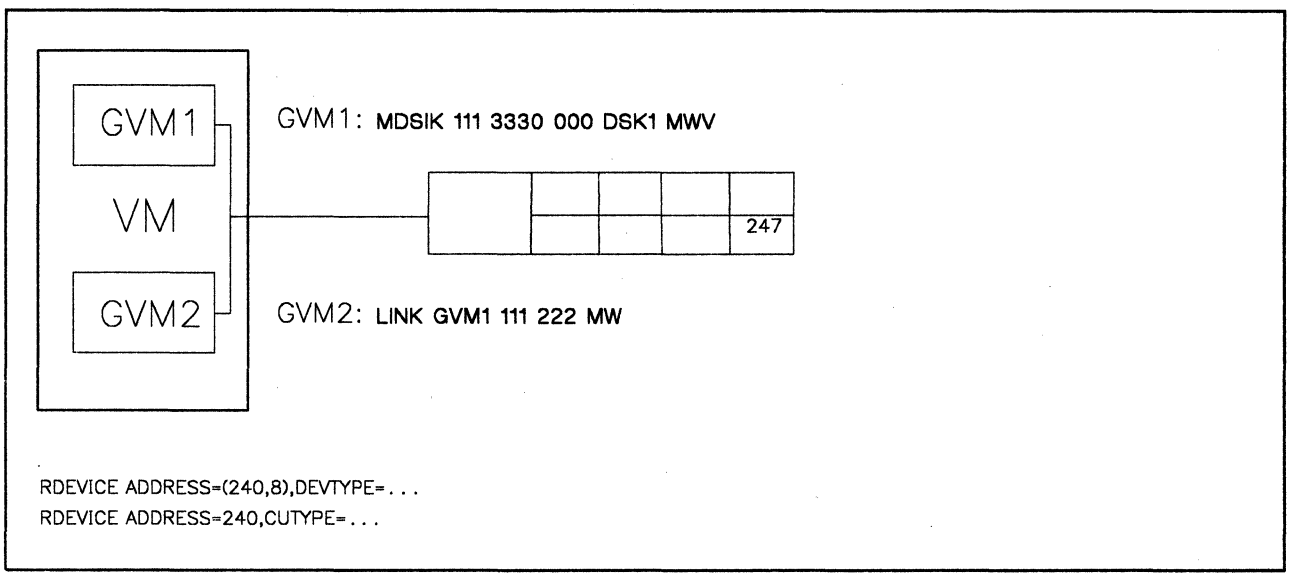


Figure 43. Virtual Reserve/Release

In Figure 43, the use of the hardware's real reserve/release facility leads to an integrity exposure. The reserve/release hardware (if present at all) cannot do its job because the read and write requests from GVM1 and GVM2 must travel along the same path.

However, VM has a software simulation facility to handle this, virtual reserve/release. It is valid only for minidisks and is specified by an access mode of MWV within the directory entry.[8]

For VSE virtual machines using MDISK/LINK you must define access mode MWV for the shared minidisk containing the Lock File. Virtual reserve/release ensures that the example in Figure 43 works and that the data integrity mechanism of each guest operating system can operate just as if it was not running under VM.

---

8 Note that MWV must be in the MDISK control statement. Putting the MWV in the LINK control statement in the DIRECTORY will be flagged as an error.

Virtual reserve/release executes in CP only. If a guest issues a reserve CCW to protect the device from being accessed by other operating systems on the same processor complex, CP will flag this minidisk as being reserved for that particular virtual machine. It reserves the access to the minidisk, just as the real reserve/release hardware would reserve access to the real disk.

Because the virtual reserve/release facility executes only in CP, the reserved minidisk can be of any size. It does not need to be a full pack minidisk.[9] Virtual reserve/release can work for several independent minidisks on the same volume as long as the volume ***is not shared*** with another processor complex.

If you have specified MWV in your MDISK definition, your guest can issue a reserve CCW and CP will reserve the minidisk accordingly.

RULE 2A: FOR A MINIDISK WITH VIRTUAL RESERVE/RELEASE

If the reserve/release hardware IS present and no alternate paths are online, then CP sends the reserve/release CCWs unmodified to the hardware.[10]

Because the hardware handles the CCW, a reserve/release for a minidisk will always result in a reserve/release for the whole DASD volume on which this minidisk is defined if no alternate paths are online.

So, in addition to the virtual reserve/release protection in CP, you are protected against access via other paths by the real reserve/release hardware. These other paths can lead to a different processor, or to the same one (dedicated path). This case is explained in Figure 44 on page 92.

---

9 However, there are several very good reasons for using only full-pack minidisks. These reasons will be discussed later.

10 For more information about reserve/release and alternate path support see "VM Alternate Path Support and Reserve/Release" on page 95.

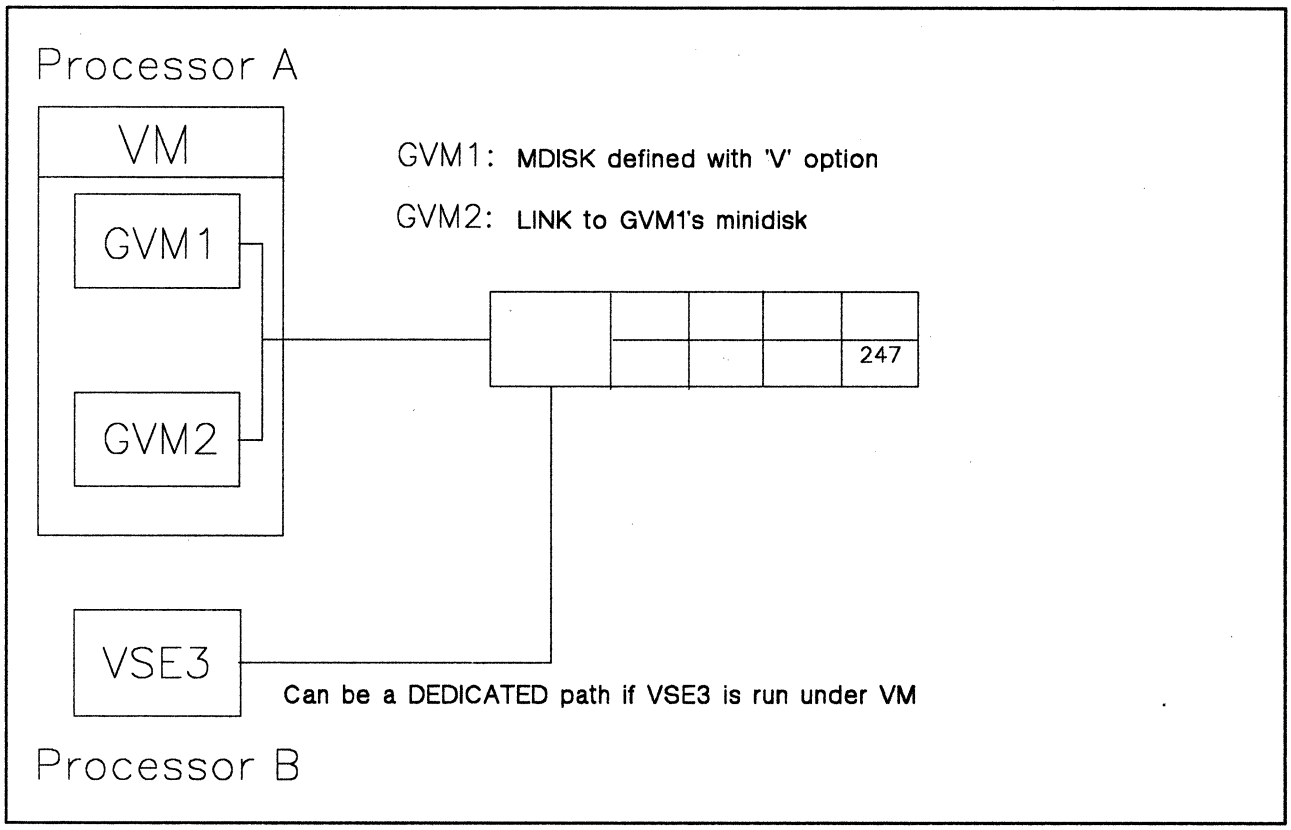


Figure 44. Virtual and Real Reserve/Release

In Figure 44 there are three operating systems with write access to the same disk. GVM1 and GVM2 are under VM and are using virtual reserve/release. VSE3 is executing natively on a separate processor. Figure 44 applies whether VSE3 is executing natively or as a guest on a separate processor.

The locking structure works as desired for all three operating systems.

1. GVM1 and GVM2 are protected against each other by CP.

2. GVM1 and GVM2 together are protected against GOS3 and vice versa by the hardware.

Note that the software mechanism works for minidisks. If you do not use full-pack minidisks the hardware still reserves the complete pack. It is advisable to use full pack minidisks only, especially if these disks are to be shared by using the hardware feature, see "Sharing Minidisks" on page 98.

VSE DASD Sharing without Hardware Switches

For sharing DASD between several VSE virtual machines, it is not necessary for the hardware to be equipped with channel switches and/or string switches. VM allows multiple access to the same DASD.

The VSE virtual machine tests and handles the environment in the following ways.

- At IPL time, VSE issues a release CCW to all of its DASD that have been defined as shared. If the release CCW returns a CC=0 (condition code 0), VSE allows that disk to be shared and uses its own software locking mechanism to control the sharing.

- VSE does not inform its operator that it has disallowed certain volumes from being protected because it received a check condition code from the release CCW sent to it at IPL time.

- When the release CCW is issued, if all volumes defined as shared return a COMMAND REJECT, VSE informs the operator that the locking facility is not active. As a result, if you want to find out if a particular volume is being protected by the VSE locking mechanism, you must issue the VSE VOLUMES command (after VSE has been initialized under VM) to find out if that volume retains the shared status you requested. If it doesn't, you have a data integrity exposure for that volume.

- Lock requests for resources on these DASD would not be written to the Lock File.

- The integrity of the VSE system is not maintained as other virtual machines would not be notified about the LOCK.

When VSE is running under VM, DASD containing the shared data may be defined as minidisks. Multiple access to the same devices under VM is provided through MDISK/LINK definitions. The shared minidisk containing the Lock File to be defined must have access mode MWV on the MDISK definition in the VM directory. If access mode MWV is defined and the DASD does not have hardware switches, VM modifies the reserve/release CCW to a sense CCW. The sense CCW will never get back an error code; therefore, VSE accepts the SHR definition for these devices.

RULE 2B: FOR A MINIDISK WITH VIRTUAL RESERVE/RELEASE

If the reserve/release hardware IS NOT present, CP modifies the reserve/release CCWs into sense CCWs before sending them to the hardware.

A sense CCW returns a condition code that is similar to that of a successful reserve or release CCW, but without doing anything. If this CCW modification were not done, the guest operating system would receive a COMMAND REJECT and would "think" the devices are not shared. For example, it would not issue reserve/release CCWs anymore. But as CP is simulating the hardware reserve/release facility, we want the guest virtual machine to believe that the facility exists.

RULE 3A: FOR A MINIDISK WITHOUT VIRTUAL RESERVE/RELEASE

If the reserve/release hardware IS present and no alternate paths are online, CP sends the reserve/release CCWs unmodified to the hardware. [11]

In this case you have no protection from other virtual machines along the same path using a LINK to the minidisk. At IPL time, the VSE guest issues a release CCW to check whether this DASD can be shared (as defined in the ASIPROC). If the reserve/release hardware is not present, the VSE guest operating system will get back a COMMAND REJECT on this first release CCW and will regard the disk as not shared. It will not include this DASD into the Shared DASD Support via the Lock File.

RULE 3B: FOR A MINIDISK WITHOUT VIRTUAL RESERVE/RELEASE

If the reserve/release hardware IS NOT present, CP sends the reserve/release CCWs unmodified to the hardware and the guest virtual machine receives a COMMAND REJECT error condition from the hardware.

---

[11] For alternate path, see "VM Alternate Path Support and Reserve/Release" on page 95.

VM Alternate Path Support and Reserve/Release

In the previous sections we spoke about path protection through real and virtual reserve/release. We will now see how we must modify our picture if an alternate path exists for the path we want to protect. Consider the case of two guest operating systems wanting to share DASD but one of the paths has an alternate channel or control unit specified (ALTCH or ALTCU in DMKRIO), and this alternate path is online.

RULE 4:

If the defined alternate path to the device is online, CP modifies the reserve CCWs into sense CCWs before sending them to the hardware. ALWAYS!

The release CCWs are sent unmodified.

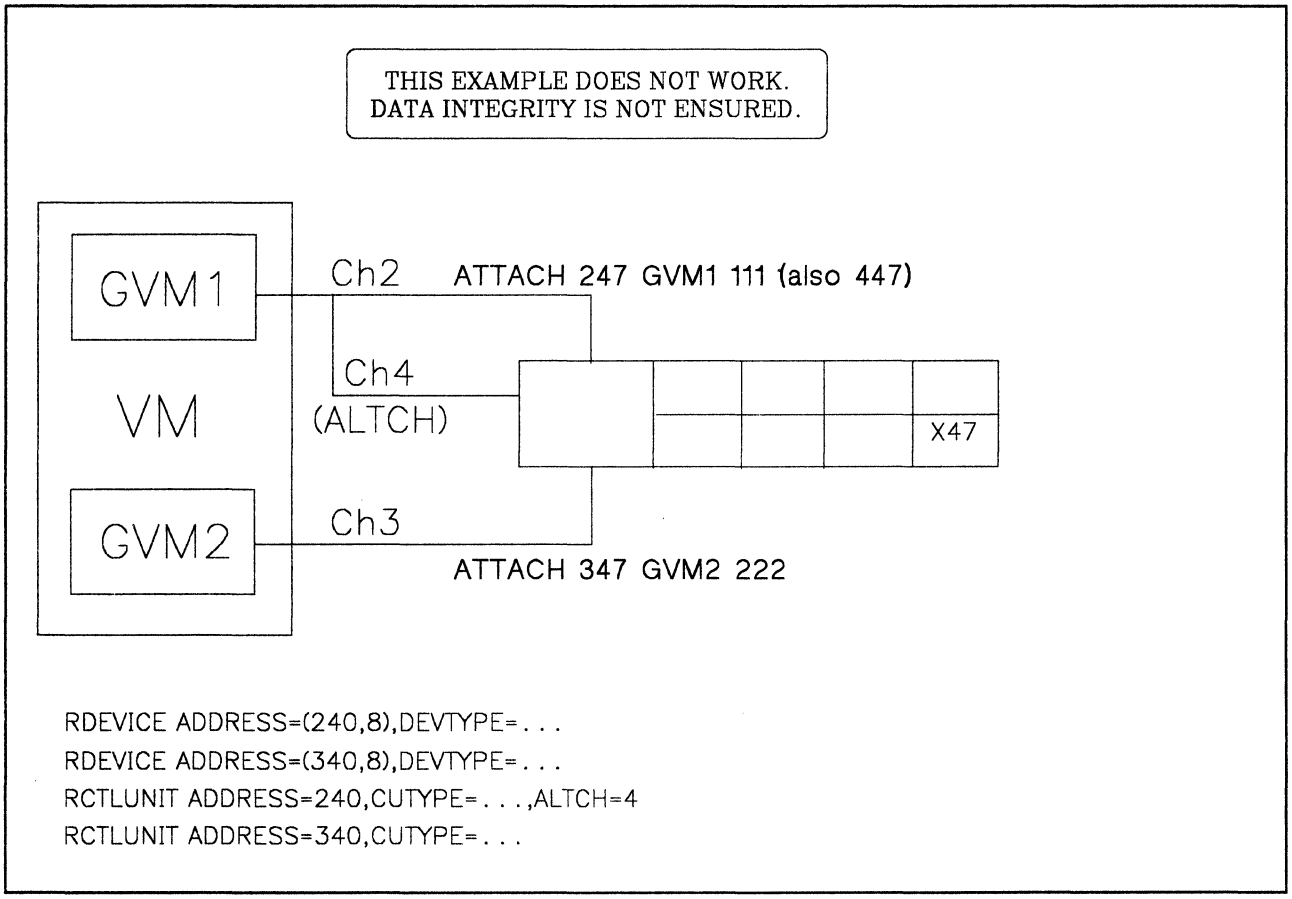


Figure 45. Alternate Path and Dedicated Disk

When you attach a device with an alternate path to a virtual machine, CP automatically considers the alternate address as also attached. The guest knows only one address. This means that the guest cannot issue a SIO(F) directly to the alternate path address.

In Figure 45, a reserve CCW issued from GVM1 doesn't go through to the hardware. As a consequence, there is no protection from GVM2 accessing the DASD while GVM1 has access to this device in write-mode.

Warning: CP does not inform the guest operating system of the change made to the reserve CCW. Therefore, the guest operating system believes that the disk is shared and protected.

To circumvent this problem you must suppress the alternate path logic, and be sure that only one path is online! You might want to switch to full pack minidisks and link to them with virtual reserve/release.

As another example, consider the case of two guest operating systems wanting to share DASD (defined as minidisk) with one physical and alternate path.

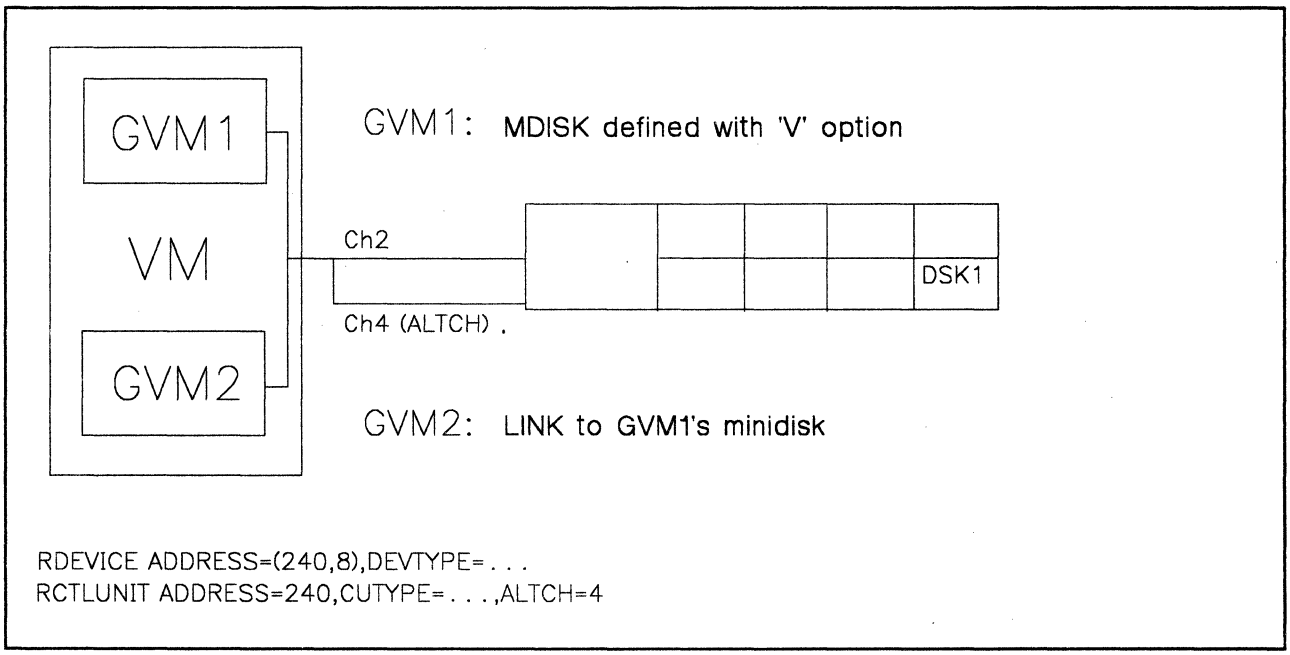


Figure 46. Alternate Path and Minidisk

The reserve CCWs are modified to sense CCWs before they are sent to the hardware. Protection is still maintained for this environment since it is all done in CP. In this case, the modification is of no importance. However, a problem arises in the following example if the DASD is to be shared with another processor or with a dedicated path on the same processor.

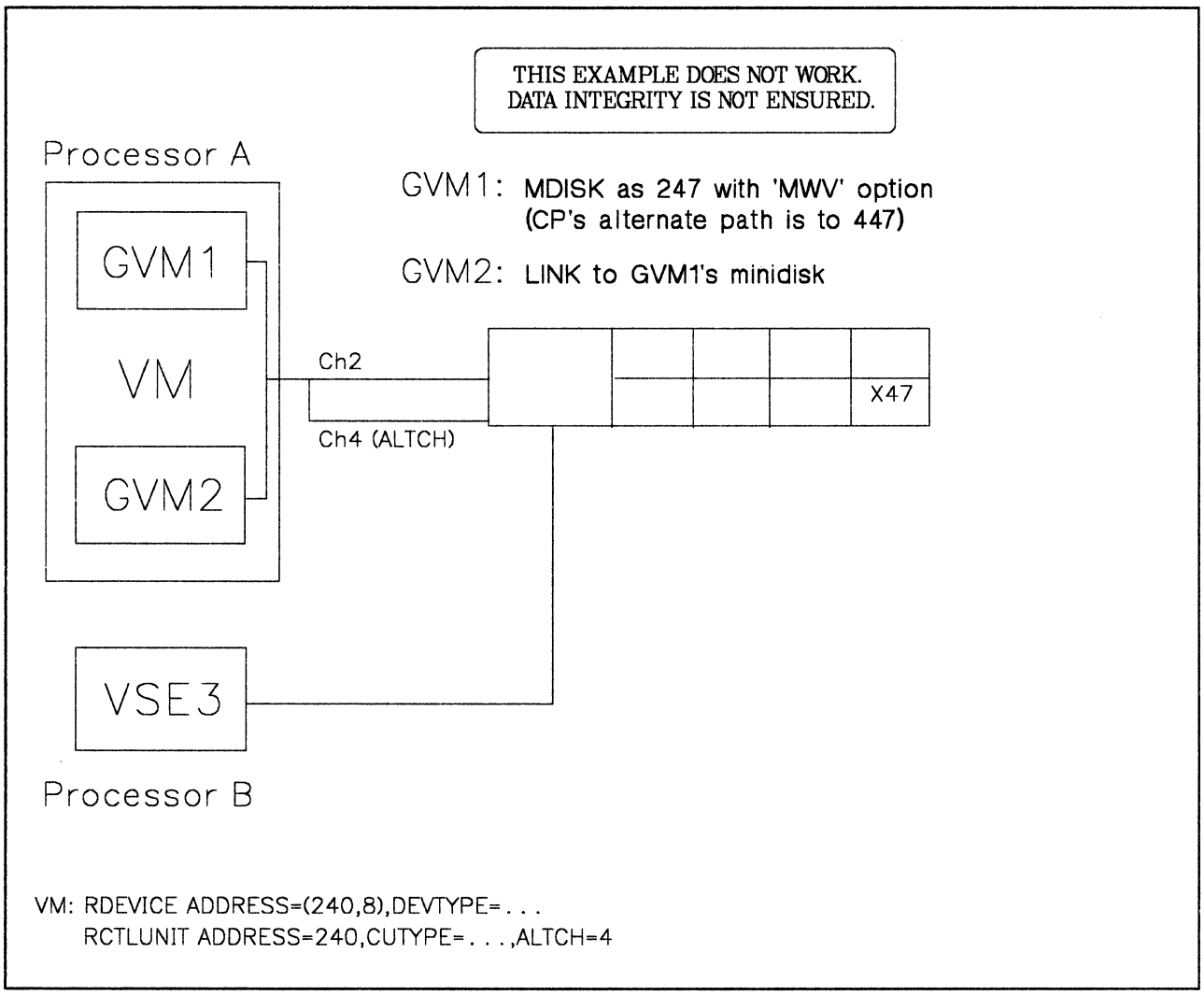


Figure 47. Alternate Path and Minidisk in Multisystem Environment

In Figure 47, there would be no protection against VSE3 accessing the minidisk while one of the guest operating systems (GVM1 or GVM2) has an access to this minidisk in write-mode. The only solution is to avoid the alternate path.

Sharing Minidisks

Assume you are running two real processors and at least one of them is running VM along with two guest operating systems, GVM1 and GVM2. In such an environment you can find several minidisks defined on one real pack. Depending on the sharing of these minidisks and the SCPs running in the guests you can run into severe problems.

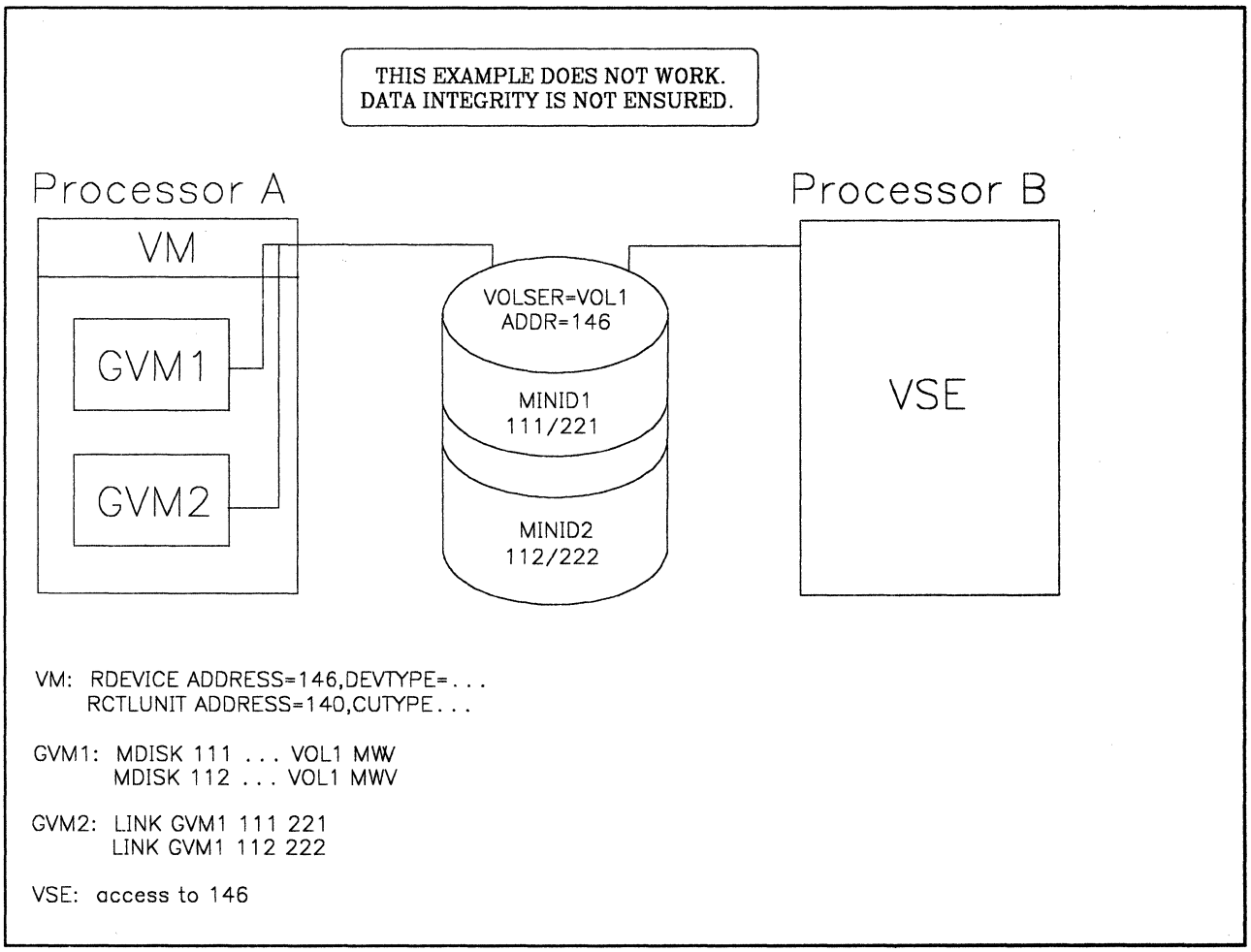


Figure 48. Sharing Minidisks

In this example we have a VM system running on Processor-A with two virtual guests sharing two minidisks, MINID1 and MINID2. Both virtual machines have been defined on real pack VOL1 (mounted on address 146) and are using the virtual reserve/release facility. ***In this example there exists only one path from Processor-A to VOL1.***

Processor-B is running a native SCP. The SCP wants to share the data on MINID1 at the beginning of the real pack **(and can use only this part of the pack)**.

The DASD definitions have to be defined in the ASIPROC specifying whether this DASD can be shared.

To ensure data integrity, the following VSE definitions should be contained in their respective ASIPROCs:

GVM1

```
ADD 111,33xx,SHR
ADD 112,33xx,SHR
 ....
```

GVM2

```
ADD 221,33xx,SHR
ADD 222,33xx,SHR
 ....
```

VSE

```
ADD 146,33xx,SHR
 ....
```

During system initialization, VSE checks to see whether the DASD you want to share are on devices that can be shared (For example, those devices where the reserve/release hardware is present). For this reason, VSE sends a release CCW to all DASD addresses defined as shared and checks the condition code. After this, VSE uses reserve/release CCWs only for the Lock File. If the reserve/release hardware IS NOT present, you need to specify MWV for all shared minidisks.

RULE 5:

If you are using a minidisk for your Lock File in a multiple-processor VSE environment no other minidisk on this particular pack must be defined as shared (with the MWV option).

Assume GVM1 and VSE are using MINID1 (111/221) for the Lock File and the third system (GVM2) is started up. Due to the definitions GVM2 sends a release CCW to MINID2. Because virtual reserve/release support was defined for MINID2, CP checks to see if this minidisk is reserved. As no system has made a reservation for this address (VSE uses reserve/release only for the DLF!) the release CCW for MINID2 is sent to the hardware thereby freeing the whole pack. **THIS CAN LEAD TO A DATA INTEGRITY EXPOSURE.**

VSE DASD Sharing with Hardware Switches but within One Processor

If switchable hardware is available, you should make use of the additional access paths for availability. The ALTCH or ALTCU definitions in the VM DMKRIO and the MDISK/LINK definitions in the directory for the VSE virtual machines would still use the primary path for the first try and then switch to the alternates.

If you have enough real channels and DASD, you can define in VM DMKRIO separate access paths and dedicate one path to one VSE virtual machine. Defining separate access paths has some performance advantages over the MDISK/LINK definition. The addresses on the additional access paths have to be varied online after VM startup, because VM will vary them offline due to the duplicate volume IDs.

Note: Data integrity can not be maintained if a dedicated pack is used for the Lock File and an alternate control unit (ALTCU) or alternate channel (ALTCH) is defined and online.

Part Two. MVS/SP under VM/SP HPO

The procedure that follows assumes that you have your MVS and VM/SP HPO systems up and running. This procedure does not provide any assistance in the tasks involved in bringing up either systems. Also, if you are not sure of all the basic functions of VM/SP HPO, please take the time to review them before you proceed.

Chapter 1. Introduction to Running MVS/SP under VM/SP HPO

You can use VM/SP HPO in many different ways. How you choose to use it depends on your hardware configuration and the kind of work that you want to do.

This section describes the basic ways you can operate VM/SP HPO. It shows you how MVS and VM/SP HPO can exist in the same real system, and how you can take advantage of your processor complex to the greatest benefit of your installation.

There are two terms that will be used throughout this section:

- **Environment** refers to the hardware configuration and the resources available to a control program.
- **Modes of operation** refers to the ways the control program can be operated to let you effectively use the environment.

This section makes distinctions between the different ways in which processors can be **generated** or **operated**. It generally does not discuss different processor models.

The Main Types of MVS Guests

There are many ways to run MVS under VM/SP HPO, but all types fall into one of two main categories:

- V=V guests, which run in the dynamic paging area of CP
- V=R guests, which run in the V=R area of CP.

MVS V=V Guests

The storage of an MVS V=V (virtual=virtual) guest is allocated from the dynamic paging area of CP.

This is the most flexible way of allocating real storage. In this way, only the active working sets of active MVS guests need to be in real storage at any given time. However, since both MVS and CP will perform paging, there is considerable CP overhead in managing real storage. CP uses **shadow tables** to reduce the overhead of double paging, but CP still needs to do work to maintain shadow tables. Shadow table support is described in Chapter 3 of this book.

MVS V=R Guests

If you have a high-priority MVS virtual machine, you will probably want to dedicate a portion of real storage for it. The real storage in which this guest runs is called the **V=R area** (virtual=real area) and the guest is called the **V=R guest**. Because the guest runs in real storage, CP does not need to process page faults for it.

You can run only one V=R guest at a time.

On processors with preferred machine assist, you can run the V=R guest as a **preferred guest** and improve its performance even more.

Where MVS Runs in the Real Storage of VM/SP HPO

The following figure shows how VM/SP HPO accommodates an MVS V=R (virtual=real) guest and more than one MVS V=V (virtual=virtual) guest. This figure focuses on the allocation of storage below the 16 Mb line. In this figure, the **V=R area** is where you can run an MVS guest in real storage. The **CP nucleus area** is where all resident CP routines are located. The **dynamic paging area** is where CP uses spool file buffers. It is also used by VM/SP HPO to run virtual machines such as MVS V=V guests or CMS applications, and contains pageable CP routines.

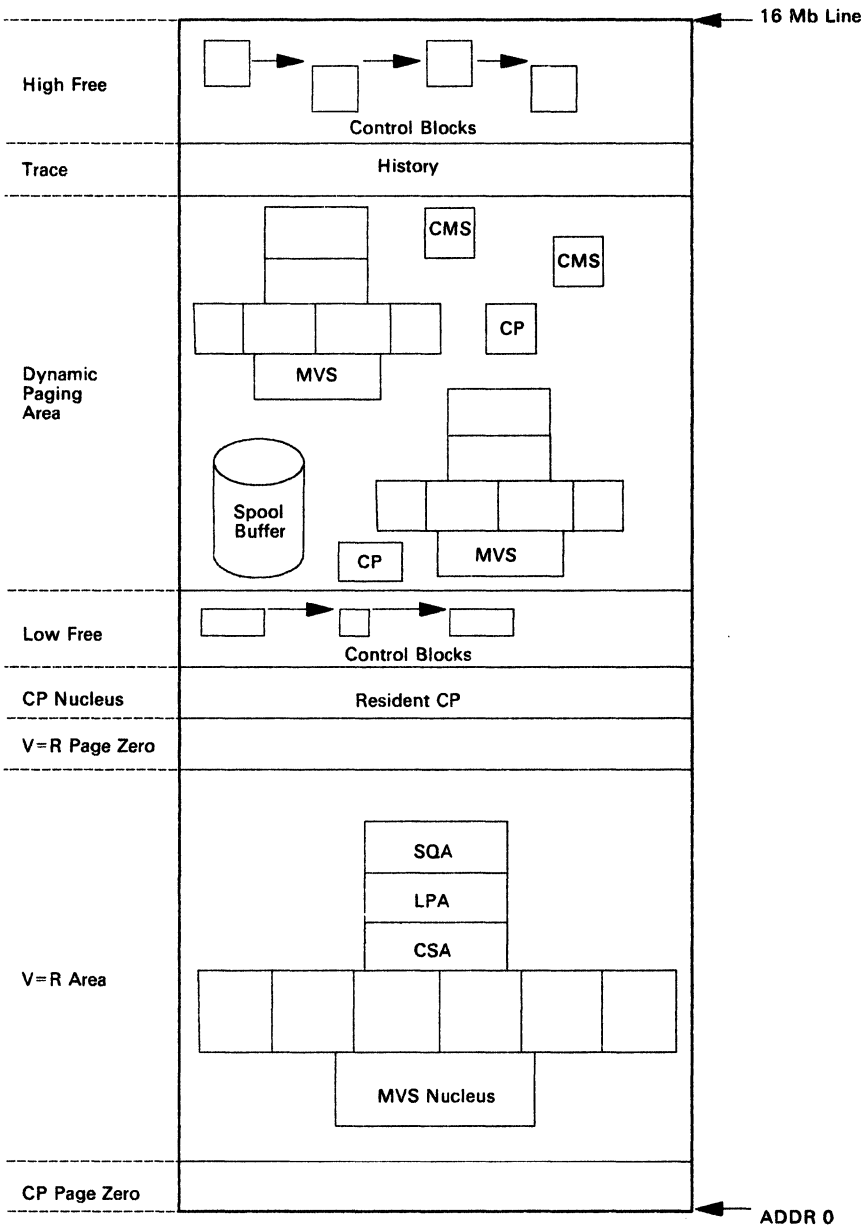


Figure 49. Comparison of VM/SP HPO and MVS Storage Layouts

Specifying How You Want CP to Use Real Storage

Use the SYSCOR macro of DMKSYS ASSEMBLE to tell CP how to maintain real storage. At system generation time, CP will build a **CORTABLE** to maintain real storage according to your specifications.

The **RSSIZE** parameter of the SYSCOR macro specifies the size of the CORTABLE— this value is the maximum real storage size that CP will use. RSSIZE is an optional parameter; if you do not specify it, it will default to the value of RMSIZE.

The **RMSIZE** parameter of the SYSCOR macro specifies the amount of real storage available to VM/SP HPO.

Using Real Storage above the 16 Mb Line

On processors with more than 16 Mb of storage, it is important to use proper values for the RMSIZE and RSSIZE parameters. The combined use of these two values determines the partitioning of real storage between CP and an MVS preferred machine assist guest.

The storage between the 16 Mb line and RMSIZE is used by CP as an extension to dynamic paging storage. CP uses this storage for V=V guest virtual machines.

The storage between RMSIZE and the top of storage is reserved for the exclusive use of the MVS preferred guest. The only guest operating system that can utilize such a split of real storage is MVS/SP Release 1.3.1 and later.

Types of Processors on Which VM/SP HPO Can Run

VM/SP HPO can **run** on the following types of processors:

Uniprocessor One processor controls real storage and I/O devices.

Multiprocessor Two processors share a common real storage.

- True multiprocessor (MP) — two processors that both have I/O capability. You can partition these into two independent uniprocessors. A 3033 MP is an example of a true multiprocessor.
- Attached processor (AP) — only one of the two processors has I/O capability. A 3033 AP is an example of an attached processor.

- Dyadic processor (DY) — can be a multiprocessor or an attached processor. You cannot, however, partition the system into two independent uniprocessors. A 3081 is an example of a dyadic processor.
- Dual processor — is similar to a dyadic processor except that the channels are integrated processors. A 4381-13 is an example of a dual processor.

How You Can Generate CP for Different Modes of Operation

You can generate CP for UP, AP, or MP modes of operation. Before you decide how to generate CP, consider how you can best use it, the performance implications of your choice, and the I/O configuration of your processor.

Using UP-Generated VM/SP HPO Systems

You should generate CP in UP mode only for use on UP processors. On AP and MP processors, you should generate VM/SP HPO to run in either AP or MP mode regardless of final operational environment.

Using AP-Generated VM/SP HPO Systems

In AP mode, VM/SP HPO controls both processors but uses only one channel set for I/O. The attached processor can be used by VM/SP HPO, or can be dedicated to the MVS guest by using single processor mode.

Other Considerations When You Use AP Mode

- VM/SP HPO **does not support** virtual AP mode. If both processors run under the control of VM/SP HPO, a virtual machine (even a preferred guest) can run only in UP mode and therefore can use no more than 50% of the processing capacity of the processor complex.

  Note: Single processor mode is not virtual AP mode since VM/SP HPO controls one processor only. The MVS guest runs natively on the other processor.

- There is only one channel set available for VM/SP HPO. Therefore, the maximum number of channels that VM/SP HPO can use for its virtual machines is 16.
- VM/SP HPO can use only one processor for I/O. But if the I/O processor fails and the hardware supports **Channel Set Switching**, an AP-generated VM/SP HPO system will attempt recovery by switching the channel set to the operational processor.
- A preferred guest with affinity set to the non-IPL processor will have exclusive use of the channels on that processor.

Using MP-Generated Systems

In MP mode, VM/SP HPO controls both processors and uses two channel sets for its I/O.

Consider running VM/SP HPO in MP mode if your installation has a heavy VM I/O load. This way, the I/O load can be routed through two channel sets and up to 24 channels.

Other Considerations When You Use MP Mode

- VM/SP HPO **does not support** virtual MP mode. Single processor mode is not virtual MP mode. In single processor mode, MVS runs in MP mode with one processor running under VM/SP HPO and the other running natively under MVS. If both processors run under the control of VM/SP HPO, an MVS virtual machine will run in UP mode and exploit the power of only 50% or less of the entire processor complex.
- There is no Channel Set Switching when you use MP mode.
- If you are running an MP VM/SP HPO system with a UP preferred guest on one processor, be careful when configuring preferred channels. Since the preferred guest must have affinity to one processor and will have exclusive use of the channels not generated in DMKRIO on that processor; the corresponding channel number on the other processor cannot be used by VM/SP HPO or MVS.

The Uniprocessor Environment

The simplest environment in which to run VM/SP HPO is on a uniprocessor. On a uniprocessor, you can run an MVS guest in four ways:

- As a V=V guest
- As a V=R guest
- As a V=R guest with preferred machine assist
- As a V=R guest with preferred machine assist and control switch assist.

Figure 50 on page 109 shows a storage layout of a UP system configured to support only V=V guests. This figure shows a single uniprocessor controlled by VM/SP HPO. This is a typical storage layout for a VM/SP HPO system on a uniprocessor with no V=R guest. You can run multiple V=V MVS guests, but the performance of these guests may not be acceptable for production work.

Figure 51 on page 110 shows a storage layout of a UP system configured to support a V=R guest. In this figure, CP owns absolute page zero, and the MVS/SP real page zero resides in the high end of the V=R area. VM/SP

HPO still owns the processor, so CP is the only code that runs in supervisor state. Guest operating systems, even a V=R guest, run in real problem state (simulated supervisor state). Note that all I/O on the system is controlled by VM/SP HPO.

Figure 54 on page 114 shows a storage layout of a V=R guest with preferred machine assist.

