GC20-1877-0
File No. S/370-01

Systems

A Guide to the IBM 4341 Processor

This guide presents hardware I/O device, programming systems, and other pertinent information about the IBM 4341 Processor that describes its significant new features and advantages. Knowledge of System/360 hardware and I/O devices is assumed. The contents of the guide are intended to aquaint the reader with the 4341 Processor and to be of benefit in planning for its installation.

First Edition (April 1979)

This guide is intended for planning purposes only. It will be updated from time to time; however, the reader should remember that the authoritative sources of system information are the system library publications for the 4341 Processor, its associated components and its programming support. These publications will first reflect any changes.

Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form has been provided at the back of this publication for readers' comments. If this form has been removed, address comments to: IBM Corporation, Technical Publications, Dept. 824, 1133 Westchester Avenue, White Plains, New York 10604. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

PREFACE

This publication assumes the reader is familiar with a System/360 Model 22 or higher. The reader should have a general knowledge of System/360 architecture, channels, I/O devices, and programming systems support. This publication highlights and discusses the significance of only those hardware, I/O, and programming systems features of the 4341 Processor that are different from those of System/360 Models 22 and up.

This publication is also designed for readers who are knowledgeable about System/370 architecture, channels, I/O devices, and programming systems. Features of the 4341 Processor that are like the same features in System/370 processors are identified. Compatibility between the architectures implemented in 4300 Processors and System/370 Processors and their programming systems support are also discussed.

CONTENTS

FIGURES

TABLES

SECTION 01: HIGHLIGHTS

The 4341 Processor is an intermediate-scale, general purpose processor. It is one of the IBM 4300 Processors and is compatible with the 4331 Processor. The 4341 Processor offers System/360- and System/370-compatible architecture, a new architecture that provides new function, and a new level of price performance for intermediate system users made possible by the use of large-scale integrated technology.

The 4341 Processor provides the range of commercial and scientific data processing capabilities offered by System/360 and System/370, as well as the advanced functions provided by System/370, such as support of virtual storage. Virtual storage support is designed to facilitate new application development and to ease entry into, and expansion of, online data processing operations.

The 4341 Processor provides a growth path for users of System/360 Models 50 and 65. It offers such installations advanced functions, significantly increased internal performance, greatly improved price performance, and hardware and programming systems compatibility. The price performance of the 4341 Processor together with its hardware and programming systems compatibility with System/370 also make the 4341 Processor a suitable growth processor for installations with a System/370 Model 125, 135, 138, or 145. In addition, the 4341 Processor can be utilized in distributed processing environments that require high performance and large-capacity storage.

Transition from System/360 Model 50 and 65 configurations to a 4341 Processor configuration can be accomplished with a minimum of effort because most System/360 user programs, I/O devices, and programming systems are upward compatible with those of the 4341 Processor.

Compatible growth from System/360 operating systems to a 4341 Processor virtual storage environment and compatible transition from System/370 operating systems can be achieved using Disk Operating System/Virtual Storage Extended (DOS/VSE) and Operating System/Virtual Storage 1 (OS/VS1), which support the 4341 Processor. DOS/VSE is a compatible extension of DOS/VS Release 34 (which utilized DOS Version 4 as a base) that provides significant new function and performance features. OS/VS1, which is based on OS MFT, is extended to support the 4341 Processor as of Release 7.

In addition to supporting virtual storage, the operating systems that support the 4341 Processor offer a large number of other capabilities and performance-oriented enhancements that are not provided by System/360 operating systems. If necessary (for transition purposes, for example), DOS Release 26 can be used in a 4341 Processor.

A virtual machine environment is supported by Virtual Machine Facility/370 (VM/370), the successor to CP-67/CMS for System/370. VM/370 Release 6 and the Virtual Machine/Basic System Extensions (VM/BSE) Release 2 program product support the 4341 Processor. While CP-67/CMS is available only to Model 67 System/360 users, VM/370 operates on all System/370 processors except Models 115, 125, 155, and 165 and supports all 4300 Processors.

VM/370 provides interactive computing via its Conversational Monitor System (CMS) component and remote spooling via its Remote Spooling Communications Subsystem (RSCS) component.

While the 4341 Processor does not have a 1400 compatibility feature, simulation of 1401/1440/1460 programs on the 4341 Processor using a DOS Release 26, DOS/VS Release 34, or DOS/VSE operating system is supported.

A 1400 simulator program can be generated using the IBM Systems 1401/1440/1460 Emulator Program (a program product).

Highlights of the 4341 Processor, when compared with System/360 Models 50 and 65, are as follows:

- Upward compatibility with most System/360 and System/370 architecture and programming systems has been maintained in the 4341 Processor through implementation of the System/370 mode of processor operation. This mode provides compatibility for System/360 and System/370 control programs and problem programs.

  The Extended Control Program Support:Virtual Storage Extended (ECPS:VSE) mode of operation, not provided in System/360 or System/370, is also implemented in the 4341 Processor. This mode is designed to provide increased processor performance when DOS/VSE is used. ECPS:VSE mode provides compatibility for System/360 and System/370 problem programs but not control programs.

- The 4341 Processor is capable of an instruction execution rate of up to 3.2 times that of the 3138 Processing Unit and 1.73 times that of the 3148 Processing Unit with identical programs and equivalent configurations running under DOS/VS Release 34, OS/VS1 Release 7, or VM/370 Release 6. This performance is achieved for OS/VS1 and VM/370 when the appropriate hardware assist is utilized in the 4341 Processor (ECPS:VS1 or ECPS:VM/370, respectively) and Models 138 and 148 (VS1 hardware assist function and VM/370 hardware assist function of the ECPS feature).

  The 4341 Processor offers balanced performance for binary, decimal, and scientific (floating-point) arithmetic instructions. In a scientific environment, the 4341 Processor has been measured to be up to 3.1 times faster than a 3148 Processing Unit running the same FORTRAN jobs.

  A reduction of up to 13 percent of total processor utilization time has been measured when DOS/VSE, running in an interactive environment, is used with ECPS:VSE mode instead of System/370 mode in effect in the 4341 Processor.

- The following are instruction processing function features of the 4341 Processor.

  Instruction processing function logic is implemented in large-scale integrated technology that provides greatly increased circuit density per chip (704 circuits maximum versus 8 circuits maximum per chip in most System/370 processors). See Section 05:05 for a detailed logic technology discussion.

  Implementation of a System/370 mode and an ECPS:VSE mode, both of which support virtual storage, is standard. The major difference between the two modes is the way in which address translation is performed to support a virtual storage environment.

  System/370 mode provides compatibility with System/360 and System/370. When System/370 mode is in effect, either the basic control (BC) mode or the extended control (EC) mode of operation will also be in effect. BC mode is the System/360-compatible mode for 4300 Processors and is architecturally equivalent to BC mode for System/370 Processors.

  The EC mode of System/370 mode provides a different PSW format and an altered, permanently assigned lower processor storage area. These changes provide additional processor control and support new functions, such as the dynamic address translation facility to

support virtual storage, that are not available in BC mode. The EC mode of System/370 mode in the 4341 processor is architecturally equivalent (with a few exceptions) to the EC mode of System/370 processors.

When System/370 and EC modes are in effect, the standard dynamic address translation (DAT) facility can be enabled to provide translation of addresses in instructions (but not addresses in channel command words) during program execution. The standard channel indirect data addressing function is provided for System/370 mode to enable an I/O buffer to span a set of noncontiguous processor storage areas when DAT is enabled.

The DAT facility can be used to support one virtual storage of up to 16 million bytes or multiple virtual storages of up to 16 million bytes each. The DAT facility in the 4341 Processor is functionally identical to the DAT facility for System/370 processors.

ECPS:VSE mode is specifically designed to be utilized with the DOS/VSE operating system to provide increased processor performance when compared to that achieved using DOS/VSE executing with System/370 mode in effect.

ECPS:VSE mode provides an alternative method for translating addresses in instructions during instruction execution and, in addition, provides for translating addresses in channel command words during I/O operations. The address translation technique used in ECPS:VSE mode permits only one virtual storage of up to 16 million bytes to be supported.

ECPS:VSE mode offers a reduction in the processor time required to perform address translation to support one virtual storage. The reduction occurs primarily because programmed channel program translation, including the use of channel indirect data addressing, are eliminated. In addition, the instruction address translation technique used in this mode is faster than the DAT facility technique used in System/370 mode.

The cycle time of the 4341 Processor varies from 150 to 300 nanoseconds.

The instruction processing function design provides increased instruction execution performance. Instruction prefetching is implemented that results in the overlap of instruction fetching with instruction execution during sequential instruction processing. In addition, an eight-byte-wide arithmetic logic unit is used that improves the performance of decimal and floating-point operations.

Improvements in the number of functions performed during the instruction cycle of instruction execution result in the faster execution of many other instructions, most of which are among the more frequently used instructions.

The standard instruction set for the 4341 Processor includes many new general purpose and control-program-oriented instructions in addition to the powerful System/360 instruction set. The standard 4341 Processor instruction set consists of the entire instruction set provided for System/370 (except for multiprocessing and direct control instructions) and several new control instructions that can be utilized only when ECPS:VSE mode is in effect.

Floating-point arithmetic, which includes extended precision operations, is standard. Precision of up to 28 hexadecimal digits, equal to up to 34 decimal digits, is provided by the extended precision data format.

An interval timer of 3.3 milliseconds resolution at location 80 in processor storage, which can improve job accounting accuracy, is standard. A 16.6-ms resolution timer is available for Models 50 and 65.

A time-of-day clock is included as a standard feature to provide more accurate time-of-day values than does the interval timer. This clock has a one-microsecond resolution.

A CPU timer and clock comparator are standard. The CPU timer provides an interval timing capability similar to that of the interval timer but has a much larger capacity than the latter and is updated every microsecond, as is the time-of-day clock. The clock comparator can be used to cause an interruption when the time-of-day clock passes a specified value. These items provide higher resolution timing facilities than the interval timer and enable more efficient timing facility routines to be used.

The standard byte-oriented operands facility permits byte boundary alignment for the operands of nonprivileged instructions, making it unnecessary to add padding bytes within records or to blocked records to align fixed- or floating-point data. In the 4341 Processor, minimal performance degradation results from the use of unaligned data.

A monitoring feature that can be used to trace user-defined program events for the purpose of debugging or statistics gathering is standard.

Program event recording is standard and is designed to be used as a problem determination aid. This feature includes hardware that monitors the following during program execution: successful branches, the alteration of general registers, and instruction fetching from, and alterations of, specified areas of processor storage. It can function only when the 4341 Processor is operating in EC mode.

The Extended Control Program Support features are standard. They provide a VS1 hardware assist function (ECPS:VS1) and a VM/370 hardware assist function (ECPS:VM/370) for the 4341 Processor. Either ECPS:VS1 or ECPS:VM/370 can be activated only when System/370 mode is in effect. The two assists cannot be used concurrently. The operator selects the assist to be active for an IML using the operator console.

ECPS:VS1 and ECPS:VM/370 increase the performance of a VS1 and VM/370 operating system (respectively) when it operates in a 4341 Processor, since these functions cause certain control program routines to execute in hardware instead of as routines written using 4341 Processor instructions.

ECPS:VS1 for the 4341 Processor is functionally equivalent to the OS/VS1 Extended Control Program Support feature for the System/370 Model 158 and 3031 Processor Complex. ECPS:VM/370 provides the same facilities as the Virtual Machine Assist feature, which is available for most System/370 processors, plus several additional facilities. It is functionally equivalent to the VM/370 hardware assist function for Models 135 Model 3, 138, 145 Model 3, and 148.

A reduction of up to 7 percent of supervisor state processor busy time has been measured when ECPS:VS1 is utilized by the 4341 Processor as compared to the same version of OS/VS1 operating without the assist activated. A reduction of up to 84 percent of the supervisor state processor time used by CP has been measured when ECPS:VM/370 is utilized as compared to the same CP operating without the assist activated.

- The following are significant storage features of the 4341 Processor.

  All storage in the 4341 Processor--processor (main), control, high-speed buffer, and local--is implemented using monolithic technology instead of discrete ferrite cores. The technology used for processor storage in the 4341 Processor provides a much denser storage chip (64K or 16K bits per chip) than is used in most System/370 processors (2K bits per chip) and a new bipolar technology is used for control storage and high-speed buffer storage (see Section 05:05).

  A two-level storage system is implemented, consisting of large processor storage used as backing storage for a smaller high-speed buffer storage. The instruction processing function works mostly with the buffer so that the effective processor storage cycle is a fraction of the actual processor storage cycle. For System/370, high-speed buffer storage is implemented only in large-scale processors (Models 155 and up and 303X processors).

  8K bytes of high-speed buffer storage are standard. Data is fetched from the buffer at a rate of 225 nanoseconds for a doubleword.

  2048K and 4096K of processor storage are available--eight times the maximum amount of main storage available on the Model 50 and four times the maximum amount available on the Model 65.

  A minimum of 14K and a maximum of 108K of highest addressed installed processor storage will not be accessible to programs. The amount of unavailable processor storage depends on the number of UCWs installed, the mode (System/370 or ECPS:VSE) in effect, and the amount of processor storage installed for ECPS:VSE mode.

  Reloadable control storage to contain all the microcode required by the instruction processing function of the 4341 Processor is standard. Use of writable, instead of read-only, control storage offers the advantages of improved system serviceability and ease of optional feature and engineering change installation.

  Error checking and correction (ECC) hardware (like that implemented in System/370), which automatically corrects all single-bit processor storage errors and detects (but does not correct) all double-bit and many multiple-bit errors, is standard. System/360 processors use parity checking for processor storage.

- The following channel features are provided for the 4341 Processor.

  Two channel groups are available for the 4341 Processor. The standard channel group consists of one byte multiplexer and two block multiplexer channels. The optional channel group consists of three block multiplexer channels or one byte and two block multiplexer channels.

  Functionally, the byte multiplexer channels for the 4341 Processor are equivalent to those for System/360 and System/370 processors. The standard byte multiplexer channel has a 16-KB/sec maximum data rate in byte mode for single-byte transfer operations and a 64-KB/sec maximum byte mode data rate for four-byte transfer operations. For burst mode operations, a maximum data rate of 1 MB/sec is possible for a buffered device.

  The optional byte multiplexer channel has a 22-KB/sec and 88-KB/sec maximum data rate for one-byte and four-byte transfer operations, respectively, and a 2-MB/sec maximum data rate for burst mode operations on buffered devices.

The two standard block multiplexer channels have a maximum data rate of 2 MB/sec each. Two of the block multiplexer channels in the optional channel group have a maximum data rate of 2 MB/sec while one has a maximum data rate of 1 MB/sec. The maximum aggregate data rate for the block multiplexer channels is 4 MB/sec when the optional channel group is not installed and 9 MB/sec when the optional channel group with three block multiplexer channels is installed.

Functionally, a block multiplexer channel in the 4341 Processor is equivalent to that for System/370 processors. A block multiplexer channel is a superset of a selector channel and is designed to increase total system throughput by permitting increased amounts of data to enter and leave the processor in a given time period. A single block multiplexer channel can support interleaved, concurrent execution of multiple high-speed I/O operations. The block multiplexer channel for the 4341 Processor can also operate in selector channel mode.

Block multiplexer channels with data rates of up to 2 MB/sec support attachment to the 4341 Processor of 3370, 3330-series, 3340/3344, 3350, and 2305 Model 2 direct access storage, which cannot be attached to Models 50 and 65. These disk devices have rotational position sensing capability and are designed to be used with block multiplexer channels.

Optionally, one Channel-to-Channel Adapter can be installed in a 4341 Processor and attached to any block multiplexer channel. The adapter can be used to connect the channel in the 4341 Processor to a channel in a System/360, a System/370, or another 4341 processor.

- A 3278 Model 2A Display Console, consisting of a cathode ray tube, keyboard, and operator control panel, is required as the operator console. Display mode and a printer-keyboard mode are standard for this console, which natively attaches to the 4341 Processor. The display console provides a faster means of display than a typewriter-keyboard device.

  The display console is used to perform manual operations that for System/360 processors are performed using switches and pushbuttons on a control panel located on the front of the processor unit. The display console is also used for operator-to-operating system communication and by the customer engineer for performing diagnostic functions.

  Up to three 3278 Model 2A Display Consoles (without the operator control panel) and/or 3287 Model 1 or 2 Printers (any combination of the two) can be natively attached to the 4341 Processor in addition to the required 3278 Model 2A Display Console. The additional 3278 displays can be used as alternate or additional consoles.

  The 3287 Printer can be used for hard-copy backup of a 3278 Display Console that operates in display mode. A 3287 Printer is recommended for hard-copy output when the display console operates in printer-keyboard mode, which is made availble to enable a System/360 operating system that utilizes a 1052 Printer-Keyboard or a System/370 operating system that uses a 3210/3215 Console Printer-Keyboard as the operator console device to execute in the 4341 Processor.

- I/O devices for the 4341 Processor include the following.

  Most I/O devices for System/360 Models 50 and 65 can be attached. In addition, several I/O devices that cannot be attached to Models

50 and 65 because of their data rate or to any System/360 processors are attachable to the 4341 Processor.

The 3505 Card Reader and the 3525 Card Punch with optional card read capability (not attachable to Models 50 and 65) can be attached to the 4341 Processor. A variety of models are available. They offer 2500-series 80-column card users configuration flexibility, new functions, high reliability, and greatly expanded error recovery facilities.

Models B1 and B2 of the 3505 Card Reader can operate at 800 and 1200 cards per minute, respectively. Significant new features for 2500-series users include Optical Mark Reading (optional) and Read Column Eliminate (standard). The latter is designed to permit the successful reading of cards containing internal perforations or other holes that normally would cause an error.

Models P1, P2, and P3 of the 3525 Card Punch can punch and, optionally, read 100, 200, and 300 cards per minute, respectively. New features of this unit for 2500-series users include automatic punch retry when an error is detected during non-read/punch operations (standard) and optional card printing. A two-line print feature and a multiline (up to 25 lines) print feature are available.

The 3203 Model 5 Printer (not attachable to Models 50 and 65), with a tapeless carriage and print speed of 1200 alphameric lines per second (with a 48-character set), can be attached. The 3203 Model 5 is a standalone version of the 3203 Model 2, which attaches to System/370 Models 115 and 125. In addition to improved price performance, the 3203 Printer offers several advantages over the 1403 Printer, such as reduced operator intervention, higher reliability, quieter operation, and a more compact design.

The high-speed 3211 Printer, with a tapeless carriage and print speed of 2000 alphameric lines per minute (for 48-character sets), is attachable. The 3211 can be installed instead of, or in addition to, 3203 Model 5 Printers when the volume of print activity in the installation is high enough to require its faster print speed or when additional print capability is needed.

The 3800 Printing Subsystem (not attachable to Models 50 and 65) can be attached to provide very high-speed printing (up to 10,020 lines per minute for 11-inch-long paper with 6 lines to the inch and up to 20,040 lines per minute with 12 lines to the inch). The 3800 is a nonimpact printer that uses an electrophotographic technique with a low-powered laser to print on single-form paper. It offers a variety of printing features not provided by other IBM printers as well as operational features such as a burster-trimmer-stacker.