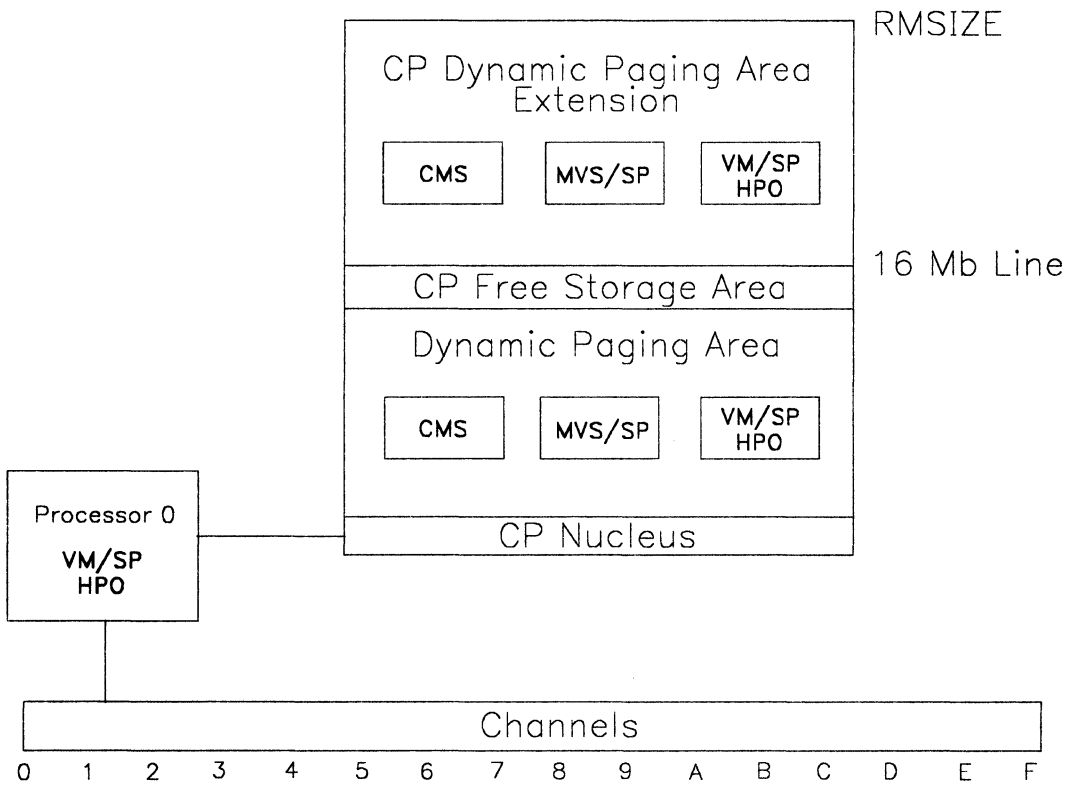


Figure 50. Storage Layout of a UP System with No V=R Area

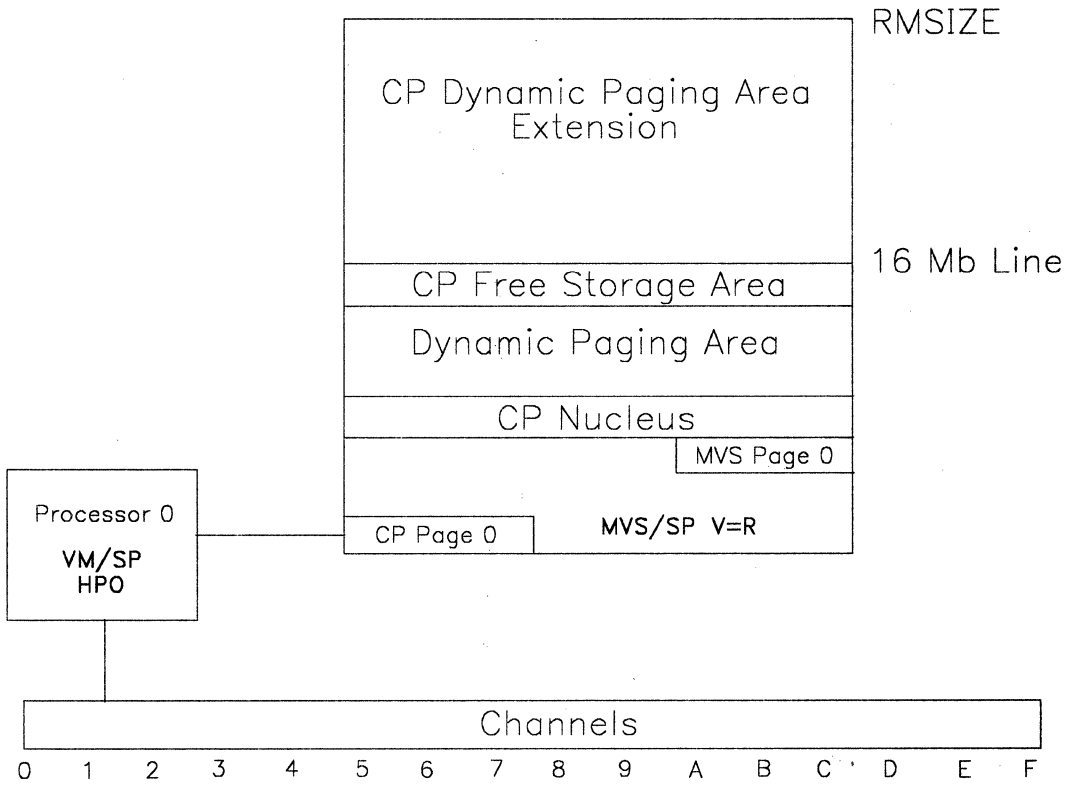


Figure 51. Storage Layout of a UP System with a Nonpreferred V=R Guest

The Multiprocessor Environment

Multiprocessors include true multiprocessors, attached processors, dyadic processors, and dual processors.

To understand the implementation of MVS/SP running under VM/SP HPO, you must understand the hardware facility called **prefixing**. Prefixing is the mechanism that allows more than one processor to use a single real storage and share the fixed storage addresses (page 0) to communicate with the hardware.

Each processor in a multiprocessor complex has a **prefix register**. The prefix register allows a range of real addresses from 0-4096 to reside in a different block of real storage than absolute addresses 0-4096. The alternate page 0's that give each processor the appearance of its own page 0 are called **prefix storage areas.**

The following are some multiprocessing considerations:

- When you generate CP as an AP system, Channel Set Switching is available in the event of failure on the IPL processor.
- Channel Set Switching is not available if you generate CP as an MP system or if you are using single processor mode.
- When you generate CP as an AP system, all the I/O is done on the side controlled by VM/SP HPO.
- You can generate CP as an AP and still run it in an MP environment, but VM/SP HPO will not use any of the I/O on the non-IPL side. The I/O on the non-IPL processor can be used by a preferred guest.
- The maximum number of channels on a UP or AP system is 16. If you need more than 16 channels, you will need an MP system.

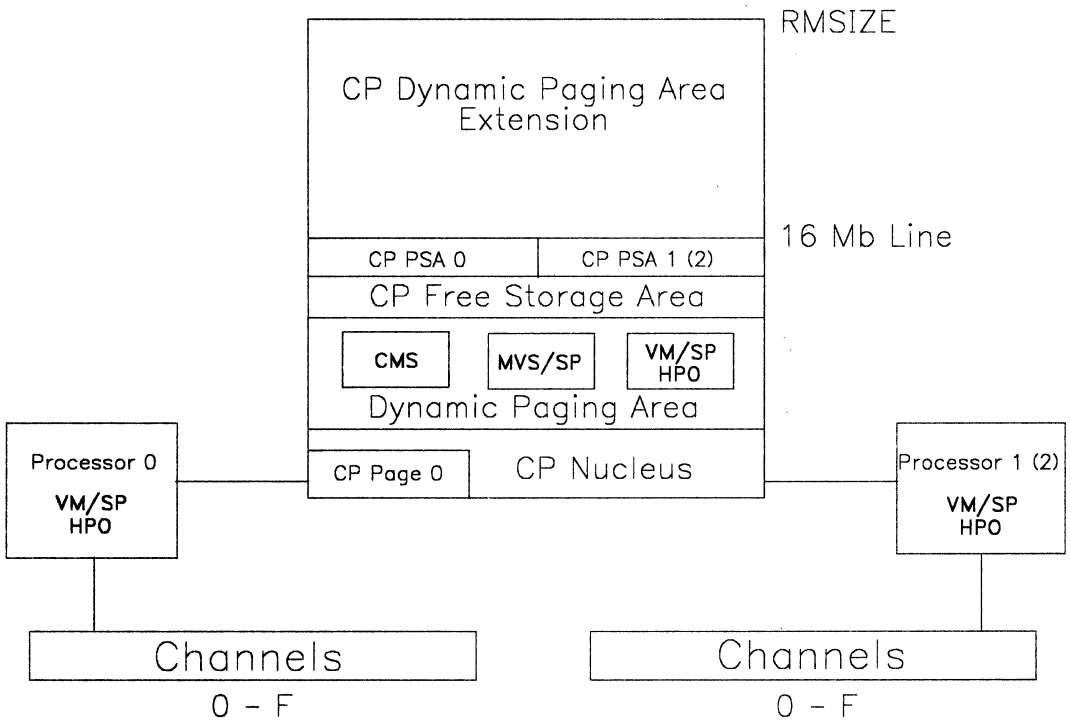
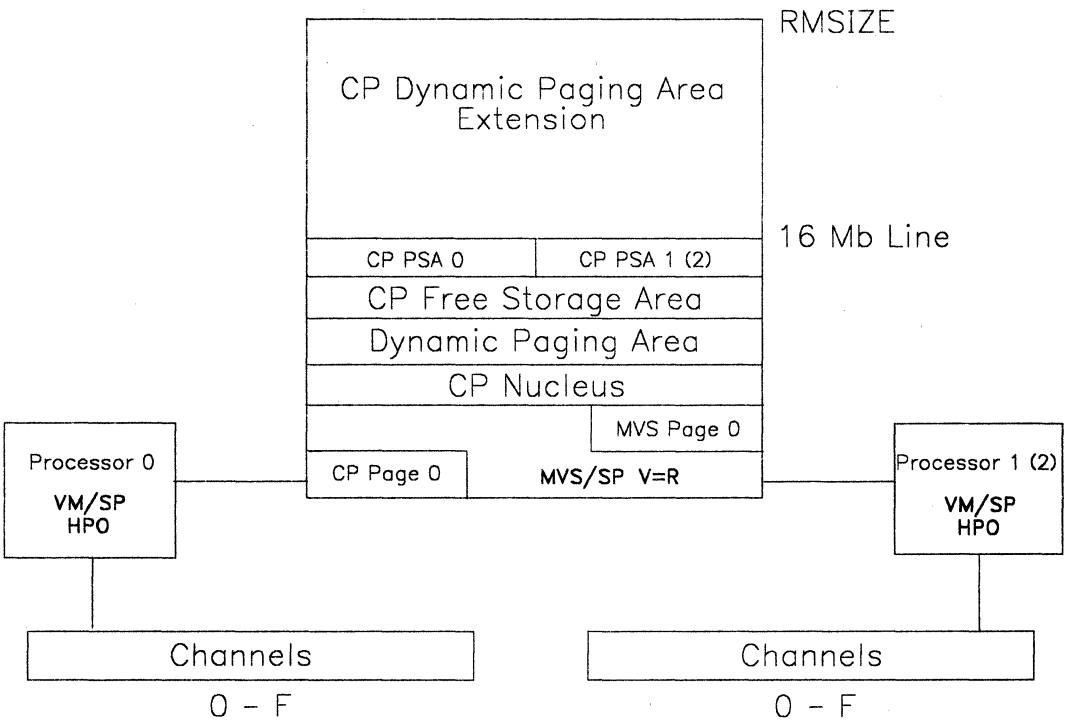


Figure 52. Storage Layout of a Multiprocessor System with No V=R Area



Note: VM/SP HPO owns absolute page 0. The MVS real page 0 resides at the high end of the V=R area. VM/SP HPO runs prefixed as it does in a multiprocessor system without a V=R area.

Figure 53. Storage Layout of a Multiprocessor System with a Nonpreferred V=R Guest

Preferred Machine Assist

Preferred machine assist is a standard hardware feature for the 308x, 3090, and 4381 processors; it is an optional feature for the 3033 processor. It allows an MVS/SP Release 1.1.1 or later guest to run under VM/SP HPO in supervisor state with direct control of its own I/O operations.

The **preferred guest** must be the V=R guest. You can use only one preferred guest at a time. As the preferred guest runs in supervisor state when it is dispatched by VM, it therefore controls the processor, the channels, and the I/O attached to those channels.

How Preferred Machine Assist Works

The MVS preferred guest is dispatched by CP to run in **real** supervisor state. This cuts down on CP overhead, as CP does not have to simulate most privileged instructions.

Preferred machine assist determines if a real interrupt should be handled by CP or MVS/SP.

The MVS preferred guest owns **real page 0** and MVS/SP gets interrupts directly. These interrupts include:

- CPU timer and clock comparator interrupts
- Program, SVC, and I/O interrupts.

Preferred Channels

You can generate VM/SP HPO without including in DMKRIO some of the physically installed channels and their devices. The exclusion of these channels identifies them as **preferred channels**. They are available to the preferred guest but to no other virtual machine or CP. The only way CP can use devices on preferred channels is that, another CP-known path must exist to them, or another CP nucleus has to be loaded with a DMKRIO that includes them.

When you generate VM/SP HPO for preferred machine assist, **do not** define in DMKRIO the channels you want to be used only for MVS I/O. MVS is running natively when it is dispatched and controls the I/O on those channels. VM/SP HPO cannot use them since it does not know about them.

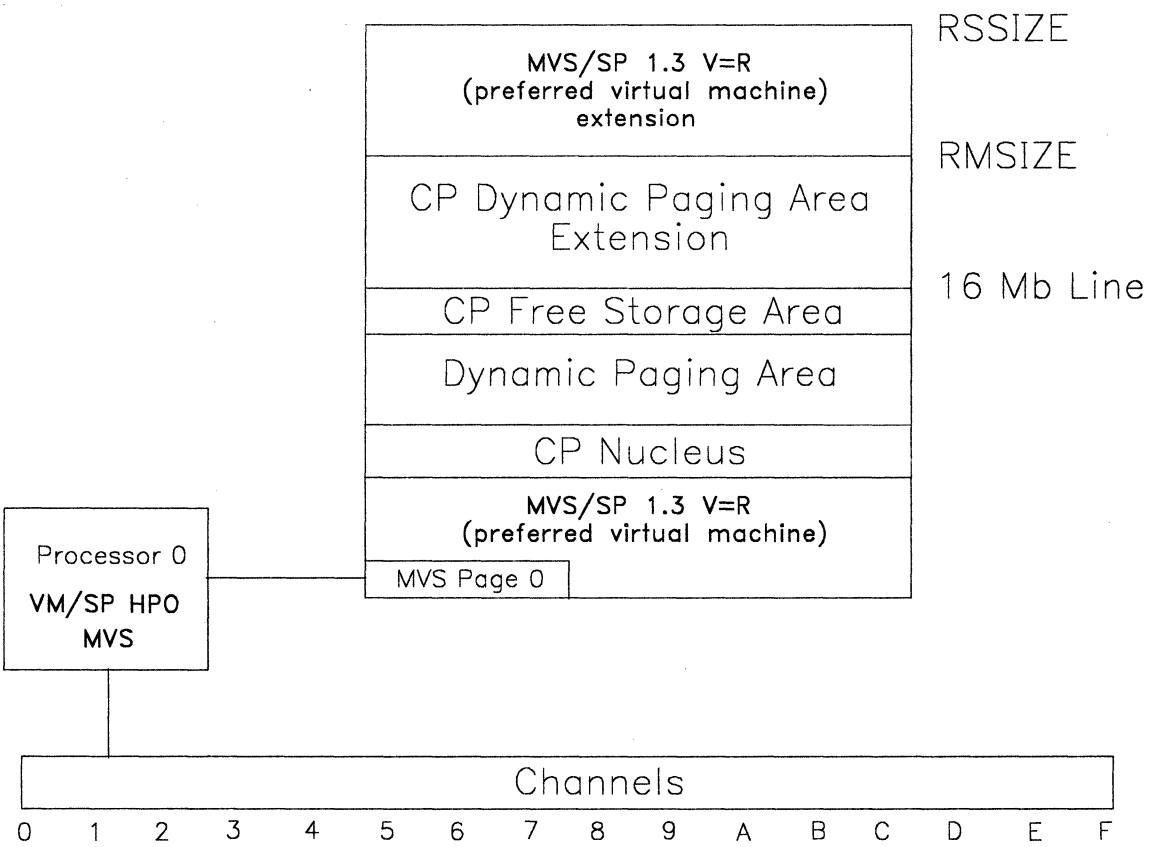
If you run VM/SP HPO with a preferred guest at times, and without one at others, consider keeping alternate copies of DMKRIO and the VM/SP HPO nucleus available. One set should have all channels defined in DMKRIO; the other would leave some undefined for exclusive use by the preferred guest.

Preferred Machine Assist Considerations

Make sure you allocate enough storage below the 16 Mb line for the preferred guest. MVS must have its entire nucleus and all its I/O buffers below the 16 Mb line.

You must SET AFFINITY for either of the two processors. If you want the preferred guest to have exclusive use of one processor and part of another, you must also run VM/SP HPO in single processor mode. If VM/SP HPO is not in single processor mode, a preferred guest will receive at most slightly less than one processor.

There is one hardware function — the interval timer — that the preferred guest cannot use even though it runs in supervisor state. VM/SP HPO needs the interval timer to control the allocation of time slices, and uses it to regain control from MVS at the end of the MVS time slice.



Note: If you generate VM/SP HPO to use less than 16 Mb of storage, the storage between its top and 16 Mb is unusable.

Figure 54. Storage Layout of Preferred Machine Assist System.

Control Switch Assist (Extension to Preferred Machine Assist)

On the 308x, 3090, and 4381 processors, the functions of preferred machine assist are enhanced by a **control switch assist**.

Preferred machine assist with the control switch assist improves the availability of the preferred guest through:

- Better handling of virtual I/O interrupts
- Optional support for IUCV and many CP DIAGNOSE codes.

I/O Interrupt Support

The support for I/O interrupts means that CP reflects interrupts that occur on CP-owned channels to the preferred guest in a more timely manner.

IUCV and CP DIAGNOSE Support

The support for IUCV and CP DIAGNOSE codes allows a preferred guest with control switch assist to use IUCV functions and many CP DIAGNOSE codes.

The **IUCV support** allows you to communicate with other users or CP in the same way you would with a nonpreferred guest.

The **CP DIAGNOSE support** allows:

- The MVS Job Entry Subsystem (JES) to issue CP commands to manipulate VM spool files
- The MSS Central Server application program to run on a production MVS system
- MVS guests to communicate using the Virtual Machine Communication Facility (VMCF).

Supported DIAGNOSE Codes

You can use the following DIAGNOSE codes when you IPL with the PMAV option:

Code	Description
00	Store Extended Identifier-Code
04	Examine Real Storage
08	Virtual Console Function
40	Cleanup after Virtual IPL by Device
4C	Generate Accounting Records
68	VMCF
6C	Shadow Table Maintenance (note 1)
78	MSS Communication
80	MSSFCALL

Notes:

1. *This gives a no-op because page zero for preferred machine assist with control switch assist is real page zero and there are no shadow tables for PMA and PMAV guests.*
2. *Using unsupported DIAGNOSE codes will give unpredictable results.*

Figure 55. DIAGNOSE Codes Supported

Running Your Preferred Virtual Machine with Control Switch Assist

With control switch assist, you can run your MVS/SP V=R virtual machine in one of two ways:

- IPL with the PMA option
- IPL with the PMAV option.

If you IPL with the PMA option, you receive all the functions of preferred machine assist and the I/O interrupt support.

If you IPL with the PMAV option, you receive all the functions of preferred machine assist, the I/O interrupt support, and the IUCV and CP DIAGNOSE support.

Note: Do not specify STOP or ATTN with PMA or PMAV.

Considerations for Dumping

If you IPL your MVS guest with the PMA or PMAV option, you must also IPL the dump program with the PMA or PMAV option. If you do not do this, the dump program cannot access the following:

- Real storage above the 16Mb line
- Any device that contains MVS paging data sets that are not on CP-owned channels.

IBM recommends that you IPL the dump program with the PMAV option for the following reason:

- DIAGNOSE X'40' is supported when you IPL with the PMAV option. It allows CP to save and restore the contents of the virtual machine page in which the IPL simulator runs.

  Because the original contents of this page are restored, the MVS stand-alone dump program can dump *all* of the preferred guest's storage.

Single Processor Mode

When you run VM/SP HPO on a multiprocessor complex, you can give an MVS guest complete control of one processor and part of another. You do this by running VM/SP HPO in **single processor mode.** MVS runs natively in one processor and under the control of CP in the other.

In single processor mode, the MVS guest runs in the V=R area. The remaining storage is shared among CP and the other virtual machines. These virtual machines are all dispatched on the VM/SP HPO processor. You must therefore be careful when you decide on the size of the V=R area. Leave enough storage apart from the V=R area to support CMS users and any other virtual machines that your installation plans to run.

You should use single processor mode when you need more than 50% of the processor complex for an MVS guest.

Note: On a 3081 or 3090 processor complex with an MVS/SP 1.3 or later guest, only a preferred machine assist guest can run in single processor mode.

To avoid TOD clock synchronization problems when running single processor mode, it is recommended that you generate CP as an AP- or MP-system and vary offline one processor to CP prior to establishing single processor mode.

Once CP is in UP mode, you can issue the CP SPMODE ON command to start single processor mode. VM/SP HPO gives up absolute page 0 and runs prefixed. At this point, you can IPL the MVS guest and run MVS in either AP or MP mode.

On an AP processor, CP overhead is avoided on the non-VM side. But overhead still exists since the I/O must be done on the VM side.

On an MP processor, you can configure all or most of the MVS I/O on the non-VM side, thereby reducing not only the CP overhead required to do I/O but also the CP overhead required to handle storage management and privileged operation simulation.

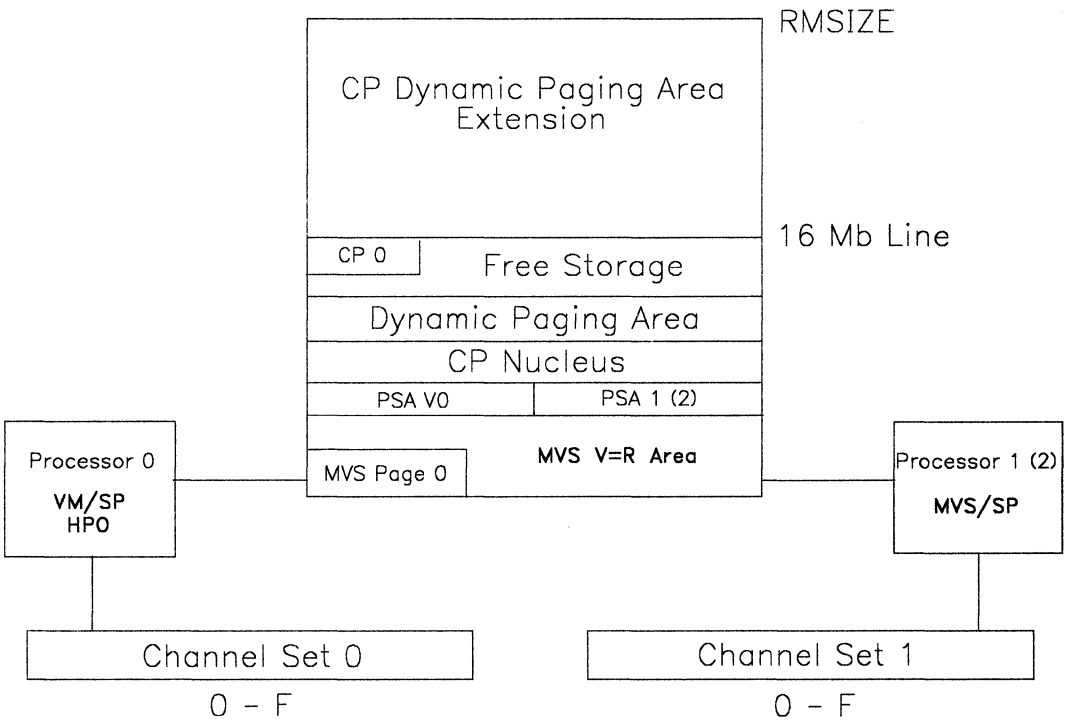
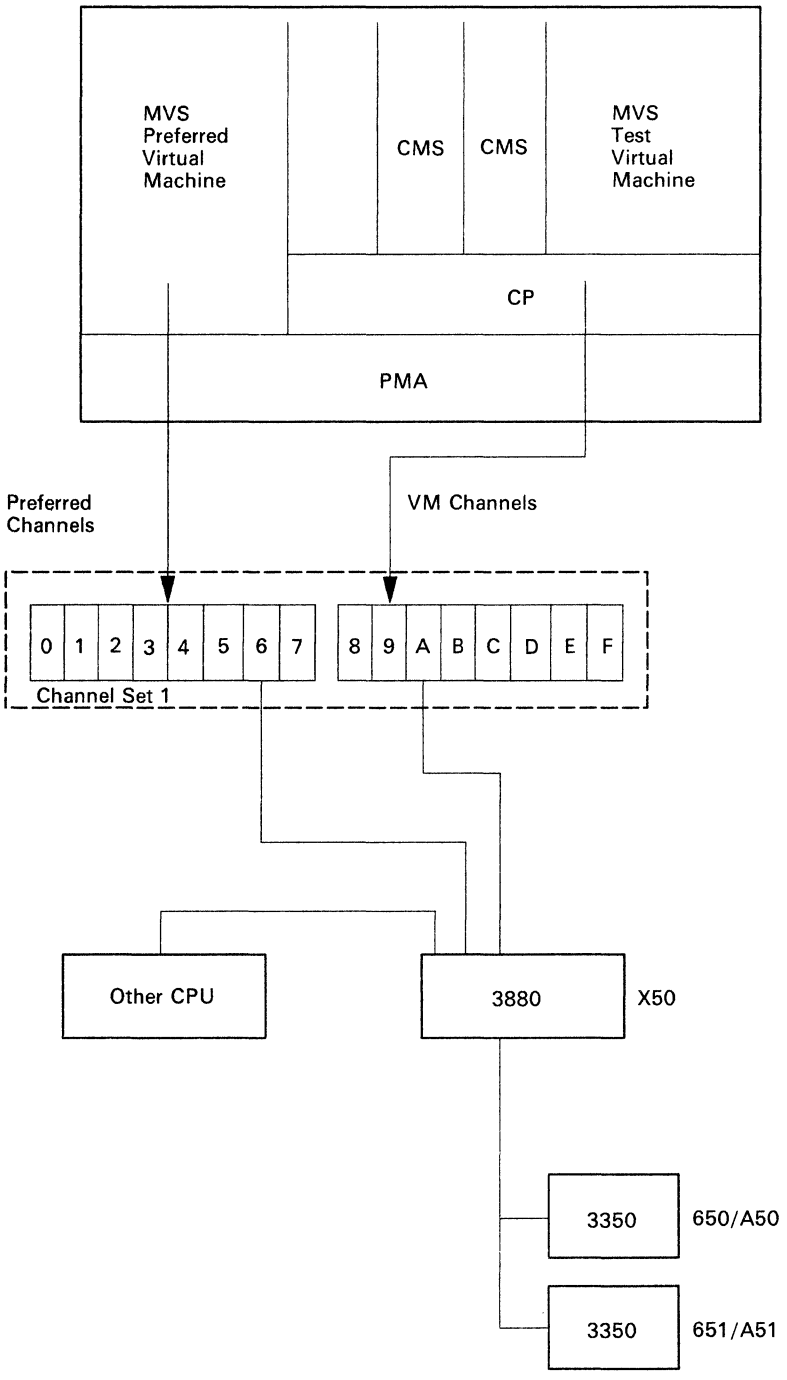


Figure 56. Storage Layout of a Single Processor Mode System (without Preferred Machine Assist) after IPL of MVS

Configuration Examples for Preferred Machine Assist Systems

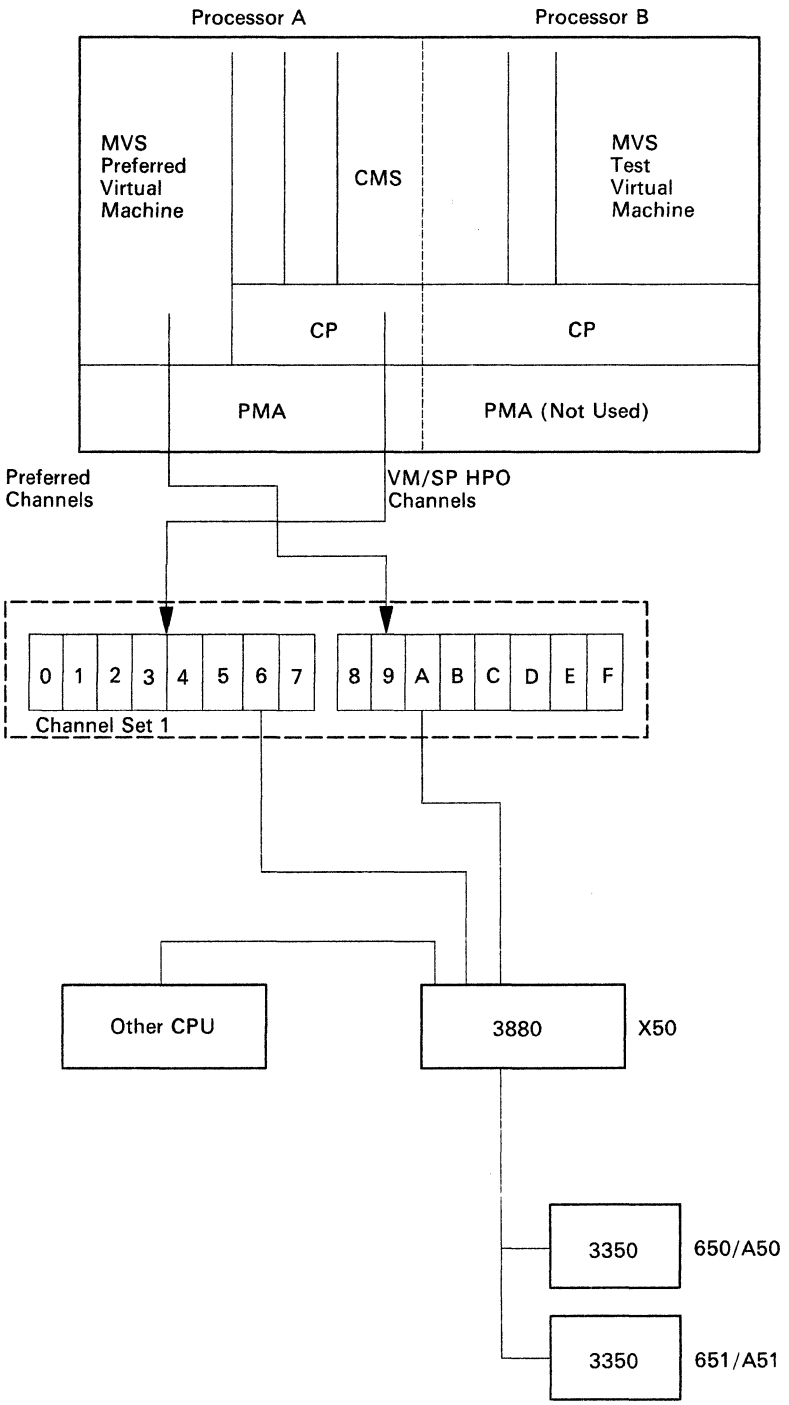
The following examples show I/O configurations with preferred channels for a UP, AP, and MP processor. The figures are simplified and do not show complete detail about string and control unit paths.

Note: Virtual machine system volumes must not exist on control units or strings that have both a virtual machine and a preferred channel path.



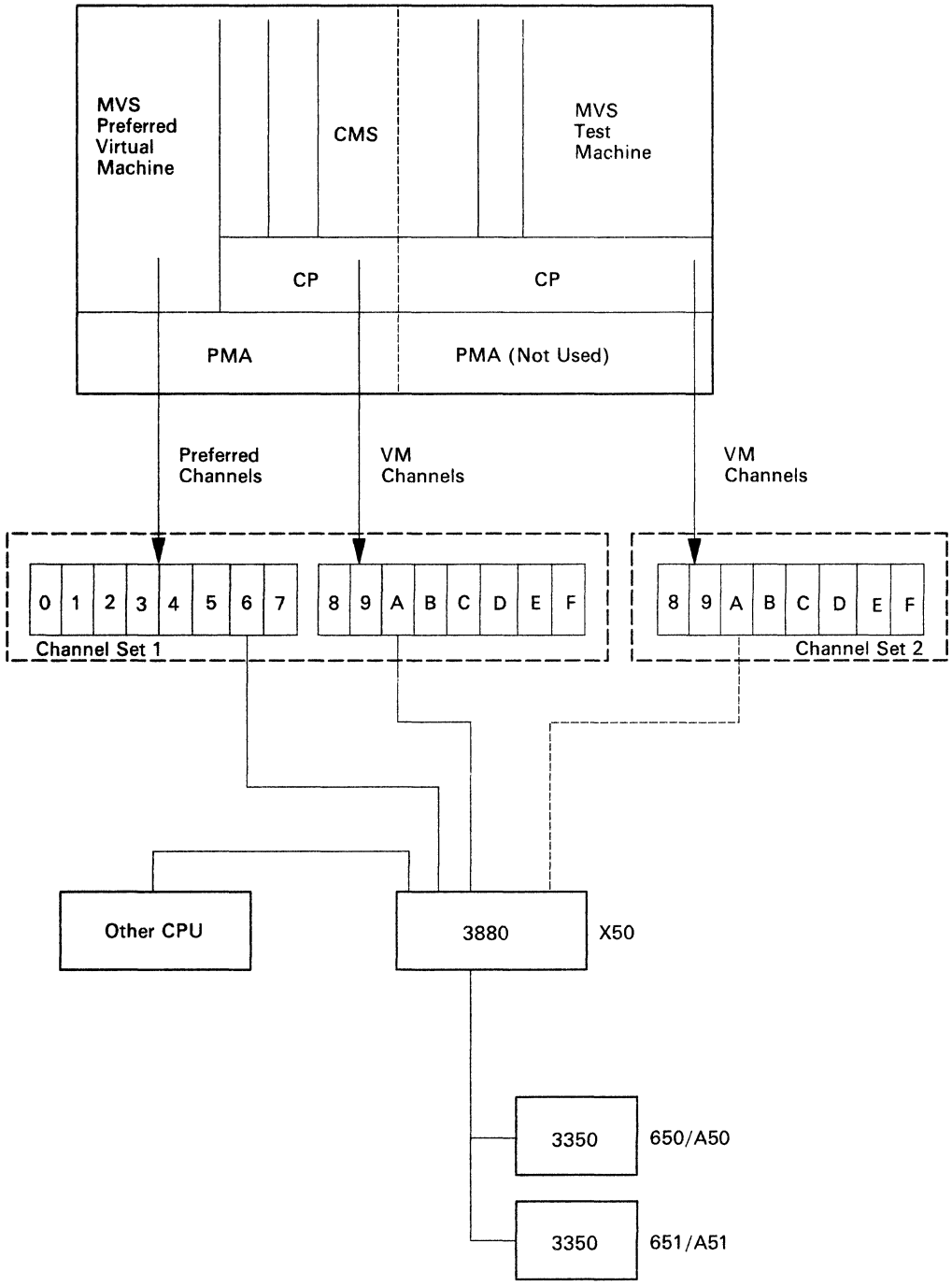
Note: Eight channels not defined in DMKRIO are used exclusively by the MVS preferred guest.

Figure 57. Configuration Example Using Preferred Channels on a UP Processor



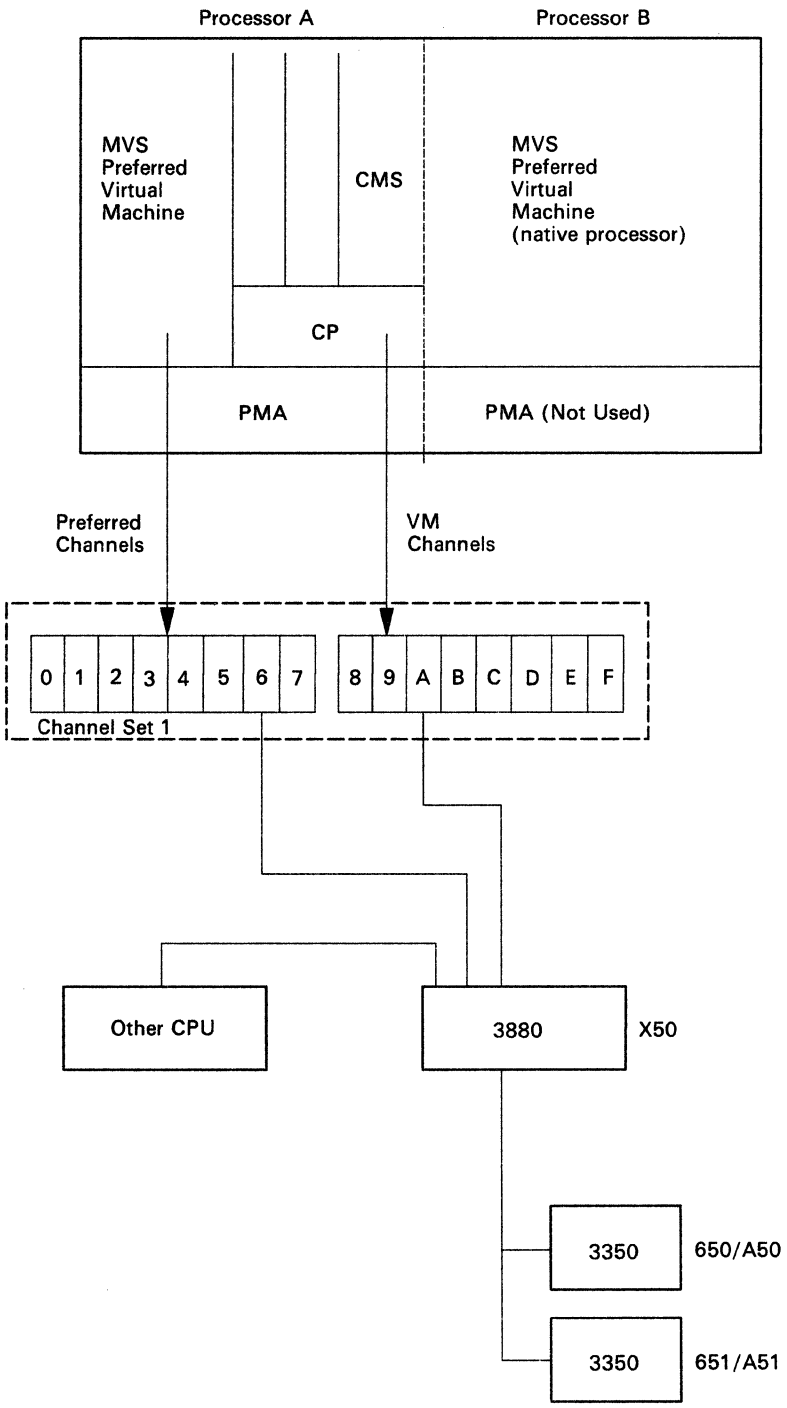
Note: VM/SP HPO uses both processors but only eight of the 16 channels. The eight channels not defined in DMKRIO are used exclusively by the MVS preferred guest.

Figure 58. Configuration Example Using Preferred Channels on an AP Processor



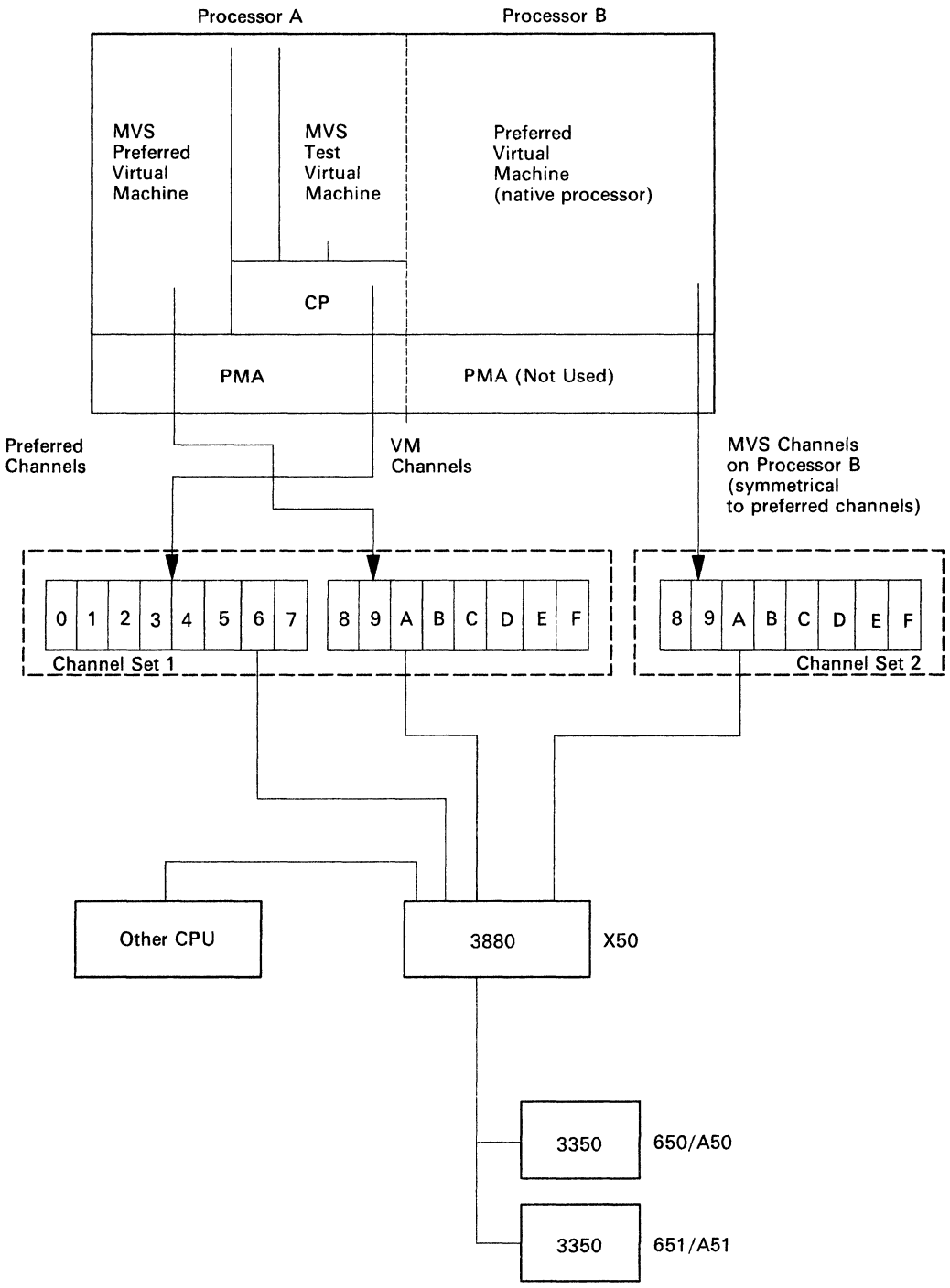
Note: The dotted line from channel set two represents a path that has been varied offline from VM/SP HPO. You must vary off one path from VM/SP HPO if you need data sharing between various MVS systems.

Figure 59. Configuration Example Using Preferred Channels on an MP Processor



Note: Only one processor can perform I/O on an AP.

Figure 60. Configuration Example Using Single Processor Mode on an AP Processor



Note: Both processors can perform I/O on an MP. In this example, channels 8 through 1 are not generated in DMKRIO. They are used symmetrically from MVS on both processors.

Figure 61. Configuration Example Using Single Processor Mode on an MP Processor

Chapter 2. Defining a Basic MVS/SP Virtual Machine

Creating Directory Entries

You need a directory definition for each MVS guest that you want to exist in the VM/SP HPO system. If you need details on directory entries beyond those in this chapter, refer to *VM/SP HPO Planning Guide and Reference.*

Directory Entry Considerations

This topic lists VM/SP HPO directory control statements in their logical order of appearance in a directory entry. It describes statements for running MVS in a virtual machine.

USER Defines a virtual machine and creates a directory entry. The USER control statement specifies the storage size of the MVS guest virtual machine and the class of CP commands that it can issue.

ACCOUNT Defines an account number and a distribution identification. The ACCOUNT control statement has no unique considerations for running MVS under VM/SP HPO.

OPTION Specifies certain options and features for MVS virtual machines.

IPL Automatically loads MVS for a guest virtual machine. The virtual address that you specify on the IPL statement should be the virtual address of the MVS system residence volume.

CONSOLE Defines the virtual console. Specify 3270 on the console control statement if you want to alternate between 3215 mode for CP commands and 3270 full- screen mode for MVS. Specify a secondary userid if you want to use the single console image facility to let another user control all messages, replies, and commands for a virtual machine after the primary user disconnects. See "Console Definitions" on page 129 for more information on defining consoles.

MDISK Describes the minidisk on a DASD to be owned by the MVS guest.

SPOOL Specifies the virtual unit record device to be used by the MVS guest if you want to use VM/SP HPO spooling.

DEDICATE Provides an MVS guest with sole use of a real device.

LINK Makes a minidisk that belongs to another user available to this guest at logon time.

SPECIAL Adds I/O devices that do not require corresponding real devices. These include virtual consoles, virtual channel-to-channel devices, pseudo timers , and communication lines.

Virtual Machine Options

VM/SP HPO provides several optional services to virtual machines. Bear in mind that you must specify some of these options when you run MVS under VM/SP HPO. You can specify these options in the OPTION control statement of the user's directory.

For more information about the OPTION control statement, refer to the *VM/SP HPO Planning Guide and Reference.*

BMX Option

Use the BMX (virtual block multiplexer) option to enhance the performance of your MVS guest. An MVS guest with this option can overlap multiple SIO requests on a specified channel path. The BMX option applies to all channels in the virtual machine except to channel 0. You can specify this option regardless of whether real block multiplexer channels are attached to the processor or not.

ECMODE Option

You must specify the ECMODE option when you run MVS under VM/SP HPO. It lets the MVS guest use the complete set of virtual System/370 control registers and the dynamic address translation feature of the System/370 processor.

STFIRST Option

When virtual machine assist is available on the machine, the STFIRST option permits an MVS guest to issue the CP SET STBYPASS command. Refer to the shadow table guidelines in the next chapter for more information.

VIRT=REAL Option

You must specify this option for an MVS guest that uses the V=R area.[12]

---

12 See *VM/SP HPO Installation Guide* for the definition of, and the requirements for, generating a V=R area.

The size of the V=R area must be as large as the largest virtual storage that the installation defines for the virtual machines that are to use the V=R option. Any number of virtual machines running under VM/SP HPO can specify the V=R option. However, only one virtual machine at a time can run in the V=R area. If a virtual machine is using the V=R area when another virtual machine with the V=R option logs on, VM/SP HPO runs the second virtual machine in V=V mode and sends a message informing the user that the V=R area is currently occupied.

PMA Option

The PMA option lets the virtual machine use preferred machine assist or preferred machine assist with control switch assist. It must be in the directory for a guest to specify the PMA option or the PMAV option on the IPL command.

REALTIMER Option

Enter the REALTIMER option if you want the virtual timer to be updated during virtual wait time as well as during virtual processor time.

This option is required for operating systems running applications (such as CICS) where certain interruptions are timer driven.

Notes:

1. *Normally, a virtual interval timer indicates only the real processor time used by the virtual machine.*

2. *Even when you specify the REALTIMER option, the virtual interval timer does not give accurate time-of-day values. This is because the virtual interval timer does not reflect real processor time that CP uses to perform required service for that virtual machine.*

370E Option

The 370E option lets an MVS guest use the MVS/System Extensions or MVS/System Product functions of VM/SP HPO.

To disable these functions for the VM/SP HPO system, system operators can issue the CP SET S370E OFF command. To disable these functions for themselves, general users can issue the CP SET 370E OFF command.

To display the ON and OFF system status of the MVS/System Extensions or MVS/System Product functions, both classes of users can issue the CP QUERY S370E command. To display the ON and OFF status of these functions for themselves, general users can issue the CP QUERY SET command.

XMEM Option

The XMEM option allows the MVS/SP (Release 3 or later) virtual machine to use cross memory facility. MVS/SP cross memory services allows a program to pass control to a different program in another address space, allowing direct data movement from one address space to another.

VM/SP HPO provides a performance enhancement for MVS cross memory services. VM/SP HPO needs the 3033 Extension Feature Enhancement to virtual machine assist, a special feature on the 3033 processor with the 3033 Extension Feature (#6850), to improve the performance of MVS/SP Release 3 cross memory services. To activate the cross memory services assist for the VM/SP HPO, the system operator must issue the CP SET S370E ON XMEM command. The operands on the S370E command are positional when you specify XMEM. The XMEM operand must immediately follow S370E ON. If XMEM is not in the OPTION control statement in the directory entry, the virtual machine user issues the CP SET 370E ON XMEM command to activate the cross memory services assist for the virtual machine.

On the 308x, MVS/SP cross memory services are available for the V=R virtual machine only when preferred machine assist is active.

The XMEM feature is available on the 4341(group 2 and 12) and 3033 processors with or without preferred machine assist.

General DMKRIO Considerations

When you run MVS under VM/SP HPO, the I/O definitions you make in DMKRIO will most likely closely parallel the definitions in your MVS stage 1 input.

- When you dedicate a device to a nonpreferred MVS guest, the *real address* of the device must appear in DMKRIO, and the *virtual address* of the device must appear in your stage 1 input. If your MVS and VM I/O definitions are parallel, you can specify the same address for the virtual and real address in the DEDICATE control statement of the directory, or with the CP ATTACH command after the user is logged on.

- When you dedicate a device to a preferred MVS guest, making the virtual and real address the same will ensure that MVS can continue to run natively if CP abends.

Note: *When you want a preferred MVS guest to have exclusive use of a channel and all its devices, you have to delete that channel definition and its device definitions from DMKRIO. A channel that VM/SP HPO does not know about is called a* **preferred channel**. *The more preferred channels you have, the better will be the overall performance of the preferred guest (because CP does not have to simulate the I/O instructions and handle the interrupts for preferred channels).*

V=V Configuration for MVS

The following example will be for an MVS V=V guest, although most of what is said will be true for V=R and preferred guests as well. Differences are pointed out where appropriate.

```
VMUSERS DIRECT    A1  F 80  TRUNC=72 SIZE=20 LINE=9 COLUMN

===== *
===== ****************************************************
===== *                      SYSTEM USERIDS                  *
===== ****************************************************
===== *
===== USER MVSVV2 PASSWORD 8M 16M G
===== ACCOUNT CMS00003 VM-FLOOR
===== OPTION REALTIMER ECMODE BMX 370E STFIRST
===== IPL 179
=====    CONSOLE 01F 3215 C
===== * Spooled unit record devices
=====    SPOOL 01C 3505 A
=====    SPOOL 01D 3525 A
=====    SPOOL 010 3211 A
===== * MVS operator's console
=====    DEDICATE CC0 CC0
===== * MVS system residence volume
=====    DEDICATE 179 MVSRES
=====    DEDICATE 1A2 1A2
=====    DEDICATE 1A3 1A3
=====    DEDICATE 170 M30PGE
=====    DEDICATE 171 M30SPL
=====    DEDICATE 172 M30LIB
=====    SPECIAL 1A0 3270
=====    SPECIAL 1A1 3270
=====    MDISK 191 3330 275 002 H34P30 MR
```

Figure 62. Sample Directory for MVS V=V Guest

Console Definitions

There are two kinds of virtual consoles you can use when running MVS under VM/SP HPO:

- The **logon console** is the virtual machine console. Use this console to enter CP commands.
- The **MVS operator console** communicates with MVS. Use this console to enter MVS commands.

You can use two separate terminals as your logon and MVS operator console, or you can use one terminal for both.

Using Separate Terminals as Logon and MVS Operator Consoles

If you use separate terminals, your directory entry should look like this:

```
CONSOLE 01F
DEDICATE CC0 CC0
```

The console at virtual address 01F is your logon console. From there, you can log on your guest virtual machine and IPL the guest. You can then disconnect the logon terminal. The terminal at real address CC0 is now your MVS operator console.

Using the Same Terminal as Logon and MVS Operator Consoles

If you want to use the same terminal as your logon and MVS operator console, your directory entry should look something like this:

```
CONSOLE CC0 3270
```

Your logon console is at virtual address CC0. The 3270 specification allows the MVS guest to share a locally attached terminal controlled by CP. The MVS guest can use the terminal in full screen mode, while CP shares the terminal and uses it as a line device.

To use this terminal as both a logon console and MVS operator console:

1. Log on the guest's virtual machine.
2. The address specified in the CONSOLE statement should match one of the console addresses defined in your MVS stage 1 input as a console or alternate console. If the CONSOLE statement in the directory does not match the address specified in the MVS stage 1 input, use the CP DEFINE command to correct it.
3. *Do not* disconnect the virtual machine.
4. Issue:

   ```
   #cp terminal conmode 3270 scrnsave on
   ```

   (You must be at a local 3270 to issue this command, and you cannot be running through a VTAM service machine.)
5. IPL the guest operating system.

MVS Stage 1 I/O Console Considerations

Consider the following before you make your directory definitions for consoles:

The terminal type and real address of the MVS operator console (in this case a 3278 at real address CC0) must be the same as specified in the MVS stage 1 I/O generation. For example:

```
IODEVICE UNIT=3278,ADDRESS=(CC0,3),MODEL=4,
                FEATURE=(DOCHAR,AUDALRM,OCKY3277,KB78KEY,SELPEN)
```

If your virtual console is not defined properly, MVS will enter a wait state because it could not find a valid console address or specification.

Spooled Unit Record Device Definitions

You may want to dedicate unit record devices to a production system. If you use VM/SP HPO spooling, MVS does not close VM spool files. Your output will not be separated unless you close the spool files manually or via a DIAGNOSE. (And when using preferred machine assist, you cannot issue a DIAGNOSE instruction, unless control switch assist is also activated.)

Although you will likely dedicate unit record devices to a single virtual machine, you may choose to use the SPOOL statement to let the devices service all virtual machines.

The virtual addresses of the unit record devices you want to use must be defined as real addresses in your MVS stage 1 I/O generation. For example, if your MVS stage 1 I/O definitions look like this:

```
IODEVICE UNIT=3505,ADDRESS=01C
IODEVICE UNIT=3525,ADDRESS=01D
IODEVICE UNIT=3211,ADDRESS=010
                FEATURE=MULTILINE
```

your VM/SP HPO directory definitions should look like this:

```
SPOOL 01C 3505 A
SPOOL 01D 3525 A
SPOOL 010 3211 A
```

where *A* is the VM/SP HPO spool class associated with the device.

You do not need corresponding DMKRIO definitions as long as your unit record devices are spooled instead of dedicated. If you were to dedicate a unit record device, the real address in the DEDICATE statement would have to correspond to a DMKRIO definition.

DASD Definitions

To give exclusive use of a DASD to one virtual machine, specify the DEDICATE statement in that user's directory. You can specify the same DEDICATE statement for more than one user, but only the first user to log on will be able to use it.

DASD Address Considerations

The virtual DASD addresses that you use in a directory must be defined in your MVS stage 1 input as real addresses, whether the volumes are dedicated or shared minidisks.

When you use the DEDICATE statement in a directory for a nonpreferred guest (like the V=V sample guest), you must include the real addresses in the directory in your DMKRIO definitions. In your directory definitions, you can specify the volume label or the volume's real address.

For example, if your stage 1 input looks like this:

```
IODEVICE UNIT=3330,ADDRESS=(170,08)
```

then your DMKRIO definition should look like this:

```
RDEVICE ADDRESS=(170,08),DEVTYPE=3330,MODEL=1
```

and your directory definitions might look like this:

```
DEDICATE 170 M30PGE
DEDICATE 171 M30SPL
DEDICATE 172 M30LIB
```

or

```
DEDICATE 170 170
DEDICATE 171 171
DEDICATE 172 172
```

Tape Definitions

Magnetic tape drives can be used by only one virtual machine at a time. Dedicate a tape drive to a virtual machine (usually a production machine) like this:

```
DEDICATE 580 580
```

where the second 580 is the real address of the tape drive.

Stage 1 I/O and DMKRIO Considerations for Tape Drives

Remember, any time you use the DEDICATE statement in a directory, the real address of the dedicated device must appear in DMKRIO ASSEMBLE:

```
RDEVICE ADDRESS=(580,16),DEVTYPE=3420,MODEL=8,FEATURE=DUALDENS
```

and the virtual address of the dedicated device (usually the same as the real address) must appear in your stage 1 I/O generation:

```
IODEVICE UNIT=3420,ADDRESS=(580,16),MODEL=8,FEATURE=(OPT 1600)
```

It is possible to switch devices such as tape drives and printers among virtual machines. Because these devices cannot be shared, the device needs to be attached to one user at a time. This can be done by issuing the CP ATTACH and CP DETACH commands.

V=R Configuration for MVS

```
VMUSERS DIRECT    A1  F 80  TRUNC=72 SIZE=20 LINE=9 COLUMN

===== *
===== ************************************************
===== *                    SYSTEM USERIDS
===== ************************************************
===== *
===== USER MVSVR PASSWORD 12M 12M BG
=====  ACCOUNT SYS00001 VM-FLOOR
=====  OPTION REALTIMER ECMODE BMX 370E VIRT=REAL
=====    CONSOLE 01F 3215 C
=====    IPL 179
===== * Spooled unit record devices
=====    SPOOL 01C 3505 A
=====    SPOOL 01D 3525 A
=====    SPOOL 010 3211 A
===== * MVS operator's console
=====    DEDICATE CC0 CC0
===== * MVS system residence volume
=====    DEDICATE 179 MVSRES
=====    DEDICATE 1A2 1A2
=====    DEDICATE 1A3 1A3
=====    DEDICATE 179 M31RES
=====    DEDICATE 170 M30PGE
=====    DEDICATE 171 M30SPL
=====    DEDICATE 172 M30LIB
```

Figure 63. Sample Directory for MVS V=R Guest

You must include the VIRT=REAL option on the OPTION statement in the directory of the V=R guest:

```
OPTION REALTIMER ECMODE BMX 370E VIRT=REAL
```

V=R Environment Storage Considerations

For a V=R guest, you must consider how the location and use of free storage can affect performance. CP free storage in VM/SP HPO is used in a way similar to the way MVS uses SQA. Control blocks and some system routines are resident in the CP free storage area. A shortage of CP free storage can impose severe performance constraints on the entire VM/SP HPO system.

In a VM/SP HPO system, both the V=R area and the CP free storage area must reside below the 16 Mb line. When you configure the storage layout of an MVS-under-VM/SP HPO system, you must achieve a proper balance between these vital areas.

Other Considerations for V=R Guest

- Improve the performance of a V=R guest by issuing the CP SET NOTRANS ON command after IPL. This eliminates CCW storage address translation for all I/O except that involving page zero; however, each CCW is still scanned by CP for validity.

- Issue the SET STBYPASS VR command to eliminate unnecessary shadow table maintenance.

- When you tune the system, remember that the V=R guest is scheduled and dispatched by VM/SP HPO just like any other virtual machine. You can use tuning options such as SET FAVOR and SET PRIORITY to better allocate processor power within a processor complex.

- Keep in mind that only the V=R guest can use the V=R area. You can specify the V=R option in the directory for several virtual machines, but only one logged-on guest can use the V=R area.

Preferred Machine Assist Configuration for MVS

```
VMUSERS DIRECT    A1  F 80  TRUNC=72 SIZE=20 LINE=9 COLUMN
===== *
===== ************************************************
===== *                    SYSTEM USERIDS                *
===== ************************************************
===== *
===== USER MVSPMA2 PASSWORD 12M 12M BG
=====  ACCOUNT SYS00001 VM-FLOOR
=====  OPTION  ECMODE BMX VIRT=REAL PMA AFF 00
=====  IPL 179
=====    CONSOLE 01F 3215 C
=====    SPOOL 011 3211 A
=====    SPECIAL 520  CTCA
===== * MVS system residence volume
=====    DEDICATE 179 MVSRES
=====    DEDICATE 010 010
=====    DEDICATE 1A2 1A2
=====    DEDICATE 1A3 1A3
```

Figure 64. Sample Directory for MVS Preferred Guest

Special Directory Considerations for MVS Preferred Guests

- Specify the PMA option directly after the V=R option. For example:

  ```
  OPTION ECMODE BMX VIRT=REAL PMA AFF 00
  ```

- Some directory options are redundant for a preferred guest. Do not specify the following:

  REALTIMER
  370E
  XMEM

- If you run CP in AP or MP mode, you must set affinity to one of the two processors for the preferred guest. You can do this in the directory with a statement like this:

  ```
  OPTION  ECMODE BMX VIRT=REAL PMA AFF 00
  ```

  where **AFF 00** sets affinity to CP 0 or processor 0.

- You cannot use minidisks for a preferred guest unless they are full-volume minidisks.

Address Rules for MVS Preferred Guest Devices

Follow these rules when defining virtual addresses for preferred guests.

Rule 1:

When you dedicate CP-known devices to the preferred guest, those devices should either:

- Have a virtual address exactly the same as the real address, **or**
- Have a virtual address that does not match any real address generated in DMKRIO.

and

- Have a virtual channel address that *is* generated in DMKRIO.

Rule 2:

When you use nondedicated CP-known devices, they should both:

- Have a virtual address that *is not* generated in DMKRIO, and
- Have a virtual channel address that *is* generated in DMKRIO.

At first this rule may sound confusing, but think about what it means. The resultant address for a nondedicated virtual device cannot map to a real address. But the channel address must be known to VM/SP HPO or the channel would be a preferred channel. For example, the addresses of 01F, 011, 191, and 520 (as defined in the previous figure) must not be generated in DMKRIO.

Channel Considerations for Preferred Guests

For the best performance, you should isolate some channels for both MVS and VM/SP HPO. MVS should have enough channels to support a stand-alone, native MVS system. Likewise, VM/SP HPO should have enough channels to handle the requirements of VM/SP HPO and all other nonpreferred guests.

General Restrictions for Preferred Machine Assist

1. Preferred machine assist supports *only* MVS/SP Version 1 Release 1 with Release 1 Enhancement or later releases. MVS/SP Version 1 Release 3 or later releases are needed to support "extended storage" above 16 megabytes. Any attempt to run another system using the preferred machine assist feature may result in CP abending.

2. The MSS Central Server Application Program, a VM/370 Release 6 Program that runs under MVS, cannot run when preferred machine assist is active unless control switch assist is also active.

3. Devices on channels that belong to the preferred machine assist virtual machine are not defined to VM/SP HPO at system generation. Because these devices are unknown to VM/SP HPO, the VM/SP HPO system operator cannot vary them online to VM/SP HPO without reinitializing the system and including the devices in DMKRIO.

4. Do not issue DIAGNOSE instructions when running a guest MVS system with preferred machine assist, unless control switch assist is also active and you IPLed the guest with the PMAV operand. Otherwise, the results of using a DIAGNOSE in a preferred machine assist environment are unpredictable.

   For example, suppose an MVS guest is running with preferred machine assist but not control switch assist. If you issue a DIAGNOSE X'80' to vary a channel offline, the DIAGNOSE instruction will go directly to the hardware and the channel will be varied offline. VM/SP HPO is unable to intercept the DIAGNOSE in this case.

5. An MVS V=R guest that was IPLed with the PMAV operand cannot issue a DIAGNOSE or IUCV instruction from above the 16 Mb line, nor can it specify on the operands of such an instruction an address above the 16 Mb line.

6. You must not have CP-owned volumes and MVS volumes on the same strings and control units when you are:

   - Running an MVS guest when VM is in single processor mode (SPMODE) on multiprocessing systems

   - Running an MVS guest with preferred machine assist (PMA) on uniprocessors or multiprocessors.

   In either of these two environments, A path must exist to the DASD volumes from a VM channel, and from a channel that VM does not know about. This can be either a channel from the MVS native processor (when running single processor mode) or a preferred channel on the VM processor.

   When the above conditions are met, a deadlock situation can occur that will prevent the MVS guest from being dispatched if MVS and CP-owned volumes exist on the same strings and control units. This can happen when an I/O error occurs on the channel that VM does not know about. When this happens, a contingent connection is established between the device, head of string, control unit, and the channel until MVS can issue a sense command to gather the error data. If, before the sense can be issued, VM must perform some service for the MVS guest, the lockout can occur. If CP must do some I/O to the CP owned volumes that share the same string or control unit that are busy because of the I/O error, MVS may be placed in a wait state, until the resulting busy condition has been cleared. Since only the MVS guest can clear the busy, by issuing the sense, a deadlock has occurred, and the MVS guest will never be dispatched.

The two types of CP services that can cause I/O to DASD devices are:

- Use of VM spooling. This can be unit record spooling or console spooling.
- The issuing of CP commands at the MVS virtual machine console.

If you cannot avoid either of these activities, you must not have VM system volumes and MVS volumes sharing the same strings and control units in the configurations mentioned above.

There are some special considerations when using strings of 3380s that start with AA4s which are connected to two storage directors of the 3880 control units. Since these devices have multiple internal paths, and can have two concurrent data transfer operations taking place, these strings can be properly configured to avoid the deadlock situation, and still have MVS and VM volumes on the same string. To do this you must segregate the MVS and VM volumes on different internal paths. In a string of 16 devices the addresses are assigned to internal paths as follows:

PATH A - xx0,xx1,xx8,xx9

PATH B - xx2,xx3,xxA,xxB

PATH C - xx4,xx5,xxC,xxD

PATH D - xx6,xx7,xxE,xxF

If you can arrange your MVS and CP-owned volumes so that they do not share an internal path, you can avoid these lockout situations.

7. If you started single processor mode on a 3033, 308x, 3090, or 4381 Model Group 13 UP-generated system to IPL with the PMA or PMAV option, the MVS guest must vary off the native side before issuing SPMODE off.

8. If you are running single processor mode on a 308x or 4381 processor and you IPL with the PMAV option, the MVS guest must issue DIAGNOSE and IUCV instructions from the CP side (not the native side). In order to accomplish this, the MVS system programmer must establish affinity for the program issuing the DIAGNOSE and IUCV instructions.

9. On a 3090 processor, an MVS guest operating system that uses preferred machine assist or single processor mode must use a release of MVS that would run natively on the processor.

10. When you are running in either single processor mode or as a preferred machine assist guest, do not use the CP PER command.

For restrictions on generating a VM/SP HPO system with a preferred virtual machine, see the *VM/SP HPO Planning Guide and Reference.* For restrictions on certain CP commands when preferred machine assist is active, see *CP for System Programming.*

Things to Do before IPLing the MVS/SP Virtual Machine

If you use the same console as the logon console and the MVS operator console, there are three CP commands you should enter before IPLing MVS. You can use the following CMS EXEC to do this automatically.

```
 CMS      EXEC       A1  F  80  TRUNC=132  SIZE=4  LINE=1  COL=1 ALT=0

===== &TRACE OFF
===== CP TERM BRKKEY PF12
===== CP TERM BREAKIN GUESTCTL
===== CP TERM CONMODE 3270 SCRN ON
```

Figure 65. Sample CMS PROFILE EXEC (VM/SP HPO - MVS/SP)

CP TERM BRKKEY PF12 lets you set the CP break key (normally PA1) to another program function (PF) key. This is helpful because MVS uses PA1 to retrieve the last command entered. If you do not set the BRKKEY to something other than PA1, you will drop into a CP READ state if you press PA1 to retrieve the last command.

CP TERM BREAKIN GUESTCTL prevents CP from interrupting the screen when a message must be displayed at your terminal. For example, if someone sends you a message, the terminal will beep — your MVS console will not be cleared. When you want to display the message, press the break key to drop into CP READ.

CP TERM CONMODE 3270 SCRN ON places your virtual console in 3270 mode. You must include this command when you run a CMS PROFILE EXEC because CMS internally sets your virtual console to a 3215 when it is IPLed. This should be the last command in the PROFILE EXEC because CMS can no longer run once the command is issued.

You should always specify the SCRN ON portion of the command. It saves the full-screen image if you must go into CP READ.

How to IPL MVS for a V=V or V=R Guest

After you have logged on your virtual machine and before you IPL MVS, make sure your virtual machine size is adequate for this particular session. The USER control statement for the V=V user was:

```
USER MVSVV2 PASSWORD 8M 16M G
```

so user MVSVV2 received 8 Mb of storage at logon. If you needed more storage in this case, you can specify up to 16 Mb. If you wanted 12 Mb, you would issue the command:

```
define storage as 12m
```

To IPL MVS in the sample V=V guest, enter the command:

```
ipl 179
```

because the directory entry for the V=V guest defined the MVS system residence volume at virtual address 179.

How to IPL MVS for a Preferred Guest

There is one difference when you IPL a preferred guest as opposed to a V=V or V=R guest. You must include the PMA option or the PMAV option on the IPL command:

```
ipl 179 pma
```

or

```
ipl 179 pmav
```

where 179 is the virtual address of the system residence volume.

The IPL device can be on a VM/SP HPO channel or a preferred channel. When you use single processor mode, it must be on the VM-owned processor.

Using a CMS Profile EXEC to Automatically IPL MVS

You can use a CMS PROFILE EXEC that IPLs MVS for you when you logon. The EXEC that you use should detach your CMS minidisks before it IPLs MVS.

```
PROFILE   EXEC       A1  F  80  TRUNC=132  SIZE=11 LINE=6 COL=1 ALT=0

===== &TRACE OFF
===== CP SET RUN ON
===== CONWAIT
===== DESBUF
===== &STACK CP DET 19E
===== &STACK CP DET 19D
===== &STACK CP DET 191
===== &STACK CP DET 190
===== &STACK CP IPL 179
===== &EXIT
```

Figure 66. Sample PROFILE EXEC for Automatic IPL of MVS/SP

Issuing the CP SET RUN ON Command

Unless you are using CP debug facilities (such as ADSTOP and TRACE commands), you should issue the CP SET RUN ON command. This allows you to enter CP commands without stopping your virtual machine.

CP Commands to Know at the MVS/SP Operator's Console

When you run MVS under VM/SP HPO, there will be times when you must simulate real processor/hardware functions. Use the following CP commands to simulate these real operator functions. Remember, to execute any CP commands, you must press the break key that you previously defined.

EXT Use the EXT command to create an external interrupt to your virtual machine. It simulates pressing the interrupt key on the real system console. You will usually do this when you have "lost" your virtual console to MVS because of a console switch or a console I/O error.

READY Use the CP READY command to set a device-end interrupt pending for a specified virtual device. This simulates "popping the plug" on a real disk device. You may have to do this on the few occasions when you mount a disk to MVS and the mount message does not disappear.

RESET Use the CP RESET command to clear all pending interrupts from a specified virtual device. You can use this command to drop a DIALed terminal from the virtual machine issuing the command.

SYSTEM Use the CP SYSTEM command to simulate the action of the RESET and RESTART buttons on the real computer console, and to clear storage. The operands of this command do the following:

- **RESET** resets all pending interrupts and conditions in the virtual machine.
- **RESTART** simulates the hardware system RESTART function.
- **CLEAR** clears virtual storage and virtual keys to binary zeros.

Chapter 3. Additional Topics When Operating an MVS/SP Virtual Machine

How to Initialize a DASD for Use by MVS

You must initialize DASD volumes before an operating system can use them. To initialize a full DASD volume or minidisk for use by an MVS guest, use the Device Support Facility.

You can initialize DASD volumes with Device Support Facility under MVS, or you can run the stand-alone version in a virtual machine. To initialize a DASD volume with the stand-alone version, the volume must be a part of the virtual machine configuration as a result of a DEDICATE statement in the directory, or the operator can issue an ATTACH command for the volume. Minidisks must have MDISK entries in the directory and the user running Device Support Facility must be properly LINKed to them.

The stand-alone version of Device Support Facility is provided on the CMS system disk (190) as file IPL DSF S. To execute the program in a virtual machine, you can perform the following steps:

1. Spool your virtual punch to yourself.
2. Punch the file IPL DSF S without a header card.
3. IPL the Device Support Facility program from your virtual reader.
4. Identify your terminal to the Device Support Facility program by pressing ENTER.
5. Respond to the DSF message to define the input device as the console.
6. Respond to the Device Support Facility message to define the output device as the console.
7. Enter the Device Support Facility control statements to perform the initialization.
8. Re-IPL CMS when Device Support Facility finishes.

The following is an example of an EXEC that will perform some of these steps for you:

```
DSF      EXEC     A1   F  80  TRUNC=132 SIZE=20 LINE=5 COL=1 ALT=0


===== &TRACE ERR
===== &TYPE **** SPOOLING DSF TO YOUR READER ****
===== CP SPOOL PUNCH * CLASS I
===== PUNCH IPL DSF * (NOH
===== &IF &RETCODE NE 0 &GOTO -ERROR
===== &IF &INDEX GT 0 PUNCH &1 &2 &3 (NOH
===== &IF &RETCODE NE 0 &GOTO -ERROR
===== CP SPOOL PUNCH NOCONT CLOSE
===== CP SPOOL PUNCH OFF CLASS A
===== CP CLOSE READER
===== CP ORDER READER CLASS I
===== CP SPOOL READER CONT CLASS * NOHOLD
===== &TYPE **** IPLING DSF
===== &TYPE **** WHEN DSF IS FINISHED, RE-IPL CMS
===== CP IPL 00C
===== -ERROR &R = &RETCODE
===== CP SPOOL PUNCH NOCONT PURGE
===== CP SPOOL PUNCH OFF CLASS A
===== &EXIT &R
```

Figure 67. DSF EXEC for an MVS Virtual Machine

Case 1: Using DSF for a Dedicated Volume

You have a 3350 DASD online at real address 160.

1. Attach it to yourself by issuing:

   ```
   att 160 *
   ```

2. Run the DSF EXEC.

3. Enter the information requested from you as the EXEC executes.

4. When you are prompted to enter the INIT command, you specify:

   UNITADDRESS the hexadecimal address of the channel and unit on which the volume is mounted.

   NOVERIFY to bypass verification of the volume serial number and owner identification.

   DEVICETYPE the type of DASD.

   VOLID the volume label name.

The following example shows how this process works:

ENTER

```
att 160 *
```

SYSTEM RESPONSE

```
DASD 160 ATTACH TO CPG      160
Ready;
```

ENTER

```
dsf
```

SYSTEM RESPONSE

```
**** SPOOLING DSF TO YOUR READER ****
PUN FILE 0011  TO  CPG      COPY 001   NOHOLD
0001 FILE  ORDERED
**** IPLING DSF
**** WHEN DSF IS FINISHED, RE-IPL CMS

ICK005E DEFINE INPUT DEVICE, REPLY 'DDDD,CUU' OR 'CONSOLE'
ENTER INPUT/COMMAND:
```

ENTER

```
console
```

SYSTEM RESPONSE

```
CONSOLE
ICK006E DEFINE OUTPUT DEVICE, REPLY 'DDDD,CUU' OR 'CONSOLE'
ENTER INPUT/COMMAND:
```

ENTER

```
console
```

SYSTEM RESPONSE

```
CONSOLE
ICKDSF -  SA  DEVICE SUPPORT FACILITIES   6.0
ENTER INPUT/COMMAND:
```

ENTER

```
init unitaddress(160) noverify devicetype(3350) volid(d33500)
```

SYSTEM RESPONSE

```
INIT UNITADDRESS(160) NOVERIFY DEVICETYPE(3350) VOLID(D33500)
ICK00700I 160 BEING PROCESSED AS LOGICAL DEV = 3350
                                 PHYSICAL DEVICE = 3350
ICK003D REPLY U TO ALTER VOLUME 160 CONTENTS, ELSE T
ENTER INPUT/COMMAND:
```

ENTER

```
u
```

SYSTEM RESPONSE

```
U
ICK01307I DEFECTIVE-TRACK LIST IN HEXADECIMAL FOR VOLUME
          D33500
ICK01310I NO DEFECTIVE TRACKS WERE FOUND.
ICK01313I VOLUME CONTAINS 150   ALTERNATE TRACKS -- 150
          AVAILABLE.
ICK01314I VTOC IS LOCATED AT CCHH=X'0000 0001' AND IS  1
          TRACKS.
ICK00001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
```

Case 2: Using DSF for a Full-Volume Minidisk

You have a 3350 defined in the directory at virtual address 460. Make sure you have a volid on the volume that contains the minidisk.

```
MDISK 460 3350 000 555 D33500 MW ALL W460 M460
```

Link to the minidisk:

```
link * 460 460 mw
```

The process now is the same as in Case 1, with one difference. Because you are initializing a minidisk instead of a dedicated volume, you **must** include the MIMIC(MINI(nnn)) parameter when you issue the INIT command. Specify nnn as the full number of cylinders on the minidisk. This will prevent DSF from trying to assign alternate tracks to the minidisk. For example:

```
init unitaddress(460) noverify-
devicetype(3350) mimic(mini(555)) volid (d33500)
```

Note: DSF cannot assign alternate tracks for a 3330, 3340, 3350, 3375, 3380, or FBA minidisk. Therefore, you cannot initialize a full-volume minidisk by issuing the INIT command without the MIMIC(MINI(nnn) parameter.

Case 3: Using DSF for a Minidisk That Is Less Than a Full Volume

You have a 3350 minidisk defined in the directory at virtual address 462, and it contains 20 cylinders:

```
MDISK 462 3350 000 020 D33500 MW ALL W462 M462
```

Link to the minidisk:

```
link * 462 462 mw
```

Then follow the procedure discussed in Case 2, but specify the INIT command with only 20 cylinders in the MIMIC(MINI(nnn) parameter:

```
init unitaddress(462) noverify-
devicetype(3350) mimic(mini(020)) volid (d33500)
```

Using CP Commands to Enhance Performance

You can use certain CP commands to aid the performance of MVS guest virtual machines. Bear in mind that improving the performance of one virtual machine may impair the performance of others.

Before using the following commands, refer to the *CP Command Reference.*

LOCK Use the LOCK command to permanently lock in real storage selected pages of a V=V guest's virtual storage. Those pages are excluded from future paging activity. Make sure you have enough page frames available before issuing this command, or you will severely degrade the performance of other virtual machines.

SET FAVOR Use the SET FAVOR command to provide a specific percentage of processor time for one or more favored MVS guests. When you specify a percentage of 100, one favored MVS guest is placed at the top of the dispatch queue and held there until it logs off.

SET MINWS Use the SET MINWS command to set the minimum working set size (in number of pages) to a high value for one or more MVS guests.

SET RESERVE Use the SET RESERVE command to reserve page frames of real storage for one or more MVS V=V guests.

SET SRM IBUFF Use the SET SRM IBUFF command to increase the amount of storage available to MVS V=V guests by limiting the size of the interactive buffer.

SET SRM PREPAGE Use the SET SRM PREPAGE command to control the number of swap sets that are swapped into main storage when the system adds a virtual machine to queue.

SET PRIORITY Use the SET PRIORITY command to improve an MVS guest's dispatching priority in relation to that of other users on the system.

Problem Determination

Problem Recognition

System failures appear in the usual way when you run MVS under VM/SP HPO. Abends, wait states, loops, and incorrect results are the basic indicators of system problems. See the *Virtual Machine Diagnosis Guide* for a comprehensive description of failure symptoms and how to handle them.

When you run MVS under VM/SP HPO, any of these system failures may occur with either MVS or VM/SP HPO in control of the processor. *One special concern* is the case where a preferred machine assist guest enters a disabled loop. This prevents the VM system operator from regaining control of the system and entering commands from the master console. It is **strongly recommended** that you confine any testing of MVS systems to V=V virtual machines.

Overview of Preferred Machine Assist

An MVS preferred guest runs in real supervisor mode and is able to issue native I/O instructions without having CP intercept and simulate these operations. In addition, MVS exclusively owns a set of preferred guest channels. These channels and their associated devices are not included in DMKRIO. Therefore, no RDEVICE, RCTLUNIT, or RCHANNEL control blocks are generated for preferred guest devices. Status information concerning I/O to these devices can only be obtained from control blocks maintained by MVS (such as UCB). However, the preferred guest can also access devices known to CP via LINK, ATTACH, or DEDICATE commands. When accessing these devices, CP controls I/O operations and maintains status information.

While operating as the preferred guest, MVS is given the system's absolute page 0, allowing I/O interrupts to be directly reflected into MVS's low storage by the hardware. During CP initialization, CP's page 0 is dynamically relocated to a page above the V=R area to let MVS use absolute page 0. A prefix register is used to provide addressability to the relocated CP page 0. This also occurs on a uniprocessor.

MVS does not have to be modified to run under VM/SP HPO. Because MVS makes no tests during initialization to determine whether it is running natively or under a VM/SP HPO system, no VM/SP HPO handshaking facilities are utilized. MVS always assumes that it is controlling a native processor complex with its associated system resources.

Use of Service Aids

Whenever an abend occurs in CP, the module DMKDMP takes a system abend dump. A system abend dump is also created whenever the system operator presses the PSW Restart key. Consider the following when obtaining a CP dump:

- Use the CP SET DUMP command to direct an abend dump to an output device and to determine the contents of the dump. SET DUMP AUTO CP will be sufficient for most CP problems. SET DUMP AUTO ALL will include guest storage as well as that of CP. **Remember**:
  - SET DUMP AUTO CP dumps only CP pages.
  - SET DUMP AUTO ALL dumps all of real storage.
  - SET DUMP AUTO V=R dumps CP storage and V=R user storage in the V=R area below the 16Mb line.
- You should always direct dumps to DASD devices. V=R guest recovery will not work if you direct dumps to printers or tapes. Be aware that timeouts may occur on the preferred guest's network even when the system dump is directed to DASD and CP has successfully restarted.

  You must define enough DUMP or TEMP space on the system to contain the system abend dump. If not, the dump will automatically go to the real system printer.
- The system operator can take a system abend dump by performing a PSW Restart. Be especially careful when attempting a PSW RESTART on a VM/SP HPO system with single processor mode active. The PSW RESTART will be reflected directly to the preferred guest unless you follow the procedure described later in this chapter, under the heading "How to Obtain a VM/SP HPO Dump in Single Processor Mode."
- When you are using preferred machine assist, a restart will be reflected to either CP or the preferred guest if it is in control when you press the Restart key. Ensure that CP is in control before you press RESTART.
- The CP system abend dump should give you enough information to resolve most CP failures. Use IPCS/E for formatting VM/SP HPO dumps. VMFDUMP will not process dumps of more than seven megabytes.

MVS Dumps

In many cases, you must gather information from the MVS preferred guest when analyzing problems. MVS provides many excellent service for aids this. For detailed information on these service aids, see the *OS/VS2 System Programming Library: Diagnostic Techniques* (GC28-0725).

How to Obtain an MVS Stand-alone Dump

You can use the MVS stand-alone dump to dump MVS storage. This may be the only way you can dump the system if MVS has entered a disabled loop. Consider these operational points when using the MVS dump:

1. Issue a CP STORE STATUS command machine before IPLing the dump to save register and PSW contents. **YOU MUST ISSUE THE CP STORE STATUS COMMAND FROM THE MVS VIRTUAL MACHINE CONSOLE, NOT FROM THE SYSTEM CONSOLE.**
2. Display the first X'20' bytes of CP's absolute page 0 to save CP's internal trace table pointers.
3. When IPLing the stand-alone dump from the preferred guest, IPL with the PMA option. This is the only way the dump program can access the preferred guest devices that contain the MVS page data sets. Also, the dump must be in PMA mode to dump storage above the 16 Mb line.