The 3410/3411 Magnetic Tape Subsystem (not attachable to a Model 65), Models 1, 2, and 3, can be attached to provide data transfer rates of 20, 40, and 80 KB/sec, respectively, at 1600-BPI density. Phase-encoded recording is used. A Model 1 subsystem can consist of from one to four tape units. Models 2 and 3 of the subsystem can have from one to six tape units. This subsystem offers improved price performance over 2400-series tape units for data rates under 120 KB/sec, a simplified tape path to speed tape setup, Dual Density and Seven Track features, a totally new compact physical design that minimizes floor space requirements, and reliability, availability, and serviceability improvements.

The 3803/3420 Magnetic Tape Subsystem is attachable. Models 3, 5, and 7 of the 3420 Magnetic Tape Unit have data rates of 120, 200, and 320 KB/sec, respectively, at 1600-BPI recording density. Phase-

encoded recording, which permits automatic correction of all single-bit read errors in flight, is used for these models. Models 4, 6, and 8 of the 3420 Magnetic Tape Unit have data rates of 470, 780, and 1250 KB/sec, respectively, at 6250-BPI recording density. The advanced recording technique used for these models provides automatic correction of all single- and double-bit read errors in flight.

This tape subsystem, at 6250-BPI density, offers significantly faster data rates than 2400-series tape units; increased tape reel capacity at 6250-BPI density; Dual Density and Seven Track features for compatibility with, and conversion of, 2400-series tape volumes; greatly reduced operator handling through implementation of such features as automatic tape threading and cartridge loading; lower cost tape switching than is provided for 2400-series tape units; and enhanced reliability, availability, and serviceability features.

3370 Direct Access Storage (not attachable to System/360 or System/370) can be attached to the 4341 Processor via 3880 Storage Control. The 3370 is very large capacity, high-speed, fixed-media direct access storage with movable heads. Data is stored on nonremovable disks utilizing fixed block architecture (FBA) instead of the count, key, data (CKD) architecture utilized for System/360 and System/370 disk devices. Fixed block recording is designed to utilize fully the block multiplexing capability (rotational position sensing is a standard facility) and it provides data mobility advantages over the self-formatting recording technique.

A 3370 drive has a data transfer rate of 1.86 MB/sec, average seek time of 20 milliseconds (ms), and full rotation time of 20.2 ms. Each 3370 drive has a capacity of 571.3 million bytes and two actuators, each of which can access half the data in the drive. A 3370 string can contain from one to four drives in one-drive increments for a string capacity of approximately 2.3 billion bytes of disk storage.

Each 3370 contains two logical drives, one per actuator. Only one actuator can transfer data at a time. However, both actuators can perform seeking and rotational positioning operations concurrently or one actuator can transfer data while the other is involved in a positioning operation. Thus, a four-drive 3370 string can have up to eight channel programs active concurrently.

Automatic error correction, rotational position sensing, and multiple requesting facilities like those implemented for CKD 33XX direct access devices (which are not implemented for System/360 direct access devices) are standard for 3370 strings.

3880 Storage Control provides functional and cost advantages over 3830 Storage Control Model 2. A 3880 Storage Control unit contains two independent storage control functions called storage directors, each of which can have up to four direct access storage strings attached (total of eight strings per 3880 unit versus four maximum per 3830 Model 2). When the two storage directors are attached to two different channels, a 3880 can handle two data transfer operations simultaneously, one for each storage director. Two- and four-channel switching is available for the 3880.

3880 Storage Control (not attachable to Models 50 and 65), when attached to the 4341 Processor, can have both CKD and FBA disk devices attached. The devices attached to one storage director must all be of the same architecture (CKD or FBA). The 3370, 3340/3344, 3330-series, and 3350 Direct Access Storage drives can be attached to 3880 Storage Control attached to the 4341 Processor. A storage director can have 3330-series and 3350 strings intermixed but

3340/3344 strings cannot be intermixed with other types of strings on the same storage director.

3330-series disk storage Models 1, 2, and 11 (not attachable to Models 50 and 65) can be attached to the block multiplexer channels in a 4341 Processor via 3830 Storage Control Models 1 and 2 and 3880 Storage Control. The 3330-series is large-capacity disk storage that offers significantly faster seeks and more than twice the data rate of the 2314 facility. The 3336 Model 1 and 11 removable disk packs are used as the storage medium. Eight Model 1 3330-series drives offer more than three times the capacity of eight 2314 drives. Automatic error correction features, rotational position sensing, and multiple requesting are provided as standard features.

The 3330-series has a data transfer rate of 806 KB/sec, average seek time of 30 ms, and full rotation time of 16.7 ms. A 3330-series Model 1 or 2 drive has a maximum capacity of 100 million bytes while a Model 11 drive has a 200-million-byte capacity. A string of from one to eight 3330-series Model 1, 2, or 11 drives can be configured. Model 1 of 3830 Storage Control can handle one string of from one to eight 3330-series drives. Model 2 of 3830 Storage Control can handle from one to four strings of from two to eight 3330-series drives each. The 3880 can have four 3330-series strings attached to each storage director.

3340 Direct Access Storage (not attachable to Models 50 and 65) can be attached to the block multiplexer channels in a 4341 Processor via 3830 Storage Control Model 2 and 3880 Storage Control. The 3340 facility is intermediate-capacity direct access storage that, because of its unique design and advanced technology, offers advantages over 2314 disk storage in addition to those provided by 3330-series disk storage. Automatic error correction features and multiple requesting are standard. Rotational position sensing is optional.

The storage medium for the 3340 is the removable interchangeable 3348 Data Module, a sealed cartridge that is never opened by the operator. In addition to the disks on which data is written, the 3348 Data Module contains a spindle, access arms, and read/write heads. The 3340 disk storage drive contains the mechanical and electrical components required to operate the 3348 Data Module.

The 3340 facility has an 885-KB/sec data transfer rate, average seek time of 25 ms, and full rotation time of 20.2 ms. A 3348 Data Module has a maximum capacity of approximately 35 million bytes or 70 million bytes, depending on the model. One model of the 3348 offers fixed heads for zero seek time to approximately 502,000 bytes maximum and movable heads for an average seek time of 25 ms to the remaining bytes in the data module.

A string of from two to eight 3340 drives can be configured. From one to four strings can be attached to the 3830 Model 2 or to a storage director in 3880 Storage Control. Any model of the 3348 can be mounted on a 3340 drive. Therefore, 3340 string capacity can vary from 70 million to 560 million bytes in increments of 35 and/or 70 million bytes.

The sealed cartridge design of the 3340 facility offers the advantages of multiple capacities per 3340 drive, increased data reliability over other removable recording media, such as disk packs, and simplified data medium loading and unloading.

3344 Direct Access Storage (not attachable to Models 50 and 65) can be attached to a 4341 Processor via 3830 Storage Control Model 2 and 3880 Storage Control. It offers significantly increased maximum

online capacity per drive for 3340 users without the necessity of program conversion. The 3344 is fixed-media disk storage. Data is recorded on nonremovable disks. The 3344 is designed to eliminate operator handling, eliminate exposure to external contamination (like the 3348 Data Module), and provide high reliability.

The 3344 has the same data transfer rate, average seek time, and full rotation time as the 3340. However, the maximum capacity of a 3344 drive is 280 megabytes, or the equivalent of four 70-million-byte 3348 Data Modules. The 3344 is a two-drive unit that attaches to the 3340 Model A2. A 3340/3344 string can contain any mixture of 3344 and 3340 units (as long as the first is a 3340 Model A2) for a maximum of eight drives with a maximum capacity of over 1.8 billion bytes.

Automatic error correction, rotational position sensing, and multiple requesting are standard in the 3344. Models are also available that contain fixed heads for zero access time to a portion of the data and movable heads for access to the balance of the data.

3350 Direct Access Storage (not attachable to Models 50 and 65) can be attached to a 4341 Processor via 3830 Storage Control Model 2 and 3880 Storage Control. The 3350 is very large-capacity, high-speed, fixed-media direct access storage. Data is stored on nonremovable disks. The 3350, like the 3344, is designed to eliminate operator handling, eliminate exposure to external contamination, and provide high reliability.

The 3350 has a data transfer rate of 1198 KB/sec, average seek time of 25 ms, and full rotation time of 16.8 ms. A 3350 drive operating in native mode has a maximum capacity of 317.5 megabytes. A 3350 string can contain from two to eight drives in two-drive increments for a maximum string capacity of over 2.5 billion bytes of online disk storage. Four 3350 strings can be attached to a 3830 Model 2 or each storage director in a 3880.

The standard Selective Format feature enables the format of each 3350 to be set during volume initialization. A 3350 drive can operate in 3350 native mode, 3330 Model 1 compatibility mode, or 3330 Model 11 compatibility mode. When operating in 3330 Model 1 compatibility mode, a 3350 drive is the equivalent of two 3330 Model 1 drives in capacity. When operating in 3330 Model 11 compatibility mode, a 3350 drive is the equivalent of one 3330 Model 11 drive in capacity. This feature enables an installation that first utilizes 3330-series disk storage to move to 3350 disk storage and obtain the price performance and functional advantages of the 3350 without program conversion.

Automatic error correction, rotational position sensing, and multiple requesting features are standard for the 3350. The 3350 is also available in fixed head models. These models provide fixed heads for zero access time to a portion of the data and movable heads for access to the balance of the data.

The 2305 Fixed Head Storage facility Model 2 can be connected to 4341 Processor block multiplexer channels via 2835 Storage Control to provide faster access to data than other disk devices for the 4341 Processor. Two disk storage modules can be contained in a facility. The Model 2 has a 1.5-MB/sec data rate, a maximum single-module capacity of 11.2 million bytes, and an average access time of 5 ms.

- The technology implemented in the 4341 Processor for logic and processor storage (which is not utilized in any System/360 or System/370 processor) provides significantly increased reliability,

greatly reduced space requirements, reduced power and cooling requirements, reduced cost, and reduced maintenance costs (see discussion in Section 05:05).

- Extensive hardware and programming systems error recovery and repair features for 4341 Processor hardware errors, not implemented for System/360, are provided to improve system availability and serviceability. These features also include facilities not implemented in most System/370 processors, such as processor diagnosis of logout data after a hardware error occurs to generate a reference code that identifies the field replaceable unit or the procedure to follow to attempt to locate the malfunction.

  In addition, inquiry into a remote data bank by the on-site customer engineer and remote diagnosis of hardware failures by IBM support center personnel are supported via the optional, no-charge Remote Support Facility. Remote maintenance facilities are not provided for System/370 Models 125 to 145 or System/360 processors.

As the highlights indicate, the 4341 Processor offers Model 50 and 65 users a wide variety of improved features and additional functions that provide improved throughput and expanded capabilities. The 4341 Processor contains many facilities that for System/370 are available only in large-scale processors. These facilities include an eight-byte-wide arithmetic logic unit, an eight-byte-wide data path, a high-speed buffer, the Remote Support Facility, microcode-controlled power sequencing, and microcode-controlled voltage and temperature monitoring.

Specifically, the 4341 Processor offers the following advantages, in addition to significantly better internal performance and price performance, when compared with Models 50 and 65.

Larger, Faster Processor Storage

Processor storage sizes of 2048K and 4096K are provided. The Model 50 can have a maximum of 512K and the Model 65 a maximum of 1024K. The effective access time to processor storage in the 4341 Processor is faster than that of Models 50 and 65 (two microseconds for four bytes for the Model 50 and 750 nanoseconds for eight bytes for the Model 65 versus 225 nanoseconds for eight bytes for the 4341 Processor). The effective access time to processor storage in the 4341 Processor is lower because of the implementation of a high-speed buffer. This improved access time increases internal performance and permits faster I/O devices to be attached to the 4341 Processor.

The availability of two processor storage sizes for the 4341 Processor, instead of multiple sizes, simplifies decisions regarding the amount of processor storage to install. The operating system to be used and applications to be installed can be selected primarily on the basis of their advantages, with less regard for their processor storage costs.

Additional processor storage can contribute significantly to system capabilities and performance. Specifically, the availability of larger processor storage for the 4341 Processor provides the ability to:

- Support more virtual storage
- Execute more or larger jobs concurrently, including new application and integrated emulator jobs
- Add and expand applications, such as graphics, teleprocessing, time sharing, and data base, that require larger amounts of storage

- Use higher level language translators and linkage editors that provide more functions and execute faster
- Execute larger processing programs without the necessity of overlay structures
- Allocate more storage to language translators and sorts to improve their execution speed
- Use more and larger I/O areas to speed up input/output operations and optimize use of direct access storage and tape media space
- Include operating system options that improve control program performance and support additional functions

Support of a Virtual Storage Environment

While the 4341 Processor has significantly more processor storage than its comparable-scale System/360 processors, it is specifically designed to support a virtual storage environment, which allows programmers to write and execute programs that are larger than the processor storage available to them. When virtual storage is supported, many of the restraints normally imposed by the amount of processor storage available in a system are eased. The removal of certain restraints can enable applications to be installed more easily and can be valuable in the installation and operation of online applications.

While many of the new hardware features and I/O devices for the 4341 Processor and the new facilities supported by operating systems that support the 4341 Processor are designed to improve system performance, a virtual storage environment provides new functions that can help improve the productivity of data processing personnel and enhance the operational flexibility of the installation (see discussion in Section 15:05).

Greatly Expanded I/O Capabilities

The fast internal performance of the 4341 Processor, together with the expanded use of multiprogramming, requires that more data be available faster than on Models 50 and 65. The 4341 Processor supports more and faster concurrent high-speed I/O operations than Models 50 and 65. It also provides the block multiplexing capability, which is not available for Models 50 and 65.

The I/O features of the 4341 Processor provide:

- Two more channels than are available for the Model 50 (six versus four) and the availability of two byte multiplexer channels
- Attachment of the 3505 reader and 3525 punch, not attachable to Models 50 and 65
- Attachment of a larger variety of printers (3203 Model 5 Printer and 3800 Printing Subsystem in addition to 1403 and 3211 Printers)
- Low-cost attachment of up to three 3278 Model 2A displays and/or 3287 Model 1 or 2 printers
- Support of synchronous data link control communications for remote units attached via the 3704/3705 Communications Controllers
- Attachment of high-speed, high-capacity, direct access devices, such as 3370, 3330-series, 3340/3344, 3350, and the 2305 Model 2 (not attachable to Models 50 and 65). The 4341 Processor can have

significantly lower cost online disk storage attached than Models 50 and 65.

- Attachment of higher speed tape units, such as the 3420 Models 6 and 8, which do not attach to the Model 50. In addition, the 4341 Processor can have more higher speed tape units attached. The Model 50 can have 3420 Model 4 tape units attached to only one channel. The 4341 Processor is not limited to having 3420 Model 4, 6, and 8 units attached to only one channel.

- Potential increases in channel throughput via use of block multiplexing and rotational position sensing to improve effective data transfer rates

- A significantly higher attainable aggregate I/O data rate than Models 50 and 65 to balance the higher performance capabilities of the 4341 Processor (maximum 9 MB/sec for the 4341 Processor with five block multiplexer channels versus 1 MB/sec and 4 MB/sec for Models 50 and 65, respectively)

| Improved Physical Characteristics

| The physical characteristics of 4341 Processor configurations are
| significantly improved over those of Model 50 and 65 configurations as a
| result of the use of new technology for logic and storage in the 4341
| Processor and new design and technology in 3370 disk storage. Although
| the 4341 Processor provides significantly more function and processor
| storage than Models 50 and 65, it requires significantly less space,
| power, and cooling than 2050 and 2065 Processing Units. Detailed
| comparisons of physical requirements are given at the end of Section
| 05:05. Similarly, space, power, and cooling requirements for the 3370
| are significantly less than for 2314/2319 and 33XX CKD direct access
| devices that attach to the 4341 Processor.

Summary

Since hardware features and programming systems support for the 4341 Processor are upward compatible with those of System/360, the 4341 Processor offers Model 50 and 65 users significantly expanded computing capabilities without the necessity of a large conversion effort. Little or no time need be spent modifying operational System/360 or System/370 problem programs or the 1400 programs currently being emulated.

Existing processor-bound System/360 programs can execute faster in a 4341 Processor because of the significantly increased internal performance of the 4341 Processor, while I/O-bound programs can benefit from the use of more processor storage, faster channel capability, block multiplexing, and faster I/O devices. The 4341 Processor also offers economical and flexible entry into communications-based applications.

The increased power and new functions of the 4341 Processor provide the base for expanded application installation and penetration of previously marginal application areas. New application installation and transition to online operations can be easier when a virtual storage environment is implemented. The greatly improved price performance of the 4341 Processor offers the System/360 and System/370 user the opportunity to widen his data processing base for a significantly lower cost than was previously possible.

For large installations that want undisrupted growth and decentralization of their data processing facilities, the 4341 Processor provides economical and easy entry into (or expansion of) distributed

data processing operations. The following outlines the distributed application environments in which a 4341 Processor can be used:

- Distributed applications (periodic data transfer, remote job entry, and pass through). The 4341 Processor can provide powerful processing capability in a department or branch location of a larger enterprise, with telecommunications connections to one or more host processors in the data processing center of the enterprise. Host-connect applications may vary from periodic transmission of summary data between the 4341 Processor and the host system to a continuous connection offering RJE and/or pass through capabilities.

  Compatible System/370 architecture allows application processor workloads to be distributed between central systems and the 4341 Processor via RJE facilities. RJE support is provided by two DOS/VSE SNA and BSC program products, VM/370 RSCS Networking, and the OS/VS1 Host Remote Node Entry System (HRNES) Installed User Program. Pass through facilities are supported by ACF/VTAME, ACF/VTAM/MSNF, and VSE/3270 Bisynch Pass Through, allowing terminals connected through the 4341 Processor to access such host facilities as IMS or TSO, or local applications such as those using CICS.

- Distributed data applications. CICS/VS Intersystem Communications (ISC) with DL/I and IMS Multiple Systems Coupling (MSC) provide support for applications accessed from either host or node systems. Data most frequently used locally may be stored on direct access storage attached to the 4341 Processor with transaction-by-transaction access to the central host data base as needed.

- Distributed network (host plus peer coupling). An SNA or VM/370 RSCS network may be enhanced through communication directly between local or remote 4341 Processors to the host or 8100 Information Systems. With a complete network, both data and processor loads can be spread to larger processors, to distributed 8100 systems, or between 4341 Processors. Transactions from the 8100 to CICS/VS are supported by the 8100 DPPX Host Transaction Facility. DPPX also supports remote job entry to OS/VS1 RES and VM/370 RSCS Networking systems.

SECTION 05: TECHNOLOGY, ARCHITECTURE, AND PHYSICAL COMPONENTS

05:05 TECHNOLOGY

INTRODUCTION

The price performance and compact size of the 4341 Processor have been achieved in large part through the use of large-scale integrated semiconductor technologies for both logic circuitry and storage. A new IBM bipolar technology is used for logic circuitry, and SAMOS (Silicon and Aluminum Metal Oxide Semiconductor) FET (Field Effect Transistor) technology is used for 64K-bit processor storage in the 4341 Processor. A new bipolar technology is used for reloadable control storage and high-speed buffer storage in the 4341 Processor.

Just as the MST (Monolithic System Technology) logic and MOSFET (Metal Oxide Semiconductor Field Effect Transistor) processor storage technologies implemented in System/370 processors represented a major technological advance over the Solid Logic Technology (SLT) logic and core storage implemented in System/360, the large-scale integrated logic and SAMOS storage technologies implemented in 4300 Processors represent a major advance over System/370 technologies.