How to Obtain a VM/SP HPO Dump With Preferred Machine Assist but not Single Processor Mode

To take a restart dump when running PMA and/or SPMODE the following cases have to be considered:

1. If the system is not running in single processor mode you must ensure that the preferred guest is not being dispatched. You can do this by stopping the processor on which CP is running (using the STOP CPx command, not the STOP key) and looking at the *prefixed* storage location 0000 of this processor. If CP has control, you should have 000C0000 00000xxx, where xxx is pointing to DMKPSADU (usually 868 or 870).

```
                              ALTER / DISPLAY
01 PRI VIRT=REAL
02 SEC VIRT=REAL      ADDRESS                    DATA
03 STORAGE KEY REAL
04 STORAGE REAL       00000000  000C0000  00000870  00000000  00F21000
 . . . . . . . . . . . . . . . .
 . . . . . . . . . . . . . . . .
```

2. If you do not have 000C0000 00000xxx, it means that the preferred guest had control when you stopped the processor. You must place the preferred guest in the dormant state, either by placing it in CP READ, or by doing a START/STOP on the processor, to get a chance to be within CP. (Use the START CPx and STOP CPx commands, not the START or STOP keys.)
3. If you have 000C0000 00000xxx at prefixed 0000, CP had control. Use the RESTART CPx command from the system console (where x is the processor on which CP was running) to get a CP RESTART dump.

Note: Any other process may result in the guest address space being cancelled and no CP dump taken.

How To Obtain a VM/SP HPO Dump in Single Processor Mode

1. To make sure that the guest virtual machine is not being dispatched, stop the processor on which CP is running (using the STOP CPx command, not the STOP key) and look at the *prefixed* storage location 0000 of this processor. If CP has control, you should have 000C0000 00000xxx, where xxx is pointing to DMKPSADU (usually 868 or 870).

```
                              ALTER / DISPLAY
01 PRI VIRT=REAL
02 SEC VIRT=REAL      ADDRESS                  DATA
03 STORAGE KEY REAL
04 STORAGE REAL       00000000  000C0000  00000870  00000000  00F21000
 . . . . . . . . . . . . . . . .
 . . . . . . . . . . . . . . . .
```

2. If you do not have 000C0000 00000xxx it means that the SPMODE machine had control when you stopped the processor. You must place the preferred guest in the dormant state, either by placing it in CP READ, or by doing a START/STOP on the processor, to get a chance to be within CP. (Use the START CPx and STOP CPx commands, not the START or STOP keys.)

3. Subtract 8 bytes from the address you found (868 will become 860; 870 will become 868) and store X 'FF' at this *prefixed* address. You can do this by moving the cursor and typing over the first byte. Do not change the other three bytes. Your console would look like this:

```
                              ALTER / DISPLAY
01 PRI VIRT=REAL
02 SEC VIRT=REAL      ADDRESS                  DATA
03 STORAGE KEY REAL
04 STORAGE REAL       00000868  FF000000  00000000  9110034A  471008A0
 . . . . . . . . . . . . . . . .
 . . . . . . . . . . . . . . . .
```

4. Use the RESTART CPx command from the system console (where x is the processor on which CP was running) to get a CP RESTART dump.

Note: Any other process may result in the guest address space being cancelled and no CP dump taken.

If you need more details about how to use the console to display and alter main storage, refer to the appropriate System/370 operating procedures publication.

SVC Dumps

MVS's SVC dumps are a useful source of information about MVS failures. At times, they provide clues to what appear to be hardware problems. **Remember**, MVS does not know that it is running under VM/SP HPO. Certain CP instruction simulation errors may appear to the MVS guest as hardware failures. For very difficult hardware-like errors, you may have to ZAP the MVS system in order to load a disabled wait PSW. This will allow you to dump the MVS system using the MVS stand-alone dump.

Error Recording and Analysis

MVS records both hardware and software errors on the SYS1.LOGREC data set. This information is critical to tracing the sequence of events that caused a system failure. Use the information contained in SYS1.LOGREC along with CP's dumps and error recording data for greater information concerning system failures.

How MVS and CP Use SVC 76

When MVS runs in a virtual machines, it uses SVC 76 to write error records to the error recording data sets. However, in a nonpreferred virtual machine CP intercepts SVC 76 and records the error in its own error recording area. Therefore, error records from MVS reside in this one centralized error recording area. To access the recorded data, use the CMS CPEREP command. For further information about error recording, formatting output from the error recording area with Service Record File devices, and CPEREP refer to *VM/SP HPO OLTSEP and Error Recording Guide.*

Error Recording in Single Processor Mode and Preferred Machine Assist

When VM/SP HPO is in single processor mode and/or preferred machine assist mode CP does not intercept SVC 76 (the error recording SVC) in the preferred guest. Thus, error records for the MVS V=R guest may be in two locations:

- The VM/SP HPO error recording cylinders when SVC 76 is issued in the VM/SP HPO processor, and
- The MVS SYS1.LOGREC data set when SVC 76 is issued in the non-VM processor.

To find all the error records that pertain to the MVS V=R guest, you must look in both locations. To put the records in chronological sequence, you can follow the time and date recorded in each record.

Note: Duplicate error records appear for channel checks reflected on the VM processor. When MVS in the IPL processor issues SVC 76 for a channel check, CP intercepts the SVC 76 and records the error in its error recording cylinders. However, CP then passes the SVC 76 back to MVS for recording in its SYS1.LOGREC data set.

CP Trace Table

The most important area in any CP dump is the CP internal trace table. This is the key to tracing the sequence of events that preceded a system failure. VM/SP HPO allows the spooling of trace data to an output device. CP does not trace I/O events to preferred guest devices because these instructions are never intercepted and simulated by CP.

MVS Trace Table

The MVS system trace table is as important as the CP trace when examining failures for preferred guests. It provides the sequence of events that preceded a system failure.

This trace table will contain all SIOs issued to preferred guest devices, as well as the interrupts received from that I/O. If the preferred guest has issued a LINK, ATTACH, or DEDICATE to a CP-generated device, you must examine both the CP trace table and the MVS trace table to understand the entire I/O event.

The External, SVC, and Program interrupt trace table entries provide additional information concerning the status of the MVS system.

CP Control Blocks

Refer to *VM/SP HPO Data Areas and Control Block Logic- CP* when examining CP control blocks. Valuable status information about a preferred guest is contained in the PSA, VMBLOK, and the MICBLOK.

When you use CP's dump routine, module DMKDMP will move MVS's absolute page 0 to DMKSLC-4096. It will move CP's page 0 to absolute page 0. This occurs in both single processor mode and non-single processor mode. For all system abend dumps, CP's PSA can be found in the first page of storage.

However, when you use the MVS stand-alone dump, there is no movement of PSAs. The MVS stand-alone dump dumps storage without changing any locations.

CP Command Restriction for Problem Determination

You cannot issue ADSTOP or TRACE commands from a preferred guest.

Trace Table Recording Facility

The Trace Table Recording Facility expands problem determination capability to service personnel and system programmers. The facility uses the CP CPTRAP command to create on a reader spool file a chronological record of selected trace table, virtual machine interface, and CP interface information. Use this facility to help analyze VM/SP HPO problems that you cannot detect with a system dump.

A CMS utility program, TRAPRED, is included as part of the CPTRAP facility. TRAPRED uses the reader file as input, and supports output to either a spooled print file or an interactive terminal display. For additional information on using the CPTRAP facility, refer to *Virtual Machine Diagnosis Guide.*

Summary of How to Approach a Diagnostic Problem

The following recommendations will help you analyze failures on VM/SP HPO:

In general, approach CP abends and disabled wait states from a CP perspective. Interrogate the abend or wait state code first. System hangs or loops may be more difficult to diagnose. Your initial question should be, "Which SCP was in control of the system at the time of the failure?"

As a general rule, assume the preferred guest was in control. Gather as much information as possible from the preferred guest. In particular, find the MVS trace table and interrogate the trace entries for an understanding of MVS/SP's perspective of the failure. Always examine all I/O control blocks — such as UCBs and IOBs — for possible hardware errors. Remember that MVS/SP controls all preferred guest devices.

Coordinate the examination of the MVS trace table with an interrogation of the CP internal trace. Understand what has occurred in CP prior to the system failure. Remember that even when MVS is running with preferred machine assist, it continues to operate as a guest machine under CP. CP still controls the scheduling and dispatching of the MVS guest and may have other virtual machines to service. Use both system trace tables to help diagnose those problems that are unable to be solved with a single SCP's trace table.

Transitions to and from Single Processor Mode

This section discusses the steps involved in placing your VM/SP HPO system into single processor mode. Special considerations and restrictions will also be discussed.

How to Put VM/SP HPO in Uniprocessor Mode

Before you can use single processor mode, VM/SP HPO must be in UP mode. If you are in AP or MP mode, the system operator must vary a processor offline. Decide which processor is to have native control of MVS and vary it offline.

How to Vary Offline a 308x, 3090, or 4381 Processor

To vary offline a 308x, 3090, or 4381 processor for single processor mode, use the CP command:

```
vary offline processr nn vlog
```

If the FORCE or VPHY option, or no option of the CP VARY command is used, the processor will be physically varied offline and will require that MVS physically vary online the processor before you can use it. This may take several minutes.

Setting Single Processor Mode On

Once the correct processor is offline, the system operator can then turn on single processor mode by issuing the CP command:

```
spmode on
```

After the system operator enables single processor mode, the virtual machine operator must then vary online the offline or second processor to the MVS virtual machine using the MVS VARY command.

Setting Single Processor Mode Off

To get out of single processor mode, the system operator must vary the second processor offline from the MVS virtual machine and issue the CP command:

```
spmode off
```

If the system programmer generated VM/SP HPO as an AP/MP system, after setting single processor mode off, the system operator must vary the second processor online to the VM/SP HPO system. The system operator uses the CP command:

```
vary online processor nn
```

VM/SP HPO automatically returns to MP mode and resumes AP or MP applications using the second processor.

Verifying Single Processor Mode

To check if VM/SP HPO is in single processor mode, issue the command:

```
query spmode
```

If single processor mode is on, the operator or user will receive a message that indicates single processor mode is active.

Restrictions for Single Processor Mode

- There must be a path from the VM processor to the MVS system residence device.
- **Do not specify** OPTIONS=(CRH) in the MVS CTRLPROG system generation macro. Single processor mode does not support either real or virtual channel reconfiguration. Thus, if an MVS system is to operate in a virtual machine and use single processor mode, do not generate the system with channel reconfiguration hardware (CRH) support.
- On a 308x or 3090 processor, the MVS VARY PROCESSOR OFFLINE command disconnects the channel set of the processor that was varied offline. That is, if an operator issues VARY PROCESSOR OFFLINE on a 308x or 3090 and the channel set of the real processor is identical to the channel set in the MVS sysgen, the processor that was varied offline will lose its channels. When the operator sets single processor mode off and varies the processor back online, that processor will not have I/O capabilities. To reconnect the channels, the operator should use the hardware reconfiguration frame on the 308x before varying the processor online.

Warnings for MVS Operators Using SPMODE or Preferred Machine Assist Without Control Switch Assist

VM/SP HPO cannot intercept the DIAGNOSE instruction for a preferred guest, or for the native processor when you are using SPMODE. 308x processors use the hardware DIAGNOSE instruction to vary channels and storage. Therefore:

- **DO NOT** use the MVS VARY command to vary offline the VM processor, channels, or storage. Varying the VM processor offline by MVS, may cause an abnormal termination of VM/SP HPO.
- Be careful about the increments of storage that you vary offline. On some processors, storage can be physically varied only in units of 4 or 8 Mbs. You can unintentionally affect VM/SP HPO's storage.
- **DO NOT** invoke Alternate CPU Recovery (ACR) to restart the processor on which MVS runs under VM/SP HPO.

- **DO NOT** attempt to change the TOD clock setting with the MVS SET CLOCK command. The TOD clock can only be changed by VM/SP HPO at the time it is IPLed.

- **DO NOT** use the MVS command QUIESCE. MVS (not CP) determines which processor the MVS V=R virtual machine dispatches to handle the QUIESCE command. If MVS dispatches the task on the MVS native processor, that processor will issue SIGP STOP and STORE STATUS to the processor CP is controlling. This puts the CP processor into a manual state. Also, the registers from the SIGP STOP and STORE STATUS commands may not be the registers of the MVS V=R virtual machine. Results are unpredictable if the MVS V=R virtual machine tries to use these registers.

- Make sure you know which function you will get when you press the RESTART key. When running a preferred guest but **not** in single processor mode, pressing the RESTART key will affect whichever control program (MVS or VM/SP HPO) is in control of the hardware at that time. Therefore:

  - If you want to force a VM restart, first hit PA1 on the VM console of the MVS machine to make sure it is stopped.

  - If you want the MVS function of the RESTART key, issue:

    ```
    #cp restart
    ```

- In single processor mode, all restart interrupts are passed to MVS. Stopping MVS will not force a VM restart.

- In single processor mode, you must ensure that the VM processor allocates enough processor cycles to the MVS guest. Otherwise, spin loops will result as the MVS guest attempts to keep pace with the native side. To prevent MVS spin loops, use the CP SET FAVOR command to ensure that the MVS guest receives the required amount of processor time.

- The MVS stand-alone dump program may not be able to function properly when it you IPL it in a virtual machine. The IPL simulator uses one page of virtual machine storage for its simulation routines. In nonpreferred MVS guests, a DIAGNOSE is issued to restore the contents of that page. Since a preferred guest cannot issue a DIAGNOSE instruction, that page is not restored. If critical MVS control blocks are located in that page, dump formatting may not be possible.

Shadow Tables

You should use shadow tables properly for the best performance of an MVS V=V or V=R guest. **Shadow tables** are page and segment tables created and used by CP to control the virtual storage of MVS.

When MVS runs under VM/SP HPO:

first-level storage is the real storage that VM/SP HPO controls

second-level storage is the virtual storage that VM/SP HPO creates and manages for a virtual machine like the MVS system control program

third-level storage is the virtual storage in which an MVS V=V guest runs and is managed by MVS.

CP uses shadow tables to map third-level storage addresses to real, or first-level, storage addresses. Without them, CP needs to perform two sets of segment and page table manipulations for V=V guests.

While shadow tables can improve system performance in some environments, they can impair it in others. CP needs to do extra work to maintain them, and in some cases, this extra work is unnecessary. For example, an MVS V=R guest does not benefit from shadow tables because its virtual addresses are equal to real addresses.

Use the following information to control CP's use of shadow tables and to improve the performance of your MVS guest.

Five Things You Need to Know About Shadow Tables

Enough Free Storage
The more shadow tables you use, the more free storage you need. Each shadow table for a 16 Mb address space requires 1024 bytes of free storage, plus storage for the related page tables.

STFIRST Directory Option
You must have this option in the directory of the guest virtual machine if you want to issue the CP SET STBYPASS command.

Specify this directory option only for virtual machines that execute debugged and tested production workloads.

Do not specify this option for guest operating systems or guests running programs that do not follow the programming restrictions for shadow table bypass. These restrictions are listed later in this section.

CP SET STBYPASS Command
This command allows V=R users to eliminate shadow tables. It allows V=V users to reduce shadow table processing. Use this command *after* you IPL the MVS guest.

CP SET STMULTI Command
This command allows V=R or V=V users to have VM/SP HPO maintain multiple shadow tables for a virtual operating system such as MVS that uses multiple segment tables. Use this command *after* you IPL the MVS guest.

CP QUERY SET Command
The response to this command displays current shadow table settings.

How to Control Shadow Tables for a V=R Guest When Not in Single Processor Mode

A V=R guest will perform better without shadow tables. You should issue this command after you IPL a V=R guest:

```
set stbypass vr
```

When you issue this command, CP doesn't create shadow tables for the V=R guest. Instead, CP uses the segment and page tables maintained by MVS.

Since issuing this command eliminates shadow tables, you do not need to issue the CP SET STMULTI command.

How to Control Shadow Tables for a V=R Guest While in Single Processor Mode

In this mode of operation, VM/SP HPO needs shadow tables to simulate virtual prefixing. Therefore, **do not** issue SET STBYPASS VR for a V=R guest while in single processor mode.

However, you should use the CP SET STMULTI command to specify a minimal number of shadow tables. You can specify a value as low as two in this environment; the default value is three. You should also use the CP SET STBYPASS *nnnnn* K command, described later in this section.

How to Control Shadow Tables for a Preferred Machine Assist Guest

Preferred machine assist does not use shadow tables. A preferred guest cannot use the CP SET STBYPASS or the CP SET STMULTI command.

How to Control Shadow Tables for a V=V Guest

V=V guests benefit by more than one shadow table because MVS has more than one address space. Use the CP SET STMULTI command to create a pool of shadow tables. The syntax of the SET STMULTI command is:

```
SET STM n USEG xx CSEG yyy
```

Where:

STM n is the number of shadow tables (one for each MVS address space) that you want CP to maintain. You can specify a number between 1 and 16.

USEG xx is the number of contiguous shadow page tables (number of segments in the user area) for the V=V guest's dynamic paging area. USEG xx can be set to 0, or range from 8 through 99.

CSEG yyy is the number of contiguous segments for the common area (at the high end of an address space) that is shared by all address spaces. CSEG yyy can be set from 0 to 128.

Setting the STMULTI n Parameter

If your MVS guest is using cross memory services, set n equal to 16.

Otherwise:

Specify a number equal to the average number of initiators that are active at one time, plus two. (You add two because there is one address space for JES and one for the master scheduler.)

Setting the USEG xx Parameter

To correctly set this parameter, you need to monitor the activity of the page table steal counter in the ECBLOK, which is an extension to the VMBLOK. You will have to experiment with different values over various time periods and use VMMAP to collect the needed data.

Start with a low USEG value and gradually increase it.

- If you observe a high increase in the counter (a three- or four-digit hexadecimal value), keep increasing the USEG value.
- When you observe a small increase in the counter (a two-digit hexadecimal value), the USEG parameter is set correctly.

Using VMMAP is the easiest way to report data about shadow table activity. However, if you want to manually locate the page table steal counter, the procedure is as follows:

1. Enter:

   ```
   #cp loc userid
   ```

   The userid in this command is the name of the user's virtual machine. The system response shows the address of the user's virtual machine block (VMBLOK).

2. Add X'0C' to the VMBLOK address to locate the pointer to the ECBLOK. This the field VMECEXT.

3. Locate the ECBLOK.

4. Add X'7C' to the ECBLOK address to locate the page table steal counter. You will find this counter in the field EXTUPTST.

Setting the CSEG yyy Parameter When Not in Single Processor Mode

Define this value to the number of 64K segments in the MVS common area.

To calculate this value:

1. Run the AMBLIST service aid program to find the beginning address of the PLPA.

2. Subtract the address found in step 1 from X'FFFFFF', and convert it from hexadecimal to decimal.

3. Divide the result from step 2 by 65,536 (64K) and round it to the nearest 64K segment.

4. Specify the decimal value in step 3 in the CSEG parameter.

This calculation represents the maximum common area size. However, specifying this size for the CSEG parameter may not provide the best performance.

For better performance, organize the MVS PLPA by packing frequently used modules together and putting them in the high address range of the PLPA. The CSEG value should represent the PLPA size from its highest address to the lower address boundary of the packed area.

Setting the CSEG yyy Parameter While in Single Processor Mode

For better performance in single processor mode, use a CSEG value that represents the maximum size of the common area.

To calculate this value:

1. Locate entry CVTSHRVM (X'1A0') in the MVS communication vector table (CVT).

2. Subtract the value in entry CVTSHRVM from X'FFFFFF' and convert the result to decimal.

3. Divide the result from step 2 by 65,536 (64K) and round it to the nearest 64K segment.

4. Specify the decimal value in step 3 in the CSEG parameter.

Using SET STBYPASS to Define the High-Water Mark

The **high-water mark** is the highest limit of the MVS nucleus where the virtual and real addresses are equal. Set this value in the SET STBYPASS ***nnnnn***K (or ***nn***M) command to reduce the overhead incurred by CP as it maintains shadow tables.

Selective Invalidation

Selective invalidation is a function of shadow table maintenance. It allows CP to selectively invalidate a shadow page table entry when a page frame is stolen or released from an MVS guest.

Selective invalidation always takes place below the high-water mark, which you can specify with the SET STBYPASS command. Above the high-water mark, selective invalidation occurs only when virtual machine assist is off.

Setting a Precise Value for SET STBYPASS

To determine the precise ***nnnnn*** K (or ***nn*** M) value follow this method:

1. Locate entry CVTPVTP (X'164') in the MVS communication vector table (CVT). This is the address of the page vector table (PVT).
2. Locate entry PVTFPFN (X'10') in the PVT. This is a half-word value that represents the relative block number (RBN) of the first page frame table entry (PFTE) in the page frame table (PFT).
   - For a pre-MVS/SP 1.3 guest, the value at PVTFPFN is left-justified and the 12 high-order bits are the 12 high-order bits of a 24-bit address. Thus a value of X'0960' at PVTFPFN becomes X'096XXX' in an address. The 12 low-order bits are zeroes, so the result is X'096000' for the address value you are looking for.
   - For an MVS/SP 1.3 or later guest virtual machine, the address calculation is different. The value at PVTFPFN is, in this case, right-justified and its 12 low-order bits are the 12 high-order bits of a 24 bit-address. For example, if PVTFPFN contains a value of X'0096', drop the first four bits (the first zero) and begin the 24-bit address with PVTFPFN's last 12 bits. The address is now X'096XX'. The 12 low-order bits are zeroes so the resulting address is X'096000'.
3. Take the X'096000' and convert it to decimal. The result is 614,400.
4. Divide the decimal value 614,400 by 1024. The result will be the ***nnnnn*** K value you are looking for, in this case 600K.
5. Issue the command:

   ```
   cp set stb 600k
   ```

*Note: The nearer the value for **nnnnn** K (or **nn** M) to the virtual machine size, the greater the reduction in CP overhead.*

Shadow Table Restrictions for V=R Guests

When shadow tables are eliminated, the following restrictions apply:

- The virtual system's real page zero must map only to its virtual page zero. Otherwise, STBYPASS VR will be set off and shadow tables will be used instead.
- No virtual machine segment or page tables can start in a relocated page. This means that VM/SP HPO control register 1 and the segment table entries cannot point to the first 4K of storage.
- The system *cannot* use the relocated page table entry in these ways:
  - By looking at its contents, or
  - By executing a load real address (LRA) instruction on virtual page zero (normally mapped to real page zero), except for using the condition code returned by the LRA instruction.
- The virtual operating system must have only one page table entry for its real page zero. If multiple address spaces are used, the page table must be shared by each address space that uses real page zero.

  Note: Once relocated by the SET STBYPASS VR command, the virtual operating system must continue to use the relocated page table entry without changing its contents or moving its location.
- Any dump taken of the virtual operating system may contain a relocated page table entry for page zero. Thus, any program designed to automatically read and interpret dumps must handle this condition.

Shadow Table Restrictions for V=V Guests

When using shadow table bypass for the V=V user, the following restrictions apply:

- Below the high-water mark, the virtual operating system must map each virtual address, starting from location zero, to its real address.
- When multiple segment tables are used by a virtual operating system, the page tables that correspond to the area below the high-water mark must be common to all segment tables.
- To *invalidate* entries below the high-water mark, the virtual operating system must specify DIAGNOSE code X'10' to release the virtual pages. However, the operating system is not restricted when validating entries below this mark.
- After setting shadow table bypass for the V=V user, the shadow tables are invalidated and rebuilt. Shadow table bypass should be set before using the SET STMULTI command. Otherwise, the STMULTI command will be reset by setting shadow table bypass.

AUTOLOG Facility

AUTOLOG is a convenient way to log on MVS production systems with many I/O devices that run under VM/SP HPO. The I/O devices needed by these production systems require considerable contiguous storage space for VM/SP HPO I/O control blocks. If smaller users log on before these larger production systems, there may not be enough contiguous storage space available for the required I/O control blocks. The logon of the production virtual machine will still be completed, but the I/O control blocks may not be established, and there may not be enough I/O devices to run the production system and its application programs.

To avoid this problem, log on such virtual machines immediately after IPLing VM/SP HPO. Do either of the following:

- Have the system operator issue the CP AUTOLOG command before enabling user terminals, or
- Define the AUTOLOG1 virtual machine in the directory entry. The AUTOLOG1 virtual machine is automatically logged on immediately after VM/SP HPO is IPLed and can be used to log on and IPL virtual machines that need substantial contiguous storage.

Using the CP AUTOLOG Command

Before enabling user terminals, the VM system operator can issue the CP AUTOLOG command for each production virtual machine that requires substantial contiguous storage. The directory entry for the userid indicated by the CP AUTOLOG command must contain an IPL statement for the desired operating system. For more information about the CP AUTOLOG command, refer to the *CP Command Reference.*

Defining AUTOLOG1 in the Directory

To use AUTOLOG1 to log on several virtual machines, define directory statements to load CMS for the AUTOLOG1 userid. The CMS PROFILE EXEC would contain several CP AUTOLOG commands. Each AUTOLOG command initiates one virtual machine containing a production operating system. Each directory entry referred to by the CP AUTOLOG command must contain an IPL statement.

The CP AUTOLOG command in the PROFILE EXEC IPLs the virtual machine. You then gain access to the virtual machine by doing one of the following:

- Logging on with the userid specified in the CP AUTOLOG command
- Issuing the CP SEND command through the secondary user's console
- Issuing the CP DIAL command and specifying the guest userid.

Multiple Systems With AUTOLOG1

In the following figure, AUTOLOG1 initializes CMS in a virtual machine.

```
USER AUTOLOG1 PASSWORD 512K 1M ABG
     ACCOUNT ACCTNO BIN1
     IPL CMS
         CONSOLE 009 3215
         SPOOL 00C 2540 R
         SPOOL 00D 2540 P
         SPOOL 00E 1403
         LINK MAINT 190 190 RR
         LINK MAINT 19E 19E RR
         LINK MAINT 19D 19D RR
         MDISK 191 3330 1 1 UDISKA WR RPASS WPASS
```

In the following figure , the CMS PROFILE EXEC has a CP AUTOLOG command for each virtual machine to be IPLed . In this way, the production virtual machines are automatically logged on in disconnect mode by the CMS PROFILE EXEC. Each userid identified by the CP AUTOLOG command must also have an IPL CMS statement in its directory entry. The last CP command in the PROFILE EXEC logs off AUTOLOG1.

```
       TRACE E;
       ADDRESS COMMAND;
       CP SPOOL CONSOLE START;
       CP SET EMSG ON;
/* The following message will inform the operator that the guest */
/* operating systems are being autologged.                       */
       CP MSG OP The guest MVS virtual machines are being autologged.
       CP AUTOLOG MVSUSER PASSWD1;
       CP AUTOLOG MVSUSER2 PASSWD2;
       CP AUTOLOG MVSUSER3 PASSWD3;
       CP ENABLE ALL
       CP LOGOFF
       EXIT
```

The AUTOLOG1 directory entry and PROFILE EXEC permit the MVSUSER, MVSUSER2, and MVSUSER3 virtual machines to log on to the system in disconnect mode. You access these virtual machines through their secondary user's consoles, if any, or by logging on with the userid of MVSUSER, MVSUSER2, or MVSUSER3 along with the appropriate password.

MVS V=R Virtual Machine Recovery

When an MVS V=R guest is running and CP abends, VM/SP HPO allows that guest to resume operation after a CP abend without requiring an IPL. This is called **V=R recovery**.

During a warm start, VM/SP HPO restores pending interrupts, dedicated I/O devices, I/O control blocks, real storage, single processor mode (including AP/MP support if active), and preferred machine assist support.

VM/SP HPO *does not restore* any changes that you made to your virtual machine configuration using the CP SET or DEFINE command. For V=R recovery to work, you should not issue CP commands such as DEFINE STORAGE for the V=R guest.

Note: Because MVS/SP does not support the use of the Vector Facility in 370 mode, V=R recovery does not save Vector status and Vector registers.

What You Must Do Before CP Abends

For V=R guest survival to work, you must do the following *before CP abends*:

- Ensure that the CP dump facility is directed to disk. If not, use the CP command SET DUMP AUTO to allocate the CP abend dump to disk.
- Issue the CP command SET NOTRANS ON for the V=R guest after the IPL of MVS. (This ensures that MVS does not require CP for CCW translation.)
- Ensure that the V=R virtual machine is logged on when VM/SP HPO abnormally terminates.
- Ensure that module DMKVRR is present in the system. You can do this by checking the nucleus map — DMKVRR is normally in the V=R load list.

When V=R Guest Survival Will Not Work

V=R guest survival **is not possible** if:

- The V=R guest was the running user that caused the abend.
- The V=R guest was in IOWAIT or EXWAIT.
- **You use RESTART** to request a dump and VM/SP HPO is unable to dump, checkpoint, and re-IPL the system. V=R recovery is designed to recover from system abends. A system abend is defined as a situation where CP itself has decided to request the dump and re-IPL.

Preferred Machine Assist Guest Survival

When the system is operating with preferred machine assist active, the MVS V=R guest has more recovery power.

In cases where CP would normally enter a disabled wait state, if a preferred guest is running, CP may be able to allow it to continue operation in native mode. This can provide time for an orderly shutdown of the MVS system, to allow a re-IPL of the VM/SP HPO system to be scheduled to restore VM/SP HPO operations.

Guest Survival with Control Switch Assist

If a preferred machine assist with control switch assist guest issues DIAGNOSEs following a survival incident, the DIAGNOSEs are performed on the hardware. This gives unpredictable results.

Reinitializing After V=R Recovery When Using Control Switch Assist

Because the functions provided by control switch assist support are not recovered, you must reinitialize the following after a V=R recovery:

- MSS Central Server application
- Any user application that uses IUCV or VMCF.

System Activity Display Frames

On 3033, 308x, 3090, and 4381 processors, you can use system activity display (SAD) frames to display processor and channel utilization. The SAD frame is a bar graph that the operator can display at his console.

The system updates the SAD frame every few seconds. When you run VM/SP HPO on a multiprocessor system and are not in single processor mode, the SAD frame will always show 100% utilization for both processors. This is because VM/SP HPO is in an active wait state during which it is continually looking for work.

To display the time you are spending in an active wait state, examine the SAD frame for the percentage of time spent under PSW key 3 in supervisor state. No IBM system runs under PSW key 3 in supervisor state; therefore any time spent in this mode is active wait time. Consult the operator's guide of the appropriate processor for more information on SAD frames.

Note: For an MVS guest running with preferred machine assist, the activity of any authorized user (such as TSO) that runs under PSW key 3 will be recorded as "wait" time. To obtain accurate values of processor utilization for this user, examine the MONITOR records.

Multiple-Access Virtual Machines

Multiple-access programs execute in a virtual machine and directly control terminals. These terminals do not have to be supported by VM/SP HPO as virtual operator consoles, but they can be of any type supported by the program executing. These programs use lines that are either dedicated to the virtual machine (by the directory entry) or assigned to the virtual machine dynamically.

For example: Figure 68 shows two multiple-access systems (controlled by virtual machines MVS1 and MVS2). While each system controls real 3277s by using part of the real 3272, the real 3272 appears to both virtual machines as though they each have sole control of it. (The virtual system consoles of MVS1 and MVS2 are not shown.)

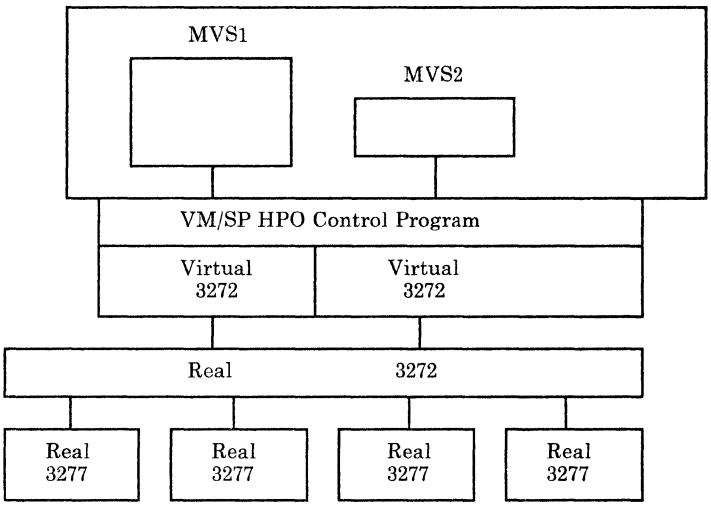
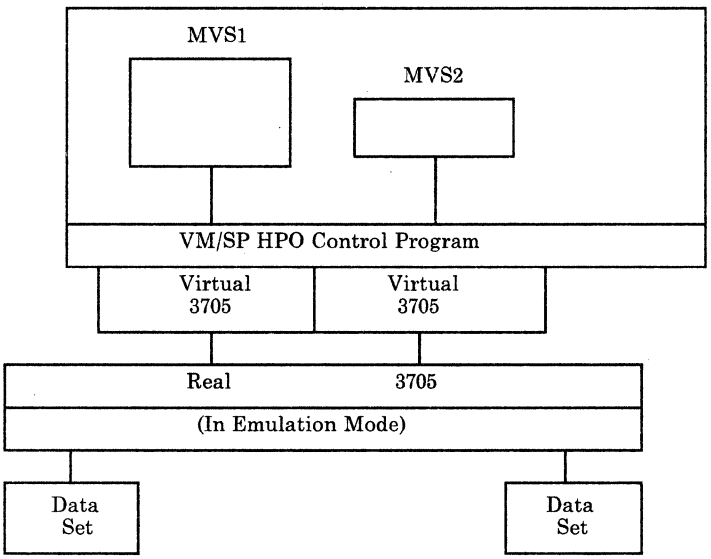


Figure 68. Virtual Devices: Local 3270 Terminals

A subset of the lines of a real transmission control unit (TCU) can be defined as virtual lines for a virtual machine, as shown in Figure 69.



Note: Two lines on the real 3705 are defined as virtual lines for two virtual machines named MVS1 and MVS2. The remaining lines support virtual operator consoles.

Figure 69. Virtual Devices: Remote Terminals

As shown in Figure 70, the virtual machine operating system may be one like MVS, running TSO, IMS, or CICS.

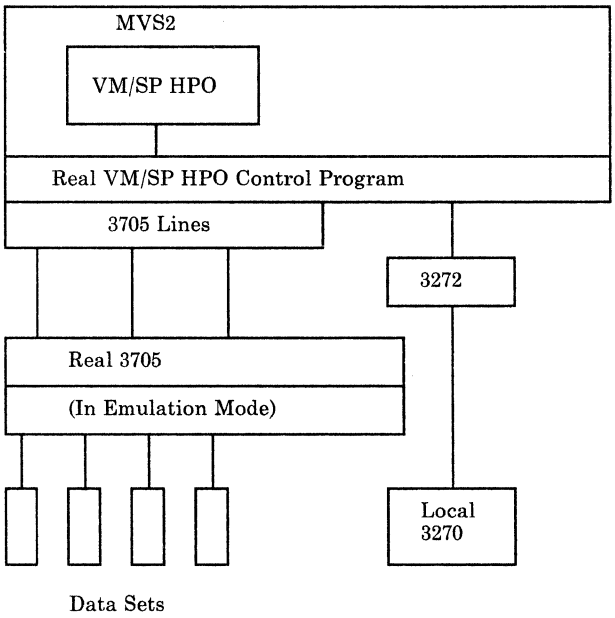


Figure 70. A Virtual VM/SP HPO Multiple-Access System

To assign a real line as a virtual line, the terminals supported by the virtual machine's operating system are of the same type as those supported by VM/SP HPO as virtual system consoles. To make this assignment, define the virtual lines either in the virtual machine's directory entry (via the SPECIAL control statement) or add them to the logged-on virtual machine (via the CP DEFINE command).

To connect a terminal supported by both VM/SP HPO and a multiple-access system, use the CP DIAL command. Such terminals can be on either non-switched or switched lines. To connect a terminal to the virtual machine, issue:

```
dial testvm
```

The VM/SP HPO system matches the terminal type to an equivalent virtual line that is available and enabled (in this example, 070, 080, 081, 082, or 083). Once a connection is made, the virtual machine controls the terminal to which it is logically connected (in this example, the VM/SP HPO virtual machine). It remains connected until one of the following happens:

- The user logs off using standard logoff procedures.
- The virtual machine is forcibly logged off.
- The user issues one of the following CP commands: RESET, SYSTEM RESET, SYSTEM CLEAR, or IPL.

Once dropped, the user is then free either to logon to VM/SP HPO or to use the CP DIAL command to contact another multiple-access system.

Dial-up terminals supported by a multiple-access system may be of a different type than those supported by VM/SP HPO as virtual system consoles. Such terminals must be on switched lines, and the CP DIAL command cannot be used. Users must dial the multiple-access system's telephone numbers directly.

As shown in Figure 71, a communications system can be tested by using multiple virtual machines in place of multiple real machines. For example: While there exists a single two-line 3705 on the real machine, the virtual 3705 units could each be defined as a one-line 3705.

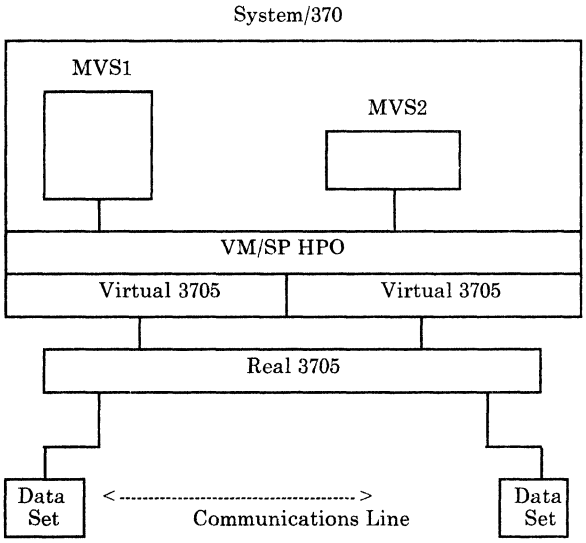


Figure 71. A Communications Test System

Figure 72 illustrates a virtual transmission control unit running remote 3270 units.

When the terminals supported by the multiple-access system are not those supported by VM/SP HPO as virtual operator consoles, the real line appearances must be one of the following:

- Defined in the directory entry for the virtual machine via the DEDICATE control statement; for example:

  ```
  DEDICATE vaddr raddr
  ```

  where **vaddr** is the virtual address, and **raddr** is the real address of the appropriate line appearance on the real transmission control unit.

  ---- or ----

- Attached to the virtual machine by an operator; for example:

  ```
  attach raddr to vm1 as vaddr
  ```

  Where **raddr** is the real address of the appropriate line appearance on the real transmission control unit, and **vaddr** is the address of the line appearance as generated in the virtual machine operating system.

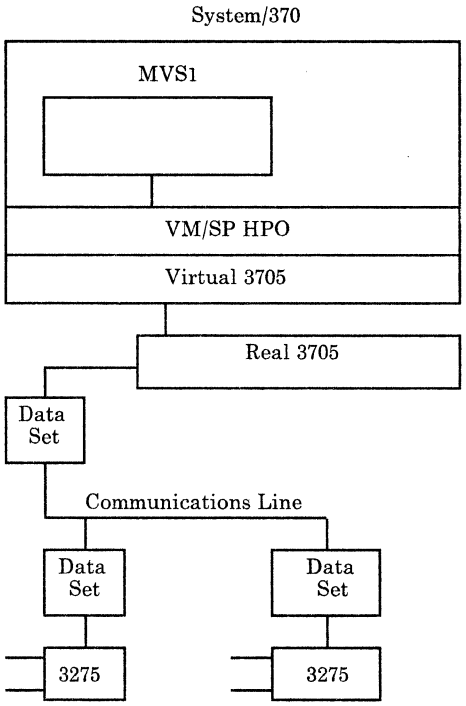


Figure 72. A Virtual 2703 TCU Controlling Remote 3270 Terminals

If you will be using terminals to log on as a TSO user under MVS, you might want to use the IBM Pass-Through Virtual Machine Facility (PVM) or the DIAL command to access the MVS guest. When you DIAL an MVS virtual machine, you logically attach a terminal to it so that MVS can communicate with the terminal.

In order to DIAL your MVS virtual machine, go to a free VM terminal (one with a VM logo) and clear the screen. Once you are in CP READ, enter:

```
dial userid
```

where userid is the MVS guest machine.

This attaches your terminal to the specified userid. VM/SP HPO will select the first virtual graphics device available that you previously defined in the CP directory with the SPECIAL control statements.

Once DIALed, the virtual machine is in control of the terminal.

The following directory entry represents a multiple-access TSO system configured to handle one to four concurrent remote terminals and one local 3270.

```
USER TSOSYS PASSWORD 12M 12M BG
  ACCOUNT SYS00001 VM-FLOOR
  OPTION REALTIMER ECMODE BMX 370E VIRT=REAL
    CONSOLE 01F 3215 C
    IPL 179
 * Spooled unit record devices
    SPOOL 01C 3505 A
    SPOOL 01D 3525 A
    SPOOL 010 3211 A
 * MVS operator's console
    DEDICATE CC0 CC0
 * MVS system residence volume
    DEDICATE 179 MVSRES
    DEDICATE 1A2 1A2
    DEDICATE 1A3 1A3
    DEDICATE 179 M31RES
    DEDICATE 170 M30PGE
    DEDICATE 171 M30SPL
    DEDICATE 172 M30LIB
 * These four entries define communication paths
    SPECIAL 080 2702 IBM
    SPECIAL 081 2702 IBM
    SPECIAL 082 2702 IBM
    SPECIAL 083 2702 TELE
```

Figure 73. A Multiple-Access Virtual Machine

Unsupported Devices

You can use some I/O devices that VM/SP HPO does not support. An unsupported device is one whose device type is not permitted in the DEVTYPE operand of the RDEVICE macro instruction. To use an unsupported device, you must attach or dedicate the device to a virtual machine. The device cannot, therefore, be shared among users. However, VM/SP HPO supports these dedicated devices only under these conditions:

- No timing dependencies exist in the device or the program.
- No dynamically modified channel programs exist in the access method, except when OS/VS TCAM Level 5 is used.
- No special functions need to be provided by VM/SP HPO.
- No other CP restrictions are violated. (Refer to the restrictions list in the *VM/SP HPO Planning Guide and Reference*).
- The device is generated into the VM/SP HPO nucleus (by using the RDEVICE macro instruction with the appropriate CLASS operand).

Analyzing Performance

Use these tools to analyze the performance of MVS under VM/SP HPO:

- Real Time Monitor Program, 5796-PNA (also known as SMART).
- Virtual Machine Monitor Analysis Program, 5664-191 (also known as VMMAP).

For **short-term** study and problem solving, use SMART. SMART provides an online realtime display of performance indicators.

For **long-term** trend analysis or **capacity planning**, use VMMAP. VMMAP is also helpful to analyze system bottlenecks. It reports on delays for resources, both on a virtual machine basis and a system-wide basis. When using VMMAP, the monitor classes USER, SCHEDULE, PERFORM, and DASTAP will yield basic data. The RESPONSE class is useful for studying CMS command response.

Within the VM/SP HPO environment, there are many commands you can use to gather performance information. INDICATE commands provide a broad overview of how system resources are being used.

When you analyze the MVS environment, remember that you have two operating systems running in a single processor. Both VM/SP HPO and MVS are vying for the basic system resources, such as processor, I/O, storage, and paging; both are generating their own accounting information; and both are supplying their own performance information.

MVS under VM/SP HPO Operating Environments

Remember that the MVS SCP is unaware that it is running as a guest under VM/SP HPO. What the MVS guest thinks is real time is actually virtual time. When the MVS guest is executing, elapsed time references are accurate. But when VM/SP HPO dispatches another virtual machine and later redispatches the MVS guest, MVS does not realize that it had stopped running.

3480 Restrictions

1. When you run MVS under VM/SP HPO, it must use all 3480 devices in MVS's 3420 Compatibility Mode. This applies to all VM/SP HPO environments: V=V, V=R, and single processor mode.

2. An MVS guest (V=V and V=R) cannot issue any of the Assign-related CCWs to a 3480 device. These CCWS are:

   - Set Path Group ID
   - Sense Path Group ID
   - Assign
   - Unassign
   - Control Access.

Chapter 4. MVS Virtual Machines Sharing DASD

Hardware for DASD Sharing

Sharing DASD means accessing the same DASD via different paths. In most cases these paths lead from two or more different processors. Two base configurations are possible:

- You have a common control unit which is equipped with a **two-channel switch** (or **four-channel switch**), or
- You have separate control units. In this case, the **head-of-string** needs a **string switch** feature.[13]

---

[13] Technically, the 3380-AA4s do not have a string switch feature. However, the equivalent function of a string switch is contained within the dynamic path selection (DPS) feature. This is the **only** portion of DPS that VM/SP HPO supports and uses.

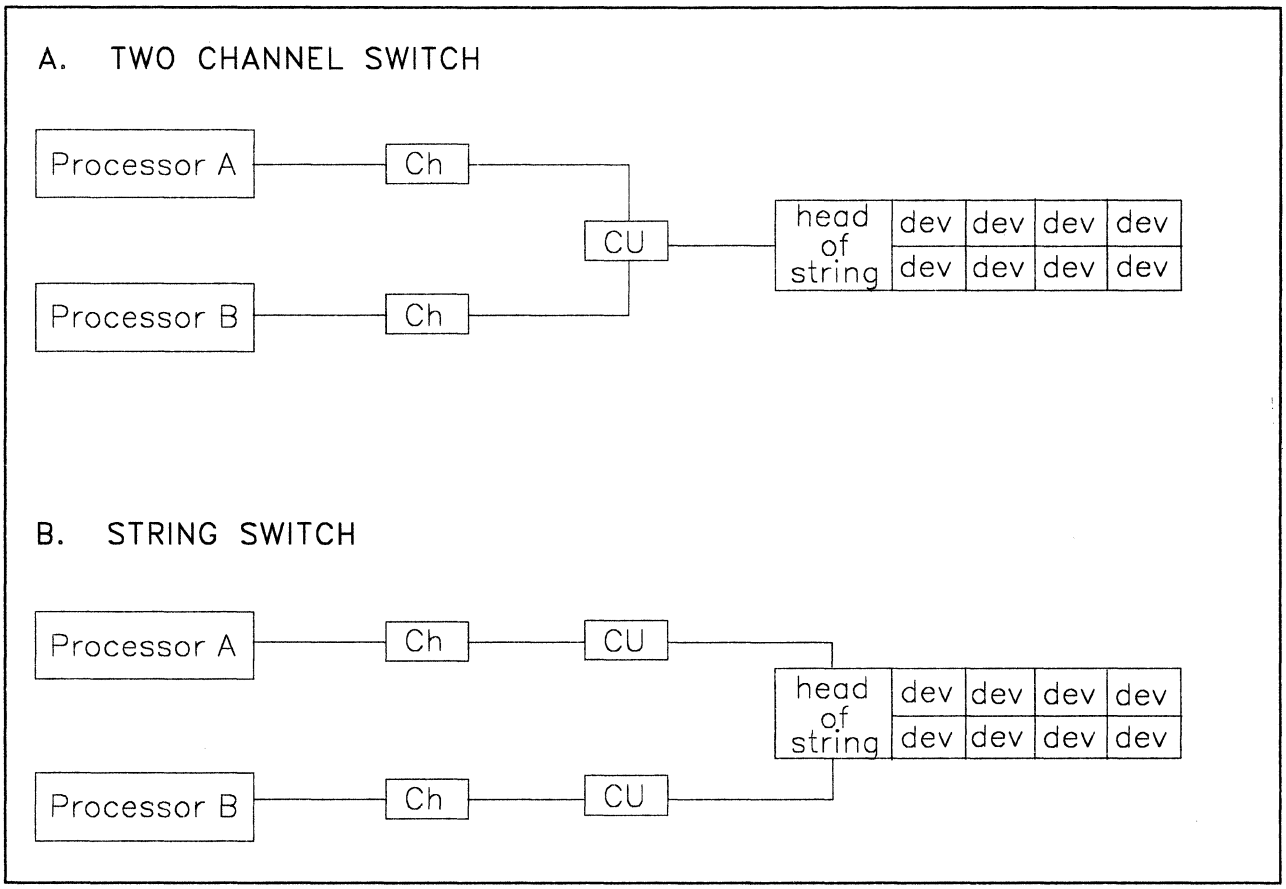


Figure 74. Two-Channel Switch and String Switch

From a DASD sharing standpoint, the two configurations are similar. The VM/SP HPO software supporting shared DASD doesn't see a difference. It is the path to the DASD that matters. Therefore, the following figures do not show channels and control units.

Sharing DASD is not restricted to just accessibility from different paths and different systems. It also means that it must be possible to share the DASD in **write-mode** from different paths. To avoid concurrent update or simultaneous writing — and destroying the DASD contents — the hardware is equipped with a special feature, **reserve/release**.

As soon as your hardware has more than one path (For example, either a two-channel switch or a string switch is installed along the path from the channel to the DASD), then it has the **reserve/release facility** as well. If only one path exists for the DASD and neither a two-channel switch nor a string switch exist along that path, the DASD device type will determine whether the hardware will reject the reserve/release commands or not.

For example, IBM 3375s, 3380s, and 3350s will not reject a reserve or release command, whether or not the switching hardware is installed along the path to the device. On the other hand, IBM 3370s and 3330s will issue a COMMAND REJECT to a reserve or release CCW if either a two-channel switch or a string switch do not exist on the path to the DASD.

Optionally, SCP software can use the hardware reserve/release facility. However, complete protection and data integrity are possible only if reserve/release CCWs are used along **all** paths.

The following figure shows how reserve/release works:

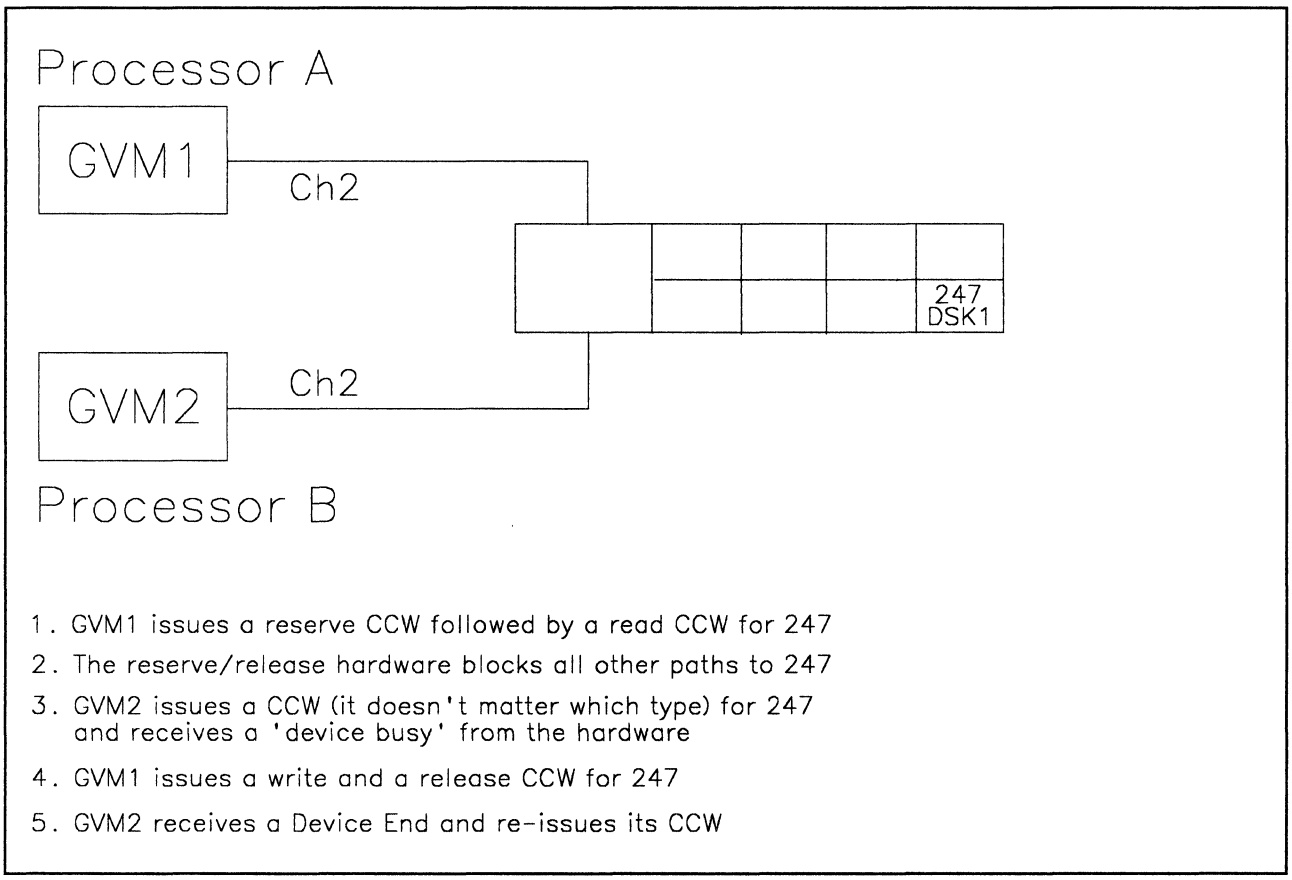


Figure 75. Reserve/Release Hardware

Reserve/Release CCWs

Reserve/release CCW commands prevent several users of the same data files from simultaneously accessing the same data.

A **reserve CCW** is an I/O command that is sent from an operating system to a channel. Its purpose is to reserve a single DASD to a particular channel on a specific processor for its own exclusive use. This is obtained through the reserve/release hardware contained in either the DASD control unit or DASD head-of-string.

Essentially, the reserve/release hardware restricts access to this particular DASD to a specific channel/control unit/head-of-string path. Therefore, a reserve CCW is treated as a path reservation.[14] Once a path is reserved the device can only be accessed through this path. All accesses to this device using a different path result in a device busy condition while the device reservation is in effect. Access to other devices on the same string is not affected by this device reservation.

A **release CCW** is sent via the reserved path only. It ends the path reservation for this user. All paths that received a device busy condition will receive a Device End (DE) indicating that the device is now available. The device can now accept I/O operations from all paths.

New IBM DASD devices such as the 3375-A1/D1 and the 3380-AA4 can complicate string switching, DASD sharing, and the reserve/release facility. However, this can be quickly clarified.

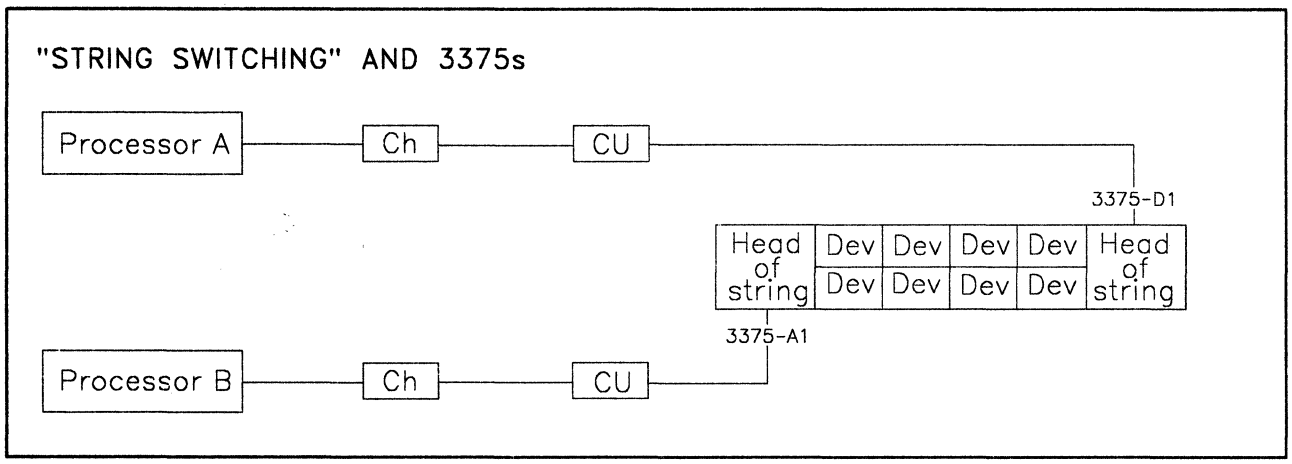


Figure 76. 3375 Configuration with Two Heads-of-String

In Figure 76, a 3375 string is connected to Processor-A through the 3375-D1 unit and to Processor-B through the 3375-A1 unit. Some confusion may arise when you try to understand how an I/O coming across the path from Processor-A through the 3375-D1 will detect a device reservation that was made from Processor-B through the 3375-A1.

---

[14] However, 3380-AA4s are an exception when the full dynamic path selection feature is used. With full support of this feature, MVS can reserve a group of paths to a device. Only MVS uses this feature, VM/SP HPO does not support nor use the full dynamic path selection feature. Also, VM/SP HPO does not allow an MVS guest to use it unless MVS runs as a preferred machine assist guest and the 3380 is only accessed via preferred channels (i.e., channels which are known only to the preferred guest and are not defined in DMKRIO).

The actual DASD device does not have any reserve/release status information or logic of its own; all of the device reservation status and logic is handled through the 3375 head-of-string.[15] The question is: for any 3375-A1/D1 string, how does one head-of-string know the reservation status contained in the other head-of-string?

Look at the example in Figure 76 on page 180. When a reserve CCW is sent across the path from Processor-B through the D1-unit, the head-of-string updates its device status information and internally sets up the path reservation to that device so that no other path to that device can access it. Also, the D1-unit will signal this path reservation status to its corresponding A1-unit so that the A1-unit can update its reservation status for that shared device. Thus, both heads-of-string support the device reservation.

The release CCW for this device must be made across the same path as the original reserve CCW in order to be accepted by the hardware. When the D1-unit receives the release CCW for this device, it will release the path reservation, update its own device status information, and inform the A1-unit of this status change. Once again, this device is accessible from all paths.

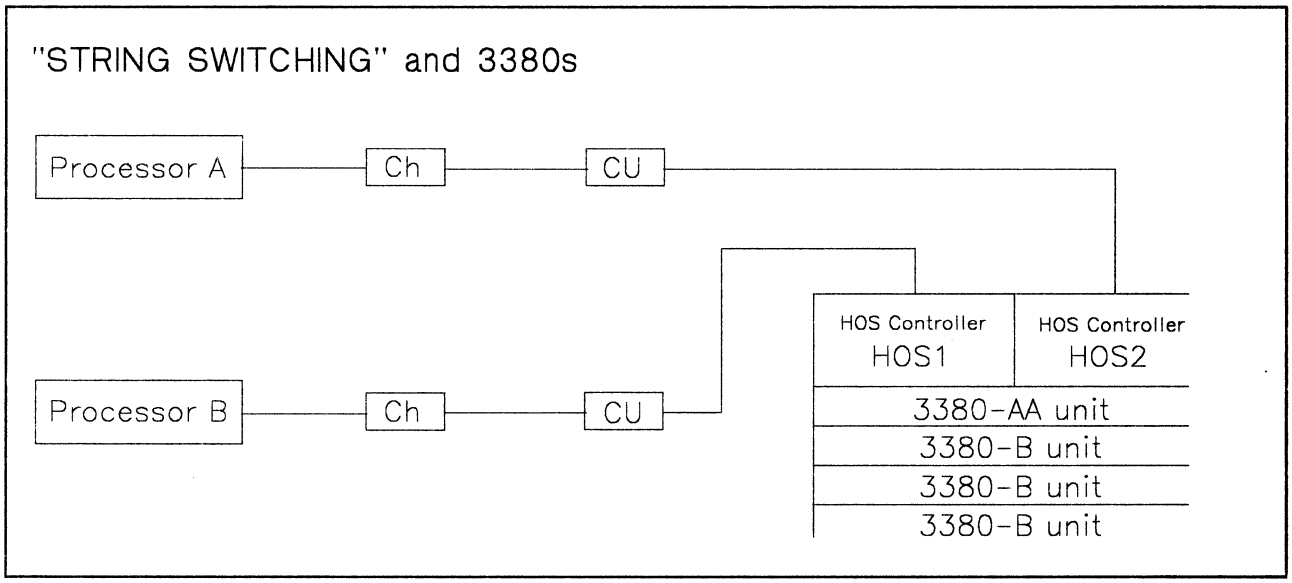


Figure 77. 3380-AA4 Configuration (Two Heads-of-String: HOS1, HOS2)

---

15 This is one of the main reasons why VM/SP HPO does not have to be concerned with whether the DASD sharing is implemented through a string switch or a two-channel switch. The actual reservation status has to be maintained in the head-of-string mechanism. This applies to IBM DASD such as 3350s, 3375s, and 3380s.

From a conceptual viewpoint, the 3380-AA4 handles the hardware reserve/release logic in a VM/SP HPO environment just like the 3375-A1/D1 combination described in the previous example (see Figure 76 on page 180). Specifically, comparing Figure 77 on page 181 with the 3375-A1/D1 example, the 3380 Head-of-String Controller, HOS1, corresponds to the 3375-A1 and HOS2 corresponds to the 3375-D1.

Apart from the software, both HOS1 and HOS2 can access any device on that 3380 string just as the 3375-A1 and the 3375-D1 can access to any device on their 3375 string. Functionally, the synchronization of the device reservation status is the same for 3375-A1/D1 combinations and 3380-AA4s.

VM/SP HPO and Dynamic Path Selection

It is important to note that this similarity between the 3380-AA4s and the 3375-A1/D1s does not exist under MVS/SP. Under MVS/SP, the 3380-AA4s can use the full dynamic path selection (DPS) feature which includes the hardware support of "system-related reserve." This feature allows a group of paths to be reserved by MVS through the 3380-AA4 hardware rather than just a single path.

VM/SP HPO does not support the full DPS feature. This prevents VM/SP HPO from using the "system-related reserve." An MVS guest virtual machine can only use DPS if that MVS virtual machine is running as a preferred guest under VM/SP HPO, **and** all of its paths to the 3380-AA4 originate from preferred channels (that is, those channels not defined in DMKRIO but known to the MVS preferred guest).

Summary of Reserve/Release CCW Support under VM/SP HPO

The following table summarizes VM/SP HPO's support of reserve/release CCWs if issued by a guest operating system such as MVS. Columns 1 through 4 of the table describe cases, depending on whether alternate paths are online, whether the reserve/release hardware feature is present, and whether virtual reserve/release has been defined for the shared DASD. Columns 5 through 7B summarize VM/SP HPO's support of the particular case.

Device Type	Alternate Path Online?	Resv/Rel Hardware Present?	Virtual Resv/Rel Defined?	What is Sent to Hardware?	Error Condition From CCW?	Integrity Problems with Links?	Integrity Problems with Multiple Paths?
Ded [1]	No	yes/no	-	reserve	no/yes	-	no
Ded [2] [5]	yes	yes	-	sense	no	-	yes
Mdisk [1]	no	yes	no	reserve	no	yes	no
Mdisk [1]	no	yes	yes	reserve	no	no	no
Mdisk [3] [1]	no	no	no	reserve	yes	yes	no
Mdisk [4]	no	no	yes	sense	no	no	no
Mdisk [5]	yes	yes	no	sense	no	yes	yes
Mdisk [5]	yes	yes	yes	sense	no	no	yes

Figure 78. Summary of VM Reserve/Release CCW Support

Notes:

1. *Normal Operation. The command is passed unchanged to the hardware.*
2. *When the VM/SP HPO system has been generated with alternate path support for those devices, and these alternate paths are online, then CP does not allow the real reserve CCW to be sent to the hardware. This action causes VM/SP HPO to avoid a possible channel lockout. VM/SP HPO does not return any indication that the device was not physically reserved to the operating system issuing the CCW.*
3. *Without the two-channel switch special feature, VM/SP HPO sends the reserve/release CCW unchanged to the hardware. However, the hardware rejects the command and does not reserve the device.*
4. *Before sending the command to the hardware, VM/SP HPO changes reserve CCWs to sense CCWs, and places a virtual reserve on the minidisk. The real device is not reserved. The virtual reserve prevents other operating systems running under the same VM/SP HPO system from accessing the minidisk. However, these same virtual operating systems can virtually reserve other minidisks located on the same real volume. Because the reserve/release hardware is not present along the path to the DASD devices, VM's virtual reserve/release processing modifies the reserve CCW to a sense CCW. If the reserve CCW had not been modified, it would have been rejected by the hardware.*
5. *When alternate paths to a device are online, VM/SP HPO changes the reserve/release CCW to a sense CCW to prevent a possible channel lockout. In an MP environment, a symmetric alternate path is automatically defined. If that symmetrical alternate path is online the reserve CCW is changed to a sense CCW* **in all cases.**

To better understand how to use the table, look at the explanation of the following example. This case is fully covered later on, so for now, concentrate only on the interpretation of the table entries.

The following example shows how to interpret the first line of the reserve/release table:

Column Number	Explanation
Column 1	The DASD is either DEDICATEd or ATTACHed to a virtual machine. The column heading "DEVICE TYPE?" refers to the way in which the DASD is defined to the virtual machine, either as a minidisk (MDISK) or a DEDICATEd volume.
Column 2	No alternate paths are defined and online to this device.
Column 3	This column **must** be interpreted with Column 6: • When column 3 is Y (YES), column 6 is N (NO). This means that if the reserve/release hardware exists somewhere along the path to the device, no error condition will be returned by the hardware to CP when a reserve or release CCW is issued by a guest virtual machine to this shared device. • When column 3 is N (NO), column 6 is Y (YES). This means that if the reserve/release hardware **does not exist** along the path to the device, a COMMAND REJECT will be returned by the hardware to CP that will reflect this error to the guest virtual machine which issued the reserve or release command.
Column 4	Virtual reserve/release support is not relevant in this case.
Column 5	When a guest virtual machine issues a reserve command to the device, the reserve is sent **unmodified** to the hardware.
Column 6	Covered in the discussion of column 3.
Column 7	This column is not applicable to this case because one cannot link to a DEDICATEd volume.
Column 8	Because the reserve CCW is always passed to the hardware, there are no problems with having multiple paths to this device online. For example, there can be more than one path online to this device either from the same or from a different system as long as it is not defined as an alternate path.

Real Reserve/Release

Real reserve/release support allows several operating systems, such as MVS, to have data protection on a full volume. This data protection exists whether the operating systems are running as virtual machines under VM/SP HPO or on other processors. The hardware reserves the device when a reserve CCW command is executed.

CP does not issue reserve/release CCWs for itself — neither does CMS. CP issues them only on behalf of guest operating systems (for example, MVS) that issue reserve/release CCWs. Most questions about sharing DASD under VM/SP HPO have to do with the fact that CP does **not** send the guest's reserve/release CCWs to the hardware *in all cases*; or in other words, CP will *modify* the CCWs issued by a guest *in some cases*. The following explanation starts with the easiest case:

Remember that the reservation works for the path to the DASD, whether it comes from different processors or not.

Consider the case in Figure 79 on page 186. Two virtual machines use a **dedicated path** each to the same string of DASD. To do this, you must tell CP that you have each device (and control unit) twice.[16]

---

16 At IPL time, CP will see two paths and two addresses for this string of DASD. Because the second path is not defined to VM/SP HPO as an alternate path, CP will vary offline the higher addressed devices, i.e., 340-347. However, CP will allow you to immediately VARY ONLINE addresses 340-347 and ATTACH those devices to GVM2.

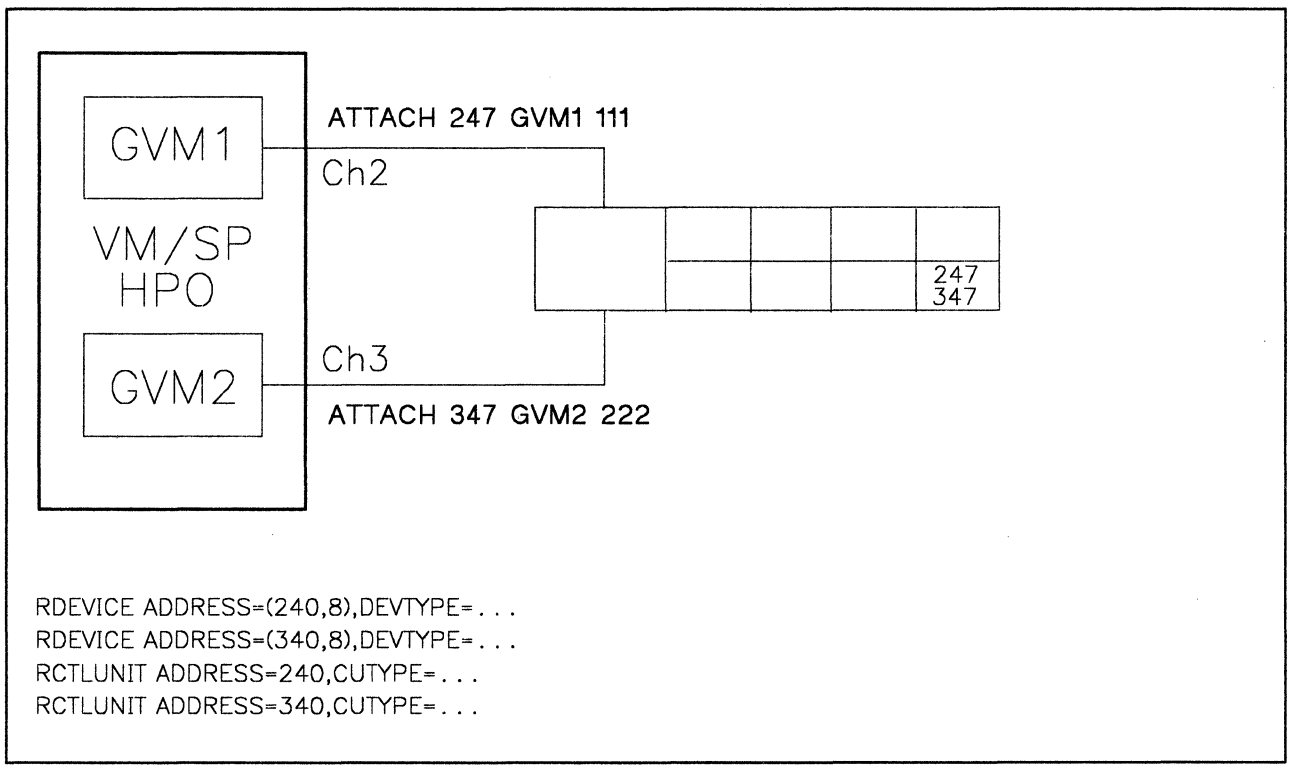


Figure 79. Reserve/Release on Dedicated Paths (Example 1)

Figure 79 yields the first rule for DASD sharing under VM/SP HPO (row 1 in the table).

RULE 1: FOR DEDICATED OR ATTACHED DASD

If the reserve/release hardware IS present and no alternate paths are online, then CP sends the reserve/release CCWs unmodified to the hardware. [17]

If all paths to a DASD device are dedicated, and if the guest operating systems use reserve/release, then there is no VM/SP HPO data integrity problem. Running natively, or using a dedicated path under VM/SP HPO are functionally equivalent modes of operation in this respect.

Consider another case:

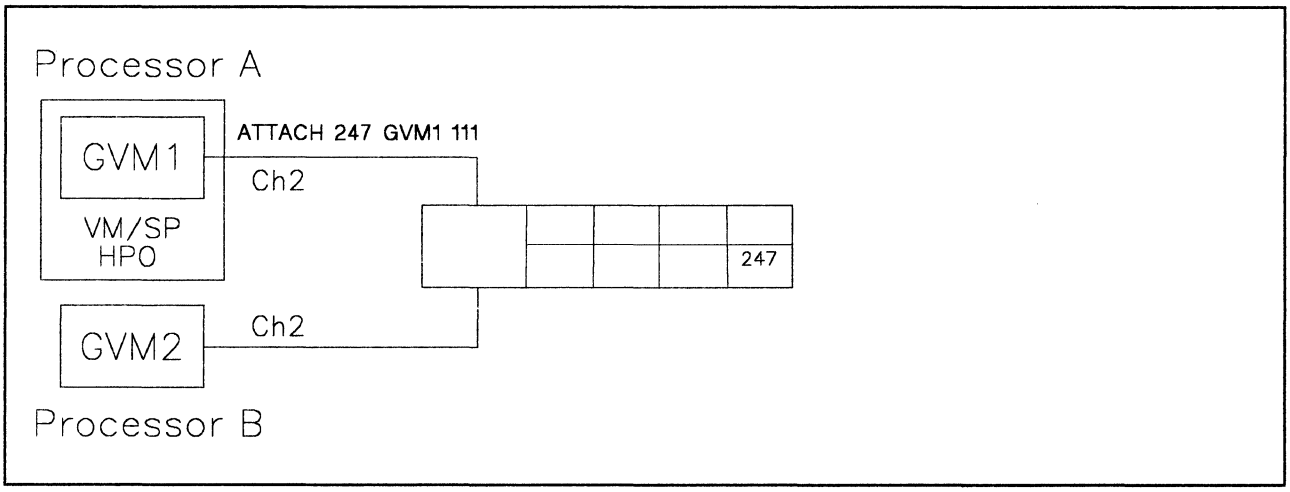


Figure 80. Reserve/Release on Dedicated Paths (Example 2)

In Figure 49 on page 105, only one virtual machine, GVM1, is sharing DASD with an operating system, GVM2, on another processor. The DASD is dedicated to the virtual machine. According to Rule 1, CP sends the reserve/release CCWs unmodified to the hardware. Data integrity is ensured.

---

[17] For more information on reserve/release and alternate paths see "VM/SP HPO Alternate Path Support and Reserve/Release" on page 191.

Virtual Reserve/Release

Virtual reserve/release support allows several operating systems, such as MVS, to run as virtual machines under the same VM/SP HPO operating system and to have data protection when using the same data files on the same minidisk.

By using virtual reserve/release support, one operating system running in a virtual machine can prevent other operating systems running under the same VM/SP HPO system from accessing the reserved minidisk. However, a minidisk protected by virtual reserve/release support may not be protected from access by an operating system running on other processors.

Consider the case of two guest operating systems wanting to share DASD when **only one physical path exists** from your processor to the DASD device. Under VM, in order to share this device it must be defined as a **minidisk** for one of the guests. The other guest will have to issue a CP LINK command to gain access.

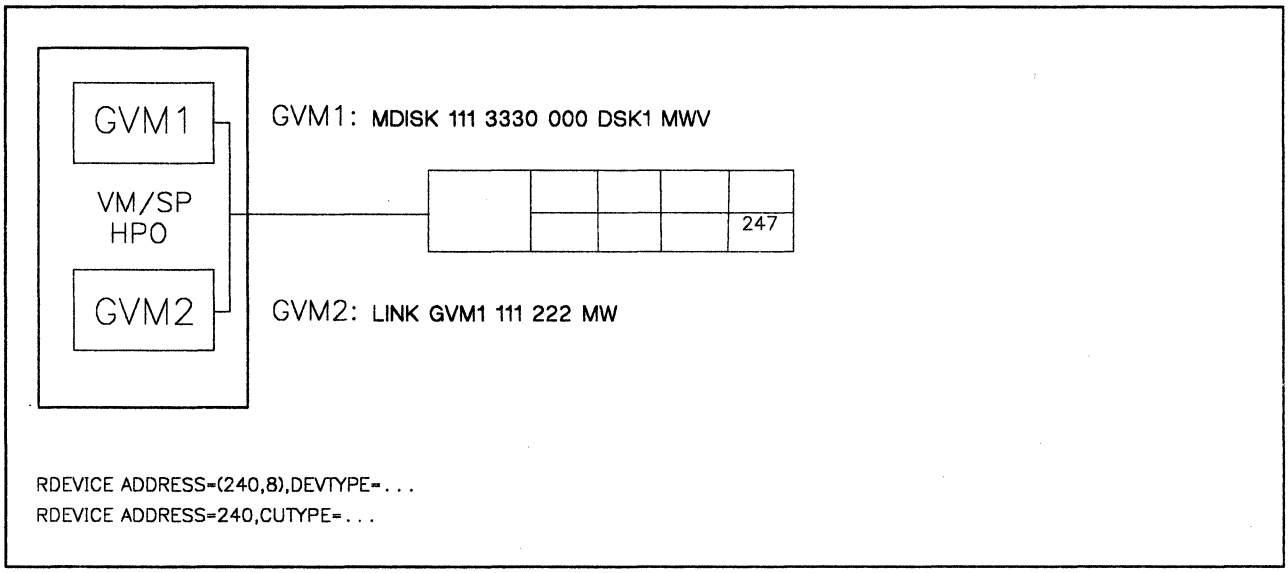


Figure 81. Virtual Reserve/Release

In this example (see Figure 81), the use of the hardware's **real** reserve/release facility leads to an integrity exposure. The reserve/release hardware — if present at all — cannot do its job since the Read and Write requests from GVM1 and GVM2 must travel along the same path.

However, VM/SP HPO has a **software simulation** facility for this, virtual reserve/release. It is valid for minidisks only and is specified by a special access mode, **MWV** (see Figure 81 on page 188).[18]

Virtual reserve/release ensures that the above example works and that the data integrity mechanism of each guest operating system can operate just as if it were not running under VM/SP HPO.

Virtual reserve/release executes in CP only. If a guest issues a reserve CCW to protect the device from being accessed by other operating systems on the same processor complex, CP will flag this minidisk as being reserved for that particular virtual machine. It reserves the access to the minidisk, just as the real reserve/release hardware would reserve access to the real disk.

Since the virtual reserve/release facility executes only in CP, the reserved minidisk can be of any size. It does not need to be a full-pack minidisk. [19]

Virtual reserve/release can work for several independent minidisks on the same volume as long as the volume is not shared with another processor complex.

If you have specified MWV in your MDISK definition, your guest can issue a reserve CCW and CP will reserve the minidisk accordingly. One question remains: which CCW is sent to the hardware?

This brings up the next rule (row 4 in the table):

RULE 2A: FOR A MINIDISK WITH VIRTUAL RESERVE/RELEASE

If the reserve/release hardware IS present and no alternate paths are online, then CP sends the reserve/release CCWs unmodified to the hardware. [20]

Since the hardware handles the CCW, a reserve/release for a minidisk will always result in a reserve/release for the whole DASD volume on which this minidisk is defined if no alternate paths are online.

So, in addition to the virtual reserve/release protection in CP, you are protected against access via other paths by the real reserve/release hardware. These other paths can lead to a different processor, or to the same one (dedicated path).

---

18 Note that "MWV" must be in the MDISK statement. If you put it in the LINK statement it has no effect. You will not get a syntax error if you put it in the LINK statement.

19 However, there are several good reasons why you should use full-pack minidisks. These reasons are discussed later in this section.

20 For more information about reserve/release and alternate path support see "VM/SP HPO Alternate Path Support and Reserve/Release" on page 191.

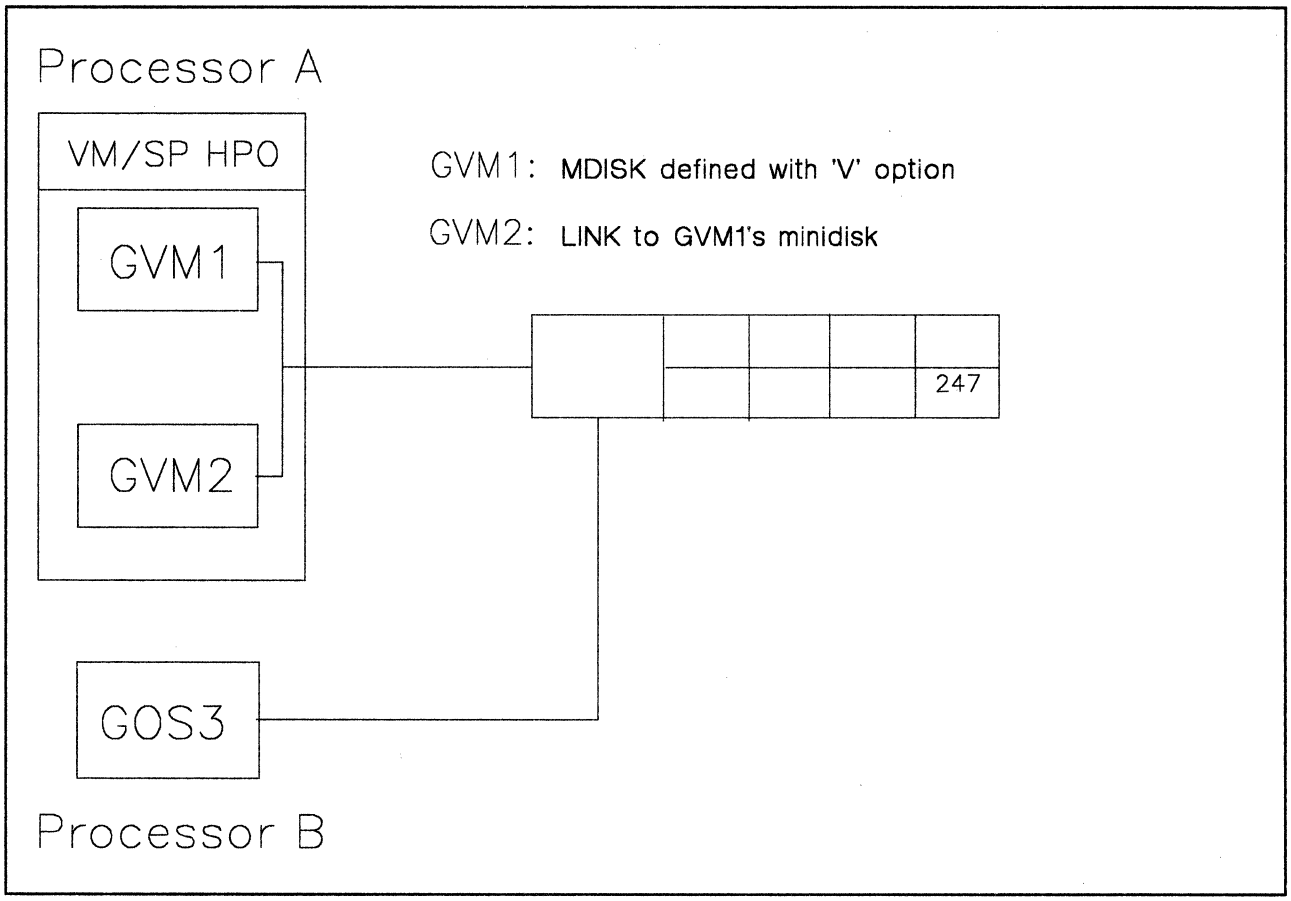


Figure 82. Virtual and Real Reserve/Release

An example for this case is given in Figure 82. Three operating systems have write access to the same disk. GVM1 and GVM2 are under VM/SP HPO using virtual reserve/release. GOS3 is executing natively on a separate processor. If GOS3 runs under the same VM/SP HPO also, consider it using a dedicated path.

The locking structure works properly for all three:

1. GVM1 and GVM2 are protected against each other by CP.

2. GVM1 and GVM2 together are protected against GOS3 and vice versa by the hardware.

Note that the software mechanism works for minidisks. If you do not use full-pack minidisks, the hardware still reserves the complete pack. You may want to use full-pack minidisks only, especially if these disks are to be shared by using the hardware feature (See also "Sharing Minidisks" on page 195).

Now look at the case where the reserve/release *hardware* is not present. CP finds out about the presence or absence of this hardware feature by issuing a release CCW to Count-Key-Data (CKD) devices[21] when either CP is being IPLed or when DASD devices are varied online by the VM/SP HPO system operator.

The following rule applies (row 6 in the table):

RULE 2B: FOR A MINIDISK WITH VIRTUAL RESERVE/RELEASE

If the reserve/release hardware IS NOT present, CP modifies the reserve/release CCWs into sense CCWs before sending them to the hardware.