An advanced computer-based engineering design system, an improved manufacturing process, and denser packaging approach are utilized in the design and fabrication of logic chips and modules for the 4341 Processor, which contain several times more circuits than the MST logic chips and modules utilized in System/370 processors. The very high density of the logic chip used in the 4341 Processor, 704 circuits maximum per chip, and its packaging for the 4341 Processor, up to nine chips in a multilayer ceramic logic module, result in the following advantages of this logic over MST logic:

- Higher reliability per logic circuit
- Faster circuit speed
- Significantly reduced space requirements
- Greatly reduced cost
- Reduced power requirements and less heat dissipation. (The latter reduces the amount of cooling required.)
- Reduced maintenance costs

A 4341 Processor will have either all 64K-bit or all 16K-bit storage chips in its processor storage. The 64K-bit chip utilizes SAMOS technology and is manufactured by IBM. The 16K-bit chip is built by other manufacturers. This chip is mounted in storage modules and storage array cards that were designed by IBM and are manufactured and tested to IBM specifications. The space, power, and heat specifications for a 4341 Processor are the same whether it utilizes the 64K-bit or 16K-bit storage chip. This section discusses the SAMOS technology of the 64K-bit storage chip only.

The SAMOS technology used for the 64K-bit processor storage chip also provides greatly improved chip density compared to the processor storage chip used in most System/370 processors, which is a 2K-bit chip. The higher density of the SAMOS storage chip used in the 4341 Processor is achieved by (1) utilizing fewer elementary components to form an individual storage cell (that can contain one bit of information) than

are utilized on a 2K-bit chip for System/370 and (2) reducing the physical size of the elementary components. The smaller component size results from the SAMOS technology process that is used to produce the storage chips.

A new manufacturing facility is also used to produce SAMOS storage chips. The facility is designed to minimize the number of contaminants to which silicon wafers are exposed during the chip production process. The wafers travel through an enclosed air track production line on an air cushion. The air in this track is ultra clean, containing fewer than ten contaminant parts per ten billion parts of air. The entire process is monitored by computers (over 60 IBM System/7 processors). The new manufacturing facility is designed to improve chip yield, which reduces storage cost, and to improve the reliability of storage chips.

The higher density of the storage chip utilized, the SAMOS technology and design of the chip, and the new manufacturing facility result in the following advantages for 64K-bit chip processor storage in the 4341 Processor compared to System/370 processor storage:

- Higher reliability (lower failure rate per bit of storage)
- Significantly reduced space requirements
- Greatly reduced storage cost
- Reduced power requirements and less heat dissipation
- Reduced maintenance costs

The advantages listed above also apply to 4341 Processors with 16K-bit chips in their processor storage. The magnitude of the technological advances that the 704-circuit logic chip and SAMOS storage represent can best be illustrated by comparison to the technology utilized in System/360 and System/370. The discussion of the 704-circuit logic and SAMOS technologies will explain in detail how the advantages listed above are achieved.

SYSTEM/360 TECHNOLOGY

System/360 utilized SLT for logic circuitry and wired, discrete ferrite cores for processor storage. As shown in Figure 05.05.1, SLT circuits were implemented on 1.27-centimeter (half-inch) ceramic squares called substrates. Metallic lands on the substrate formed interconnections onto which the components were soldered. These components consisted of transistors and diodes, which were integrated on silicon chips about the size of a pinhead, and thin film resistors. An SLT chip usually contained one type of component, and several chips and resistors were needed to form one circuit. In general, an SLT substrate contained four chips and a single circuit.

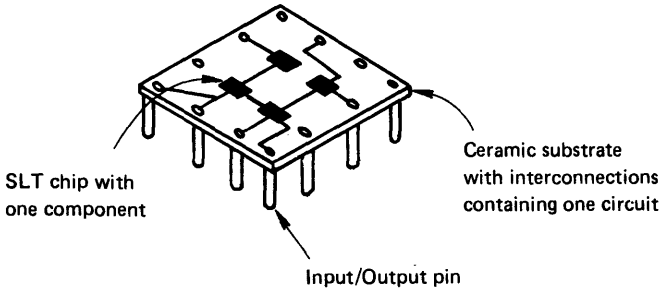


Figure 05.05.1. SLT substrate (shown four times its actual size)

SYSTEM/370 TECHNOLOGY

MST

The monolithic system technology used in System/370 was a breakaway from the hybrid circuit design concept of SLT. MST also makes use of a 1.27 by 1.27 centimeter (half-inch-square) ceramic substrate with metal interconnections onto which silicon chips are placed. However, in monolithic logic circuitry, large numbers of elementary components, such as transistors, diodes, and resistors, are integrated on a single chip.

The MST logic chip used in System/370 processors is approximately 2 millimeters (about 3/32 of an inch) square and contains over 100 elementary components, which can form up to eight interconnected circuits on the chip. This compares to a single component on an SLT chip.

Of the eight possible circuits per MST logic chip, an average of six are actually utilized in the MST logic implemented in System/370 processors. The speed of a circuit on an MST logic chip for System/370 processors is eight to twelve nanoseconds.

In MST, one logic chip is usually mounted on a substrate. (In a few instances, a substrate contains two chips.) MST logic modules, each consisting of one substrate, are mounted on circuit cards (the field replaceable unit), which are in turn mounted on circuit boards (as in SLT logic).

MST logic offers the following advantages over SLT:

- MST logic circuitry is intrinsically more reliable because many circuit connections are made on the chip, significantly reducing the number of external connections.
- Faster circuit speeds can be obtained because the path between circuits is considerably shorter.
- Space requirements for logic circuitry are greatly reduced by the significantly higher density of components per chip.
- Processor cost is reduced because the amount of wiring for interconnections and the number of modules and boards required for logic are reduced.

Monolithic Storage

Monolithic storage design incorporates the same concepts described for monolithic logic. Thus, monolithic storage, unlike core storage, can be batch fabricated. However, instead of logic circuits, storage cells that are used to contain information bits are implemented on a chip.

The processor storage chip used in most System/370 processors is a 2,048 (2K) bit chip that utilizes MOSFET technology. This monolithic storage array chip is 3.88 by 4.52 millimeters (approximately 5/32 by 6/32 of an inch) and contains a large number of interconnected circuits that form 2K storage cells and associated support circuitry on the chip. The most dense processor storage chip used in System/370 (the 3033 Processor) is a 4096 (4K) bit chip.

For System/370, two 2K-bit storage array chips are mounted on a 1.27-centimeter (half-inch) square substrate, and a pair of substrates is packaged in a storage array module to provide 8K bits per module. Storage array modules are mounted on a storage array card, which is the

field replaceable unit. In outward appearance, therefore, monolithic storage resembles monolithic logic circuitry.

The MOSFET processor storage in System/370 processors is static rather than dynamic. The differences between static and dynamic storage are discussed later in this subsection under "SAMOS Storage Technology".

The following are the general advantages of a static monolithic storage over core storage:

- Faster storage speeds can be obtained because of (1) the shorter paths between storage circuitry and (2) the nondestructive read-out capability of monolithic storage. Since core storage read-out is destructive, a regeneration cycle is required after a read and is also used prior to a write. This type of regeneration cycle is not required for static monolithic storage.

- Storage serviceability is enhanced because storage is implemented in accessible, easily replaceable cards, each of which is a functional storage component. Diagnostic routines can be written that need identify only the failing storage card, which can be replaced in a matter of minutes. Storage increments can also be field-installed more rapidly.

- Space requirements for processor storage are significantly reduced. Dense bit packaging per chip is achieved by the use of monolithic technology and by the fact that the regularity of a storage pattern lends itself to such packaging.

- Storage costs are reduced because production costs are reduced by the ability to batch fabricate processor storage.

4341 PROCESSOR TECHNOLOGY

The basic objective of large-scale integrated logic technology and high-bit-density storage technology is to bring the physical elements that make up the logic and storage in a processor physically closer together. The distances between logic circuits or storage bits can also be reduced by the packaging approach used for a logic or storage module.

When elements are brought closer together, the amount of wiring required to form the specific logic circuitry or storage size required by a processor is reduced, faster circuit speeds can be obtained, and circuit or bit reliability is improved. The failure rate of circuits and storage bits is related to the length and location of wired connections. For example, circuits connected on a chip are more reliable than circuits connected off the chip.

In the 4341 Processor, for example, wiring to accomplish the logic for the instruction processing function occurs at several levels. First, elementary components (transistors, diodes, and resistors) on a chip are connected to form circuits, which are then interconnected at the chip level. Additional circuit connections are made at the logic module level (that is, within the substrate) and at the card level.

With the use of large-scale integrated logic technology and high-bit-density storage technology, the total amount of wiring and the number of logic cards and storage cards required in the 4341 Processor are significantly reduced from that required by the technology implemented in System/370 processors. Separate logic boards and separate processor storage boards and the cabling between them are eliminated entirely for the 4341 Processor because only ten cards are required to contain the logic for

the instruction processing function in the 4341 Processor and only four or eight cards are required for processor storage.

Logic Physical Design and Advantages

The logic chip used in the 4341 Processor is 4.57 by 4.57 millimeters (approximately 3/16 of an inch square) and contains over 7000 elementary components (resistors, diodes, and transistors), as compared to over 100 on an MST logic chip approximately 2 millimeters (about 3/32 of an inch) square. The 7000 elementary components on the chip can be connected to form 704 logic circuits. The nominal speed of a circuit on this logic chip is three nanoseconds.

Figure 05.05.2 shows the size of the 704-circuit logic chip relative to a paper clip, while Figure 05.05.3 shows an MST logic chip on the left and 704-circuit logic chip on the right surrounded by salt crystals. The relative size of these chips can be seen in the latter figure.

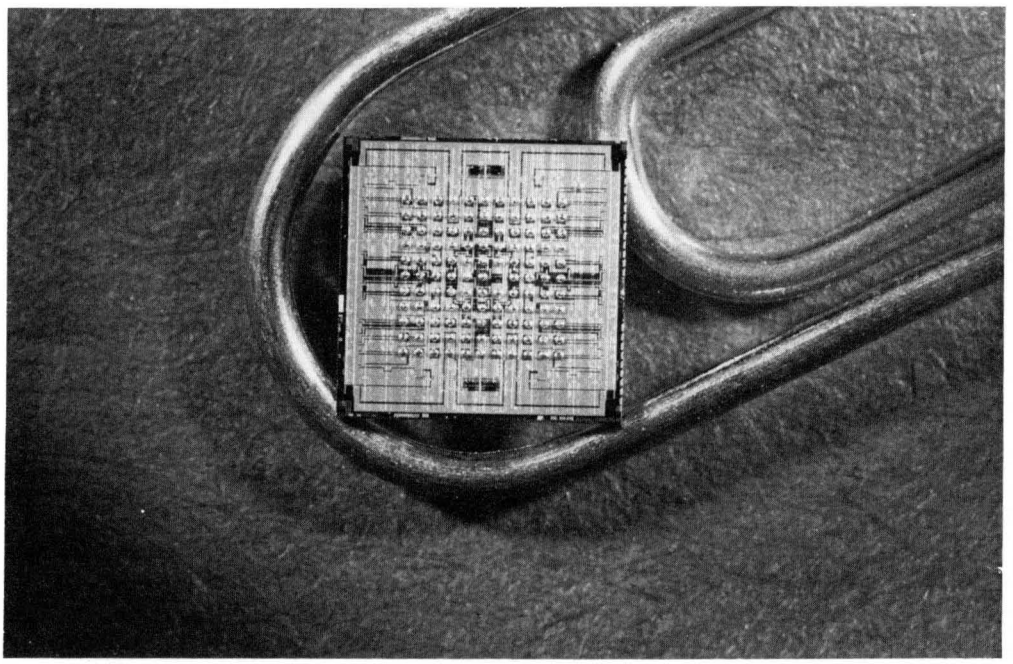


Figure 05.05.2. The 704-circuit logic chip resting on a paper clip

Of the 704 circuits available on a single chip, an average of 616 are actually utilized in the logic implemented in the 4341 Processor. The high circuit utilization is made possible in part because three layers of wiring are used for interconnections on the chip itself. A logic chip in the 4341 Processor contains up to 2.13 meters (seven feet) of wire that interconnects the elementary components and circuits on the chip. Only one layer of wiring is used on an MST logic chip.

Despite the greatly increased density of a logic chip in the 4341 Processor, a higher percentage of the circuits available on the chip are used than are used on the MST logic chip. For the 4341 Processor, average circuit utilization on a logic chip is over 87 percent, compared to 75 percent average utilization for the MST logic chip in System/370 processors.

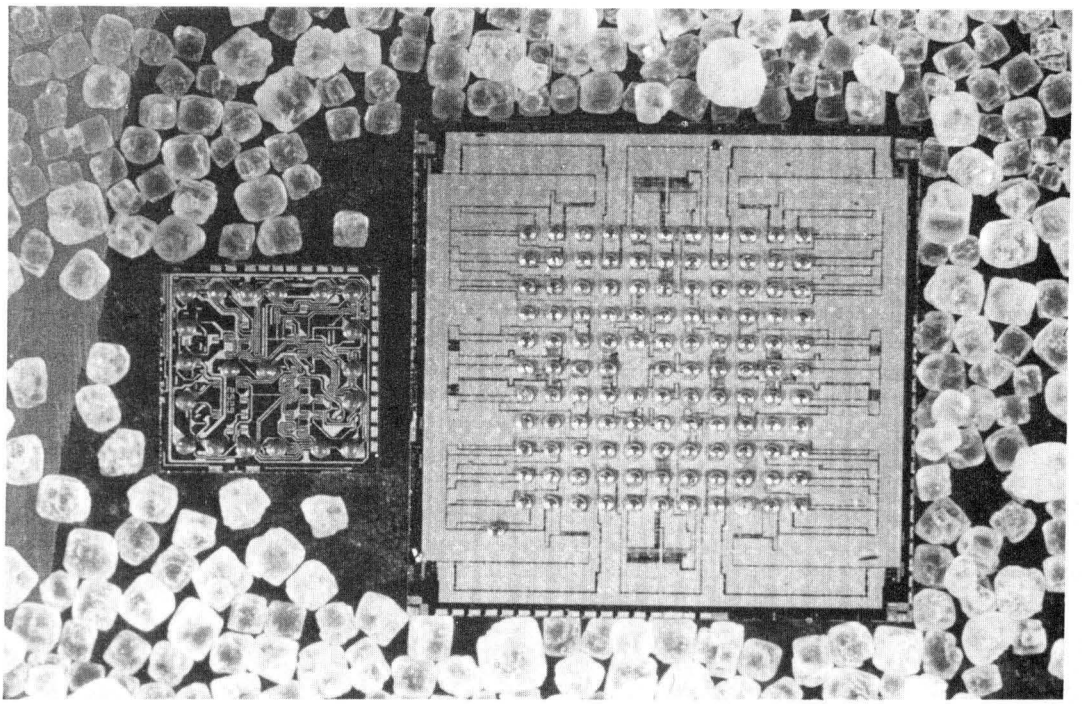


Figure 05.05.3. An MST and 704-circuit logic chip surrounded by salt crystals

In the 4341 Processor, three different sized modules are used for logic: MLC (multilayer ceramic) 50, MLC 35, and MC (metalized ceramic) 28 modules. The MLC modules have up to 23 layers of ceramic available to accommodate circuit interconnections within the ceramic itself.

The MLC 50 module is 50 by 50 millimeters (approximately 2 inches square), can contain up to nine logic and array chips, and has 361 I/O pins. The MLC 35 module is 35 by 35 millimeters (approximately 1.5 inches square), can contain up to nine logic and array chips, and has 196 I/O pins. The MC 28 module is 28 by 28 millimeters (slightly over one-inch square), contains one chip, and has 116 I/O pins. The type of module used for a specific function is selected to optimize I/O pin utilization and cost.

An MLC 50 module for the 4341 Processor has approximately 4.8 meters (15.7 feet) of wiring contained within the ceramic substrate. In MST logic, a single-layer ceramic is used and no interconnection wiring exists within the substrate for a single-chip module.

The larger ceramic module size and use of multilayer wiring within the substrate enable several logic chips to be packaged in a single logic module. For the 4341 Processor, up to nine chips are mounted on a single MLC 50 substrate, depending on the function performed by the module. On the average, six chips are contained within each MLC 50 module for the 4341 Processor. The circuits on one chip within a module are interconnected to circuits on one or more other chips via the wiring contained within the substrate.

The MLC 50 logic module used in the 4341 Processor is shown in Figure 05.05.4 mounted on a card (the FRU) without and with its cap. The substrate can accommodate surface wiring between chips. When necessary, this wiring is used to make an engineering change to an MLC 50 module at

the plant of manufacture. This capability also exists for MLC 35 modules.

MLC modules are mounted on circuit cards 11.32 by 17.46 centimeters (approximately 4.5 by 7 inches) in size. Logic cards in the 4341 Processor are mounted on a board that contains other functions in addition to instruction processing function logic.

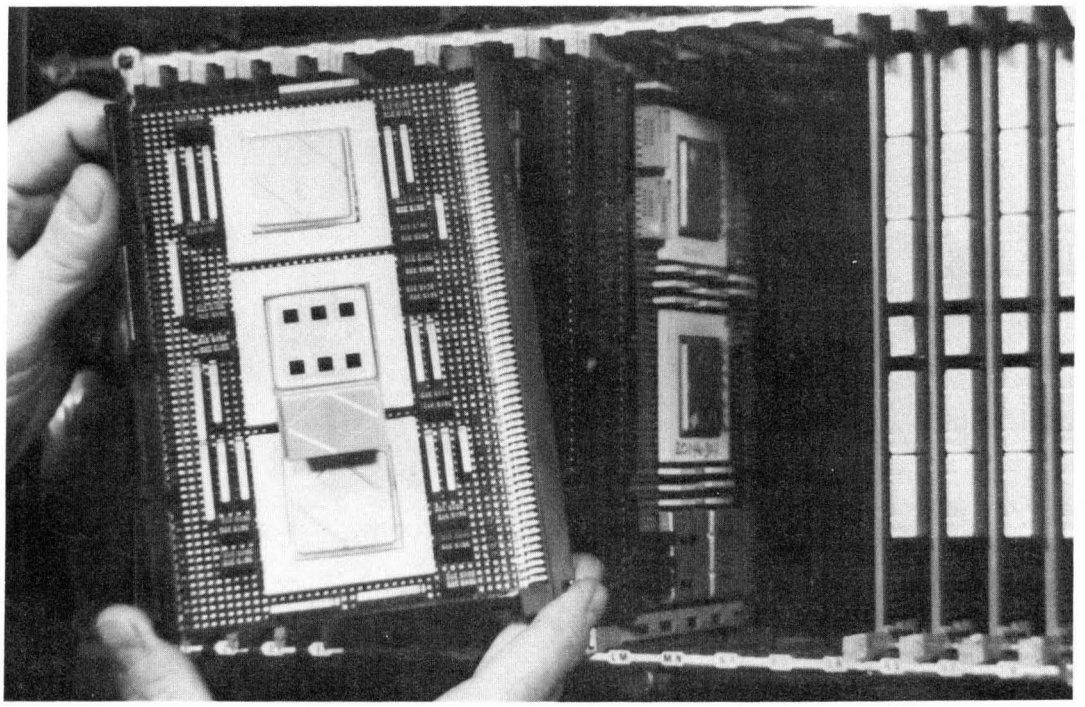


Figure 05.05.4. The MLC 50 module used for instruction processing function logic in the 4341 Processor without and with its cap

A new card and board design are used to contain MLC modules. The new design is required to accommodate the higher density of I/O pins on an MLC module (which is required to handle the higher circuit density per module). The number of pins in an MST logic module is 16.