A sense CCW returns a condition code (CC) which is similar to that of a successful reserve or release CCW. The difference is that the sense CCW doesn't do anything else. If this CCW modification was not done, the guest operating system would receive a COMMAND REJECT and would think of the devices as "not shared", and therefore, it would not issue reserve/release CCWs anymore. But as CP is simulating the hardware reserve/release facility, you want the guest virtual machine to believe that the facility exists.

IBM makes a similar recommendation for MVS/SP. For all shared DASD:

- Use either full-pack minidisks with virtual reserve/release defined, or
- Use dedicated packs.

At IPL time, MVS/SP issues a release CCW to all of its DASD that have been generated as shared.

At IPL time, CP issues a release CCW to all of its Count-Key-Data (CKD) DASD and tapes.

VM/SP HPO Alternate Path Support and Reserve/Release

Alternate path support lets you define alternate paths to a tape or DASD unit on the VM/SP HPO processor. This option supports the two-channel switch and the string switch features.

---

21 CP does not issue a release CCW to Fixed Block Architecture (FBA) devices. Instead, CP checks an extension to the device control block to see if the reserve/release facility is present for these devices.

Define alternate paths in DMKRIO for devices that a virtual operating system is to use. When you do this, VM/SP HPO will map I/O requests from a virtual address associated with the virtual machine to one of the real paths to the device as defined in DMKRIO. Refer to the section, "Alternate Path Support", in the *VM/SP HPO Planning Guide and Reference* for an explanation on the definition of alternate paths.

As a rule, alternate path and reserve/release support are mutually exclusive. There is, however, one exception to this rule. At the minidisk level, VM/SP HPO provides virtual reserve/release support in the form of a software locking mechanism. As long as virtual machines under VM/SP HPO use virtual reserve/release between them and as long as other real processors do not share the volume, alternate paths can be defined for the real device.

The previous sections spoke about path protection through real and virtual reserve/release. You will now see what to modify if an **alternate path** exists for the device you want to protect.

The following rule applies (rows 2, 7, and 8 in the table):

RULE 3:

If the defined alternate path to the device is online, then CP modifies the reserve CCWs into sense CCWs before sending them to the hardware.

The release CCWs are sent unmodified.

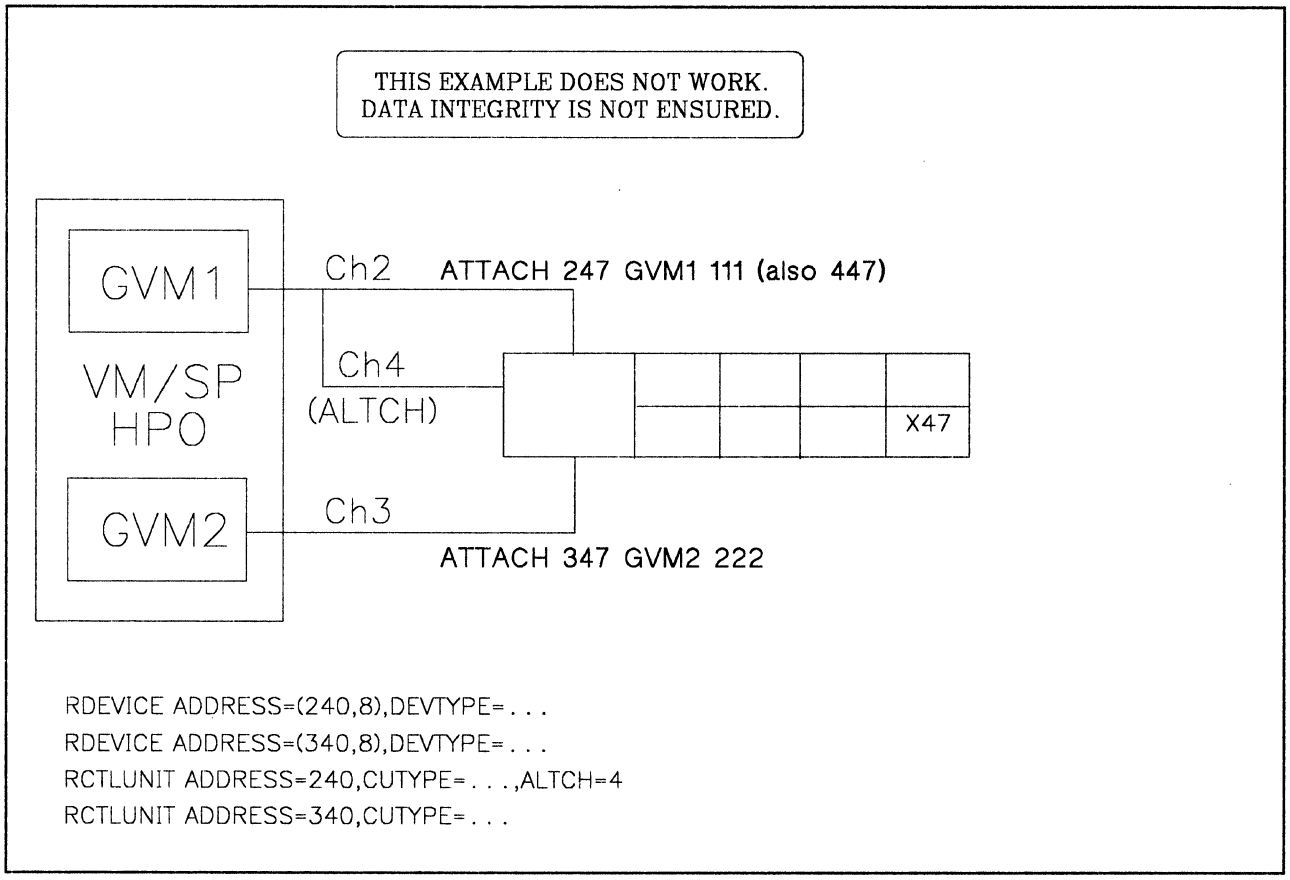


Figure 83. Alternate Path and Dedicated Disk

Look at Figure 83. You saw this one before during the discussion of dedicated paths, but now one of the paths has an alternate channel or control unit specified (ALTCH or ALTCU in DMKRIO), and this alternate path is online. When you attach a device with an alternate path to a virtual machine, CP considers automatically the alternate address as attached too. The guest knows only one address. This means that the guest cannot issue a SIO(F) directly to the alternate path address.

In this example, no reserve CCWs from GVM1 come through to the hardware. As a result, there is no protection from GVM2 accessing the DASD while GVM1 has access to this device in write-mode. This is row 2 in the table.

An additional problem is that CP does not inform the guest operating system of its changing the reserve CCW. So, the guest believes that the disk is shared and protected.

Why does this alternate path restriction exist? Consider that GVM1 issues a reserve for 247. The hardware would execute this reserve. GVM1 has no way of knowing whether his request was actually executed for 247 *or* 447. It is CP that decides which path to use. The same is true for all subsequent I/Os: if they do not come via the reserved path (which path will it be?) they will be blocked from completing. Also, the release may not work for the

same reason. To prevent this possible channel lockout, CP does not send any reserve CCW if an alternate path is online.[22] Release CCWs are still sent, but have no effect.

Is there a circumvention? Yes. In all cases you must suppress the alternate path: be sure that only one path is online. You might want to switch to full-pack minidisks and links and use virtual reserve/release. But in this case you don't use the path via Channel 4 at all.

This means that the alternate path must not be **online**. The alternate path can be defined in order to use the hardware reserve/release support. It can be defined in DMKRIO (ALTCH or ALTCU). As long as the alternate path is not online, reserve CCWs are not modified by CP. However, to be on the safe side, turn the hardware switches off before you IPL VM/SP HPO.

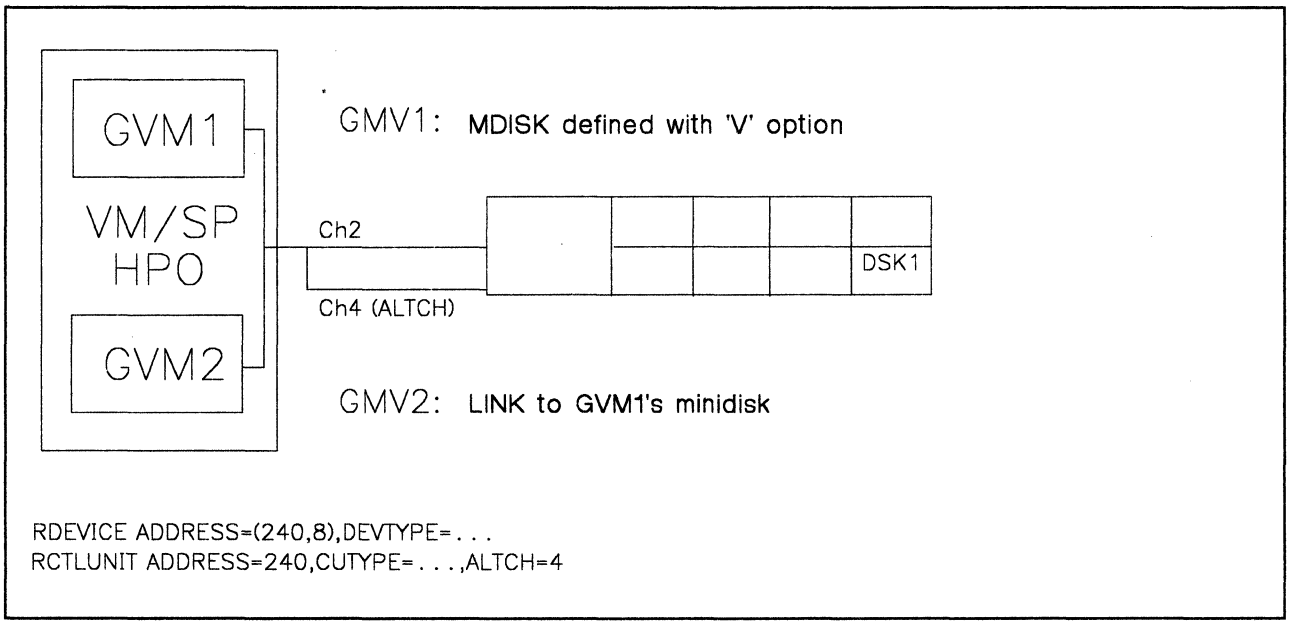


Figure 84. Alternate Path and Minidisk

Consider another example, Figure 84. It deals with a minidisk for which virtual reserve/release is requested. Again, one of the previous figures is modified to include an alternate path. Although the reserve CCWs are modified to sense CCWs before they are sent to the hardware, the protection is maintained for this environment since it is all done in CP. In other words: you don't care about the modification. This is row 8 in the table.[23]

However, a problem arises if the DASD is to be shared with another processor (or dedicated path), as explained in Figure 85 on page 195.

---

22 MVS has a more elegant solution to this problem. During the time a reserve is active, MVS suppresses all alternate paths to the reserved device.

23 Row 7 in the table is not important as you can always specify virtual reserve/release.

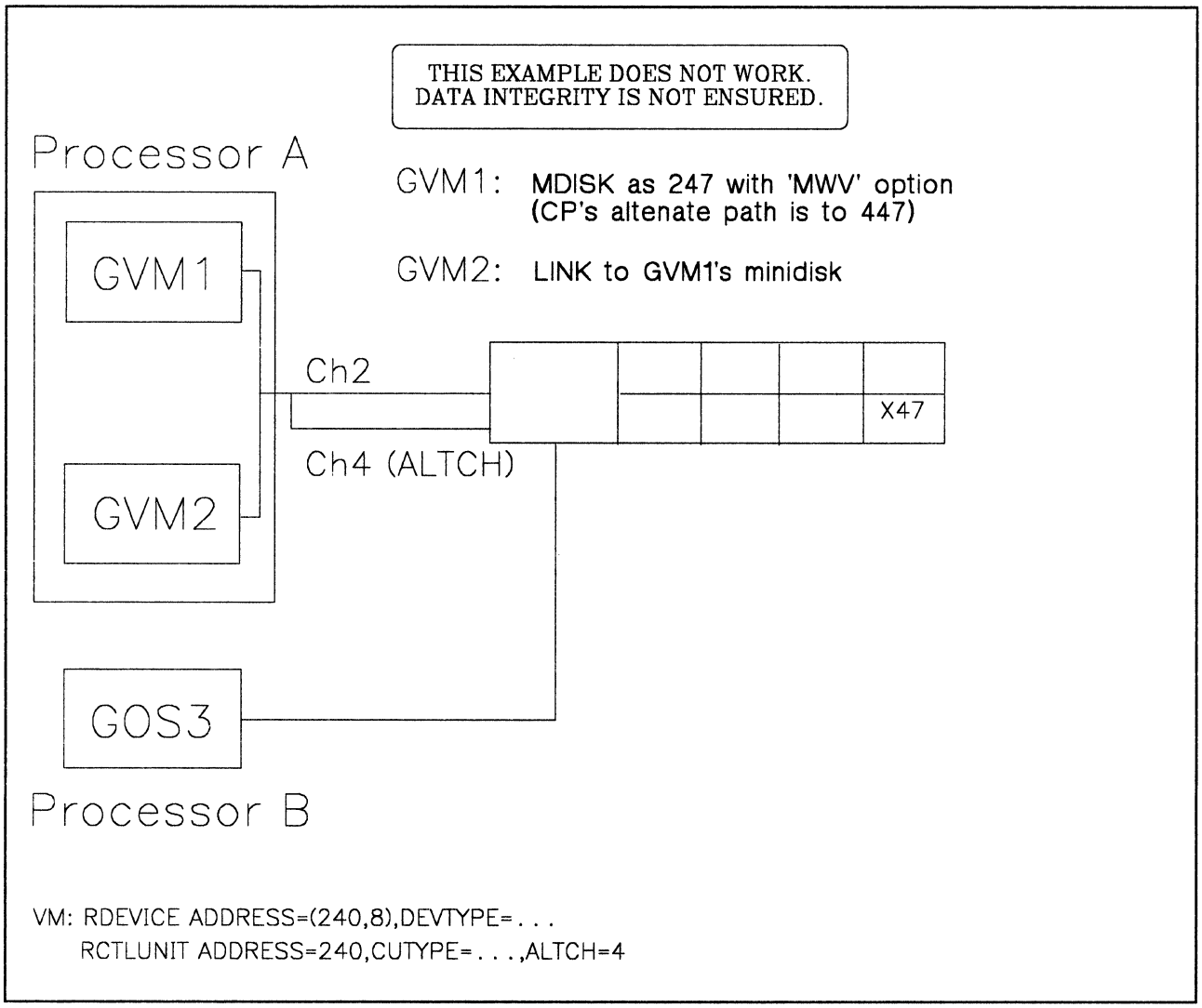


Figure 85. Alternate Path and Minidisk in a Multisystem Environment

In Figure 85, you would have no protection against GOS3 accessing the minidisk while one of the guest operating systems has an access to this minidisk in write-mode. The only solution is to avoid the alternate path, so that CP on Processor-A will not modify the reserve CCW to a sense CCW.

Sharing Minidisks

Assume you are running two real processors and at least one of them is running VM/SP HPO along with two guest operating systems, GVM1 and GVM2. In such an environment you can find several minidisks defined on one real pack. Depending on the sharing of these minidisks and the SCPs running in the guests, you can run into severe problems.

In order to follow the details of this example, refer to Figure 86 on page 196.

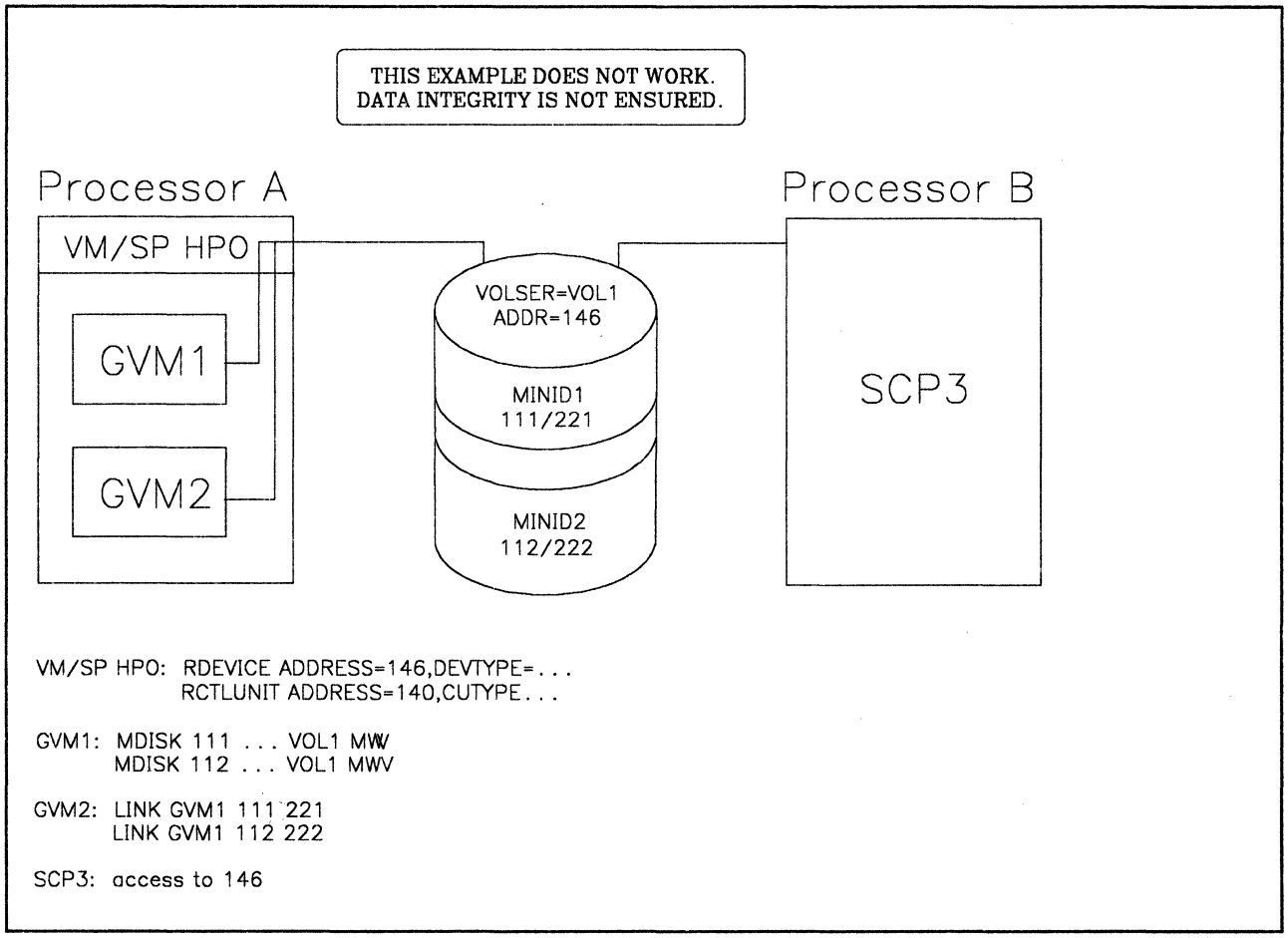


Figure 86. Sharing Minidisks

In this example, a VM/SP HPO system is running on Processor-A with two virtual guests sharing MINI1 and MINI2, which have been defined on the real pack, VOL1, mounted on address 146. For the two guests, the virtual reserve/release facility is used. There exists only one path from Processor-A to VOL1.

Processor-B is running a native SCP which wants to share the data in the minidisk MINI1 at the beginning of the real pack (and obviously can only use this part of the pack!).

When you run **MVS/SP**, the definitions for DASD sharing are part of the I/O generation. MVS systems use ENQ/DEQ which provides protection only within one software system. But the ENQ/DEQ is automatically converted to a reserve/release if the resource to be protected is found on a DASD which has been defined in the I/O-generation as shared.

This leads to the following rule:

RULE 4:

In an MVS/SP environment, you can define ONLY ONE minidisk per physical pack as shared.

Consider Figure 86 on page 196. Assume that all three systems (GVM1 and GVM2 under VM/SP HPO on Processor-A and SCP3 on Processor-B) are running. GVM1 issues an ENQ for MINI1 which is converted into a reserve CCW since all devices have been defined as shared. The hardware reserves the pack VOL1. Any access from SCP3 to VOL1 will result in a device busy condition. Now, if GVM2 issues an ENQ/reserve for MINI2, then CP will pass it to the hardware. This does not cause a problem. But it is possible for GVM2 to issue a DEQ/release for MINI2 while GVM1 still needs the protection on MINI1. Nevertheless, CP will pass the release CCW to the hardware and thus free the pack which will allow access to VOL1 for SCP3. This creates a data integrity exposure.

VM/SP HPO Multiprocessor Considerations for Shared DASD

We have been discussing shared DASD in the context of a uniprocessor (UP) environment. When we discussed cross-system sharing of DASD, it was assumed to be between two loosely coupled UPs. This chapter addresses the tightly-coupled or multiprocessor environment for shared DASD. We will use the generic term MP to refer to all multiprocessor complexes whether they are described as a a true multiprocessor, a dyadic processor, or a dual processor.

For the following discussion, refer to Figure 87 on page 199.

From a hardware standpoint, an MP can be configured either **symmetrically** or **asymmetrically.**

Symmetric Multiprocessor Configurations

A **symmetric configuration** means that for every path from CP0 to a device (for example, X'120') across CS0 (Channel Set 0) there is a **matching path** from the other processor, CP2, at the **same address** (X'120') across its own channel set, CS1 (Channel Set 1), to the **same device.** In general, this symmetry can be achieved for DASD devices through either a shared control unit with a two-channel switch or through a shared DASD head-of-string with a string switch going to two different control units. However, for an MP system, VM/SP HPO **always** assumes that this

symmetry is achieved **only** by the existence of a two-channel switch, and not through a string switch. This important assumption is shown in Figure 87 on page 199.

However, if you desire DASD symmetry through a string switch and two real control units, then specify ALTCU = "primary cuu" on the RDEVICE macro:

```
RDEVICE ADDRESS=(120,8),...ALTCU=120
```

This will cause VM/SP HPO to generate two real control unit blocks for X'120' instead of one control unit block pointed to by two channels. This is a very special case which can be used to reflect the real I/O configuration to VM/SP HPO. When you use this particular specification, CP will modify the reserve CCWs to sense CCWs.

In Figure 87 on page 199 you can see that the DMKRIO entries for X'120' in a symmetric MP environment look exactly like the entries for a UP. However, when DMKRIO is assembled with an MP control file (for example, DMKH34M CNTRL for VM/SP HPO R3.4), an alternate channel is automatically generated for the control unit at X'120'. It is just as if the ALTCH parameter was specified on the RCTLUNIT macro for X'120', except that instead of pointing to an alternate channel on the same processor it points to the matching channel (in our case, Channel 1) on the other processor (CP2). This occurs whether or not that matching path from the other processor exists.

As a result, the following rule exists for MP-generated systems:

RULE 5:

If you generate CP as an MP, then a symmetric alternate channel is AUTOMATICALLY defined for each control unit. Consequently, ALL restrictions and rules for alternate and reserve/release support apply when this symmetric alternate channel path IS ONLINE.

The key qualification in Rule 5 is:

If the automatically generated alternate channel path to a device is online then the reserve CCWs will be changed to sense CCWs. When this condition is met data integrity will not be maintained for cross-system sharing (i.e., DASD sharing between the MP Complex and another processor) of DASD.

Furthermore, in the case where symmetric paths to a DASD device are online, the reserve CCWs will be changed to sense CCWs. This will preclude the use of the hardware reserve/release facility used with DEDICATEd/ATTACHed volumes for shared DASD within a processor complex.

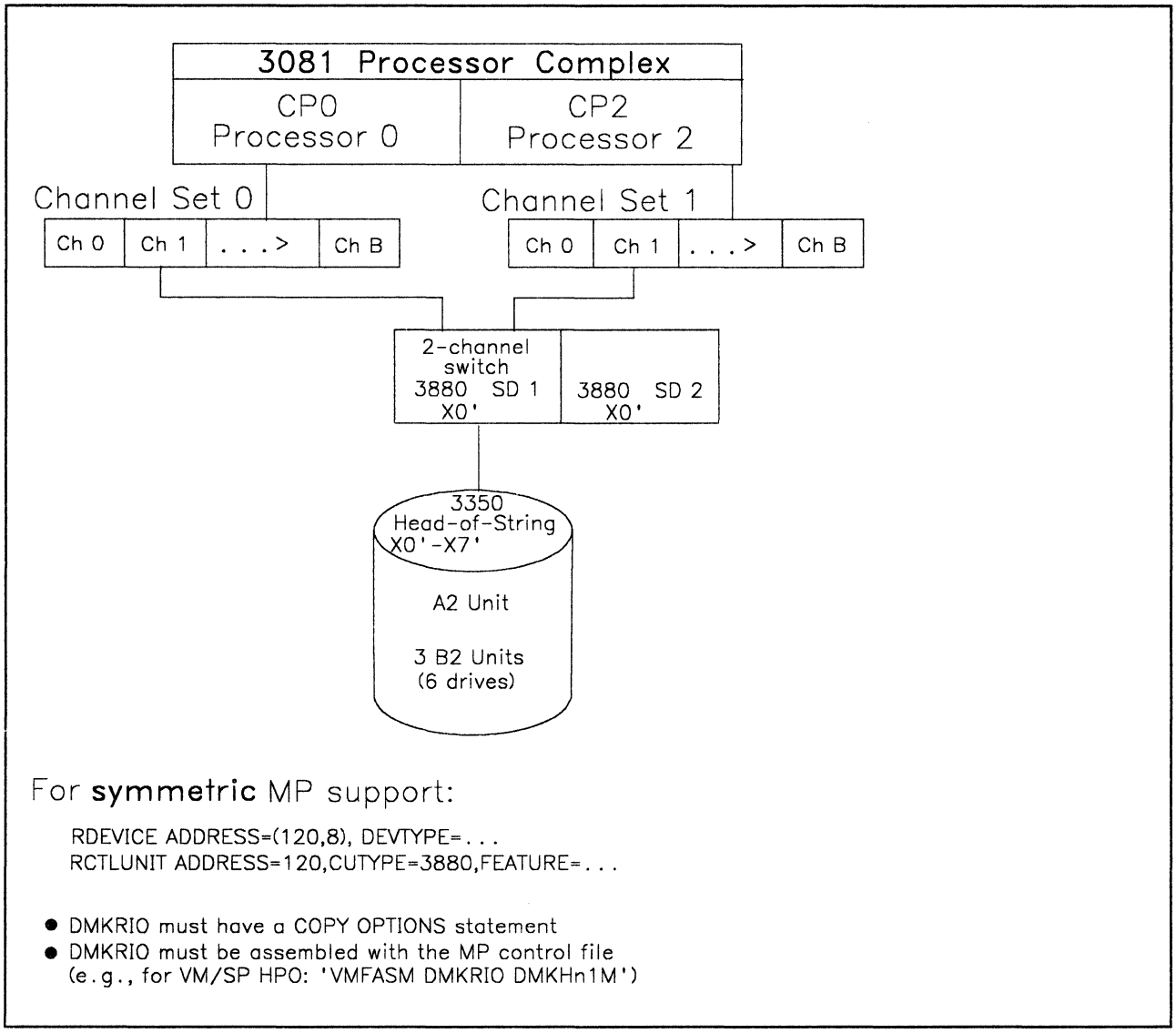


Figure 87. VM/SP HPO Symmetric DASD Configuration Support

Using Figure 87 as an example and assuming that the paths to X'120' from CP0 and CP2 are **online** we can summarize the DASD sharing exposures as follows:

1. Virtual reserve/release will maintain data integrity for a shared minidisk (defined with the **MWV** access mode) **within** an MP processor complex. However, since the reserve CCW will be changed to a sense CCW, data integrity will not be maintained for cross-system sharing, i.e., DASD sharing between the MP complex and another processor.

2. If X'120' is ATTACHed or DEDICATEd, the reserve CCW will be changed to a sense CCW and cross-system data integrity will not be maintained.

If **only one path** to X'120' is online [24] for the MP complex, then the reserve CCW will be sent to the hardware and not changed to a sense. In this case, cross-system data integrity can be maintained.

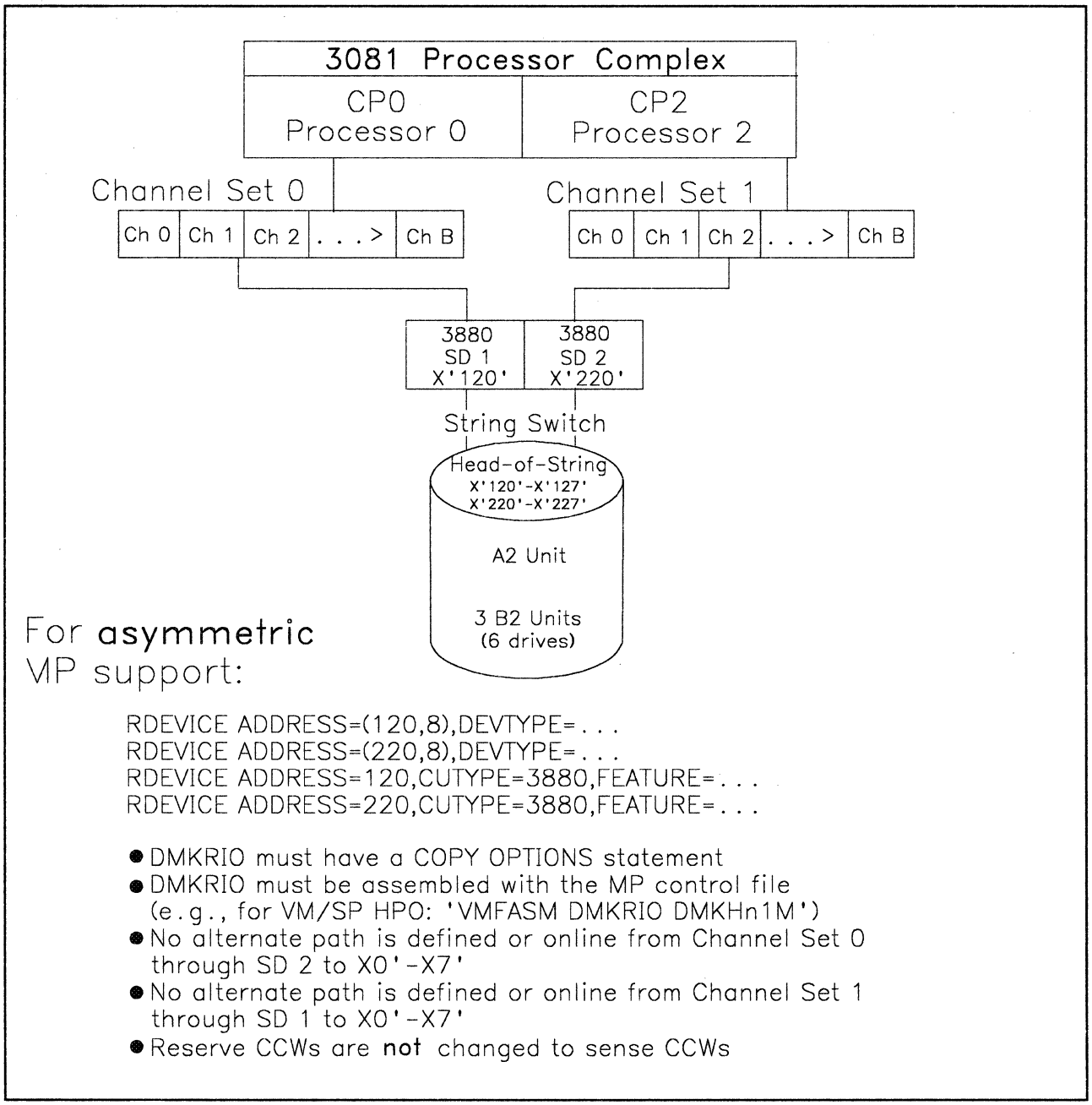


Figure 88. VM/SP HPO Asymmetric DASD Configuration Support

[24] It does not matter if the only online path to X'120' is from CP0 or CP2.

Asymmetric Multiprocessor Configurations

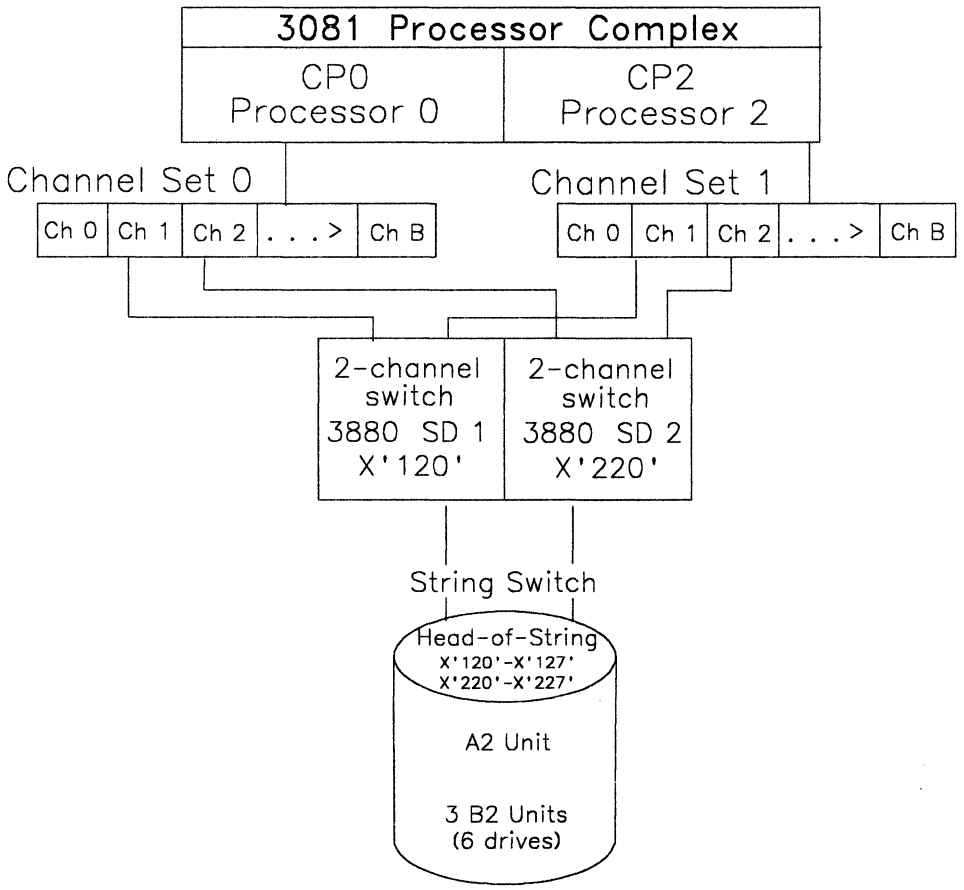
In an asymmetric MP environment, the asymmetry is achieved through the hardware. The VM/SP HPO system will still generate a control block for the alternate channel on the other processor whether or not that symmetric path actually exists.

In Figure 88 on page 200 you can see an asymmetric DASD configuration and the required DMKRIO statements. In this case, there is only one path to each device on the string from each processor. The same string of DASD is accessed by both processors at different addresses: CP0 accesses the string as X'120' to X'127' and CP2 accesses the string as X'220' to X'227'. At IPL time, CP will vary X'220' to X'227' offline for this reason: X'220' to X'227' have duplicate volid's and they are not defined on an alternate path. X'220' through X'227' can be varied back online and used as minidisks or as DEDICATEd/ATTACHed volumes.

The asymmetrically defined configuration in Figure 88 on page 200 will not cause VM/SP HPO to change reserve CCWs to sense CCWs. Therefore, data integrity can be maintained within the MP complex whether virtual reserve/release support or the hardware reserve/release support is used. Also, since the real reserve will not be modified with either virtual reserve/release or DEDICATEd/ATTACHed volumes, cross-system sharing between the MP complex and another system can take place with data integrity.

THIS EXAMPLE DOES NOT WORK. DATA INTEGRITY IS NOT MAINTAINED IF ALTERNATE PATHS ARE ONLINE.

However, data integrity can be maintained within the processor complex with online symmetric paths if virtual Reserve/Release support is used.




```
RDEVICE ADDRESS=(120,8),DEVTYPE=. . .
RDEVICE ADDRESS=(220,8),DEVTYPE=. . .
RDEVICE ADDRESS=120,CUTYPE=3880
RDEVICE ADDRESS=220,CUTYPE=3880
```

Figure 89. VM/SP HPO Symmetric DASD Configuration Support

If all defined paths in Figure 89 are online, then data integrity can only be maintained within the MP complex if virtual reserve/release is used. In no case will data integrity be maintained in a cross-system environment unless all alternate paths are offline.

Part Three: VM under VM

The procedure that follows assumes that you have your main VM system up and running. This procedure does not help you bring up that system. If you are not sure of the basic functions of VM, please take the time to review them.

Chapter 1. Introduction to Running VM under VM

VM provides an easy, convenient way to run guest operating systems. When you run VM (second-level) under VM (first- level) you get the functional equivalent of a real processor, main and auxiliary storage, and I/O devices. Because VM is simulating these functions, the simulated system is referred to as a "virtual" machine. This virtual machine is equivalent to an IBM System/370 computing system. When you run a system **second-level**, the system is running as a virtual machine under the Control Program (CP) of your first-level VM system.

CP manages the resources of the real computing system in such a way that many people can use the VM system at the same time. These virtual machines execute independently of each other and each can use a different operating system, or different releases of the same operating system. The operating systems themselves execute as though they were controlling real devices and storage.

Running VM second-level provides greater flexibility in managing the system. In this environment you can:

- Test new application programs
- Test new releases of VM
- Test new maintenance and modifications
- Train operators and system programmers.

These activities can be independent of any other work that is running in the system. This independence is achieved through the resource management provided by CP.

One of the biggest advantage of running a second-level VM system is the ability to generate a new system without disturbing the normal production activity. System programmers can log on to their own virtual machines and go through the generation steps at their own pace while the daily work is being processed. The System Product Editor can be used to create and update the files that are used during system generation. When the system is tested, it can be placed online, replacing the previous version with minimal disruption to the production activity.

Note: Although you can use a second-level VM system to test new VM releases, service, and modifications, do not use it to create a back-level system.

Performance Considerations When Operating a Second Level VM System

Due to various factors, performance characteristics are difficult to predict when VM is running second-level. Some of these factors are:

- The frequency of real interruptions
- The frequency and type of privileged instructions
- Whether the virtual machine assist or VM/370 extended control-program support is on the machine and enabled
- The frequency of START I/O (SIO) instructions
- The amount of fixed-head paging space
- The location of the paging areas on DASD
- The size of the virtual machine.

The above factors can be broadly classified into two groups:

1. Configuration factors
2. Operating system workload factors.

Configuration Factors Influencing Performance

When running VM second-level, there is an increased need for real storage, DASD space, processor speed, and so on. The overhead incurred from VM's increased need for dispatching, scheduling, and paging is relatively small in comparison to the increase in overhead from simulating privileged instructions.

When VM operates first-level, it runs directly on its own hardware and manages its resources through the use of privileged instructions, such as Start I/O (SIO) and Load Program Status Word (LPSW). When executing in second level, VM dispatches the second-level VM system in problem state, and any privileged instruction issued by the second-level virtual machine causes a real privileged instruction exception interruption. This interruption transfers control to the first-level VM system, which simulates the instruction. The amount of work done by the first-level system in analyzing and handling a second-level virtual machine-initiated interruption depends upon the type and complexity of the interruption.

The following hardware configuration factors also influence the performance of a second-level virtual machine:

- The amount of real storage available
- The speed, capacity, and number of paging devices
- The amount of channel and control unit competition and the arm rivalry affecting each paging device
- Whether virtual machine assist or VM extended control-program support is installed on the hardware and enabled
- Interference between system paging devices and devices for processing a user's I/O requests.

Notes:

1. *Virtual machine assist support has been specifically designed to reduce CPs overhead associated with simulating privileged instructions. For more information on virtual machine assist refer to the Planning Guide and Reference.*
2. *VM/370 extended-control program support (ECPS: VM/370) is a hardware assist function that provides support over and above that provided by virtual machine assist. It improves VM performance by reducing VM's real supervisor state time which is needed to support second level virtual machines.*

Workload Factors Influencing Performance

The following workload factors influence the performance of a second level virtual machine:

- The total number of active virtual machines
- The type of work each virtual machine is doing, especially the amount of I/O processing required.

By measuring and evaluating the effects of these workload factors on a specific configuration, you can anticipate their effect on performance. After measuring the performance of the VM second-level machine, the system programmer can use the VM performance options. These options create a special performance environment for one or more virtual machines. The options allow the system programmer or system operator to redistribute system resources, either to balance them or to favor a particular virtual machine over another.

The following options are available to only one virtual machine at a time:

- Reserved page frames [25]
- Virtual=Real option.

The following options are available to as many virtual machines as desired:

- Favored execution with a specified percentage
- Basic favored execution (without a specified percentage)
- Priority
- Virtual machine assist
- VM/370 extended control-program support (ECPS: VM/370)
- Locked pages
- QDROP OFF.

For basic information about these options, refer to *CP for System Programming*. For details about the use of these options, refer to the *CP Command Reference*.

VM/SP HPO 3.4 and later releases have added additional running options, (i.e., SNAPQTIME, MINWSS, and IBUFF). Please refer to the *CP for System Programming* for information on these options.

---

25 VM/SP HPO 3.4 and later releases allows multiple virtual machines to utilize reserve page frames.

Chapter 2. Defining the First- and Second-Level VM Systems

When preparing your second-level system you can either create a new CP nucleus, or make a copy of the the first-level system's CP nucleus. You will need to create a new CP nucleus if you are testing:

- New devices
- New licensed programs that require shared segments
- New service levels of VM.

If you need to create a new CP nucleus, you must use the VMFLOAD or SPGEN procedure. The VMFLOAD procedure creates a CP nucleus in your reader that can be loaded onto your second-level system's SYSRES device. In this chapter we show the steps involved in bring up a second-level system when a new CP nucleus is generated using VMFLOAD.

To copy the first-level system's CP nucleus to the DASD you'll be using for your second-level system, you can use the DASD Dump Restore (DDR) if:

- The SYSRES volume of the first-level system is the same as it is in the second-level system, and
- The SYSRES packs have the same layout — for example, the same values for SYSNUC, SYSWRM, and so forth.

Make sure that the DMKSYS text deck used in this procedure describes the second-level system's SYSRES.

Preparing the First-Level VM System

LOGON to your VM system as MAINT (or a userid that has access to the DIRECTORY and can issue the CP DIRECT command). Using XEDIT, edit the first-level system's directory file. You will need to create the directory entry that will be running the second-level VM system.

In the directory entry, of the first-level system that will be IPLing the second-level system, the following options should be included:

- ECMODE ON is required to run a second-level system.
- The storage should be defined with a minimum of 8M to improve performance by reducing the paging done by the second-level VM system.
- The channels should be defined in block (BMX) mode rather than in selector mode. This will improve performance for the second-level VM system.
- The line-end character should be changed to "%" (or any special character of your choice) to distinguish between issuing first-level and second-level CP commands.
- The REALTIMER option should be included so that the virtual interval timer can be updated during virtual wait state.
- The console should be defined.

First-Level System Directory

VMUSERS DIRECT is the name of our directory containing the entry for the second-level system. Figure 90 on page 211 shows the first-level system's directory entry for the virtual machine that will be operating the second-level system.

```
VMUSERS DIRECT    A1  F 80  TRUNC=72 SIZE=32 LINE=9 COLUMN

===== *
===== ****************************************************
===== *             SYSTEM-RELATED USERIDS
===== ****************************************************
===== *
===== USER VMTEST password 8M 16M GB 64 %
===== OPTION ECMODE REALTIMER BMX
===== ACCOUNT 1 SYSPROG
=====   CONSOLE 009 3215
=====   SPOOL C 3505
=====   SPOOL D 3525
=====   SPOOL E 3211
===== * Link to Executable CMS Code
=====   LINK MAINT 190 190 RR
===== * Link to System Definition Files and Locally Applied Service
=====   LINK MAINT 295 192 RR
===== * Link to HELP Code
=====   LINK MAINT 19D 19D RR
===== * Link to Program Products (Y-Disk)
=====   LINK MAINT 19E 19E RR
===== * Link to Base CP Code and CP Service
=====   LINK MAINT 194 194 RR
=====   LINK MAINT 294 294 RR
===== * Following Entries Are for Base CMS Code and CMS Service
=====   LINK MAINT 293 293 RR
=====   LINK MAINT 193 193 RR
===== * Following Entries Are the Test VM System.
=====   MDISK 123 3380 010 015 TESTPK MR
=====   MDISK 124 3380 025 015 TESTPK MR
===== * Following Entry Is the User's R/W 'A-disk'.
=====   MDISK 191 3380 040 020 TESTPK MR
=====   SPECIAL 060 3270
```

Figure 90. VM Directory Entry for a Second-Level VM System

Note: At logon, as the directory control statements for the user are processed, CP checks the devices represented by each MDISK, CONSOLE, DEDICATE, LINK, SPECIAL, and SPOOL statement for a possible conflict with the interface to the virtual control unit. This conflict can occur because the virtual control unit cannot support two different subchannel protocols at the same time (shared and nonshared). For each directory control statement that violates the restriction, CP sends an error message and does not create the virtual device. To avoid this problem, refer to the "Configuration Aid" in the Planning Guide and Reference.

The USER Control Statement

```
USER VMTEST password 8M 16M GB 64 %
```

USER VMTEST — the user statement defines the userid as VMTEST.

password — Select a unique nonrestricted password.

8M 16M — The lower limit of storage is set to 8M, with a maximum of 16M.

GB — we have given our first-level system user two user classes (G and B). Class G (general) users control the functions associated with the execution of their virtual machine. User class B (resource) was assigned so that the second-level virtual machine user can issue CP ATTACH and DETACH commands. Please refer to the appropriate *CP Command Reference* for a summary of the CP commands allowed by privilege classes.

You can give your VMTEST userid the user classes that meet your needs, but we recommend that class A not be assigned. Privileged class A users can issue a shutdown command and accidentally shutdown the first-level system.

64 — 64 is the default for the CP priority dispatcher. If the priority setting is not specified the line-end character would default to the system-defined values.

% — The line-end character for the first-level system is set to "%". We'll use the percent sign "%" to communicate with the first-level CP system, and the (default) pound sign "#" to communicate with second-level CP system. This eases communication between first and second-level CP systems.

The OPTION Control Statement

VM provides several optional services to virtual machines. You can specify these options with the OPTION control statement in the VM directory, or with the CP SET command.

```
OPTION ECMODE REALTIMER BMX
```

ECMODE

The ECMODE option is required for all VM operating systems running second-level. When the ECMODE option is specified for a virtual machine, the saved segments of the virtual operating system can be shared. Setting the ECMODE option does not alter the ECMODE bit of the user's PSW. The ECMODE option allows the virtual machine to use the complete set of control registers and the dynamic address translation feature of the System/370.

If the ECMODE option is not specified in the directory, before you IPL the second-level system, you can issue the CP SET command as follows:

```
%cp set ecmode on
```

REALTIMER

The REALTIMER option causes the virtual interval timer to be updated during virtual wait state. With the REALTIMER option in effect, a virtual interval timer reflects virtual processor time and virtual wait time, but not CP time used for services for that virtual machine (such as privileged instructions execution). The more services a virtual machine requires from CP, the greater the difference between the time represented by the interval timer and the actual time used by (and for) the virtual machine. The larger the number of active virtual machines contending for system resources, the greater the difference between virtual machine time and actual elapsed time.

If this option is not specified in the directory entry, you can obtain this timing facility by issuing the CP SET command with the TIMER operand. For example to turn on the timing facility, issue:

```
%cp set timer real
```

To turn off the option, issue:

```
%cp set timer off
```

BMX

The virtual block multiplexer (BMX) option allows the second-level system running in a virtual machine to overlap multiple SIO(F) requests on a specified channel path. When the BMX option is given control, it applies to all channels in the virtual machine except channel 0. This option can be specified regardless of whether block multiplexer channels are attached to the processor. The CP DEFINE command can redefine the channel mode for a virtual machine.

If this option is not specified in the directory entry, you can redefine the channel mode of operation by issuing the CP DEFINE command. For example,

```
%cp define channels bmx
```

The CONSOLE Control Statement

```
CONSOLE 009 3215
```

The CONSOLE statement specifies the console address and device type. The virtual device created by any console statement requires a nonshared virtual control unit. CP does not allow the mix of subchannel protocols, shared and nonshared, on a single virtual control unit. Refer to the "Configuration Aid" in the *Planning Guide and Reference* for the list of protocols used by specific devices.

If the first-level system's configuration is used for the second-level system's operation, the console addresses must match. If the first-level system configuration is not used for the second-level system operation, the addresses must match whatever configuration is specified in the second-level VM system's DMKRIO.

If you specify in the console control statement

```
CONSOLE 010 3270
```

You can alternate between 3215 mode for CP commands (first-level) and 3270 full-screen mode for the second-level system (on its operator's console).

In the console control statement you can specify a secondary userid. For example:

```
CONSOLE 010 3270 OPERATOR
```

When the primary userid is running in disconnected mode, the secondary userid receives all CP messages. Specifying OPERATOR as the secondary userid, gives the operator added flexibility in an environment where several virtual machines are used. The operator can control several disconnected virtual machines (via CP SEND command) from one physical terminal.

The SPOOL Control Statement

```
SPOOL C 3505
SPOOL D 3525
SPOOL E 3211
```

The SPOOL statements specify the unit record addresses. If the first-level system configuration **is used** for the second-level system operation, the unit record addresses for the first and second-level VM system must match. If the first-level system configuration **is not used** for the second-level system's operation, the spool addresses must match whatever configuration is specified in the second-level VM system's DMKRIO.

You can use the CP DEFINE command to add unit record devices. For example, you can add a printer to your second-level virtual machine by issuing:

```
%cp define printer vaddr
```

The printer is added at the address specified by **vaddr**.

The SPECIAL Control Statement

```
SPECIAL 060 3270
```

The SPECIAL statement allows a user to DIAL into the second-level system.

The SPECIAL control statement defines a virtual device type and virtual address. Terminal addresses defined in this way don't have to be available on the system since they are not real addresses.

You can use the CP DEFINE command to create a temporary virtual graphic device for the second-level virtual machine. For example:

```
%cp def graf cuu
```

The **cuu** is the hexadecimal virtual address for the device. After you define the graphic device, you must issue the CP DIAL command in order to use it. The device must be supported by the virtual machine's operating system.

Upon Completion of Directory Changes: If you made additions or changes, you must file the new directory and issue the CMS DIRECT command. The DIRECT command processes the directory file to see if it follows the required format and also writes it to the DIRECTORY cylinder of the first-level system you are creating. To actually change or swap the current active VM directory, you must have write access to the system-owned (system residence or IPL device) volume that contains the current directory up to and including the directory cylinders, or the volume that is to contain the new directory.

Note: Issuing the DIRECT command causes the system to search the directory for logon passwords that match the list of restricted passwords contained in the RPWLIST DATA file. All passwords that match are changed to NOLOG in the directory before the directory is placed online.

ENTER

```
direct vmusers
```

Format/Allocate Session

Before the second-level VM system can use the CP disks for the virtual system residence, paging, and spooling volumes, you must format and allocate space for them. Because a virtual disk is being formatted, the cylinder or block specification should reflect the size of the virtual disk being used.

It is up to you to allocate minidisks on VM in a manner that minimizes arm contention and physical overlap. Information about defining and allocating minidisks is found in the *Planning Guide and Reference.* Information on the Format/Allocate program is found in *CP for System Programming.*

We will be formatting and allocating two 15-cylinder 3380 minidisks (MINRES and MINTMP).

Format the minidisks as follows:

123 as MINRES	124 as MINTMP
PERM 000 - 005	PERM 000 - 000
DRCT 006 - 007	TEMP 001 - 012
PERM 008 - 014	TDSK 013 - 014
PERM 015 - 884 [26]	PERM 015 - 884 [26]

After formatting the volumes, ALLOCATE space on them for:

DRCT A directory on your test VM system.

PERM Nucleus area.

PERM Warm start area.

PERM Error recording area.

PAGE or TEMP Paging space.

TEMP Spooling space.

DUMP or TEMP Dump space.

Note: The Format/Allocate program should be used with care since it destroys existing data (if any). It is strongly recommended that a user's minidisks and temporary minidisks not begin on real cylinder zero of CP-owned volumes, because information critical to CP is stored in that cylinder. When running the Format/Allocate program in a virtual machine, the virtual machine must have write access to the volumes being formatted.

---

[26] We allocated the unused cylinders beyond the extent of the minidisks as permanent space (PERM). This is necessary because the minidisk is smaller than the real device; by allocating your minidisk as permanent space, your second-level system will not try to use the area outside the minidisk. Otherwise, the virtual system attempts to use temporary space beyond the size of the virtual disk, resulting in the real system reflecting either seek checks or command rejects to your second-level system.

The following is an example of our Format/Allocate session. Prompts guide you through the execution. Some prompts and messages might be changed slightly from the actual display in order to make them easier to read here.

Log on to VMTEST and IPL CMS.

ENTER

```
ipl cms
```

SYSTEM RESPONSE

```
VM/SP Release n mm/dd/yy hh:mm:ss
Y (19E) R/O
D (192) R/O
Ready;
```

ENTER

```
spool punch *
```

By spooling the punch to yourself, you send a physical card deck to your virtual card reader.

ENTER

```
punch ipl fmt s (noh
```

This puts the Format/Allocate program into your virtual card reader.

SYSTEM RESPONSE

```
PUN FILE 0092  TO  VMTEST   COPY 001   NOHOLD
Ready;
```

The file number used with the CP ORDER command should be the same number as was received in the previous system response. In our example, we used 92.

ENTER

```
order reader 92
```

SYSTEM RESPONSE

```
0001 FILE  ORDERED
Ready;
```

ENTER

```
ipl 00c clear
```

This causes the stand-alone Format/Allocate program to be loaded. In this example, 00C is the address of the virtual card reader. The actual address used in this command depends on the address of the card reader for this virtual machine.

SYSTEM RESPONSE

```
VM/SP FORMAT/ALLOCATE PROGRAM - VM
ENTER FORMAT OR ALLOCATE:
```

ENTER

```
format
```

SYSTEM RESPONSE

```
FORMAT FUNCTION SELECTED
ENTER DEVICE ADDRESS (CUU):
Enter--->123
```

123 is the virtual disk address on which the new system will be built (MINRES).

```
ENTER DEVICE TYPE:
Enter--->3380

ENTER START CYLINDER (XXX OR XXXX) OR "LABEL":
Enter--->000

ENTER END CYLINDER (XXX OR XXXX):
Enter--->014

ENTER DEVICE LABEL:
Enter--->minres

FORMAT STARTED
FORMAT DONE
000 NO. PAGE RECORDS WITH READ-CHECK ERRORS
ENTER FORMAT OR ALLOCATE:
Enter--->format

FORMAT FUNCTION SELECTED
ENTER DEVICE ADDRESS (CUU):
Enter--->124

ENTER DEVICE TYPE:
Enter--->3380

ENTER START CYLINDER (XXX OR XXXX) OR "LABEL":
Enter--->000

ENTER END CYLINDER (XXX OR XXXX):
Enter--->014

ENTER DEVICE LABEL:
Enter--->mintmp

FORMAT STARTED
FORMAT DONE
000 NO. PAGE RECORDS WITH READ-CHECK ERRORS
ENTER FORMAT OR ALLOCATE:
Enter--->allocate
```

```
ALLOCATE FUNCTION SELECTED
ENTER DEVICE ADDRESS (CUU):
Enter--->123

ENTER DEVICE TYPE:
Enter--->3380

ENTER DEVICE LABEL:
Enter--->minres

ENTER ALLOCATION DATA FOR VOLUME MINRES
TYPE CYL CYL
.... ... ...
Enter--->perm 000 005
Enter--->drct 006 007
Enter--->perm 008 014
Enter--->perm 015 884
Enter--->end

ALLOCATION RESULTS
PERM 000 005
DRCT 006 007
PERM 008 014
PERM 015 884
DEVICE 123 VOLUME MINRES ALLOCATION ENDED

ENTER FORMAT OR ALLOCATE:
Enter--->allocate

ALLOCATE FUNCTION SELECTED
ENTER DEVICE ADDRESS (CUU):
Enter--->124

ENTER DEVICE TYPE:
Enter--->3380

ENTER DEVICE LABEL:
Enter--->mintmp

ENTER ALLOCATION DATA FOR VOLUME MINTMP
TYPE CYL CYL
.... ... ...
Enter--->perm 000 000
Enter--->temp 001 012
Enter--->tdsk 013 014
Enter--->perm 015 884
Enter--->end


ALLOCATION RESULTS
PERM 000 000
TEMP 001 012
TDSK 013 014
PERM 015 884
DEVICE 124 VOLUME MINTMP ALLOCATION ENDED
ENTER FORMAT OR ALLOCATE:


ENTER

%cp ipl cms
```

IPLing CMS gets you out of the Format/Allocate program and back to CMS.

SYSTEM RESPONSE

```
VM/SP Release n mm/dd/yy hh:mm:ss
Y (19E) R/O
D (192) R/O

Ready;
```

Setting Up the System Definition Files

If you are running VM second-level for maintenance purposes or application testing, it is preferable that you use the same system definition files (DMKRIO, DMKSNT, and DMKSYS) used by the first-level system. Using the same files ensures that the testing environment matches the first-level system's configuration. The second-level VM directory will differ from the first-level VM directory in that:

- The second-level system's directory needs to have only a subset of the users defined for the first-level VM system.
- The minidisk addresses and volume labels will not match.

If you are running VM under VM for other reasons than the ones stated above, you can use the same system definition files if both the first and second-level VM system residence volumes have the same DASD type, the same volume identification, and the locations of the nucleus, error, checkpoint, and warm start cylinders are identical. You may not want to use the same system definition files for both first and second level systems if you are going to test new device support or new discontiguous shared segments.

For our example we have created a new nucleus for the second-level VM system. We copied the system-definition files from the first-level system's MAINT D-disk (192) onto VMTEST A-disk (191). These files will later be modified on the VMTEST A-disk to describe the test system's environment. To copy the system definition files, issue the following series of commands:

```
copyfile dmkrio assemble d = = a
copyfile dmksnt assemble d = = a
copyfile dmksys assemble d = = a
copyfile vmusers direct   d = = a
copyfile dmkfcb assemble d = = a
copyfile dmkbox assemble d = = a
```

Updating DMKRIO for the Second-Level System

XEDIT the DMKRIO file and change the RDEVICE, RCTLUNIT, and RIOGEN to match VMTEST's configuration.

Note: When preparing the RDEVICE and RCTLUNIT entries, refer to the "Configuration Aid" in the Planning Guide and Reference.

The intent of the following example is to show the necessary changes to DMKRIO in order to match VMTEST's configuration. This example in not a complete DMKRIO. All changes/additions made to our DMKRIO file have been commented.

```
RIO      TITLE 'DMKRIO 3380'
DMKRIO   CSECT
                          .
                          .
                          .
                          .
RDEVICE ADDRESS=001,DEVTYPE=3262,MODEL=5,CLASS=(T,A)
RDEVICE ADDRESS=004,DEVTYPE=3203,CLASS=A,MODEL=5
RDEVICE ADDRESS=00A,DEVTYPE=3505,CLASS=A
RDEVICE ADDRESS=00B,DEVTYPE=3525,CLASS=A
RDEVICE ADDRESS=00C,DEVTYPE=3505,CLASS=B        <--- These three entries match
RDEVICE ADDRESS=00D,DEVTYPE=3525,CLASS=(A,C,B) <--- the spool statements in
RDEVICE ADDRESS=00E,DEVTYPE=3211,CLASS=(A,D,F) <--- the first-level directory.
RDEVICE ADDRESS=00F,DEVTYPE=4245,CLASS=A
EJECT
                          .
                          .
                          .
                          .
RDEVICE ADDRESS=009,DEVTYPE=3215           <--  console address for line mode.
RDEVICE ADDRESS=010,DEVTYPE=3278,MODEL=2A <--  console address for full-screen
RDEVICE ADDRESS=015,DEVTYPE=3278,MODEL=2A        mode.
RDEVICE ADDRESS=016,DEVTYPE=3278,MODEL=2A
EJECT
                          .
                          .
                          .
                          .
RDEVICE ADDRESS=(060,32),DEVTYPE=3277      <----  This line matches the SPECIAL
RDEVICE ADDRESS=080,DEVTYPE=UNSP,CLASS=GRAF       statement. It allows other
RDEVICE ADDRESS=(090,3),DEVTYPE=3278,MODEL=3      full-screen users of the
RDEVICE ADDRESS=(093,3),DEVTYPE=3278,MODEL=4      second-level system.
RDEVICE ADDRESS=(0B0,2),DEVTYPE=2701,ADAPTER=BSCA
EJECT
                          .
                          .
                          .
                          .
RDEVICE ADDRESS=(120,16),DEVTYPE=3380   <-----  This matches the system you
RDEVICE ADDRESS=(140,2),DEVTYPE=3350            want to IPL.
RDEVICE ADDRESS=(150,8),DEVTYPE=3330,MODEL=1
RDEVICE ADDRESS=(160,8),DEVTYPE=3330,MODEL=1
RDEVICE ADDRESS=(170,8),DEVTYPE=3330,MODEL=11
RDEVICE ADDRESS=(190,16),DEVTYPE=3380   <-----  This is the first-level
EJECT                                           system's minidisk that
                                                can be accessed by
                                                the second-level system.
```

```
                          .
                          .
                          .
                          .
RCTLUNIT ADDRESS=000,CUTYPE=3262
RCTLUNIT ADDRESS=008,CUTYPE=2821  <---- control unit for unit record devices.
RCTLUNIT ADDRESS=010,CUTYPE=3274  <---- control unit for RDEVICE 010.
RCTLUNIT ADDRESS=018,CUTYPE=2501  <---- control unit for RDEVICE 01F.

                          .
                          .
                          .
                          .
RCTLUNIT ADDRESS=050,CUTYPE=3274,FEATURE=16-DEVICE
RCTLUNIT ADDRESS=060,CUTYPE=3272,FEATURE=32-DEVICE <-- control unit for
RCTLUNIT ADDRESS=080,CUTYPE=3274,FEATURE=32-DEVICE     RDEVICE 060.

                          .
                          .
                          .
                          .
RCTLUNIT ADDRESS=120,CUTYPE=3880,FEATURE=16-DEVICE <-- control unit for
RCTLUNIT ADDRESS=140,CUTYPE=3830,FEATURE=32-DEVICE     RDEVICE 120.
RCTLUNIT ADDRESS=160,CUTYPE=3830,FEATURE=16-DEVICE
RCTLUNIT ADDRESS=170,CUTYPE=3830,FEATURE=16-DEVICE
RCTLUNIT ADDRESS=190,CUTYPE=3880,FEATURE=16-DEVICE <-- control unit for
                          .                            RDEVICE 190.
                          .
                          .
                          .
                          .       --------------------  These are the alternate
                          .       |                     system consoles.
                                  V
RIOGEN CONS=009,ALTCONS=(010,015,016)
END             |
                |
                -----  009 is the primary VM system console.
```

Updating DMKSNT for the Second-Level System

The DMKSNT file must be changed to reflect the following:

- MINRES cylinder 0 page 11, through cylinder 1 page 47 is used for Saved Systems.
- The CMS entry contains:
  - SYSNAME = CMS2 (The name of the second-level named, saved CMS segment.)
  - VSYSADR = 190
  - SYSCYL = 000 the "real" starting cylinder of CMS SYSRES at second level

- VSYSRES=MNT190 the first-level CMS 190 volume label

- 190 CMS SYSRES is a minidisk at first-level, but a "real" drive 190 at second-level.

```
*
SNT      TITLE 'DMKSNT       VM REL n       3380 SAMPLE'
         SPACE
*
                              .
                              .
                              .
                              .
*********************************************************************
*
*        THE FOLLOWING ENTRIES ARE BASED ON THE INFORMATION PROVIDED
*        IN THE PLANNING GUIDE AND REFERENCE.
*
*********************************************************************
*
         SPACE
DMKSNT   CSECT
         SPACE
*
*              .
*     HEX LOAD ADDRESS FOR SEGMENT  239 = EF0000
*     THE SPACE FOR CMS IS ALLOCATED ON MINRES, AS FOLLOWS:
*    ( THE ALLOCATIONS ARE BASED ON 150 PAGES/3380 CYLINDER )
*          CYL 0, PAGE 11 TO CYL 2, PAGE 13 (303 PAGES)
*         302 PAGES FOR CMS, 1 FOR CP INFORMATION.
*
*********************************************************************
CMS2     NAMESYS    SYSNAME=CMS2,
                    SYSVOL=MINRES,
                    SYSSTRT=(000,11),
                    SYSPGNM=(0-8,14-34,3824-4095)
                    SYSPGCT=302,
                    SYSHRSG=(239-255)
                    SYSSIZE=256K,
                    VSYSADR=190,
                    SYSCYL=000,
                    PARMRGS=(0-15)
                    VSYSRES=MNT190
         EJECT
                              .
                              .
                              .
                              .
         END
```

Updating DMKSYS for the Second-Level System

We have updated our CP System Control File (DMKSYS) and have highlighted the changed entries. For specific information on each macro refer to the *Planning Guide and Reference*.

```
SYS     TITLE   'DMKSYS   FOR 3380      VM'
        PRINT   NOGEN
DMKSYS  CSECT
        SYSOWN  MINRES,MINTMP <-- CP owned volumes for 2nd-level SYSRES.
        SYSRES  SYSVOL=MINRES, <-- Minidisk label for 2nd-level system.
                SYSRES=123,   <--- Minidisk address for 2nd-level system.
                SYSTYPE=3380, <--- Minidisk device type for 2nd-level system.
                SYSCLR=NO,
                SYSNUC=012,  <---  Minidisk start cylinder for 2nd-level
                                   system.
                SYSWRM=(002,1), <- Warm start cylinder for 2nd-level system.
                SYSERR=(003,2), <- Error recording cylinders for 2nd-level
                                   system.
                SYSCKP=(005,1) <-- Checkpoint cylinder for 2nd-level system.
        SYSMON
        SYSJRL
        SYSCOR  RMSIZE=16M,
                AP=NO,
                MP=NO
        SYSOPR  SYSOPER=OPERATOR,
                SYSDUMP=OPERATOR <-- Use OPERATOR userid for second-level
                                     system
        SYSACNT USERID=DISKACN2, <-- Use DSKACN2 userid for second-level
                                     system
                OUTPUT=READER,
                CLASS=C,
                LIMIT=100
      . SYSTIME ZONE=4,
                LOC=WEST,
                ID=EDT
        SYSFORM
        SYSPCLAS
        SYSID   DEFAULT=VM2NDL    <------  We have changed the identifier for
        SYSORD                             the second-level system.
        SYSMIH
        SYSFCN
        SYSLOCS
        END
```

Updating DMKFCB for the Second-Level System

Changes to this file may be necessary to ensure consistent printer interface between first and second-level VM. For our example, no changes were necessary. If you need to make changes to DMKFCB, refer to the DMKFCB module prologue for specific information and *CP for System Programming*.

Updating DMKBOX for the Second-Level System (Optional)

Changes to the DMKBOX file are optional, but you may decide to change it to distinguish it from the first-level system's. We used XEDIT and changed the logo for our second-level system to the following:

Note: Any maintenance done to this system may cause DMKBOX to be changed.

```
BOX      TITLE 'DMKBOX     (CP)     VM    * VIRTUAL MACHINE SYSTEM
                           .
                           .
                           .
                           .
                           .
NBOXLIN1 DC    CL46'    WELCOME TO THE WORLD OF SECOND LEVEL      '
NBOXLIN2 DC    CL46' THIS IS A 2ND LEVEL SYSTEM RUNNING UNDER VM  '
         DC    CL46'                                              '
         DC    CL46'              22222222222222                  '
         DC    CL46'            2222222222222222                  '
         DC    CL46'          222             222                 '
         DC    CL46'          222             222                 '
         DC    CL46'                          222                 '
         DC    CL46' VVV          VVV         222 MMMM       MMMM  '
         DC    CL46'  VVV        VVV          222  MM MM     MM MM  '
         DC    CL46'   VVV      VVV           222   MM  MN   MM  MM  '
         DC    CL46'    VVV   VVV            222    MM   MM MM   MM  '
         DC    CL46'     VVV VVV           222     MM    MMM    MM  '
         DC    CL46'      VVVVV        222        MM     M     MM  '
         DC    CL46'       VVV       222         MM           MM  '
         DC    CL46'                 222                        '
         DC    CL46'               2222222222222222             '
         DC    CL46'             22222222222222222              '
NBOXWDTH EQU   NBOXLIN2-NBOXLIN1                                 @V6GPBRE
NBOXLINS EQU   (*-NBOXLIN1)/NBOXWDTH                             @V6GPBRE
         EJECT
*****************************************************************
                           .
                           .
                           .
                           .
                           .
         END
```

Preparing the Second-Level VM System

When creating the directory entry for the second-level system, you must consider both the general and unique requirements for the test. For general details about specifying directory entries, refer to the *Planning Guide and Reference.*

Note: VM does not check for overlapping extents in the MDISK statement. Therefore, you must ensure that minidisk extents defined in the VM directory do not overlap each other or in the case of 3330, 3340, and 3350 disks do not overlap the 'alternate track' cylinders.

IF OVERLAP CONDITIONS EXIST, FILE DATA DAMAGE IS INEVITABLE. *You can use the DISKMAP EXEC to check for overlaps and gaps between minidisks. DISKMAP EXEC is described in the Installation Guide.*

Second Level System Directory

These are the entries we have made, but you can add or delete entries to suit your own testing needs.

The first non-comment line in the directory has to be the address (123) and the volume label (MINRES) of where the directory will be written.

```
*
DIRECTORY 123 3380 MINRES
*
*************************************************************
                          .
                          .
                          .
                          .
```

As shown below, MAINT2 has minidisks 123 and 124 with 15 cylinders each. In the first-level directory entry from VMTEST, address 190 was assigned as a minidisk; since the DMKRIO for the second-level system also has address 190 defined, the second-level system will use address 190 as a full DASD. MNT190 is the first-level volume label to CMS and the "real" volume label of the second-level system.

We have also created a one cylinder minidisk on the MINRES pack at address 191. Label TST191 corresponds to the 191 minidisk of the VMTEST user at first-level. The second-level system will get the first-level 191 as MAINT2 291.

```
*
*****************************************************************
*             SYSTEM RELATED USERIDS
*****************************************************************
*
USER MAINT2 password 4M 16M ABCDEFG 64
 ACCOUNT 1 SYSPROG
 IPL 190
 CONSOLE 01F 3215
 SPOOL 00C 2540 READER *
 SPOOL 00D 2540 PUNCH  A
 SPOOL 00E 1403 A
 MDISK 123 3380 000 015 MINRES MW
 MDISK 124 3380 000 015 MINTMP MW
 MDISK 191 3380 009 001 MINRES MW
 MDISK 291 3380 000 003 TST191 MW
 MDISK 190 3380 000 045 MNT190 MW  <---- The sizes specified in the following
 MDISK 19E 3380 000 884 MNT19E MW  <---- three minidisk statements depend on
 MDISK 19D 3380 000 055 MNT19D MW  <---- sizes specified in the first-level
         |                  |             system's directory entry; therefore,
         |                  |             they will vary for your system.
         |                  |
         |                  |
         |                 ------  This column represents the second-
         |                         level volume labels.
         |
         --  This column represents the second-level
             virtual machine's virtual addresses.
```

Our OPERATOR directory entry has links to MAINT2's minidisks with a one cylinder minidisk on MINRES.

```
USER OPERATOR password  3M 16M ABCDEG
 ACCOUNT 2 OPERATOR
 IPL 190
 CONSOLE 009 3215 T MAINT
 SPOOL 00C 2540 READER *
 SPOOL 00D 2540 PUNCH  A
 SPOOL 00E 1403 A
 LINK MAINT2 190 190 RR
 LINK MAINT2 19D 19D RR
 LINK MAINT2 19E 19E RR
 MDISK 191 3380 008 001 MINRES MR
```

The CMS1 userid also has links to MAINT2's minidisk with one cylinder of its own on MINRES. The CMS1 user will IPL CMS2 saved segments when CMS2 has been saved.

```
USER CMS1 password 1M 3M G
 ACCOUNT 101 USER01
 IPL CMS2
 CONSOLE 009 3215
 SPOOL 00C 2540 READER *
 SPOOL 00D 2540 PUNCH  A
 SPOOL 00E 1403 A
 LINK MAINT2 190 190 RR
 LINK MAINT2 19D 19D RR
 LINK MAINT2 19E 19E RR
 MDISK 191 3380 010 001 MINRES MR
```

We have added one more userid named DISKACN2 with links to MAINT2's minidisk with a one cylinder minidisk on MINRES. The DISKACN2 will IPL CMS2 saved segments when CMS2 has been saved. (See "Saving Second-Level CMS" on page 238.)

```
USER DISKACN2 password 1M 3M BG
 ACCOUNT DISKACNT DISKACNT
 IPL CMS2
 CONSOLE 009 3215
 SPOOL 00C 2540 READER *
 SPOOL 00D 2540 PUNCH  A
 SPOOL 00E 1403 A
 LINK MAINT2 190 190 RR
 LINK MAINT2 19D 19D RR
 LINK MAINT2 19E 19E RR
 MDISK 191 3380 011 001 MINRES MR
```

File the new directory. You may want to use DISKMAP EXEC to check for overlaps and gaps between minidisks. Then issue the CMS DIRECT command.

Note: When you issue the DIRECT command, all restricted passwords in the directory file are changed to NOLOG.

ENTER

```
direct vmusers
```

As the directory source file is processed, VM checks for a protocol conflict in those statements that describe virtual devices. If a conflict is detected, CP sends an error message. Only the effects of the T-disk option of the MDISK statement and the CONSOLE, SPECIAL, and SPOOL statements are considered. The effects of the DEDICATE, LINK, and MDISK statements depend on the real device configuration at LOGON time.

Building the Second-Level System's CP Nucleus

With the minidisk defined in the directory of the second-level system, the nucleus for the second-level system is ready to be built. Before we can use the VMFLOAD command, we must set up the proper disk search order.[27]

To get system definition files:

ENTER

```
access 191 a
```

SYSTEM RESPONSE

```
A (191) R/O
Ready;
```

To get the VM CP Service Program:

ENTER

```
access 294 b/a
```

SYSTEM RESPONSE

```
B (294) R/O
Ready;
```

To get base VM CP code:

ENTER

```
access 194 c/a
```

SYSTEM RESPONSE

```
C (194) R/O
Ready;
```

Query your disks to make sure everything is in order.

ENTER

```
query disk
```

---

[27] HPO users should refer to the *HPO Installation Guide* to establish the proper disk search order.

SYSTEM RESPONSE

```
LABEL   CUU M  STAT CYL TYPE BLKSIZE    FILES   BLKS USED-(%) BLKS LEFT  BLK TOTAL
TST191  191 A    R/W   5 3380  1024        9        296-13        2029       2325
MNT294  294 B/A  R/O  20 3380  1024      487       6273-67        3027       9300
MNT194  194 C/A  R/O  21 3380  1024      577       5400-55        4365       9765
MNT190  190 S    R/O  35 3380  1024      216      11489-71        4786      16275
MNT19E  19E Y/S  R/O 100 3380  1024      216      11489-71        4786      16275
Ready;
```

We now must assemble the current level of the DMKBOX (if changed), DMKRIO, DMKSNT, and DMKSYS files with the VMFASM command. The VMFASM procedure updates the specified assembler source files according to the entries in the control file and assembles the updated source.[28] When the VMFASM procedure completes, you will have a new text deck that can be used to recreate the CP nucleus.

CHANGES TO DMKRIO:

ENTER

```
vmfasm dmkrio dmksp
```

If no errors were found, the system responds:

```
NO UPDATE FILES FOUND
ASMBLING DMKRIO

ASSEMBLER (XF) DONE
NO STATEMENTS FLAGGED IN THIS ASSEMBLY
DMKRIO TEXT A1 CREATED
Ready;
```

If errors were flagged, XEDIT the file, make the corrections, and reissue the VMFASM command.

---

[28] If you are generating an HPO system, refer to the *VM/SP HPO Installation Guide* to set up the proper control files for the release of HPO you will be using.

CHANGES TO DMKSNT:

ENTER

```
vmfasm dmksnt dmksp
```

If no errors were found, the system responds:

```
NO UPDATE FILES FOUND
ASMBLING DMKSNT

ASSEMBLER (XF) DONE
NO STATEMENTS FLAGGED IN THIS ASSEMBLY
DMKSNT TEXT A1 CREATED
Ready;
```

If errors were flagged, XEDIT the file, make the corrections, and reissue the VMFASM command.

CHANGES TO DMKSYS:

ENTER

```
vmfasm dmksys dmksp
```

If no errors were found, the system responds:

```
NO UPDATE FILES FOUND
ASMBLING DMKSYS

ASSEMBLER (XF) DONE
NO STATEMENTS FLAGGED IN THIS ASSEMBLY
DMKSYS TEXT A1 CREATED
Ready;
```

If errors were flagged, XEDIT the file, make the corrections, and reissue the VMFASM command.

CHANGES TO DMKFCB:

ENTER

```
vmfasm dmkfcb dmksp
```

If no errors were found, the system responds:

```
NO UPDATE FILES FOUND
ASMBLING DMKFCB

ASSEMBLER (XF) DONE
NO STATEMENTS FLAGGED IN THIS ASSEMBLY
DMKFCB TEXT A1 CREATED
Ready;
```

If errors were flagged, XEDIT the file, make the corrections, and reissue the VMFASM command.

CHANGES TO DMKBOX:

ENTER

```
vmfasm dmkbox dmksp
```

If no errors were found, the system responds:

```
NO UPDATE FILES FOUND
ASMBLING DMKBOX

ASSEMBLER (XF) DONE
NO STATEMENTS FLAGGED IN THIS ASSEMBLY
DMKBOX TEXT A1 CREATED
Ready;
```

If errors were flagged, XEDIT the file, make the corrections, and reissue the VMFASM command.

We can now create the IPLable CP load deck via the VMFLOAD command. Spool the virtual punch to your card reader.

ENTER

```
spool punch *
```

The above command sends the load deck to the virtual reader.

```
spool printer *
```

The above command sends the load map to the virtual reader.

Now, you can issue the VMFLOAD command.

ENTER

```
vmfload cpload dmksp[29]
```

When the VMFLOAD procedure is complete, the system responds with:

```
SYSTEM LOAD DECK COMPLETE
PUN FILE 0301  TO VMTEST   COPY 001   NOHOLD

Ready;
```

When the message SYSTEM LOAD DECK COMPLETE appears, the load deck is in your virtual reader.

ENTER

```
order reader 301
```

SYSTEM RESPONSE

```
0001 FILE ORDERED
```

---

[29] VM/SP HPO users should refer to the *VM/SP HPO Installation Guide* for the correct load list and control file.

ENTER

```
spool reader hold
```

The next step is to write the nucleus on the system residence volume by IPLing the load deck.

ENTER

```
ipl 00c clear
```

When the next message appears, the nucleus has been written on the system residence volume.

SYSTEM RESPONSE

```
NUCLEUS LOADED ON MINRES --- STARTING CYL/BLK=012    , LAST CYL/BLK USED=013
CP ENTERED; DISABLED WAIT PSW '00020000 00000012'
```

The PSW of X'00020000 00000012' informs you that everything worked. Once the IPL is complete, close your card reader and printer.

ENTER

```
close 00c
```

The above command closes your card reader.

```
close 00e
```

The above command closes your printer.

When you close your printer, the system puts the load map into your reader. The load map contains the real address locations of all the modules in the CP nucleus; this information is used to debug any problems occurring with CP.

```
PRT FILE 0302  TO  VMTEST   COPY 001   NOHOLD
```

ENTER

```
ipl cms
```

SYSTEM RESPONSE

```
VM/SP Release n mm/dd/yy  hh:mm:ss
Y (19E) R/O
D (192) R/O
Ready;
```

Query your reader to confirm that both the load deck and the load map are in your reader.

ENTER

```
query reader all *
```

SYSTEM RESPONSE

```
ORIGINID FILE CLASS RECORDS  CPY HOLD DATE  TIME     NAME     TYPE     DIST
VMTEST   0301 A PUN 00026421 001 NONE mm/dd 09:45:22 LDT      DMKSAVNC VMTEST
VMTEST   0302 A PRT 00009939 001 NONE mm/dd 09:48:17                   VMTEST
Ready;
```

Spoolid 301 contains the load deck and 302 contains the load map. When you issue the READCARD CPNUC MAP command, it reads the first file from your reader; therefore, order your reader to make the load map the first file.

ENTER

```
order reader 302
```

SYSTEM RESPONSE

```
0001 FILE  ORDERED
Ready;
```

Query your reader to confirm the reordering of the files.

ENTER

```
query reader all *
```

SYSTEM RESPONSE

```
ORIGINID FILE CLASS RECORDS  CPY HOLD DATE  TIME     NAME     TYPE     DIST
VMTEST   0302 A PRT 00009939 001 NONE mm/dd 09:48:17                   VMTEST
VMTEST   0301 A PUN 00026421 001 NONE mm/dd 09:45:22 LDT      DMKSAVNC VMTEST
Ready;
```

Issue the READCARD CPNUC MAP command to create a CMS file from the load map (spoolid 302) generated during the CP nucleus build.

ENTER

```
read cpnuc map
```

SYSTEM RESPONSE

```
RECORD LENGTH IS '150' BYTES.
Ready;
```

IPLing the Second-Level VM System

With few exceptions, IPLing a second-level system is similar to IPLing a first-level VM system. You must verify that the virtual machine configuration matches (by issuing a QUERY VIRTUAL command) or is a subset of the DMKRIO defined for the second-level system. Once this is done, you can IPL the virtual disk containing the CP nucleus. In our example, disk 123.

Note: Attention handling varies with the type of terminal used. Refer to the Terminal Reference for a list and description of the terminals supported by VM.

IPL VM in the normal fashion, responding where required.

```
ENTER

ipl 123 clear
SYSTEM RESPONSE

VM Release n , Service Level nnnn ; created on mm/dd/yy at hh:mm:ss

It is now hh:mm:ss EDT day mm/dd/yy
Change TOD clock (YES|NO) :
Enter--->no
```

The second-level system cannot set the time-of-day clock. Therefore, always reply "no" to the change time-of-day clock question.

```
DMKCPI971I System is uniprocessor generated
DMKUDR476I System directory loaded from volume MINRES
DMKCPI974I No valid override file; using system defaults
Start ((WARM|CKPT|FORCE|COLD) (DRAIN))|(SHUTDOWN) :
Enter--->cold
```

Because this is a test system there are no data or accounting information to be recovered. Therefore, you can perform a cold start unless some specific function requires a warm start.

```
DMKCPJ952I  04096K SYSTEM STORAGE

DMKCPJ957I STORAGE SIZE = 04096 K,          NUCLEUS SIZE = 404 K,
           DYNAMIC PAGING SIZE = 03380 K,   TRACE TABLE SIZE  = 060 K,
           FREE STORAGE SIZE = 0252 K,      VIRTUAL=REAL SIZE = 00000 K
hh:mm:ss FILES:  NO RDR,  NO PRT,  NO PUN
hh:mm:ss FORMATTING ERROR RECORDING AREA
```

The above system response is the normal response when you IPL a new system.

```
RRRR....RING....GGGG

DMKCPI966I Initialization complete
```

Once you IPL the second-level system, you are automatically logged on as the system operator in line mode. At this point you can enable graphic display devices to allow other users to dial into this system and log on to the second-level system. See "Enabling Terminals for a Second-Level VM System" on page 241.