The new card and board design also improves reliability, since each logic card is held in place by screws at each end of the card to prevent the card from coming loose.

The large space reduction that results from the use of the 704-circuit logic chip and the multichip, multilayer ceramic substrate can be seen by calculating the number of MST logic modules required to contain the same number of circuits as an MLC 50 logic module for the 4341 Processor. Assuming the maximum capacity of 704 circuits per chip is utilized and six chips on an MLC 50 logic module (approximately 4200 circuits), 700 MST logic modules (at six circuits per module) are required for the same number of circuits.

Figure 05.05.5 shows one MLC 50 module and the 700 MST logic modules that provide the equivalent number of circuits (approximately 4200). Figure 05.05.6 shows the relative space requirements for the number of MST logic cards (23) required to contain the same number of circuits as are contained on one MLC 50 module with six chips.

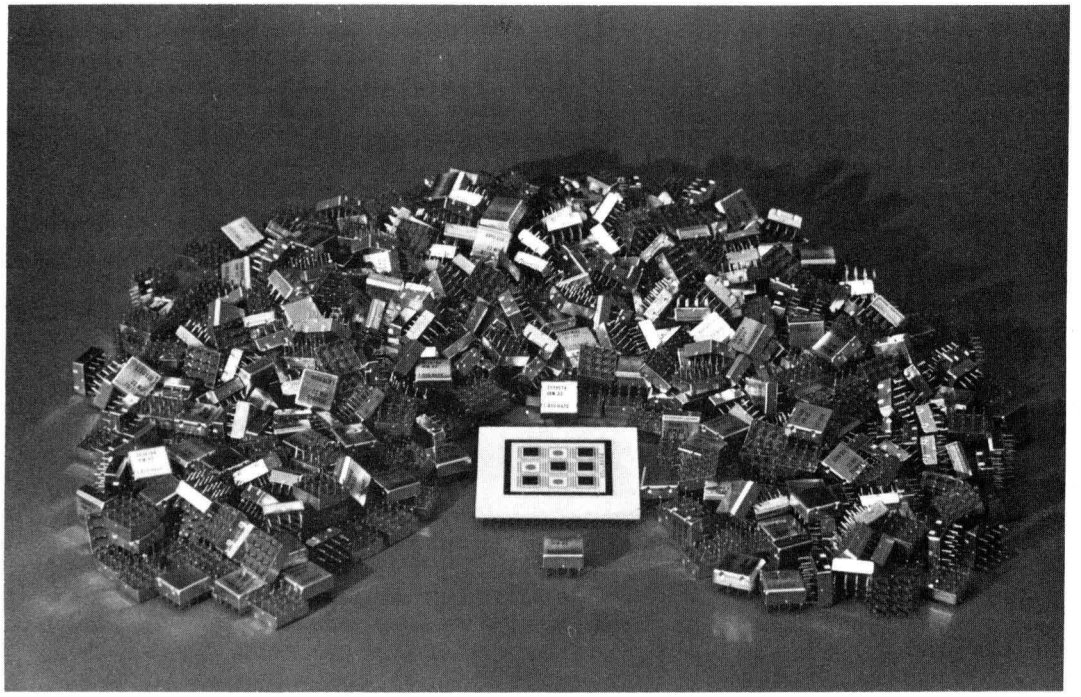


Figure 05.05.5. One MLC 50 logic module containing approximately 4200 circuits and 700 MST logic modules containing the same number of circuits

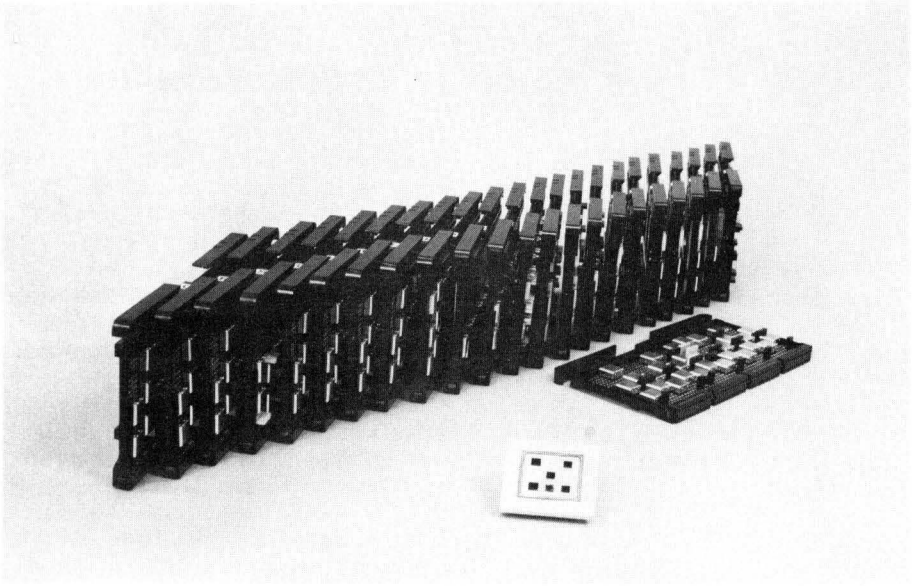


Figure 05.05.6. One MLC 50 logic module with six chips and 23 MST logic cards with the same number of circuits

The advantages of the logic technology used in the 4341 Processor over MST logic technology are the result of (1) the use of large-scale integrated technology to provide smaller elementary components on a chip and, thus, significantly increase circuit density per chip, (2) the use of three levels of wiring on the chip to increase the percentage of available circuits actually used per chip, (3) significant module packaging improvements (the use of wiring within the ceramic substrate

and multiple chips per module), (4) the use of cards with higher I/O (pin) density, and (5) the use of a higher grid density board (2.5-millimeter versus 3.18-millimeter grid density).

The density and packaging improvements bring logic circuits closer together. The higher chip density enables many more circuits to be connected at the chip level. The multiple-chip, multilayer ceramic module enables circuits to be connected at the module instead of the card level.

When only one logic chip is mounted on a single-layer ceramic substrate for MST logic modules, circuit connections are made at the chip level and, via pin soldering, at the card level. There is no circuit connection at the module level for most MST modules. Therefore, for a given number of circuits, a great deal less total wiring and shorter wire lengths are required for circuit connection when 704-circuit, instead of MST, technology is used. Specifically, approximately 305 meters (1000 feet) of wire for circuit connections are saved by the utilization of one MLC 50 logic module instead of 700 MST logic modules (equaling about a 96 percent reduction).

The reduction in the total amount of wiring required and the shorter wire lengths possible using the 704-circuit chip technology and its packaging for the 4341 Processor provide the following advantages over MST technology:

- A significant reduction in the amount of space required to contain the logic circuits of a processor. The 4341 Processor contains approximately 106,500 logic circuits while the System/370 Model 138 contains approximately 18,000. In the 4341 Processor, only 27 cards are required for the 106,500 circuits while 120 cards are required for the 18,000 circuits in the Model 138.

- A significant reduction in logic circuitry cost because physical packaging costs are so much less and production costs for the fabrication of logic circuits have been reduced compared to production costs for MST logic circuits

- Faster circuit speeds (3 to 5 nanoseconds in the 4341 Processor versus 8 to 12 nanoseconds for MST) because of a shorter signal distance (shorter wire lengths)

- Increased reliability because (1) many times the number of circuit connections are made at the chip level and (2) circuit connections are made at the module level, reducing the number of circuit connections made at the card level. Circuit connections made at the chip and module level are more reliable than those made via pin soldering at the card level. Thus, a circuit at the MLC module level is approximately 30 times more reliable than an MST circuit at the module level.

- Reduced power requirements because of the smaller physical size of the transistors on a logic chip and the shorter wire lengths. The use of less power results in a reduction in the amount of heat generated so that less cooling is required. (See comparison with System/360 and System/370 processors under "Summary" at the end of this subsection.)

- Reduced maintenance costs because of increased reliability and improved serviceability. Increased reliability should result in fewer failures in logic circuits and, thus, less maintenance time. As a result of the increase in the reliability of both logic circuits and storage bits in the 4341 Processor, no preventive maintenance is scheduled for 4341 Processors except replacement of the filters in the cooling blowers (once a year).

Serviceability is improved because the diagnostic procedures for the 4341 Processor are designed to locate the field replaceable unit more quickly. First, the use of scoping for problem determination is eliminated as the high circuit density of the logic chip precludes its use. Instead, microcoded procedures, some of which are invoked during processing at the time a failure occurs, are used to locate the failing FRU (see discussion of reference code generation in Section 50). Second, the presence of fewer logic cards in a processor makes locating the failing FRU faster.

Logic Fabrication and Design

The high circuit density and high circuit utilization of MLC 35 and 50 logic modules are made possible by improvements in the fabrication procedure that is used to produce logic chips, the new packaging design for logic modules already described, and the use of a computer-based engineering design system that automates the logic design procedure from the chip level to the board level.

The fabrication of MLC logic modules that perform specific logic functions is basically composed of three production procedures. The first procedure produces logic chips that perform specific logic functions (as required by logic designers). The second produces a multilayer substrate with the specific wiring required by specific logic chips. The third procedure combines the appropriate chips with a substrate to produce a capped logic module and tests for correct operation of the module.

The first procedure consists of two basic processing steps. The first process produces a number of identical logic chips with elementary components on a single silicon wafer. The second process personalizes the chips by interconnecting the components on each chip to form circuits that perform the specific functions desired by a logic designer.

The two processes used in the first chip production procedure are improved over those used in the fabrication of MST logic. First, the increase in the density of the logic chip is made possible by several improvements in the first process that produces logic chips on silicon wafers (improvements in process control and photolithography precision).

Second, the productivity of the first process is improved by (1) the use of larger silicon wafers than are used to produce MST logic chips (it is more productive to process one large silicon wafer than two small ones) and (2) by the fact that the circuits on the logic chip are smaller (more total circuits are produced per silicon chip). This results in less production cost per logic circuit and contributes to reduced processor cost.

The second chip production process, which connects components and circuits on a chip, utilizes electron beam direct exposure at several processing steps to connect circuits instead of the optical mask technology utilized in the production of MST logic chips. The advantages of utilizing the electron beam in the second process are the following:

- Different types of chips can be produced from the same silicon wafer. That is, logic chips of different personalities can be produced from one wafer.
- The production of new logic parts during the design of the logic for a new processor is faster because the time required for the construction of new optical masks is eliminated.

The end result of using the electron beam is reduced cost for logic circuits, which ultimately results in lower processor cost.

The procedure that produces multilayer ceramic substrates is IBM-designed. A substrate has a certain amount of predefined wiring and other wiring that personalizes the substrate to perform the functions required by the logic chips it contains. The personalized wiring is designed using the engineering design system.

The engineering design system that automates the logic design procedure was an extremely important element in the development of the logic in the 4341 Processor. This system makes it practical to utilize the high circuit density of the logic chip and offers several advantages to the logic designer.

Before implementation of the engineering design system, the physical wiring patterns required to connect the components on chips, cards, and boards to perform specific logic functions were designed manually with the help of display units. That is, the logic designer determined not only the logical interconnections, but also the physical wiring patterns required.

The engineering design system, which has been in continual development within IBM for some time, is a generalized logic design system that is now utilized throughout the entire IBM corporation. This system is programmed to handle various technologies and enables the technology designer and processor logic designer to operate independently of one another.

Prior to the existence of the technology used for the 704-circuit chip, the engineering design system was used to design wiring at the chip level (for the densities available) and at the card level for the logic used in System/370 processors. The design system has been updated to handle a chip with a 704-circuit density and to design wiring at all physical levels: chip, module, card, and board. The design system is now capable of producing all the physical wiring design required to implement the logic for a given processor. The design system is also improved in that it automatically generates the test patterns required to check the finished logic chips for proper electrical connection.

While the design system eliminates the need to manually design physical circuit wiring at all levels, it also provides the logic designer with the capability of manually intervening in the design process, if necessary.

For logic that utilizes the 704-circuit chip, the physical wiring patterns required to connect circuits from the chip to the board level to perform specific logic functions are designed by the engineering design system using input from the logic designer, who determines the logical interconnections. In addition, the design system utilizes a master slice and open part number approach that offers the advantages of greater logic design flexibility, quick verification of logic design, and more rapid logic design completion.

The first process in the fabrication of the 704-circuit logic chip produces a master slice with a specific part number. Optical masks are used in this process. A master slice is a single silicon wafer that contains several logic chip areas, each of which contains the identical optically defined configuration of elementary components. The chips on a master slice are designed by the technology designer.

Each chip on the master slice used for the logic in the 4341 Processor contains over 7000 components, which can be connected during the second chip fabrication process to form up to 704 circuits. This 704-circuit chip master slice is used in the production of logic chips

for 4341 and 4331 Processors and the IBM System/38 processor. Master slices are produced in quantity and stored as inventory until required by logic designers and for processor production. Having a master slice available with the components already existing on the chip speeds up the logic design process.

When circuits with specific logic functions are required during the design of a new processor, the logic designer utilizes the engineering design system to personalize a chip on a master slice to perform the desired functions. This personalized chip is assigned a unique part number. The logic designer provides the design system with a description of the logical functions the chip is to perform and can request almost any desired interconnection of all or part of the circuits available on the chip (704 in the case of the chip for the 4341 Processor).

The master slice and open part number approach are not utilized for MST logic. For MST, each chip that performed a unique logic function was assigned a part number. A certain number of part-numbered chips were designed and logic designers had to use this set of logic chips to develop the logic for a specific processor.

The logic chip design input supplied to the engineering design system is first checked for possible design rule violations (exceeding the chip circuit capacity, for example). The design system then designs the physical chip wiring required to accomplish the desired logic functions. The resulting wiring patterns minimize the amount of wiring required. The design system also attempts to maximize utilization of the number of circuits available on the chip. In some cases, the logic designer may have to manually design some physical wiring on the chip to maximize circuit utilization.

Once the chip design is complete, the design system identifies the electron beam and optical mask patterns needed to manufacture the specific logic chip and generates the data required to test the logic. The output from the engineering design system is a magnetic tape that is used to control the two processes that produce logic chips for a given processor. The tape also contains the automatically generated test patterns.

Once the logic chips required for a specific function are designed, the engineering design system is used to design the physical wiring required at each successive level--substrate, card, and board--based on the logic designer's input of logical connections.

SAMOS Storage Technology

Dynamic storage design. The 64K-bit chip processor storage in the 4341 Processor is a dynamic type of monolithic storage, as opposed to the static type of storage that is implemented in System/370 processors. The 64K (65,536) bit SAMOS chip utilizes one transistor per storage cell. Processor storage in the 4341 Processor is sometimes referred to as a single-cell storage. The SAMOS 64K-bit storage chip (shown in Figure 05.05.7 resting on a coin) is 6.35 by 6.35 millimeters (approximately one-quarter of an inch square) in size. Six transistors per bit are used on a 2K-bit storage chip for a System/370, which is 3.88 by 4.52 millimeters (approximately 5/32 by 6/32 of an inch) in size. The 4K-bit storage chip in the 3033 Processor utilizes four transistors per bit.

For the static 2K-bit chip storage implemented in System/370, the six transistors form a circuit. In effect, the circuit is a switch that can be in one of two states: on or off. Current is supplied continuously

to static storage cells while processor power is on. For the dynamic 64K-bit chip storage implemented in the 4341 Processor, a storage cell is implemented as one capacitor onto which a charge is stored to reflect a bit on or off condition. Current is supplied to a dynamic storage cell when a bit is read or written, rather than continuously.

For any given dynamic storage, the charges will remain on the capacitors only for a specific time interval, after which current must be resupplied to maintain the stored data. The periodic supply of current to a dynamic storage to maintain its contents is called "refreshing".

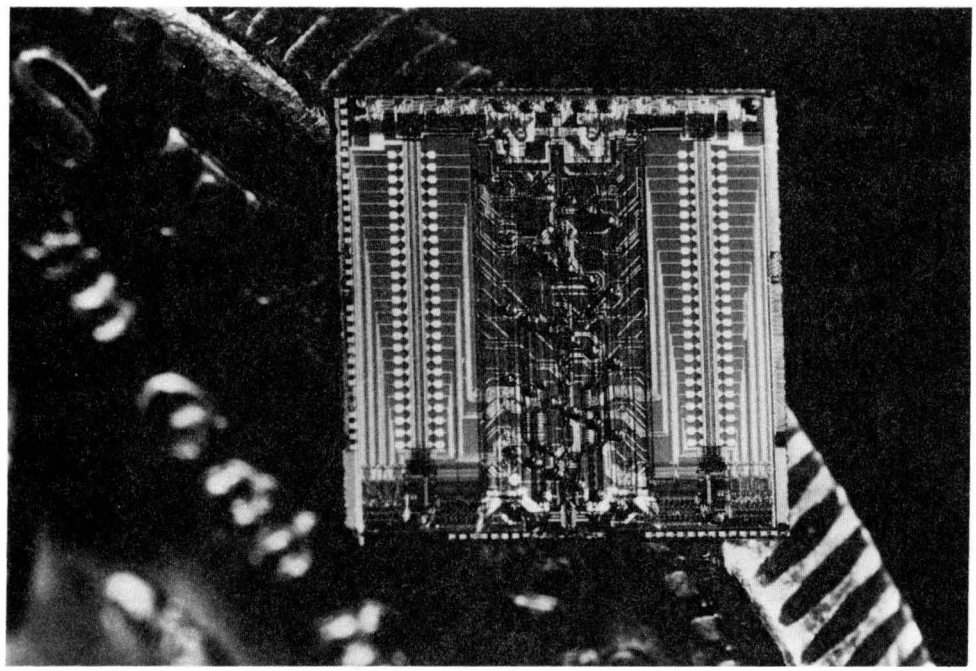


Figure 05.05.7. The 64K-bit storage chip resting on a coin

A bit in processor storage in the 4341 Processor must be refreshed every 6.1 milliseconds. In the 4341 Processor, all processor storage is not refreshed at the same time. A portion of the storage is refreshed every 24 microseconds, such that in every 6.1-millisecond interval all processor storage is refreshed.

Refreshing occurs as follows. In one refresh operation, 256 bits on every chip in one-half the processor storage installed are refreshed. This refresh operation requires 890 nanoseconds.

In the next refresh operation, which occurs 24 microseconds after the first, 256 bits on every chip in the other half of processor storage are refreshed. The next refresh operation 24 microseconds later causes a different set of 256 bits in the first half of processor storage to be refreshed, and so on, until every 256-bit group on every storage chip is refreshed during a 6.1 millisecond interval.

Processor storage cannot be accessed during a refresh operation. However, refreshing requires only 890 nanoseconds out of every 24 microseconds (24,000 nanoseconds). Thus, processor storage is unavailable less than four percent of the time. Instruction processing function operations are delayed very infrequently by storage refreshing in the 4341 Processor, since the instruction processing function works primarily with the high-speed buffer.

Continuous power is required to maintain a one or zero state in a static type of monolithic storage cell and periodic power is required to maintain a one or zero state in a dynamic monolithic storage cell. Thus, data in a monolithic storage is lost when power is turned off in a processor. Monolithic storage is therefore said to be volatile. This is not true of core storage, which retains a magnetized state when power is removed.