Saving Second-Level CMS

To load CMS into your virtual machine you can:

- IPL the CMS system disk, or
- IPL a named saved system (in our example, CMS2).

To IPL the CMS system disk (in our example, MAINT2's 190 minidisk), you need 3M of virtual storage for each user needing CMS. On the other hand, if you save a second-level CMS it can be IPLed in 1M of virtual storage.

Note: For performance reasons, user's virtual machines should be kept to a minimum size.

Using the named, saved system defined earlier in DMKSNT (CMS2), the following steps create a second-level named saved system.

When the IPL of the second-level system is complete you receive the "Initialization complete" message. At this point you must disconnect the system operator.

ENTER

```
disconn
```

SYSTEM RESPONSE

```
hh:mm:ss disconnect at hh:mm:ss EDT day mm/dd/yy


 VM/370 ONLINE--          --PRESS REQUEST KEY TO BEGIN SESSION
RRRR....RING....GGGG
```

(Press ENTER)

```
Enter one of the following commands:
LOGON userid              (Example:  LOGON VMUSER1)
DIAL userid               (Example:  DIAL VMUSER2)
MSG userid message        (Example:  MSG VMUSER2 GOOD MORNING)
LOGOFF
```

You must log on as MAINT2, because it owns the CMS system disk (190).

ENTER

```
logon maint2
```

SYSTEM RESPONSE

```
Logon at hh:mm:ss EDT day mm/dd/yy
VM/SP Release n mm/dd/yy hh:mm:ss
```

(Press ENTER)

ENTER

```
ipl 190 parm savesys cms2
```

SYSTEM RESPONSE

```
SYSTEM SAVED
VM/SP Release n mm/dd/yy hh:mm:ss
```

(Press ENTER)

SYSTEM RESPONSE

```
Ready;
```

Once the READY message appears, a copy of the second-level named, saved system is saved. Any user of your second-level system can now load CMS into their virtual machine by:

- IPLing the 190 CMS system disk belonging to MAINT2

  ```
  ipl 190
  ```

  ```
  ----- or -----
  ```

- IPLing CMS2.

  ```
  ipl cms2
  ```

Chapter 3. Additional Topics When Operating VM under VM

Operating the Second-Level Virtual Machine

The virtual machine operation at this level can be confusing. At all times it requires an awareness of what level of VM you are interacting with and what functions you are trying to perform.

Issuing CP Commands to the First-Level VM System

While running VM under VM, use CP commands to:

- Communicate with the first-level VM system.
- Query the status of virtual machine devices or spool files.
- Attach or detach devices from the virtual machine configuration.

You can communicate with CP by either:

- Pressing the ATTN key twice to force a CP read.
- Prefixing the CP command with "%CP" where "%" is the line-end character assigned in the directory entry of the first-level system. (You would use this only in line mode.)
- Pressing the PA1 key (if using 3270 console mode).

Enabling Terminals for a Second-Level VM System

Most testing can be done by initializing and running tests from the operator's virtual machine. If you want full-screen applications (such as XEDIT and FILELIST) you can enable a graphic device using the following method:

If the directory entry for the first-level system includes a SPECIAL statement, you can then use full-screen mode. You can confirm that the virtual device is defined by issuing a CP QUERY command to first-level CP.

Enter

```
%cp query virtual graf
```

SYSTEM RESPONSE

```
GRAF AT 060
```

Enter

```
#cp vary online 060
```

SYSTEM RESPONSE

```
060 VARIED ONLINE
```

Enter

```
#cp enable 060
```

SYSTEM RESPONSE

```
COMMAND COMPLETE
```

On another terminal (on the same system), issue:

```
dial vmtest 060
```

Note: If the CP DIAL command is issued without a specified address, VM connects the terminal to the first line as defined in the SPECIAL control statement; the line belongs to the specified userid. If no lines are available or if all lines are busy, VM issues an error message and does not make the connection.

The end user will remain connected until one of the following happens:

- The virtual machine logs off using standard logoff procedure.
- The virtual machine is forcibly logged off.
- The terminal is powered off/on.

The end user will also get disconnected if the CP RESET command is issued from the VSE virtual console or from a user authorized with the CP RESET command.

Once disconnected, the end user is free to use the DIAL command to connect to another userid.

Varying Devices Offline/Online

Once you IPL the second-level virtual machine, the devices that were not accessible to that machine at IPL time are considered offline. However, you can attach more devices to your machine and have them placed online as required. For example, tape drives can be attached by the first-level machine operator to the virtual machine configuration at the address that matches the configuration of the second-level CP system. You can change these virtual addresses to conform to your second-level system's DMKRIO by using the CP DEFINE command. The second-level VM operator then issues the CP VARY command.

For example, if a graphic device is offline and VMTEST needs the graphic device to be made available, notify the first-level system operator via a message:

```
#cp msg op Please attach 080 to VMTEST as 080
```

The first-level system operator would issue:

```
vary online 080
```

SYSTEM RESPONSE (to the first-level system operator):

```
080 VARIED ONLINE
```

The first-level system operator issues:

```
attach 080 to vmtest 080
```

SYSTEM RESPONSE (to the first-level system operator):

```
080 ATTACHED TO VMTEST 080
```

The second-level system operator must issue:

```
vary online 080
```

SYSTEM RESPONSE (to the second-level system operator):

```
080 VARIED ONLINE
```

The device is now ready to be used by the second-level VM system. The operator informs VMTEST that the graphic device is now attached and ready for use.

Note: If the graphic device is a VM supported terminal, the CP ENABLE command must be issued before the end user can log on to the second level system that is using the device.

Spooling Options When Running VM under VM

First-Level VM System Spooling

If your virtual machine produces a large volume of unit record output and keeps a unit record device constantly busy, you may want to dedicate a unit record device to that virtual machine. To eliminate double spooling of printer output, include a DEDICATE statement in the first- level system's directory entry, such as:

```
DEDICATE 00E 002
```

The result of having the above statement causes all output from the second-level system virtual printer 00E to go directly to the real printer at address 002.

You can also have the system operator dynamically dedicate a unit record device to your virtual machine. For example, if your first-level system has a 4245 printer, you can have the operator issue a CP ATTACH command before the second-level system is IPLed. Send the operator a message requesting that ***cuu*** be attached to VMTEST as 00F, where 00F is the address of the 4245 printer on the real system.

Enter

```
%cp msg operator Please attach real printer 00F to me as 00F
```

If the real printer at 00F is not in use by the system or any other virtual machine, the operator issues the ATTACH command.

```
attach 00F to vmtest as 00F
```

When the device is attached, VM sends a confirmation message to VMTEST.

```
PRT 00F ATTACHED
```

Second-Level VM System Spooling

If the virtual machine performs any spooling operations, second level CP is also spooling (unless it has dedicated unit record devices). This double spooling operation does not create a problem. Second level CP detects that it is running in a virtual machine and at the end of each spooled output file issues a CP CLOSE command to first-level CP. This produces real spooled output for virtual spool files.

Notice that double separators occur. For instance, the separator page on virtual printed output includes four pages — two pages for the second-level VM system and two more pages for the separator of the first-level virtual machine on which the virtual CP system is running. The extra set of separator pages can be avoided by using the START command with the NOSEP option on the second-level system.

Console Specification

Because the LOGON console for a virtual machine operates as a 3215, 3210, or 3270, either of the following methods can be used to satisfy the console requirements for your virtual machine:

1. In the DMKRIO for the second-level system you are building, define the console device as DEVTYPE 3215 or 3210 in the RDEVICE macro.

   ```
   RDEVICE ADDRESS=009,DEVTYPE=3215
   ```

2. Another approach is to attach a console-type device to your virtual machine and use that device as an alternate second-level console. For example, if your DMKRIO for address 010 defines a DEVTYPE of 3277, attach a real 3277 to your virtual machine as address 010 to function as an alternate second-level console.

   ```
   RIOGEN CONS=010,ALTCONS=(009,015)
   ```

Using Dedicated Control Statements

Use the DEDICATE control statement to provide a virtual machine with a corresponding real device at logon time. The virtual machine will have sole use of the dedicated device.

Magnetic Tapes

A device such as a magnetic tape drive can be used by only one virtual machine at a time; therefore, specify it in a directory entry with a DEDICATE statement. For example:

```
DEDICATE 190 290
```

This statement allows the second-level VM system to access the device at real address 290 via a virtual address of 190.

Remote Devices

The DEDICATE statement can be used to attach remote 3270 Information Display System Printers (3284, 3286, 3287, 3262, 3268, and 3289) to a virtual machine. For example, a directory entry can include the statement:

```
DEDICATE NETwork 06E 0102
```

where 06E is the virtual address of the device in the virtual machine, and 0102 is the resource ID as specified in DMKRIO. Remote 3270 Information Display System Printers can also be attached by the NETWORK ATTACH command. For more details, see the *CP Command Reference.*

Problem Determination for the Second-Level VM System

Error Recording and Analysis

When running VM under VM, all hardware errors are recorded in the Error Recording Area of the first-level system. To access the recorded data, use the CMS CPEREP command. For information about error recording, formatting output from the error recording area with Service Record File devices, and CPEREP, refer to the *OLTSEP and Error Recording Guide.*

Dump Procedure

When running VM under VM there are three levels of storage:

first-level storage	real storage
second-level storage	virtual storage that the first-level VM system creates and manages for a virtual machine
third level storage	virtual storage that the second-level VM system creates and manages when running in a virtual machine

To place a dump of the second-level system into its virtual reader, you must:

1. Specify a test system's userid in the SYSDUMP operand of the SYSOPR system generation macro instruction in DMKSYS.
2. Initialize the second-level CP system by assuming the SET DUMP AUTO CP command (class B) by default.

Note: The second-level system's userid in the SYSDUMP operand should be OPERATOR, rather than the default of OPERATNS. OPERATOR helps you to readily identify your dumps. It also makes the dump immediately available to the OPERATOR virtual machine user for processing.

For more debugging information, refer to the *Diagnosis Guide.*

Trace Table Recording Facility

Problem determination capability is expanded to service personnel and system programmers with the Trace Table Recording Facility. The facility uses the CP command CPTRAP to create a chronological record of selected trace table, virtual machine interface, and CP interface information on a READER spool file. Use of the facility is intended for the analysis of VM problems that escape detection using a system dump.

A CMS utility program (TRAPRED) is included as part of the CPTRAP facility. TRAPRED uses the READER file as input, and supports output to either a spooled print file or an interactive terminal display. For additional information on using the CPTRAP facility, refer to the *Diagnosis Guide.*

VM/Interactive Problem Control System (VM/IPCS)

System programmers can use the Interactive Problem Control System (IPCS) component of VM to reduce the time, effort, and expense of solving software problems. VM/IPCS improves communications between users and the IBM Support Center. With the VM/IPCS facility you can diagnose, report, and manage problems.

Diagnosing Problems: You can diagnose a problem from any VM supported terminal without waiting for the hard-copy problem data. This is because you can look at disk-resident problem data as it occurs (CP and CMS ABEND dumps) and all virtual machine dumps that the VMDUMP command takes.

Reporting Problems: You can reduce the amount of hard-copy problem data and find available fixes for the system faster. This is because IPCS finds duplicate problems on the system and similar problems that IPCS previously met. IPCS gives you a standard way to report problems.

Managing Problems: You can track and manage problems until they are resolved. With problem and data management facilities, you can:

- Update individual dump reports and their summaries.
- Display and print problem reports and status reports.
- Print a hard-copy Authorized Program Analysis Report (APAR) and move the problem data to tape so a user can submit it to IBM.

Refer to the *Diagnosis Guide*, for more information about this VM component.

Backup/Restore Procedure for the Second-Level VM System

Creating a DDR Utility Tape

VM supplies a utility known as the VM DASD Dump/Restore (DDR) program. This program allows the user to dump, restore, copy, or print VM user minidisks and system volumes. The DDR program has five functions:

1. Dumps part or all of the data from a direct access storage device (DASD) to tape.
2. Transfers data from tapes created by the DDR dump function to a DASD. The DASD must be the same DASD that originally contained the data.
3. Copies data from one device to another of the same type.

4. Prints selected parts of DASD and tape records in hexadecimal and EBCDIC on the virtual printer.

5. Displays selected parts of DASD and tape records in hexadecimal and EBCDIC on the terminal.

For further information on the command format and options, see *CP for System Programming*.

It is recommended that the first file of any DASD backup tape contain a copy of the stand-alone DDR program. This ensures that the system can be restored with the same level of the DDR utility used to make the copy. Also, it is recommended that another tape contain all the VM utilities (such as DDR, Format/Allocate, and the Device Support Facility). This second tape should contain copies of the utilities you have found to be useful. This tape should be used in emergency situations such as when encountering a bad copy of the DDR utility on a backup tape.

For now, let's concentrate on putting the stand-alone DDR utility on tape. First, locate a tape that can be used. Remember, this procedure will destroy any information already on the tape; therefore, choose the tape carefully. Mount the tape on a tape/drive in read/write mode. Log on as VMTEST and attach the tape drive as 181 to VMTEST. To actually put the DDR utility on the tape, issue the following series of commands:

```
ipl cms
cp rewind 181
filedef input disk ipl ddr s
filedef output tap1 (den 6250
movefile input output
cp rewind 181
```

The FILEDEF commands define the input as coming from the file IPL DDR on the CMS system disk, and the output as going to the tape at virtual address 181. The MOVEFILE command then takes a copy of the IPL DDR program and puts it on the tape. By rewinding the tape after the MOVEFILE is completed, you can issue, IPL 181 to verify the results of the MOVEFILE. You now have a stand-alone DDR utility tape.

Using the CMS DDR Command

When running in CMS, you can use the command version of the DDR utility. The command version is functionally equivalent to the stand-alone version. The invocation of and termination of the two versions are the only differences. The command version is invoked by simply issuing the command DDR. The stand-alone version is invoked by loading the tape with the CP IPL command. Both versions are terminated by supplying a null line as input. However, the command version will return to the CMS environment when it terminates, whereas the stand-alone version will end and return to the CP environment.

DASD/Dump Restore (DDR) and the Second-Level System

Now that we have created the stand-alone DDR utility tape we can proceed with creating a backup copy of the MINRES DASD. MINRES contains the important system related areas, such as the cylinders for the CHECKPOINT, ERROR and WARM Start areas, the area reserved for the System Name Table (DMKSNT) entry for CMS2, and several minidisks belonging to such users as OPERATOR, DISKACN2 and MAINT2. To maintain the most current level of your system simply backup this entire volume. You then will be able to recover the second-level system completely if the need arises.

The process of creating a backup copy of the MINRES volume is simple. The copy will be made on the same tape as the DDR utility. Make sure the tape containing the DDR program, created in the last section, is still on the drive and attached to the VMTEST userid.

Make sure you start the tape at the beginning.

ENTER

```
cp rewind 181
```

When the tape is positioned at the beginning, load the DDR program into storage.

ENTER

```
cp ipl 181 clear
```

SYSTEM RESPONSE

```
VM/SP DASD DUMP/RESTORE PROGRAM - VM
ENTER CARD READER ADDRESS OR CONTROL STATEMENTS
```

Request that the message be printed on the terminal.

ENTER

```
sysprint cons
```

We want the input to come from the system volume MINRES at virtual address 123.

ENTER

```
input 123 3380 minres
```

Specify you output device type and tape density (in this example, 3420 abd 6250).

ENTER

```
output 181 3420 (mode 6250
```

The entire volume is going to be saved.

ENTER

```
dump 000 014
```

SYSTEM RESPONSE

```
ENTER NEXT EXTENT OR NULL LINE
```

ENTER

```
< null line >
```

The null line was entered because all the extents had already been supplied.

SYSTEM RESPONSE

```
DUMPING MINRES
```

Following this message, the system informs you which cylinders or blocks are being written to tape. When the message END OF DUMP appears on the terminal, a null line should be entered to end the DDR program.

When all the information has been dumped to tape, the tape must be rewound and unloaded from the tape drive. You should note the date and time of the DDR backup on the tape and also keep a record of which tape was used for the backup. You may, sometime, need to restore your system, and by keeping records of when backups were made and what tapes were used you can hasten your recovery efforts.

Now that we have successfully saved a copy of the MINRES volume, we can do the same for the MINTMP volume. You can detach the tape at address 181 and put a second tape on that drive. Perform the same steps for creating a stand-alone DDR utility as was done before we saved the MINRES volume. When you supply the INPUT statement to DDR, specify:

```
input 124 3380 mintmp
```

The other DDR statements are identical and when the message END OF DUMP appears on the terminal, the MINTMP volume has been saved.

Restoring Your Second-Level System

Restoring the second-level system from the DDR tape is similar to the process we followed when creating the backup copy of the MINRES volume. When we created the backup copy, the INPUT statement specified virtual address 123, and the OUTPUT statement specified virtual address 181; these addresses will be reversed in the restoring procedure. (VMTEST owns the minidisk at virtual address 123, which contains the full system volume MINRES.) Attach the tape containing the MINRES volume and make sure the tape is rewound.

ENTER

```
cp rewind 181
```

When the tape is positioned at the beginning, load the DDR program into storage.

ENTER

```
cp ipl 181 clear
```

SYSTEM RESPONSE

```
VM/SP DASD DUMP/RESTORE PROGRAM - VM
ENTER CARD READER ADDRESS OR CONTROL STATEMENTS
```

Request that the information be printed on the terminal.

ENTER

```
sysprint cons
```

Specify your input device type and tape density (in this example, 3420 and 6250).

ENTER

```
input 181 3420 (mode 6250
```

Output is to the system volume MINRES at virtual address 123.

ENTER

```
output 123 3380 minres
```

Restore everything onto tape.

ENTER

```
restore all
```

After the RESTORE command is entered, the message, RESTORING MINRES will appear on the terminal. This eventually is followed by information regarding which cylinders or blocks are being written to DASD. When the message END OF RESTORE appears on the screen, a null line must be entered to end the DDR program.

When all the information (on the tape) has been restored to the MINRES volume, the tape will be rewound and unloaded from the drive. The old system will be restored and ready for operation.

The steps for restoring the MINTMP volume are identical to those we just performed to restore the MINRES volume. Detach the tape containing the MINRES backup and attach the tape containing the MINTMP backup. After DDR has been loaded from the MINTMP backup, issue the same set of input statements to the DDR program as you did for the MINRES restoration ***with one exception***. Because we are restoring MINTMP, the OUTPUT statement should be:

```
output 124 3380 mintmp
```

When the END OF RESTORE message appears, the MINTMP volume has been restored, and your second-level system contains exactly what it did when you saved it earlier.

Bibliography

Prerequisite Books

Your prerequisite and corequisite books are based on the VM operating system you are using (VM/SP or VM/SP HPO).

IBM Virtual Machine/System Product (VM/SP)

Introduction, Order No. GC19-6200.

Installation Guide, Order No. GC24-5237.

Planning Guide and Reference, Order No. SC19-6201.

IBM Virtual Machine/System Product High Performance Option (VM/SP HPO)

Introduction, Order No. GC19-6200.

Installation Guide, Order No. SC38-0107.

Planning Guide and Reference, Order No. SC19-6223.

Corequisite Books

IBM Virtual Machine/System Product (VM/SP)

CP for System Programming, Order No. SC19-6224

CP Command Reference, Order No. SC19-6211

CMS User's Guide, Order No. SC19-6210

CMS Macros and Functions Reference, Order No. SC19-6209

Operator's Guide, Order No. GC19-6202

Terminal Reference, Order No. GC19-6206

Data Areas and Control Block Logic-CP, Order No. LY20-0896

IBM Virtual Machine/System Product High Performance Option (VM/SP HPO)

CP for System Programming, Order No. SC19-6224

CP Command Reference, Order No. SC19-6227

Operator's Guide, Order No. GC19-6225

Diagnosis Guide, Order No. LY24-5241

System Facilities for Programming, SC24-5288

IBM VSE/SP

General Information, Order No. GC33-6176

Planning, Order No. SC33-6177

Installation, Order No. SC33-6178

System Use, Order No. SC33-6174

Networking, Order No. SC33-6180.

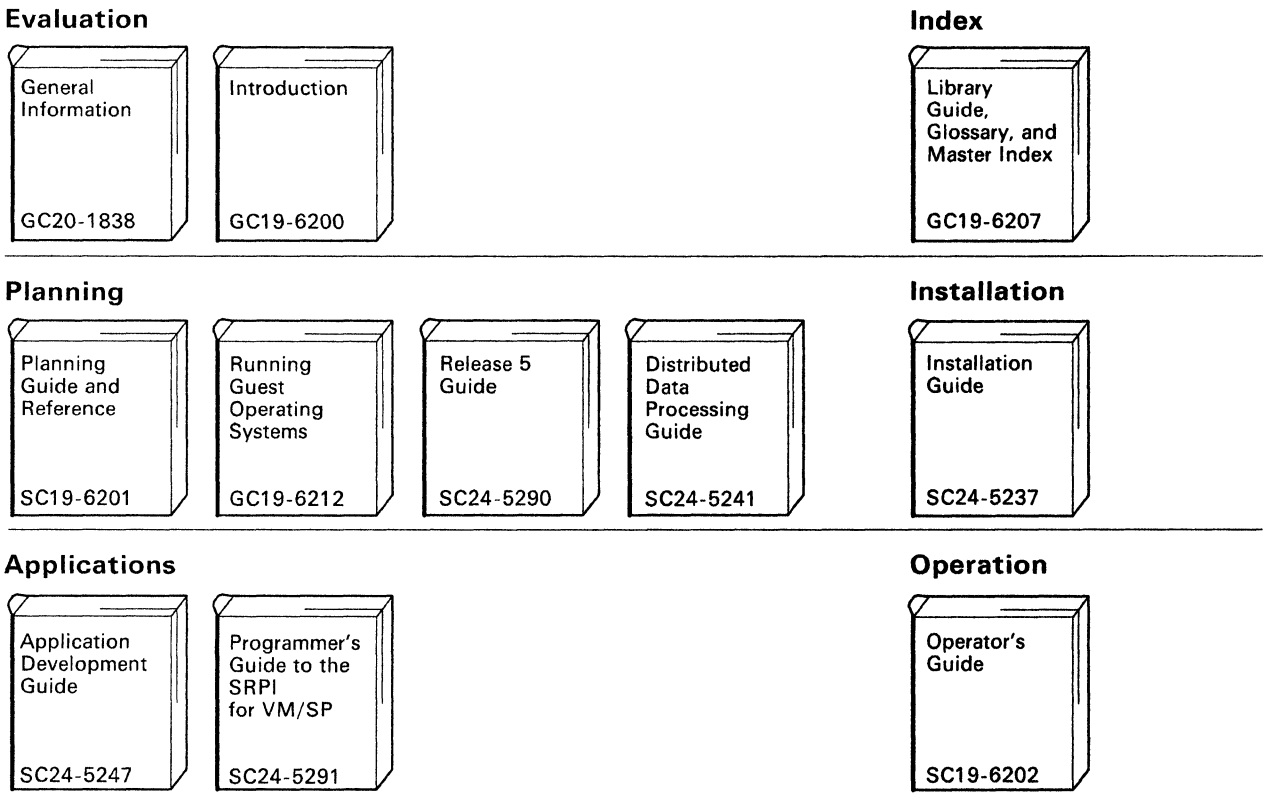
Associated Books

IBM VM/SP HPO Release 4.2 OLTSEP and ERROR Recording Guide, Order No. ST00-1901

Environmental Recording Editing and Printing (EREP) Program, Order No. GC28-1178

Device Support Facility User's Guide and Reference, Order No. GC35-0033

The VM/SP Library (Part 1 of 3)



Reference Summaries **To order all of the Reference Summaries, use order number SBOF-3242**

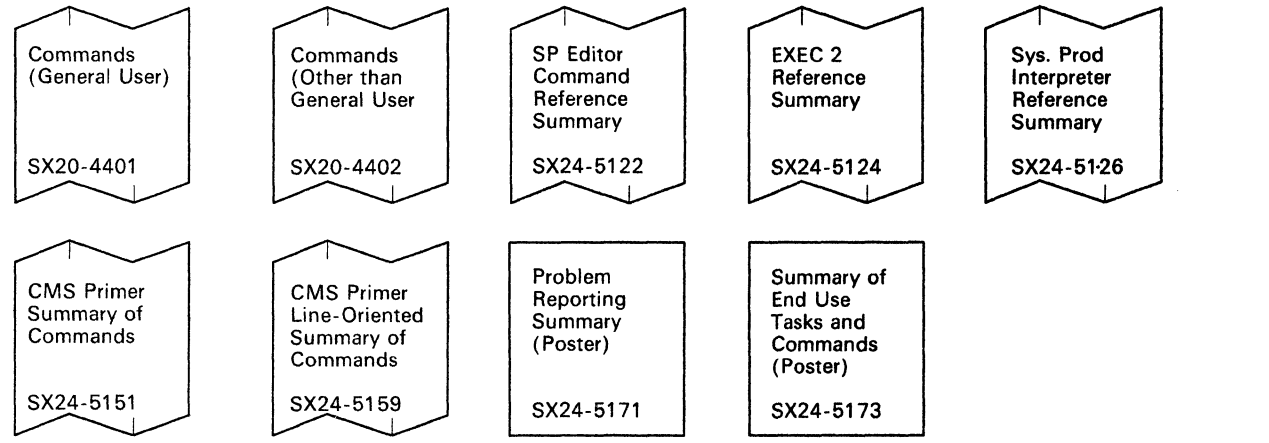
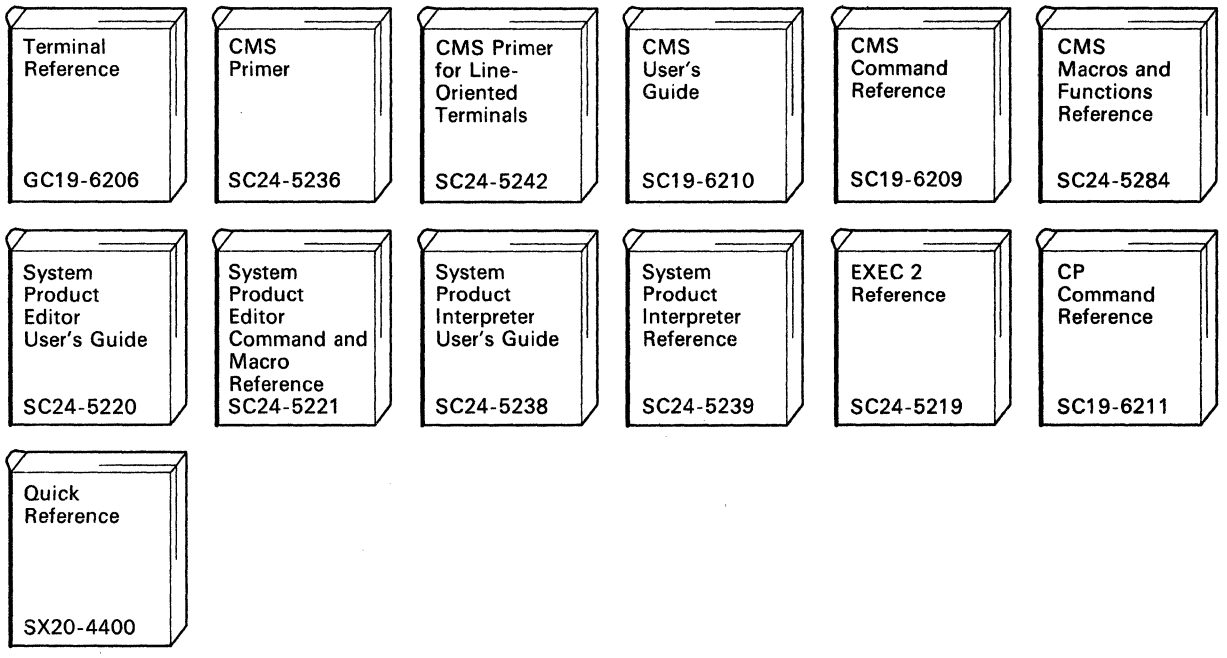


Figure 91 (Part 1 of 3). VM/SP Library

The VM/SP Library (Part 2 of 3)

End Use



Diagnosis

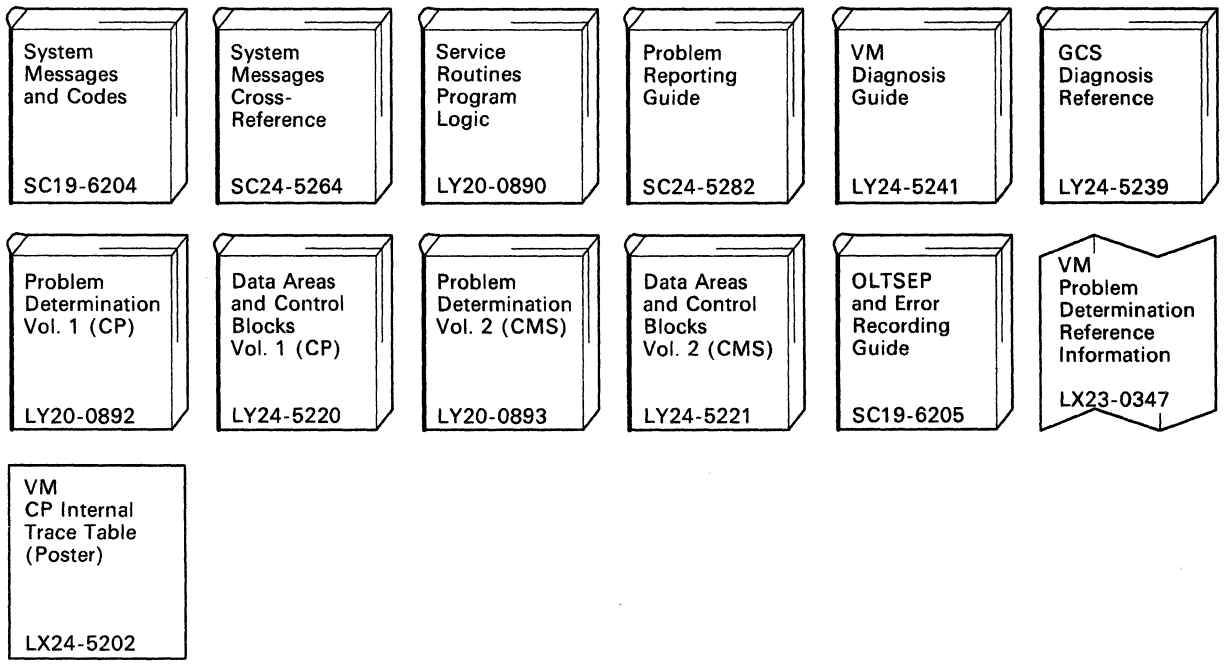
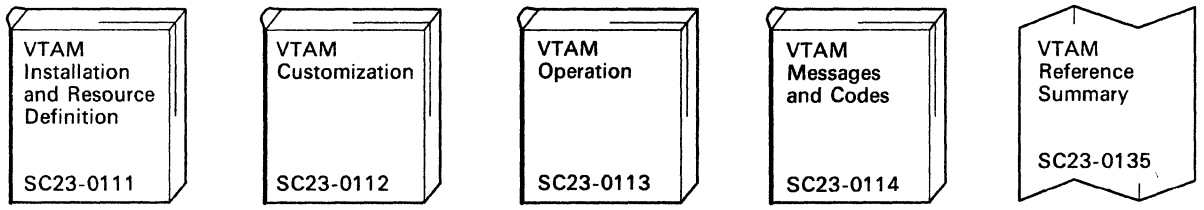


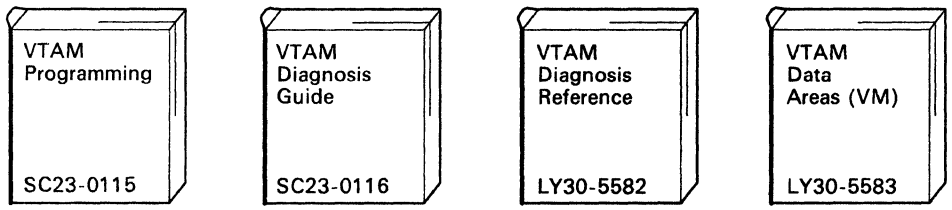
Figure 91 (Part 2 of 3). VM/SP Library

The VM/SP Library (Part 3 of 3)

Administration

Auxiliary Communication Support





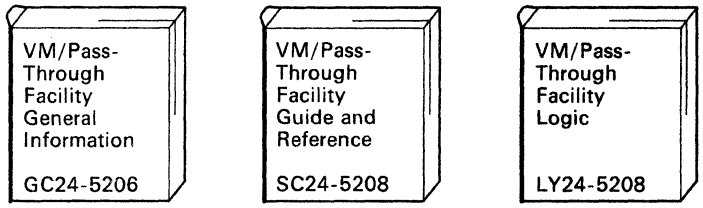


Figure 91 (Part 3 of 3). VM/SP Library

The VM/SP HPO Library

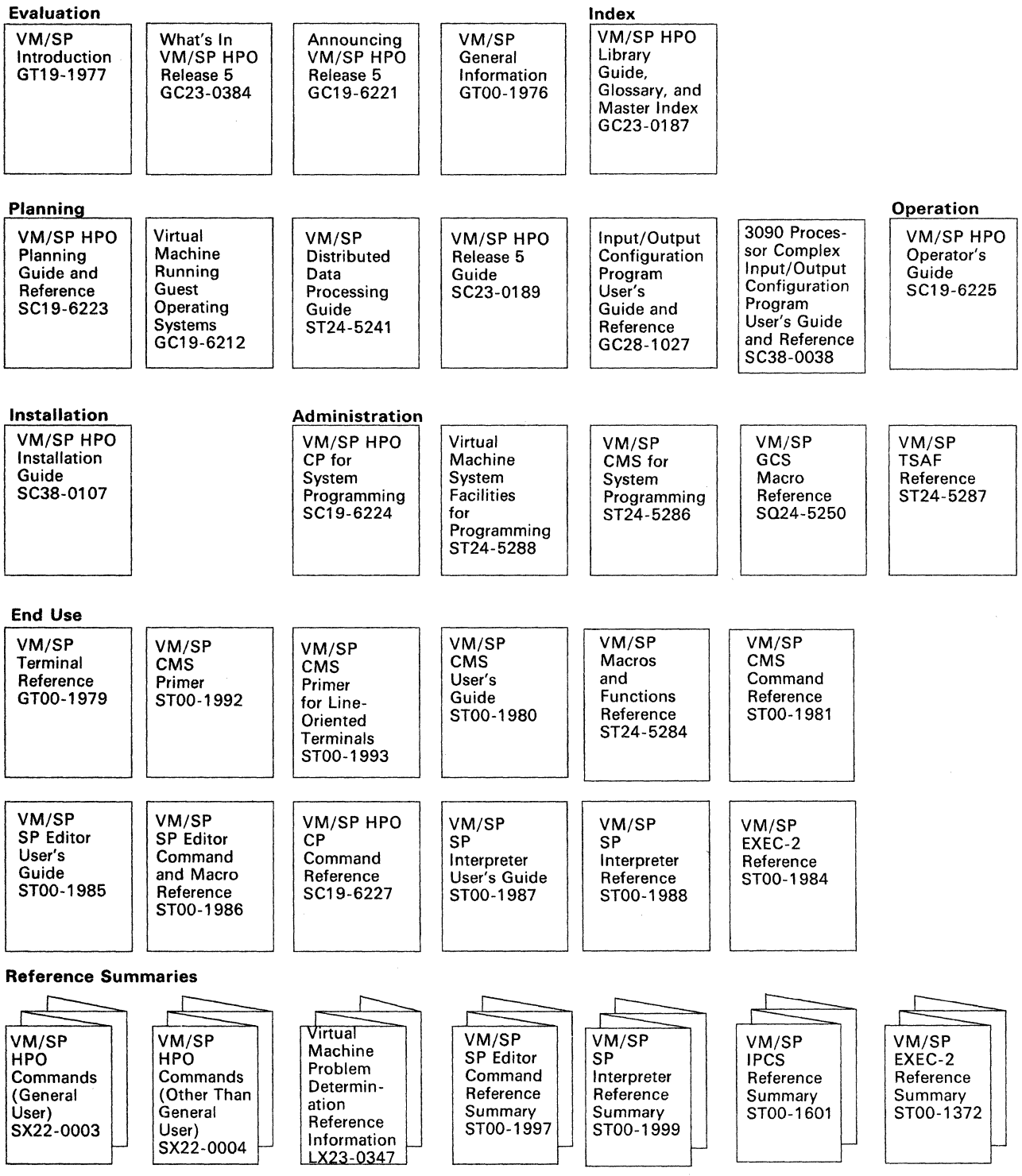


Figure 92. VM/SP HPO Library (Part 1 of 2)

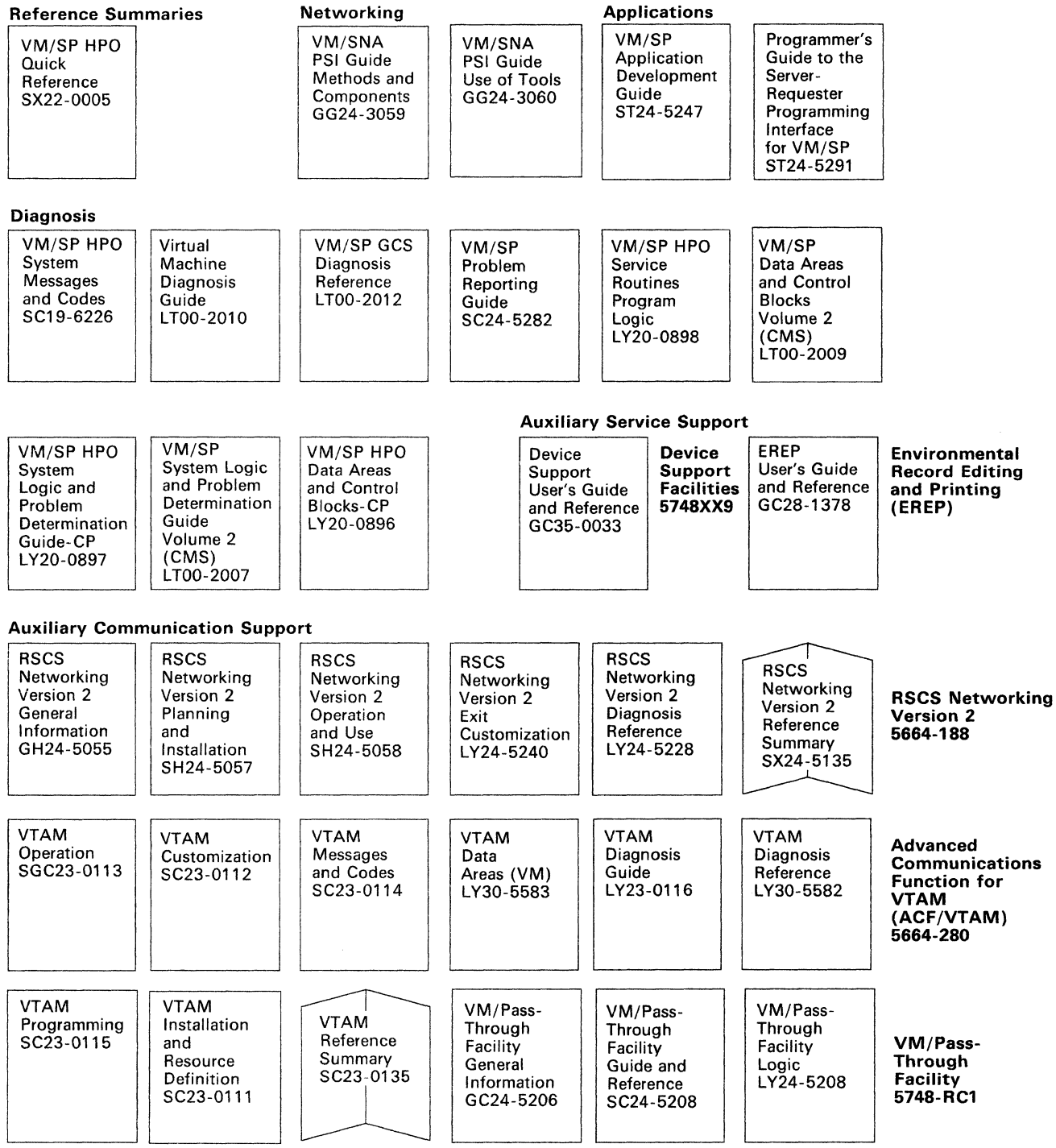


Figure 93. VM/SP HPO Library (Part 2 of 2)

Index

Special Characters

A

B

C

D

E

N

O

P

Q

R

S

T

U

V

W

X

Z

Numerics

Virtual Machine
Running Guest
Operating Systems

READER'S COMMENT FORM

Order No. GC19-6212-5

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

How did you use this publication?

[ ] As an introduction
[ ] As a reference manual
[ ] For another purpose (explain) _____

[ ] As a text (student)
[ ] As a text (instructor)

_____

Is there anything you especially like or dislike about the organization, presentation, or writing in this manual? Helpful comments include general usefulness of the book; possible additions, deletions, and clarifications; specific errors and omissions.

Page Number: Comment:

What is your occupation? _____

Newsletter number of latest Technical Newsletter (if any) concerning this publication: _____

If you wish a reply, give your name and address: _____

_____

_____

IBM branch office serving you _____

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

Note: Staples can cause problems with automatic mail-sorting equipment. Please use pressure-sensitive or other gummed tape to seal this form.

GC19-6212-5

Reader's Comment Form

Fold and Tape | Please Do Not Staple | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Department 52Q MS 458
Neighborhood Road
Kingston, New York 12401

Fold and Tape | Please Do Not Staple | Fold and Tape

IBM®

PRINTED IN U.S.A. GC19-6212-5

GC19-6212-5

Virtual Maching Running Guest Operating Systems (File No.S370/4300-34) Printed in U.S.A. GC19-6212-5