The advantages of dynamic monolithic storage over static monolithic storage are significantly increased bit capacity per chip and reduced power and cooling requirements. The bit capacity of a chip is improved by the use of fewer components per cell and the power utilization is less because current is not supplied continuously to each storage cell.

The use of less power results in less heat generation and the need for less cooling. Reduced heat generation also aids storage reliability, since the operation of electronic devices can eventually be impaired by continuous exposure to heat.

Processor storage. The 64K-bit chip used in processor storage in the 4341 Processor provides several advantages as a result of its technology, design features, and high density. The 64K-bit chip contains three functional areas: two storage array areas separated by a support area. Each storage area contains 32K usable bits plus additional (redundant) bits.

The redundant bits (approximately 2500 per chip) are provided to enable a chip to be utilized even though some bits within a storage area do not function properly. During the final testing of a 64K-bit chip, functional redundant bits are substituted for any nonfunctional bits via the programming of on-chip circuitry in the support area of the chip.

The two storage array areas occupy approximately 30 percent of the total surface of the storage chip. The area between the two storage arrays on a 64K-bit chip provides normal storage support functions (chip timing and addressing functions, for example) and two registers that speed up reading of storage bits.

Each of the two storage areas has its own dedicated support circuity and register in the support area of the chip. Thus, a 64K-bit chip is usable even if only one of the two storage arrays and its associated circuitry are functional, since each storage area can operate independently from the other. The use of two independent storage arrays per chip and redundant storage bits on the chip improves chip yield, which results in reduced processor storage cost.

The two high-speed, eight-bit registers, one for each storage array, that are implemented in the support area are provided to reduce the time required to access bits in the storage arrays, which are contained in low-speed areas on the chip. Eight bits from a storage array area can be placed in its associated register in 405 nanoseconds. A bit can then be read out of the register in 100 nanoseconds. Thus, 1205 nanoseconds are required to access eight sequential bits. Without implemention of the eight-bit register, 405 nanoseconds would be required to access the first bit and 1 microsecond would be required to access each additional bit, for a total of 7.4 microseconds for eight sequential bits.

The implementation of storage support circuitry on the chip in the support area, rather than off the chip, aids storage reliability. The high density of the chip also aids reliability, since many more circuit connections are made at the chip level than for the 2K-bit storage chip used in System/370.

The reliability of the 64K-bit chip is also improved by the SAMOS technology. The technology is designed to protect the storage cells on

a chip from contamination and minimize the potential for charge leakage after data is stored.

First, the chip has a double layer of insulation to help ensure good coverage of the chip surface (one layer of oxide covered by one layer of silicon nitride instead of a single oxide layer). Second, the chip utilizes a layer of polysilicon that acts as a surface field shield. The shield is designed to provide the very low current leakage level required by the high density of the chip.

Space requirements for processor storage in the 4341 Processor are greatly reduced because of the denser packaging of storage array modules as well as because of the high density of the storage chip. A storage array module for the 4341 Processor is 2.54-centimeters (one-inch) square and contains one or two substrates. A substrate contains four storage chips. The use of one or two substrates per module enables one or two storage array areas per chip to be utilized.

A storage array module for the 4341 Processor contains either (1) two substrates with eight chips containing 16 storage arrays or eight chips containing 8 storage arrays or (2) one substrate with four chips containing eight storage arrays. Thus, a storage module contains 256K or 512K bits.

A storage array module for the 2K-bit processor storage chip used in System/370 processors contains two substrates, each of which contains two storage chips (four chips per module that provide 8K bits). Thus, a storage array module for the 4341 Processor is either 32 or 64 times as dense as a storage array module with 2K-bit chips for System/370 processors.

The processor storage array card for the 4341 Processor, which like the logic card is 11.32 by 17.46 centimeters (approximately 4.5 by 7 inches) in size, contains a mixture of the types of storage array modules described and provides 512K bytes of processor storage. Thus, a 4341 Processor contains four or eight processor storage cards for 2048K or 4096K bytes of processor storage, respectively. A processor storage card containing 512K bytes is shown in Figuire 05.05.8.

When the 2K-bit chip is used for processor storage in System/370, 36 cards 10.8 by 17.78 centimeters (4.25 by 7 inches) in size are required for each 1024K bytes. The volume of space required for each 1024K bytes of processor storage implemented in 2K-bit chips is 30 times greater than the requirement for each 1024K bytes of processor storage implemented in 64K-bit chips or 14,158 cubic centimeters versus 475 cubic centimeters (864 cubic inches versus 29 cubic inches). The large reduction in the number of storage cards required for a given size makes locating the failing FRU faster.

The new card and board design used for logic in the 4341 Processor is also used for processor storage. Therefore, processor storage array cards in the 4341 Processor are held in place by screws at each end to prevent them from coming loose and, thus, aid reliability.

The reduction in space requirements that results from the use of large-scale integrated logic technology and high-density SAMOS storage technology in the 4341 Processor is illustrated in Figure 05.05.9. This figure shows one megabyte of 2K-bit-chip processor storage and the logic for the Model 138 and two megabytes of 64K-bit-chip processor storage and the logic for the 4341 Processor.

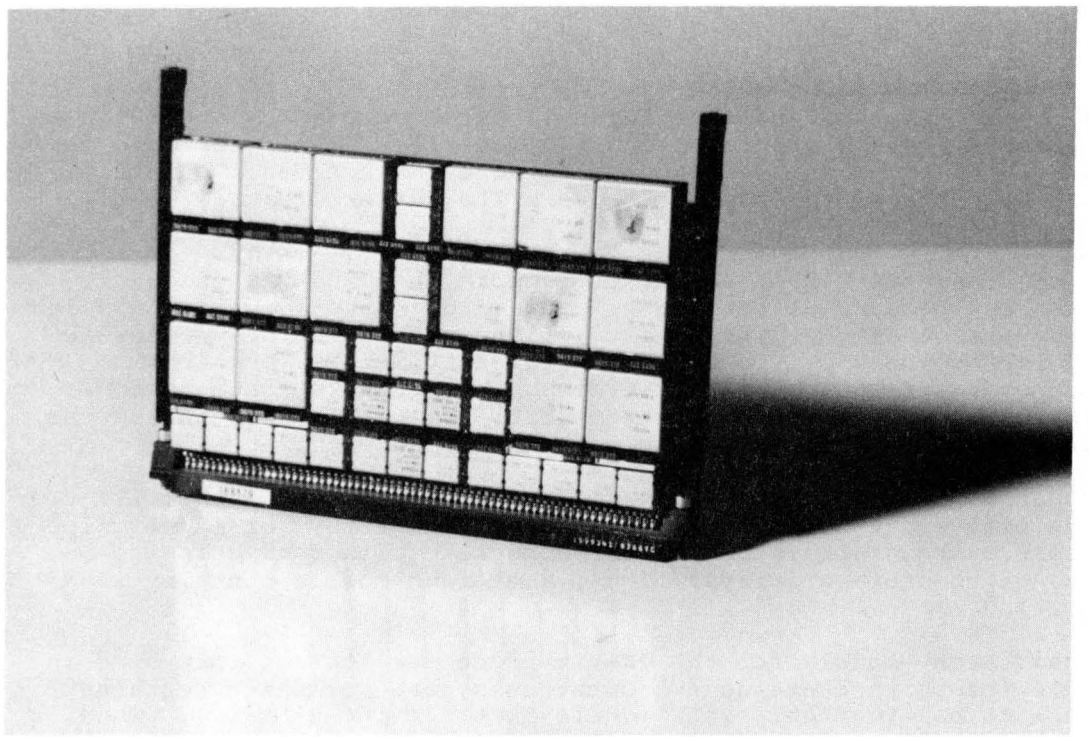


Figure 05.05.8. A processor storage card containing 512K bytes

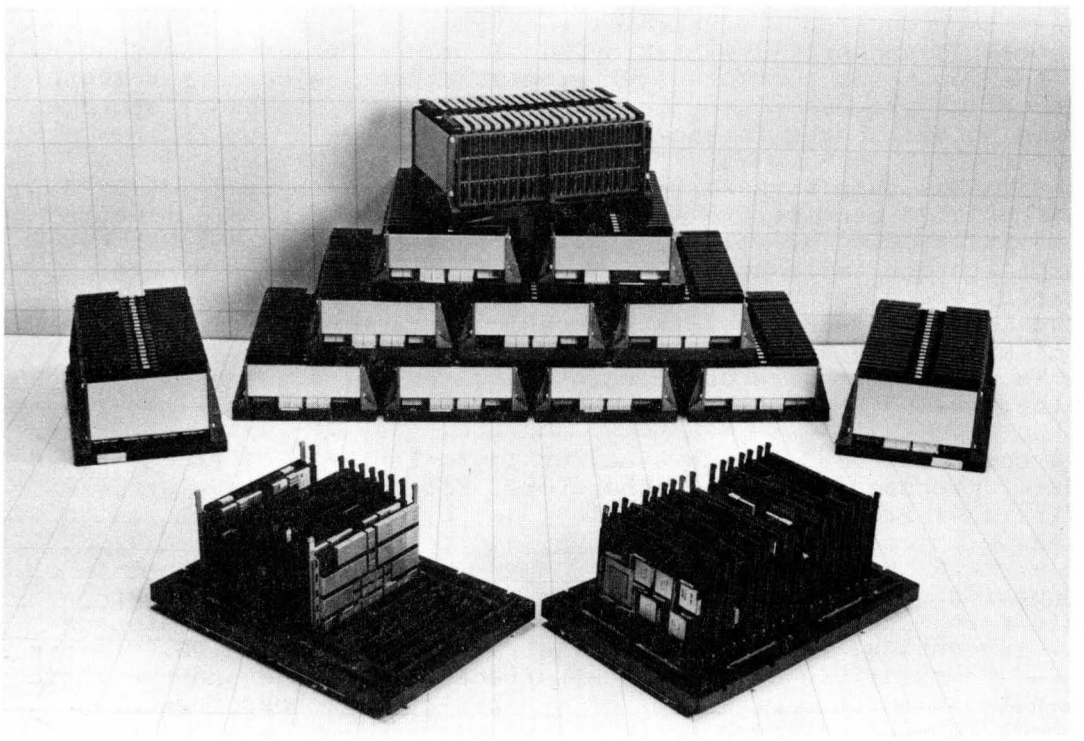


Figure 05.05.9. Logic, control storage, and one megabyte of processor storage for the Model 138 and logic, control storage, and two megabytes of processor storage for the 4341 Processor

Control storage and buffer storage. A new high-performance static bipolar technology is used for the control storage and buffer storage in the 4341 Processor to provide the faster access required by these types of storage. The capacity of the bipolar chip is 256 bytes (2304 bits), which is more than twice as dense as the chip used for control storage in Models 138 and 148, which contains 1024 bits per chip. The new chip also contains functions that are designed to increase its access speed. Access time at the card level for the control storage in the 4341 Processor is 55 nanoseconds, while the control storage in Models 138 and 148 has a 75-nanosecond access time at the card level.

For the high-speed buffer, two chips are placed on a 2.54-centimeter (one-inch) square substrate and there is one substrate in a module. Two cards are required for the 8K-byte buffer storage in the 4341 Processor. For control storage, two chips are placed on a substrate but a module contains two substrates. Four cards are required for control storage in the 4341 Processor.

Summary

The significant impact made on the physical and environmental characteristics of the 4341 Processor by the use of large-scale integrated logic technology and a 64K-bit or 16K-bit storage chip can best be shown by comparison with the same characteristics of other appropriate System/360 and System/370 processors.

The 4341 Processor requires significantly less power than System/360 Models 50 and 65 despite its greatly increased function and larger processor storage sizes. Specifically, a 4341 Processor with 4096K bytes of processor storage and attached 3278 Model 2A console and 3287 Printer requires approximately 20 percent the amount of power (measured in kilovolt amps) as a Model 65 processor (2065 Processing Unit) with 1024K of processor storage and approximately 63 percent the amount of power as a Model 50 processor (2050 Processing Unit) with 512K bytes of processor storage.

The power requirement for a 4341 Processor with 4096K bytes of processor storage and attached 3278 Model 2A and 3287 is approximately 45 percent of that for a System/370 Model 138 processor (3138 Processing Unit) with 1024K bytes of processor storage and less than one-third that of a System/370 Model 148 (3148 Processing Unit) with 2048K bytes.

The reduction in power requirements for a 4341 Processor causes it to dissipate less heat, as measured in BTUs per hour. The 4341 Processor with 4096K bytes and attached 3278 Model 2A and 3287 generates approximately 22 percent the amount of heat as a Model 65 with 1024K, 60 percent the amount of heat as a Model 50 with 512K, 40 percent the amount of heat as a Model 138 with 1024K, and 28 percent the amount of heat as a Model 148 with 2048K. Thus, less air conditioning is required for a 4341 Processor.

Space savings in square feet of floor space also result when a 4341 Processor replaces a Model 50 or 65. A 4341 Processor with 4096K bytes of processor storage (not including the 3278 Model 2A, 3287, or a Channel-to-Channel Adapter) requires 88 percent less space than a Model 65 with 1024K bytes and three channels and 45 percent less space than a Model 50 with 512K bytes.

A 4341 Processor with 4096K bytes (without a 3278 console, 3287 Printer, or Channel-to-Channel Adapter) requires 84 percent the amount of space as a Model 138 with 1024K bytes and 56 percent the amount of space required by a Model 148 with 2048K bytes. Note also that the height of the 4341 Processor is one meter (39.36 inches) or about two-thirds that of a Model 138/148 processor.

05:10 DESIGN OBJECTIVES AND ARCHITECTURES

DESIGN OBJECTIVES

The basic design objectives embodied in the 4341 Processor provide System/360 users with a growth system in the intermediate-system range that incorporates several improvements and additions to System/360 architecture. The 4341 Processor provides System/360 users with many new functional capabilities, significant performance improvements, and features to enhance system availability and serviceability. Availability and serviceability improvements and some new function is also provided for System/370 users. This progress has been achieved under the following conditions:

- The architecture implemented in 4300 Processors is upward compatible with that of System/360 so that most user-written problem programs for System/360 Models 22 and up will operate without modification in a 4341 Processor with either System/370 or ECPS:VSE mode in effect.
- The architecture implemented in 4300 Processors is upward compatible with that of System/370 so that most user-written problem programs for System/370 processors will operate without modification in a 4341 Processor with either System/370 or ECPS:VSE mode in effect.
- Programming systems support of the 4341 Processor is based on certain operating systems that support System/370. These System/370 operating systems were developed using System/360 operating systems as a base, namely, DOS Version 4 and OS MFT, to provide upward compatibility for control programs when System/370 mode is utilized.
- Most System/360 and System/370 I/O devices can be used in a 4341 Processor configuration.

ARCHITECTURES IMPLEMENTED

Two architectures are implemented in the 4341 Processor: System/370 and 4300 Processor architecture. The mode of processor operation selected during an initial microcode load (IML) of the 4341 Processor determines the architecture that is functional. When System/370 mode is selected, System/370 architecture with certain modifications, is functional. When ECPS:VSE mode is selected, 4300 Processor architecture is functional.

System/370 Architecture

System/370 architecture is an extension of System/360 architecture. The following are the more significant facilities that are implemented in System/370 but not System/360 processors:

- Extended control mode of operation to support new facilities (such as dynamic address translation and program event recording), as well as a System/360-compatible basic control mode of operation. ASCII mode is not implemented in System/370.
- Sixteen control registers to enable and disable and control the operation of new facilities
- Expanded instruction set, including many additional general purpose and processor control instructions
- Additional hardware timing facilities (time-of-day clock, CPU timer, and clock comparator) and a higher resolution for the interval timer

- Monitoring facility and program event recording for statistics gathering and problem determination
- Byte-oriented operands for certain instructions
- More external interruption types to support new features
- Hardware correction capabilities and expanded machine check interruption levels and masking to improve availability and serviceability
- Expanded logouts to processor storage (both processor independent and dependent) after machine checks and channel errors to aid recovery and serviceability
- Channel retry data provided in a limited channel logout to aid in programmed recovery after channel errors
- Block multiplexer channels to improve I/O throughput
- Dynamic address translation and channel indirect data addressing hardware to support virtual storage available for intermediate as well as large-scale processors. (Dynamic address translation is provided only for the large-scale Model 67 in System/360.)
- Store status facility to obtain processor status data after hardware errors
- System/370 Extended Facility/Feature for large-scale processors

The System/370 architecture implemented in 4300 Processors does not include the following facilities that are defined for optional implementation in System/370 processors:

- Extended machine check logout (that processor-dependent data logged beginning at the address specified in control register 15--normally location 512), the processor-dependent logout to locations 256 to 351, and the processor-dependent I/O extended logout (that data logged beginning at the address in the word at location 172)
- Direct Control (READ DIRECT and WRITE DIRECT) instructions. The external signals facility in 4300 Processor architecture provides the six external interruption lines included in the System/370 Direct Control facility without the two instructions READ DIRECT and WRITE DIRECT.
- System/370 Extended Facility
- Multiprocessing (includes SET PREFIX, STORE PREFIX, SIGNAL PROCESSOR, and STORE CPU ADDRESS instructions)
- Certain processor dependencies

System/370 architecture as implemented in 4300 Processors provides the ability to execute (1) all System/370 control and problem programs that are not time-dependent or System/370 processor-dependent and (2) all System/360 control and problem programs that are not time-dependent or System/360 processor-dependent (see specific compatibility constraints later in this subsection).

System/370 architecture as implemented in 4300 Processors provides dynamic address translation and channel indirect data addressing facilities to support one or multiple virtual storages, each of which can be up to 16,777,216 bytes in size.

The advantage of System/370 mode is that its address translation facility allows for the support of multiple virtual storages, a capability that is required to support multiple virtual machines. See Section 15 for a detailed discussion of the operation and advantages of both address translation facilities and Section 18 for the advantages of virtual machines.

4300 Architecture

The 4300 Processor architecture is essentially System/370 architecture with certain simplifications and enhancements. The 4300 Processor architecture is simplified in that it does not contain the five optional items listed above that are not provided in the System/370 architecture supported by 4300 Processors.

The 4300 Processor architecture is enhanced in that it provides an alternative to the dynamic address translation and channel indirect data addressing facilities for support of virtual storage to improve performance. Both 4300 Processor architecture and the System/370 architecture defined for 4300 Processors are improved in that they provide for processor malfunction analysis, using processor-dependent logout data (reference code generation) to aid processor serviceability.

The 4300 Processor architecture includes all System/370 architecture functions except the following:

- Dynamic address translation and channel indirect data addressing facilities
- Store status
- Processor-dependent machine check and I/O extended logouts
- Direct Control instructions
- System/370 Extended Facility
- Multiprocessing

The 4300 Processor architecture provides the following functions that are not implemented in System/370 architecture (for 4300 Processors or System/370 processors):

- An internal mapping function that translates virtual storage addresses in both instruction processing function programs and channel programs to processor storage addresses during instruction and channel program execution. This mapping function can support one virtual storage of 16,777,216 bytes maximum in size. Additional instructions to support this translation function are provided for control program use. This function is an alternative to the dynamic address translation and channel indirect data addressing facilities of System/370 architecture.
- A machine save function that preserves the state of the processor and the contents of the first 2048 bytes of processor storage. This function is an alternative to the store status function of System/370 architecture.

The 4300 Processor architecture provides the ability to execute System/360 and System/370 problem programs that are not time dependent or processor dependent. System/370 control programs that support virtual storage and System/360 control programs cannot execute correctly in the 4341 Processor when ECPS:VSE and EC modes are active.

The advantage of ECPS:VSE mode (4300 Processor architecture) over System/370 mode is that when only one virtual storage is required, ECPS:VSE mode provides a reduction in the amount of processor time required for address translation functions. Specifically, programmed address translation for channel programs is eliminated and the internal mapping function utilized for address translation is faster than the dynamic address translation facility.

While different control program support is required to support the two different modes (System/370 and ECPS:VSE) of 4341 Processor operation, 4300 Processor architecture was designed to ensure that problem program compatibility would exist between the two modes. This compatibility enables System/360 and System/370 problem programs to execute with either ECPS:VSE or System/370 mode active in a 4341 Processor (subject to the compatibility constraints discussed later in this subsection and any appropriate operating system constraints).

SYSTEM/360 PROGRAM COMPATIBILITY WITH THE 4341 PROCESSOR

For both System/370 and ECPS:VSE modes, two other modes of processor operation, basic control mode and extended control mode, are also implemented, as determined by bit 12 of the current PSW. When a 4341 Processor operates in BC mode, the contents, layout, and function of permanently assigned processor storage locations 0 to 127 are identical to these locations in System/360 Models 22 and up (except 44 and 67) with the exception of the use of PSW bit 12. BC mode essentially is the System/360-compatible mode of 4300 Processors.

When EC mode is operative in the 4341 Processor, the format of the PSW is altered and the number of permanently assigned locations extends beyond processor storage address 127. Changes to the PSW consist of removal of certain fields to create space for additional mode and mask bits that are required for 4341 Processor functions that are not implemented in System/360. The removed fields are assigned to locations above 127 and to a control register.

EC mode is effective when PSW bit 12 is a one. BC mode is effective if this bit is a zero. BC mode is established during initial program reset for both System/370 and ECPS:VSE modes. Therefore, a control program must turn on bit 12 of the PSW in order to cause EC mode to become operative. As a result, control and problem programs written for System/360 (Models 22 and up except 44 and 67) can be run without modification in BC mode in a 4341 Processor operating in System/370 mode that has a comparable hardware configuration, with the following exceptions:

1. Programs that depend on facilities that are not defined in the System/370 architecture for 4300 Processors (READ DIRECT, WRITE DIRECT, and tightly-coupled multiprocessing intructions, etc.)

2. Time-dependent programs. (They may or may not run correctly.)

3. Programs that depend on results defined in the System/370 Principles of Operation (GA22-7000) to be unpredictable or processor-dependent

4. Programs that use unassigned fields in processor formats (instruction formats, for example) that are not explicitly made available for program use

5. Programs that depend on interruptions caused by errors, such as unassigned operation codes or command codes

6. Programs that use PSW bit 12 as an ASCII bit. (ASCII mode is not implemented in 4300 Processors.)

7. Programs that depend on storage locations that are assigned to fixed functions, such as the machine-check-save area in lower processor storage. (The fixed logout area in locations 0 to 511 in the 4341 Processor is larger than that for System/360 processors.)

8. Programs that, for I/O operations, do not take into account the effects of channel prefetching, command retry, and the operation code assignment for HALT DEVICE

9. Programs that depend on data in storage after power has been turned off and then restored

SYSTEM/370 PROGRAM COMPATIBILITY WITH THE 4341 PROCESSOR

Control and processing programs written to operate in BC mode in System/370 processors can execute in a 4341 Processor operating in BC and System/370 modes without modification subject to the constraints listed above for System/360 programs, except item 6 (since ASCII mode is not implemented in System/370 either), plus one additional constraint. They cannot depend on the processor and channel identifications provided by the instructions STORE CPU ID and STORE CHANNEL ID.

Control programs written to operate in System/370 processors with EC mode and dynamic address translation enabled can operate on a 4341 Processor without modification with EC mode, dynamic address translation, and System/370 mode in effect, subject to the same constraints indicated for System/370 BC mode programs.

System/370 control programs that depend upon dynamic address translation hardware cannot execute in a 4341 Processor operating in ECPS:VSE mode. However, the problem programs used with a System/370 control program that requires dynamic address translation hardware can execute without modification in a 4341 Processor that is operating in either System/370 or ECPS:VSE mode (with an appropriate control program) subject to the same constraints indicated for System/370 BC mode programs.

05:15 PHYSICAL AND LOGICAL COMPONENTS

The physical components of a 4341 Processor configuration are the 4341 Processor, a 3278 Model 2A Display Console as the operator console, and I/O devices. The 4341 Processor, which is air cooled, is shown in Figure 05.15.1. The 3278 Model 2A Display Console and 3287 Printer are shown to the left of the 4341 Processor. A string of 3370 disk drives is shown to the left of the magnetic tape units on the right.

The functional components physically contained within the frames of the 4341 Processor are the instruction processing function, all processor storage, the storage control function, channels, and the support processor subsystem. If the optional Channel-to-Channel Adapter feature is installed, a frame is added to the 4341 Processor frames shown in Figure 05.15.1. Figure 05.15.2 shows the logical components of the 4341 Processor.

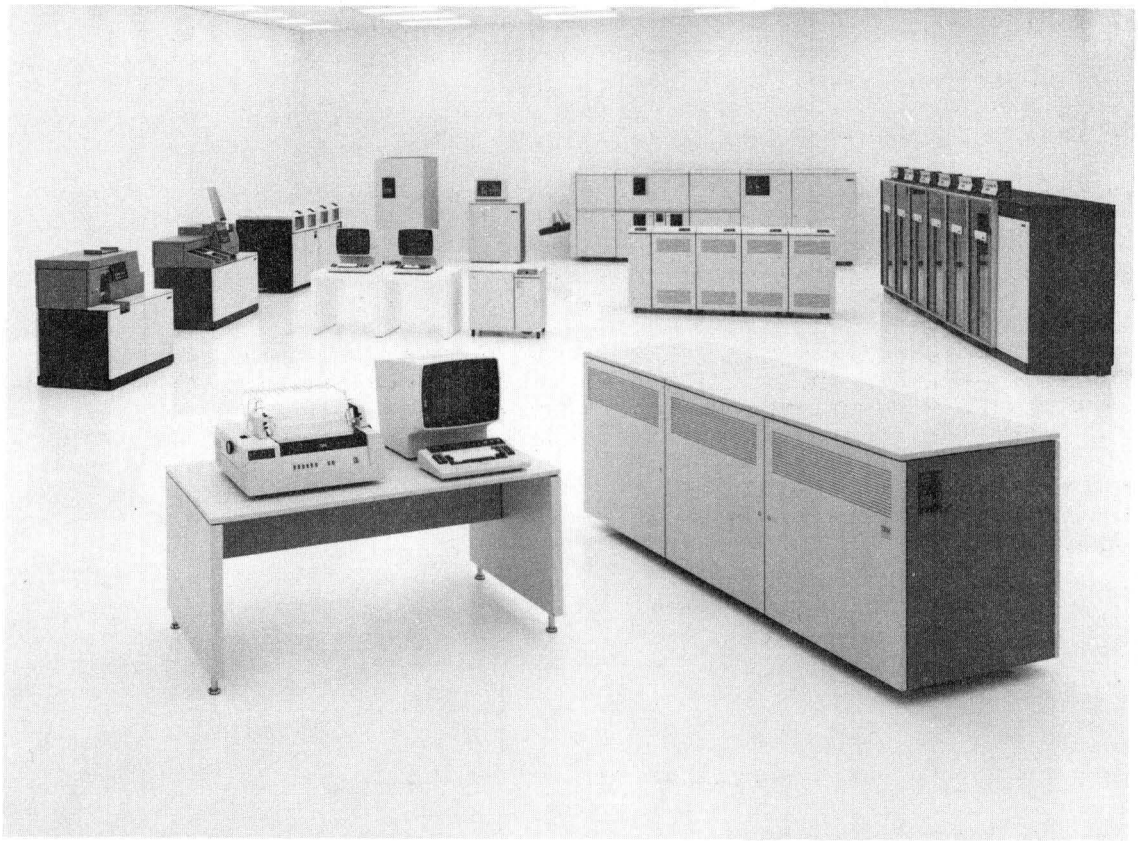


Figure 05.15.1. The 4341 Processor (design model)

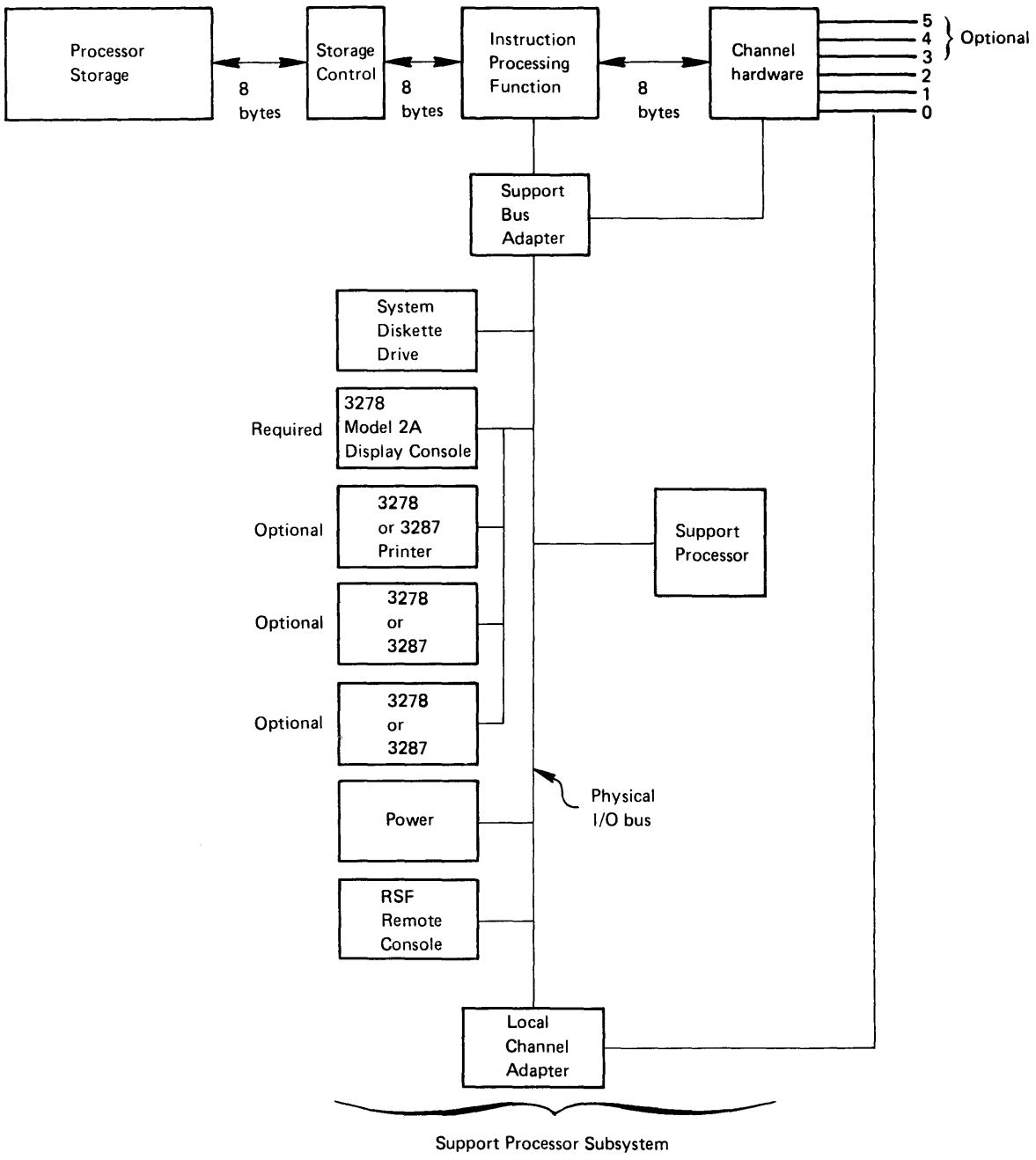


Figure 05.15.2. Logical components in the 4341 Processor

SECTION 10: THE 4341 PROCESSOR

10:05 THE INSTRUCTION PROCESSING FUNCTION

GENERAL DESCRIPTION

The instruction processing function contains all the elements necessary to decode and execute the instructions in the instruction set for the 4341 Processor. I/O instructions are partially processed by the instruction processing function and partially processed by channel hardware. Extensive parity checking is done within the instruction processing function to ensure data validity.

All instruction execution functions and most channel operations are microcode controlled. Microinstructions are four bytes in length. Control storage for the residence of all instruction processing function microcode is standard.

Certain basic control and service functions are provided for the 4341 Processor by the support processor, a component of the support processor subsystem, instead of by the instruction processing function. The support processor is a microcoded controller with its own control storage. The support processor also handles I/O operations for the operator console device and up to three other display consoles and/or printer devices that are directly attached to the 4341 Processor. In addition, the support processor controls diagnostic facilities (see discussion in Sections 10:15 and 50:15).

The instruction processing function in the 4341 Processor has a variable-length cycle time. Cycle time varies from 150 nanoseconds to 300 nanoseconds, depending on the instruction. The data path within the instruction processing function is eight bytes wide, which is the widest data path implemented in large-scale System/370 processors.

Elements included in the instruction processing function to perform instruction execution are instruction buffers for instruction prefetching, an eight-byte-wide arithmetic logic unit, an eight-byte-wide byte shifter, a bit shifter, and external registers.

The instruction processing function of the 4341 Processor includes facilities not implemented in System/370 intermediate-scale processors that are designed to speed up instruction execution. First, instruction fetching is performed during instruction execution such that, during sequential instruction processing, instruction fetching is overlapped with instruction execution. Unoverlapped instruction fetching usually occurs only when a successful branch instruction is processed.

Second, each instruction type has one 150-nanosecond instruction cycle during which several functions are performed. The following are performed during the instruction cycle: instruction decoding, selection of the microcode required to execute the instruction, calculation of the required storage address using base register and displacement values for instructions that reference storage, fetching of the contents of the register 1 specification in RR- and RS-type instructions, testing for any interruptions, and complete execution of Branch On Condition (BC and BCR) instructions that are not successful (that is, no branch is taken).

Third, the microcode for several of the most frequently used instructions (such as LOAD, MOVE CHARACTERS, LOAD MULTIPLE, STORE MULTIPLE, LOAD ADDRESS, and STORE) is optimized for faster execution.

Fourth, the use of an eight-byte-wide arithmetic logic unit for floating-point and decimal arithmetic instructions enables floating-point additions and subtractions that involve values with equal exponents and decimal additions and subtractions to be performed with the same speed as binary additions and subtractions. Usually, additions and subtractions performed using binary arithmetic operate much faster than when decimal or floating-point arithmetic is used.

Fifth, decimal arithmetic operations are performed significantly faster in the 4341 Processor than in intermediate-scale System/370 processors and execute as fast as decimal operations in certain large-scale System/370 processors.

Sixth, an eight-byte-wide instead of a one-byte-wide shifter is utilized. This shifter speeds up the execution of shift instructions, since a shift can be performed in one cycle instead of multiple.

The shifter is also used to align data that is not on the proper boundary. The use of the shifter instead of microcode for the alignment function eliminates in the 4341 Processor nearly all the performance degradation that is experienced in most System/370 processors when data is not aligned on the correct boundary. In the 4341 Processor, no performance degradation occurs when alignment is performed within a doubleword. Some degradation occurs when the unaligned data required spans two doublewords, since both doublewords must be fetched to obtain the needed data.

The multifunction instruction cycle, eight-byte-wide arithmetic logic unit, and eight-byte-wide shifter give the instruction processing function the ability to execute 32 of the 4341 Processor instructions (such as the RR-type and certain other instructions) in two cycles (one instruction and one execution cycle). Execution time for 31 of these instructions is only 300 or 375 nanoseconds. TEST UNDER MASK executes in 450 nanoseconds.

Eight-byte external registers are included in the instruction processing function. These hardware registers provide data links between instruction processing function microcode and channel or instruction processing function hardware. The external registers contain such items as the instruction counter, PSWs, the time-of-day clock, storage address registers, the channel storage address register, interruption registers, the next instruction buffer register, and status registers.

The instruction processing function accesses a data local storage area of 128 doublewords as required during the execution of instructions. This data local storage contains certain control registers, the general registers, the floating-point registers, six channel work areas, save areas, and work areas.

A trace array of 32 entries is included in the instruction processing function to trace the addresses of executed microcode. The array is always updated during instruction execution and can be set to operate in one of two modes. In the default mode, the trace array contains the addresses of the last 32 microinstructions executed. In the other mode (which the customer engineer can set using the operator console), the trace array contains the addresses of the last 32 microinstructions that caused switching from one microcode module to another. The trace array is provided to aid in error detection and recovery. The array helps to indicate the cycle that caused the error when a machine check occurs.

Unlike System/360 Models 50 and 65, the 4341 Processor contains reloadable control storage (RCS) for instruction processing function microcode residence instead of read-only control storage. The use of writable storage for control functions adds to the advantages of using a

read-only storage instead of conventional circuitry. It provides improved serviceability and simplifies extensions of functional capabilities of the processor.

Serviceability is enhanced because of the speed and ease of engineering change installation--the new microcode need only be loaded into RCS--and because more extensive diagnostics can be provided without the necessity of adding additional control storage (control storage is available to be used for diagnostic residence). Functional capability is extended by the ability to more easily support different architectures and features in one system.

The address translation facilities provided for System/370 and ECPS:VSE modes are discussed in Section 15. ECPS:VS1 and ECPS:VM/370 are discussed in Sections 30 and 18, respectively. Other significant new features of the instruction processing function of the 4341 Processor for Model 50 and 65 users are discussed in the remainder of this subsection.

CONTROL REGISTERS

The program states in which the 4341 Processor is operating are reflected in the current program status word (PSW) and in processor status indicators called control registers, just as in System/370 processors. These control registers up to 16 control registers, 0-15, can be addressed. Certain control registers are used only when EC mode is in effect. Control registers are program-addressable only when the processor is in the supervisor state.

A control register can be set with the LOAD CONTROL instruction, and its contents can be placed in processor storage with the STORE CONTROL instruction. Additional status indicators contained in control registers are required in order to support new functions. A control register is 32 bits in size.

Note that control register assignments for functions that are implemented in both 4300 Processors and System/370 processors are the same. Control register bits that control functions not supported in 4300 Processors (multiprocessing, extended machine check logouts, etc.) are unassigned in 4300 Processors for compatibility purposes.

BASIC CONTROL MODE

As indicated previously, the contents, layout, and function of fixed locations 0-127 in 4300 Processors and System/370 processors that are operating in BC mode are identical to these locations in most System/360 processors with the exception of bit 12 in the PSW, which specifies EBCDIC or ASCII mode in System/360 processors and BC or EC mode in 4300 Processors and System/370 processors. ASCII mode is not implemented in 4300 Processor or System/370 architecture, nor was the mode bit supported by IBM programming systems provided for System/360 processors, because System/360 USASCII-8 did not become the ASCII standard.

However, ASCII-encoded tapes are supported by certain DOS/VS, DOS/VSE, and OS/VS language translators and service programs. That is, ASCII-mode tapes are accepted by certain DOS/VS, DOS/VSE, and OS/VS language translators and service programs as input and converted to EBCDIC for processing. The capability of writing ASCII-mode tapes is also provided.

To improve system availability and serviceability, implementation of the machine check class of interruption for the 4341 Processor, like the implementation for System/370 processors, is considerably altered from

its implementation in Models 50 and 65 (see Section 50). However, the other four interruption classes (I/O, SVC, program, and external) operate in the same manner in Models 50 and 65 and the 4341 Processor except for the (1) expansion of external interruption masking, (2) expansion of channel masking, and (3) addition of program and external interruptions to support new features in the 4341 Processor. Imprecise interruptions do not occur in the 4341 Processor.

Five external subclass mask bits, which allow selective masking of external signals (2-7), interval timer, CPU timer, clock comparator, and operator console interruptions, are provided in control register 0. When the PSW external mask bit is off, the processor is disabled for all external interruption types. When the PSW external mask bit is on, an external interruption occurs for an external interruption type only if its associated subclass mask bit is on also.

Execution of the SET SYSTEM MASK (SSM) instruction is under the control of the SSM mask bit in control register 0. When the SSM mask bit is on, an attempt to execute an SSM instruction causes a program interruption without execution of the SSM instruction. When the SSM mask bit is off, SSM instructions are executed as usual.

This SSM interruption is implemented to enable existing programs that were written for System/360 processors or for System/370 BC mode of operation to execute correctly in EC mode without modification of the system mask field addressed by existing SSM instructions. When an SSM interruption occurs, the contents of the BC mode format system mask indicated by the SSM instruction can be inspected and the appropriate EC mode mask bits can then be set by an SSM simulation routine.

EXTENDED CONTROL MODE

Extended control mode is a major facility that is not implemented in System/360 architecture. Facilities that depend on which mode is in effect are discussed below and apply to System/370 and ECPS:VSE modes unless otherwise noted. Any item not covered operates identically in BC and EC modes.

When a 4341 Processor operates in System/370 and EC modes, it operates exactly like a System/370 processor operating in EC mode, except for the basic architecture implementation differences previously listed in Section 05:10.

Change in PSW Format

When a 4341 Processor operates in EC mode, the format of the PSW differs from its BC mode format. Both PSW formats are shown in Figure 10.05.1. In EC mode, the PSW does not contain individual channel mask bits, an instruction length code, or the interruption code for a supervisor call, external, or program interruption. The channel masks are contained in control register 2, and the other fields are allocated permanently assigned locations in the fixed lower processor storage area above address 127.

Removal of the fields indicated provides room in the EC mode PSW for control of new features that are unique to EC mode (such as dynamic address translation for System/370 mode and program event recording) and for the addition of summary mask bits (such as channel and I/O masks). Use of a single mask bit to control the operation of an entire facility (such as program event recording) or an entire interruption class (such as I/O and external) simplifies the coding required to enable and disable the processor for these interruptions.

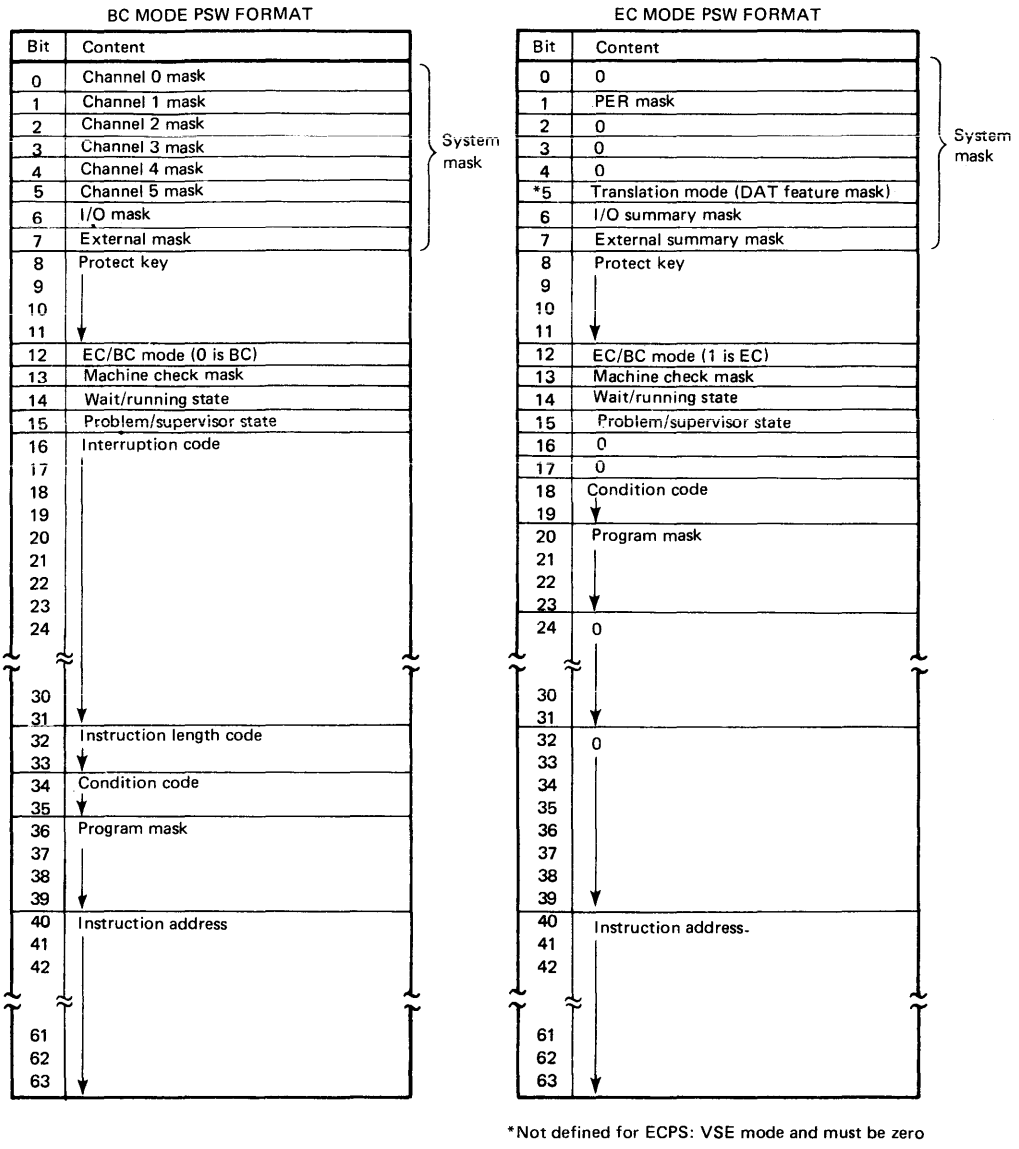


Figure 10.05.1. BC and EC mode PSW formats

Note that the BC and EC mode PSW formats shown in Figure 10.05.1 are the same for System/370 and ECPS:VSE modes with one exception. Bit 5, which enables/disables the dynamic address translation facility, must be zero for ECPS:VSE mode.

Change in Permanently Assigned Processor Storage Locations

When a 4341 Processor operates in EC mode, the number of permanently assigned locations in lower processor storage is increased to include fields for storing instruction length codes, interruption codes (for supervisor call, external, and program interruptions), program event recording data, the I/O address for an I/O interruption, and an exception address for the address translation capability.

The fixed storage layout for BC mode is shown in Figure 10.05.2 and for EC mode in Figure 10.05.3. The format of locations 0 to 511 is the same for 4300 Processors and System/370 processors for fields implemented in both. System/370 processors implement additional

processor-dependent fields, such as a region code in locations 252 to 255, that are reserved fields in 4300 Processors. The access exception field shown in Figure 10.05.2 is not implemented in System/370 processors for BC mode operations. Locations 0 to 127 are the same for 4300 Processors and System/360 processors.

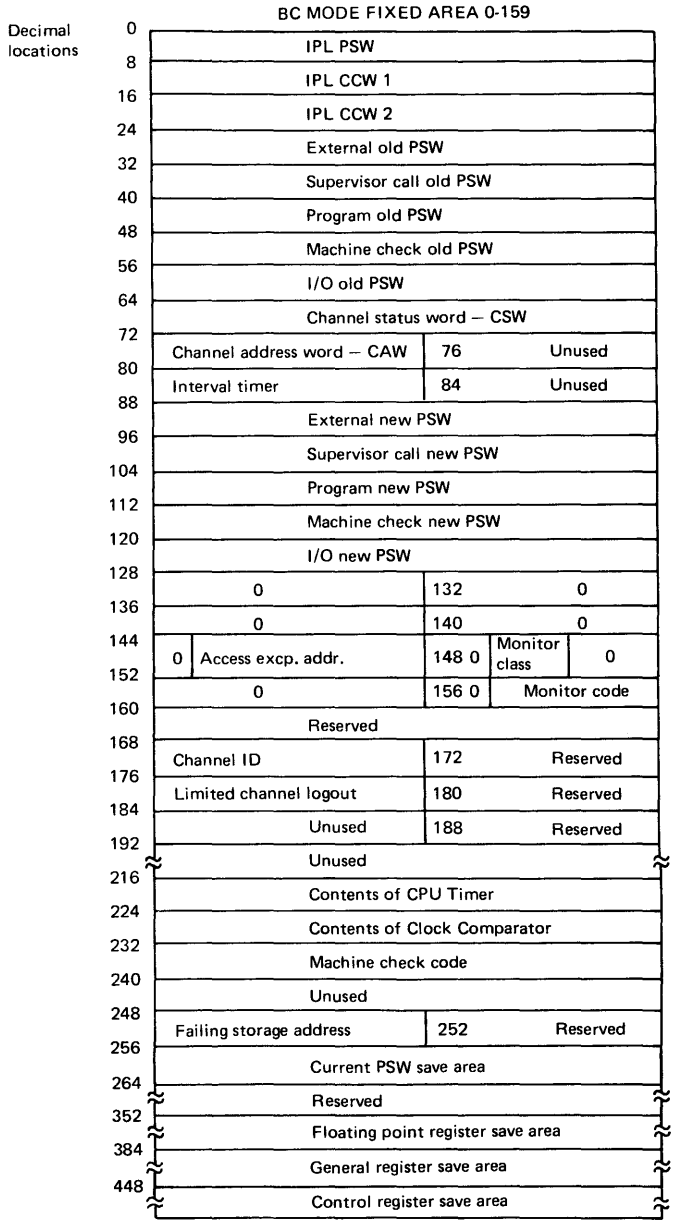


Figure 10.05.2. BC mode fixed processor storage locations 0 to 511

Channel Masking Changes

When a 4341 Processor operates in EC mode, interruptions from each channel are controlled by the summary I/O mask bit (bit 6) in the current PSW and an individual channel mask bit in control register 2. In the 4341 Processor, bits 0 to 5 in control register 2 are assigned to control channels 0 to 5, respectively. Both the summary mask bit and the appropriate individual channel mask bit must be on in order for an interruption from a given channel to occur. In BC mode, interruptions

from channels 0 to 5 are controlled only by the channel mask bits (bits 0 to 5) in the current PSW.

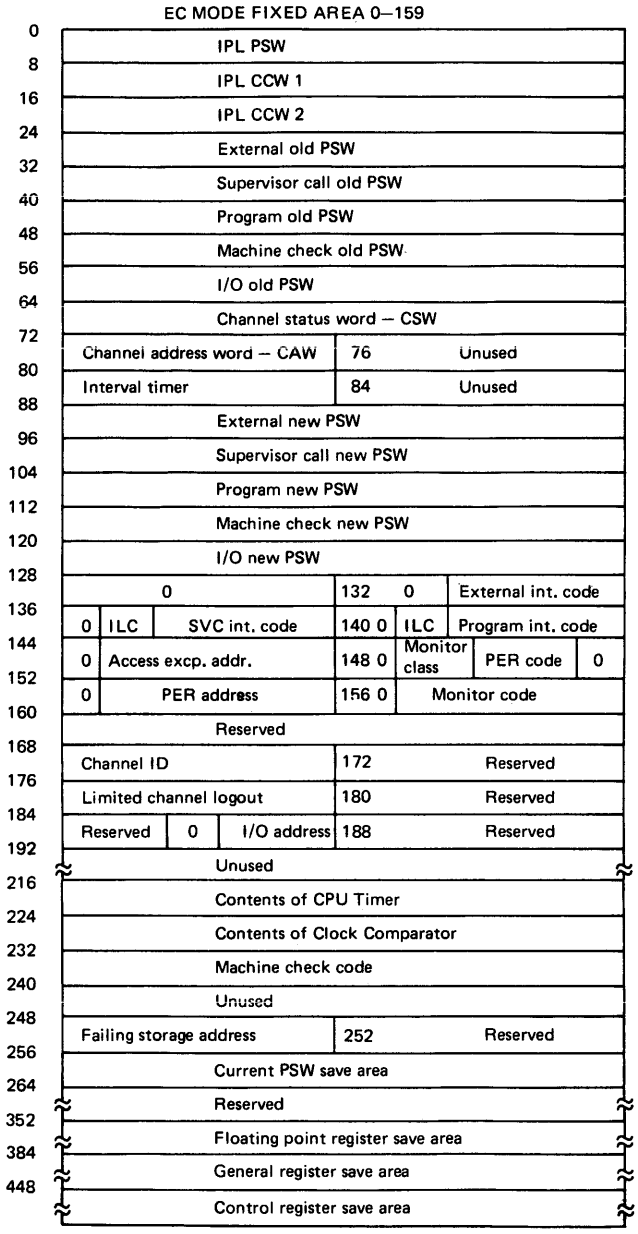


Figure 10.05.3. EC mode fixed processor storage locations 0 to 511

Expansion of Storage Key Size

The size of the storage key associated with each 2K storage block for store and fetch protection is seven bits (as in System/370) instead of five bits, as in System/360. The two additional bits (reference and change) are included for use with address translation and are discussed in Section 15:10. The SET STORAGE KEY instruction sets a seven-bit key regardless of the mode, BC or EC, in effect. The INSERT STORAGE KEY instruction causes a five-bit or a seven-bit key to be loaded into a register, depending on whether BC or EC mode, respectively, is in effect.

Changes to Certain Instruction Definitions

As a result of the differences between the PSW format and the permanently assigned processor storage locations in EC and BC modes, the definition of certain instructions is affected. Instructions provided for System/360, System/370, and 4300 Processors whose definition is altered for EC mode are:

BRANCH AND LINK (RR, RX)	SET STORAGE KEY
INSERT STORAGE KEY	SET SYSTEM MASK
LOAD PSW	SUPERVISOR CALL
SET PROGRAM MASK	

Revised definitions of these instructions to include BC/EC mode differences are contained in the System/370 and 4300 Processor Principles of Operation publications. Programs that operate in BC mode and that use LOAD PSW and/or SET SYSTEM MASK (SSM) instructions must be modified in order to operate correctly in EC mode. The eight-byte PSW to be loaded by LPSW instructions and the eight-bit system mask to be set by SSM instructions must be changed to EC mode format. (Programs that use SSM instructions and that are executed in an OS/VS1 environment need not be so modified because the interruption for SSM instructions and an SSM simulation routine are supported.)

Programs that use the other instructions listed do not have to be changed in order to operate correctly in EC mode, unless they use other facilities that are mode-dependent. System/370 programs that operate in BC mode and that use STORE THEN OR SYSTEM MASK and STORE THEN AND SYSTEM MASK instructions (not provided for System/360) must also be modified in order to operate correctly in EC mode.

Program Event Recording

Program event recording (PER), a standard feature for the 4341 Processor, is designed to assist in program debugging by enabling a program to be alerted to any combination of the following events via a program interruption:

- Successful execution of any type of branch instruction
- Alteration of the contents of the general registers designated by the user
- Fetching of an instruction from a processor storage area defined by the user
- Alteration of the contents of a processor storage area defined by the user

The PER feature can operate only when EC mode is in effect and the PER mask, bit 1 of the current PSW, is a one. Control register 9 (bits 0 to 3) is used to specify which of the four PER event types are to be monitored. A PER program interruption is taken after the occurrence of an event only if both the PER mask bit and the respective event mask bit in control register 9 are on. Control register 9 (bits 16 to 31) also specifies which of the 16 general registers are to be monitored if monitoring of this event is specified. Control registers 10 and 11 indicate the beginning address and the ending address, respectively, of the contiguous processor storage area that is to be monitored for instruction fetching and/or alteration.

When an event that is being monitored is detected, PER hardware causes a program interruption, if the PER mask bit is on, and the identification of the type of event is stored in the fixed processor storage area (location 150). The address of the instruction associated with the event is also stored (locations 153 to 155). Program event interruptions are lost if they occur when the PER mask bit or the particular event mask bit is off. In the 4341 Processor, additional processor time is required to execute instructions when program event recording is operative.

When System/370 mode is in effect, if dynamic address translation mode is specified when PER is active, virtual storage addresses instead of real storage addresses (discussed in Section 15) are placed in the control registers to monitor references to a contiguous virtual storage area. For ECPS:VSE mode, virtual storage addresses are always used.

EXPANDED INSTRUCTION SET

The instruction set for the 4341 Processor is a superset of that provided for System/360 processors. It consists of the System/360 instruction set plus several new instructions that support System/370 and 4300 Processor architecture and provide additional functions. The standard instruction set contains all the 4341 Processor instructions (no instructions are optional).

The standard instruction set for the 4341 Processor consists of (1) all System/370 instructions except those associated with features not implemented in 4300 Processors (READ DIRECT, WRITE DIRECT, and the four multiprocessing instructions), (2) several control instructions that are valid only for ECPS:VSE mode, and (3) the MOVE INVERSE instruction. The ECPS:VSE mode instructions, discussed in Section 15:15, and MOVE INVERSE instruction are the only 4300 Processor instructions that are not also available for System/370 processors.

The STORE CPU ID instruction permits a program to determine the processor and version of the processor upon which it is operating and provides the processor serial number.

The STORE CHANNEL ID instruction can be used to identify the types of channels present in the system (selector, byte multiplexer, and block multiplexer). Selector is indicated for a block multiplexer channel when it is operating in selector mode at the time the STORE CHANNEL ID instruction is issued. Some of the other new instructions are:

- General purpose instructions

  Several general purpose instructions, which can be of benefit to both control and processing program performance, are provided.

  SHIFT AND ROUND DECIMAL provides right or left shifting of packed decimal data using a single instruction. This instruction can save from 6 to 18 bytes of instruction storage and instruction execution time for each decimal shift and round operation performed in commercial processing.

  MOVE LONG provides for the movement of up to 16 million bytes from one location in processor storage to another with a single instruction, thereby removing the System/360 limitation of 256 bytes per move. A check for the possibility of destructive overlap is made by the hardware prior to the movement of any data and the MOVE LONG instruction is not executed if operand destruction can occur. This instruction can eliminate the necessity of multiple move instructions or the inclusion of move subroutines. The format and operation of MOVE LONG facilitates efficient record blocking and

deblocking, field padding, and storage clearing, which are operations frequently performed in commercial processing.

The COMPARE LOGICAL LONG instruction can be used to compare logically two fields of up to 16 million bytes in length, thus removing the System/360 256-byte limit on byte compares. In addition, when an unequal compare occurs, the two characters that caused the inequality are identified.

The MOVE LONG and COMPARE LOGICAL LONG instructions are interruptible. Thus, when an I/O operation terminates during their execution, the interruption is taken and the channel is not held up awaiting termination of what might be a lengthy move or compare.

COMPARE LOGICAL, INSERT, and STORE CHARACTERS UNDER MASK instructions provide byte addressability within the general registers and permit nonword-size data that is not on a word boundary to be compared with data in a register, loaded into a register, and stored from a register. These three instructions can be of most benefit to control program programmers, to compiler writers, and to others who must manipulate processor storage addresses.

The MOVE INVERSE instruction is standard in the 4341 Processor. It causes bytes from the second operand to be fetched in right-to-left sequence and placed in left-to-right sequence in the first operand location. The instruction is useful for handling languages in which writing occurs right to left.

- Control instructions

STORE THEN AND SYSTEM MASK and STORE THEN OR SYSTEM MASK are two privileged instructions that affect the system mask (bits 0 to 7 in the current PSW). The STORE THEN AND SYSTEM MASK instruction provides, via a single instruction, the capability of storing the current system mask for later restoration, while selectively zeroing certain system mask bits. The STORE THEN OR SYSTEM MASK provides system mask storing and selective setting of system mask bits to ones. These two instructions simplify the coding required to alter the system mask, particularly when the existing settings must be saved.

COMPARE AND SWAP and COMPARE DOUBLE AND SWAP instructions provide the capability of controlling access to a shared processor storage area in a multiprogramming environment. Although the TEST AND SET instruction can also be used for this purpose, these compare instructions enable a program to leave a message when the shared area is in use. This message can be inspected, via a compare and swap instruction, by the other programs that share the processor storage area.

Two PSW key-handling instructions are provided. The INSERT PSW KEY privileged instruction enables a program to place in general register 2 the four-bit access control (protection) key from the current PSW. The SET PSW KEY FROM ADDRESS privileged instruction enables a program to place an access control key contained in general register 2 or processor storage in the current PSW. When a control program is requested to access a given processor storage location by a problem program, these two instructions can be used by the control program during its processing of the request to determine whether or not the problem program is authorized to access the specified processor storage location.

- Extended Precision Floating Point

  The standard floating-point feature includes extended precision operations. Extended precision is provided for use in application areas in which the precision provided by the long-form floating-point format is not large enough.

  Precision of up to 28 hexadecimal digits, equal to up to 34 decimal digits, is provided by the extended precision data format. Extended precision is achieved by using two doublewords (16 bytes) to represent an extended precision floating-point number instead of using one doubleword as is done in long-form representation. Fourteen hexadecimal digits, or up to 17 decimal digits, of precision are provided by the long floating-point format.

  Seven extended precision floating-point instructions are included in this feature. They provide addition, subtraction, and multiplication operations for extended precision data, using a pair of floating-point registers, and the ability to round from long to short form or from extended to long form. An extended precision divide instruction is not provided; however, a simulator for this operation is provided in OS/VS1.

BYTE-ORIENTED OPERANDS

The 4341 Processor supports a standard byte boundary alignment facility for processor storage, identical to that for System/370. The presence of the byte-oriented operand function allows the processor storage operands of unprivileged instructions (RX and RS formats) to appear on any byte boundary without causing a specification program interruption. Without this facility, operands must be aligned on integral boundaries, that is, on storage addresses that are integral multiples of operand lengths. Byte orientation does not apply to alignment of instructions or channel command words (CCWs).

Byte orientation can be used effectively in commercial processing to eliminate the padding bytes added within records or to blocked records to ensure binary and floating-point field alignment. The smaller physical record that results from the elimination of padding bytes requires less external storage and increases effective I/O data rates. I/O-bound commercial programs, in which throughput is in almost direct proportion to the I/O data rate, can achieve performance improvement by using byte alignment for binary and floating-point data. In the 4341 Processor, the use of byte alignment in a program only minimally degrades instruction execution performance.

A program written to use byte boundary alignment will not necessarily run on a System/360 processor that does not have the feature. Therefore, programs that are to run in both the 4341 Processor and a System/360 processor without byte orientation should be written to adhere to integral boundary rules.

MONITORING FEATURE

The monitoring feature is standard in the 4341 Processor and functionally identical to the System/370 monitoring feature. This feature provides the capability of monitoring the occurrence of programmed events. For example, monitoring can be used to perform measurement functions (how many times a routine was executed) or for tracing functions for the purpose of program debugging (which routines were executed).

The MONITOR CALL instruction is provided with the monitoring feature. Execution of this instruction indicates the occurrence of one of the events being monitored. The operands of the MONITOR CALL instruction permit specification of up to 16 classes of events, each class with up to 16 million unique types of events. The 16 monitor classes are individually maskable via mask bits in control register 8. When a MONITOR CALL instruction is executed, a program interruption occurs, if the monitor class indicated is specified, and the event identification (class and type) is stored in the lower fixed storage area.

Both the PER facility and the monitoring feature are provided for debugging purposes. The two features differ from one another in (1) the number of events that can be defined, (2) whether the events are defined by the hardware or the programmer, and (3) whether the hardware or the programmer checks for the events and causes the interruptions. When PER is used, once the events to be monitored have been designated by the user, processor hardware checks for the occurrence of the events and causes the interruption. When the monitoring feature is used, the user defines the events to be monitored (up to 16 classes with up to 16 million codes each, instead of four events), and causes the program interruption by placing MONITOR CALL instructions at the desired places within the program.

ARCHITECTURE IMPLEMENTATION ALTERATIONS

Two alterations have been made to the action taken in the 4341 Processor during the execution of certain instructions common to both System/360 processors and 4300 Processors. These alterations are also implemented in System/370 processors. The first involves all instructions that check the validity of operands involved in packed decimal operations. In the 4341 Processor, an invalid sign in an operand causes the instruction to be suppressed (never executed) rather than terminated during execution as is done on System/360 processors.

Suppression, rather than termination, of an instruction when an invalid sign occurs ensures that the data fields involved remain unchanged. Therefore, a routine that inspects the field that has the invalid sign can be executed when a program check occurs. For example, when an invalid sign results from packing an entirely blank field, the sign can be corrected by programming, and transaction deletion or program termination is avoided.

The second alteration concerns the recognition of a storage protection exception during the execution of an EDIT or an EDIT AND MARK instruction. In the 4341 Processor, a protection exception always occurs when a pattern character is fetched from a location protected for storing but remains unchanged during the edit operation. This change eliminates unpredictable processor operation during editing operations in a 4341 Processor. The occurrence of a protection exception for the situation described is processor-dependent for System/360 processors.

INTERVAL TIMER

The interval timer at decimal location 80 in the fixed processor storage area is a standard feature and has a resolution of 3.3 milliseconds instead of the 16.6-ms resolution implemented for the interval timer provided for Models 50 and 65. Its maximum time period remains 15.5 hours. For accounting routines that utilize the interval timer (those in System/360 operating systems, for example), the higher resolution of this interval timer eliminates many of the accuracy problems caused by task execution durations shorter than the 16.6-ms resolution interval.

TIME-OF-DAY CLOCK

This clock is a binary counter of 52 bits with a cycle time of approximately 143 years. It is a standard feature and functionally like the time-of-day clock in System/370. The clock is updated every microsecond. Two instructions (SET CLOCK and STORE CLOCK) are provided to set the time and to request that the current time be stored in the specified doubleword of processor storage. The time can be set only when the processor is in supervisor state and only when time-of-day clock setting is enabled using the operator console.

The time-of-day clock can be used for more accurate time stamping than the interval timer. More accurate time of day can be maintained because, during normal system operation, the clock stops only when processor power is turned off. Execution of time-of-day clock microdiagnostic tests and an error in the clock also invalidate the clock time.

The interval timer cannot be as accurate as the clock for time-of-day maintenance because it is not updated when the processor is in the stopped state, and its updating may be omitted under certain conditions of excessive system activity. The 15.5-hour cycle time of the interval timer is also a restriction. The time-of-day clock better answers the timing needs of teleprocessing and realtime applications and has the capacity to handle special switchover situations, such as midnight, New Year's eve, etc.

CLOCK COMPARATOR AND CPU TIMER

These timing facilities are a standard feature of the 4341 Processor and functionally identical to the same timing facilities in System/370. The clock comparator provides a means of causing an external interruption when the time-of-day clock has passed a time specified by a program. This feature can be used to initiate an action, terminate an operation, or inspect an activity, for example, at specific clock times during system operation.

The clock comparator has the same format as the time-of-day clock. The clock comparator is set to zero during initial program reset. The SET CLOCK COMPARATOR privileged instruction is provided to place a value that represents a time of day in the clock comparator.

When clock comparator interruptions are specified via the external interruption summary mask bit in the current PSW and the clock comparator subclass mask bit in control register 0, an external interruption occurs when the time-of-day clock value is greater than the clock comparator value. If clock comparator interruptions are masked when this condition occurs, the interruption remains pending only as long as the time-of-day clock value remains higher than the value in the clock comparator. The STORE CLOCK COMPARATOR privileged instruction can be used to obtain the current value of the clock comparator.

The use of a clock comparator instead of the interval timer at location 80 to cause an interruption when a specified time is passed offers two advantages. First, the time-of-day clock increments when the processor is in the stopped state while the interval timer does not. Hence, if a processor stop occurs during processing and the processor is restarted, the clock comparator can still cause an interruption at the time requested. The interruption caused by the interval timer in such a situation is late. Second, implementing the time-of-day clock and the clock comparator in the same format eliminates the need to convert doubleword time-of-day clock values to single-word interval timer values.

The CPU timer provides a means of causing an external interruption when an interval of time specified by a program has elapsed. The CPU timer is implemented as a binary counter with a format identical to that of the time-of-day clock; however, bit 0 of the CPU timer is considered to be a sign. Therefore, the CPU timer has a maximum time period half as large as that of the time-of-day clock. When both the CPU timer and the time-of-day clock are running, the stepping rates of the two are synchronized so that they are stepped at exactly the same rate.

The CPU timer is set to zero at initial program reset, and the SET CPU TIMER privileged instruction is provided to place an interval of time in the CPU timer. The STORE CPU TIMER privileged instruction can be used to obtain the current CPU timer value. The CPU timer decrements every microsecond. If the external interruption summary mask bit in the current PSW and the CPU timer subclass mask bit in control register 0 are on, an external interruption occurs whenever the CPU timer value is negative (not just when the timer goes from positive to negative), indicating that the time interval has elapsed. The CPU timer decrements when the instruction processing function is executing instructions and while the processor is in the wait state. It is not decremented when the processor is in the stopped state.

While providing essentially the same function as the interval timer at location 80, the CPU timer provides advantages over the interval timer as follows. Task processing intervals of less than 3.3 milliseconds can be more accurately measured because of the one-microsecond read-out resolution of the CPU timer. A pending CPU timer interruption is reset when a SET CPU TIMER instruction is issued to set a positive value in the CPU timer, eliminating the need to take an interruption in order to reset the CPU timer, as is required for the interval timer.

In addition, the amount of timing facilities processing required during a task switch can be reduced because the format of the time-of-day clock and the CPU timer are the same. Conversion of doubleword time-of-day clock values to single-word interval timer values is eliminated, and timer queues can be structured so that little of the processing required during a task switch, when the interval timer is used, is necessary.

10:10 STORAGE

The 4341 Processor has a two-level storage system--a small high-speed buffer storage backed by a large processor (main) storage. In System/360 and System/370, such a concept is implemented only in large-scale, high internal performance systems such as System/360 Models 85 and 195 and System/370 Models 155 and up. Processor storage in the 4341 Processor is buffered by 8K of high-speed buffer storage.

The use of a two-level storage system, in which the instruction processing function works mostly with the buffer, significantly reduces the effective processor storage cycle of the 4341 Processor and greatly contributes to the high internal performance of the 4341 Processor.

PROCESSOR STORAGE

The 4341 Processor is available with 2048K bytes (Model K1) or 4096K bytes (Model L1) of processor storage (where K is 1024). A Model K1 is field-upgradable to a Model L1.

Access to processor storage is made via the storage control function, which operates under the control of the instruction processing function.

The path to and from processor storage is eight bytes wide and data that enters/leaves processor storage is aligned on a doubleword boundary.

Error checking and correction (ECC) hardware provides automatic detection and correction of all single-bit processor storage errors and detection, but not correction, of all double-bit and many multiple-bit errors. ECC logic is contained in the storage control function. The ECC feature is discussed fully in Section 50.

The translation table used for ECPS:VSE mode, UCWs, and certain work areas are located in highest addressed processor storage. This storage, called auxiliary storage, is reserved for processor rather than program use and is inaccessible to all programs.

The size of auxiliary storage depends on the number of UCWs installed, mode (System/370 or ECPS:VSE) in effect, and processor storage size. The minimum size is 14K for 128 UCWs, System/370 mode, and two or four megabytes of processor storage. The maximum size is 108K for 1024 UCWs, ECPS:VSE mode, and four megabytes of processor storage.

The amount of processor storage required for UCWs varies from a minimum of 8K to a maximum of 64K for 128 and 1024 UCWs, respectively, in 2K-byte increments. For System/370 mode, 6K is required in addition to the UCW requirement. For ECPS:VSE mode, 42K or 44K is required in addition to the UCW requirement.

The size of auxiliary storage is determined during IML. When System/370 mode is in effect, the processor storage address of the first byte of auxiliary storage is calculated and placed in an address check boundary (ACB) register. Any attempt to access an address equal to or above the ACB register value during program execution with System/370 mode in effect results in an addressing exception program level interruption. The ACB register is not used when ECPS:VSE mode is in effect.

AUXILIARY STORAGE

The contents of auxiliary storage vary depending on the mode, System/370 or ECPS:VSE, in effect. During an IML, the size of the auxiliary storage area is determined and each area in it is initialized as appropriate. Auxiliary storage for System/370 mode, shown in Figure 10.10.1, is always 6K bytes plus the UCW requirement in size. It contains the following in the highest to the lowest addressed locations:

- UCW area with a minimum of 128 and a maximum of 1024 UCWs. Size varies from 8K to 64K in 2K-byte increments.
- Channel directory area of 4K bytes. This area contains one directory of UCWs for each channel (see discussion under "Device Address and Unit Control Words" in Section 10:20) and two buffer areas used by the support processor.
- Program event recording area of 256 bytes
- Support processor local channel adapter work area of 256 bytes
- Support processor command validity table of 256 bytes
- K-addressable auxiliary storage area of 1K bytes. This area contains various pointers and data fields used by the instruction processing function (pointers to the beginning of the other areas in auxiliary storage, for example).

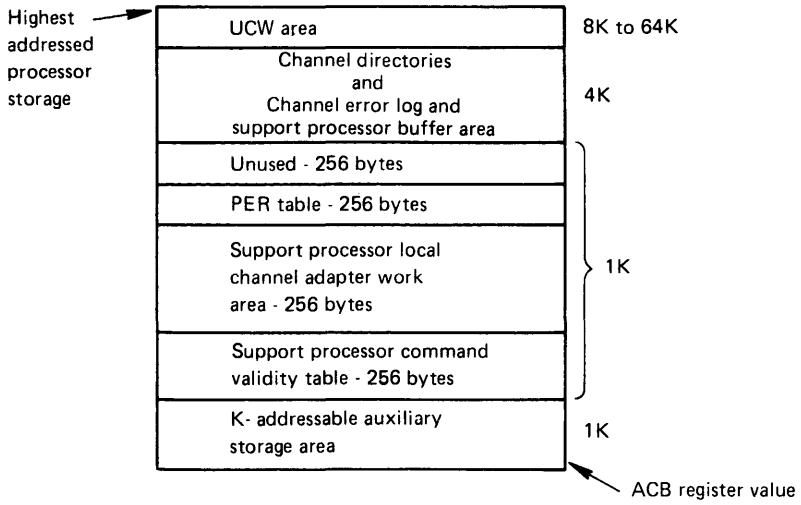


Figure 10.10.1. Layout of auxiliary storage for System/370 mode

For ECPS:VSE mode, the size of auxiliary storage depends on the amount of processor storage installed. Auxiliary storage for ECPS:VSE mode, shown in Figure 10.10.2, is 42K or 44K bytes for a two- or four-megabyte processor storage size, respectively, plus the UCW requirement and contains the following in the highest to the lowest addressed locations:

- Address translation table used to translate virtual storage addresses in instructions to real storage address during ECPS:VSE mode operations. This table is always 32K bytes (see discussion in Section 15:15).
- UCW area of 8K minimum and 64K maximum
- Channel directory area of 4K bytes
- Page management area of 2K or 4K bytes for a processor storage size of two or four megabytes, respectively (see discussion in Section 15:15).
- Machine save area of 2K bytes. Used to store the first 2048 bytes of processor storage during a machine save operation (see discussion in Section 12:10).
- Machine save area of 256 bytes for saving status information during a machine save operation
- Program event recording area of 256 bytes
- Support processor local channel adapter work area of 256 bytes
- Support processor command validity table of 256 bytes
- K-addressable auxiliary storage area of 1K bytes. The contents of this area varies slightly for System/370 and ECPS:VSE modes.

For both System/370 and ECPS:VSE modes, at IML the customer engineer can activate a microcoded instruction processing function trace facility to trace such items as PSW switching, I/O interruptions, CSWs, specific channel and device interruptions, CAWs, and I/O instructions. When activated, this trace causes a 4K-byte buffer to be defined in auxiliary

storage between the UCWs and channel directories and, thus, increases the size of auxiliary storage by 4K bytes.

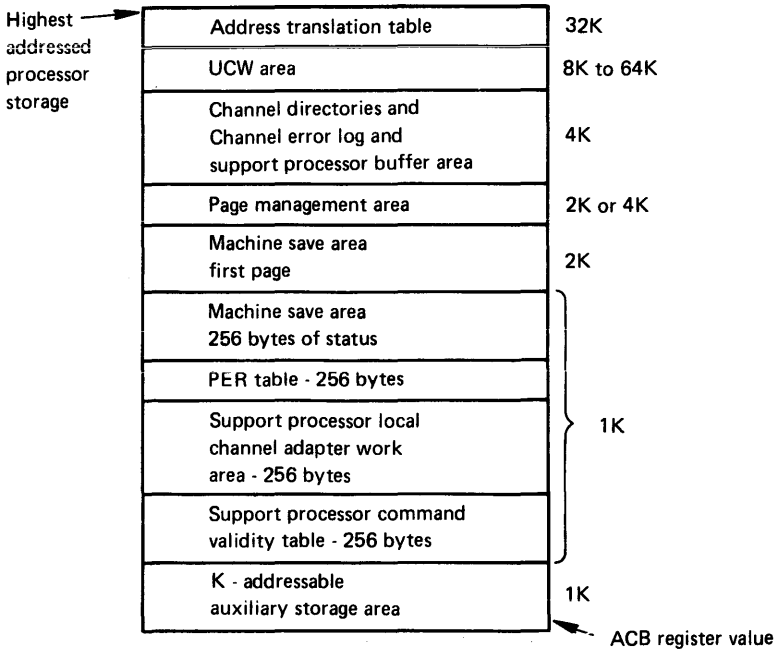


Figure 10.10.2. Layout of auxiliary storage for ECPS:VSE mode

THE STORAGE CONTROL FUNCTION

The storage control function operates under the control of the instruction processing function to handle all access to processor storage. The following components are part of the storage control function:

- High-speed buffer storage and its directory
- The processor TLB for translating virtual storage addresses in instructions to real storage addresses for both System/370 and ECPS:VSE modes (discussed in Section 15)
- The channel TLB for translating virtual storage addresses in channel programs to real storage addresses when ECPS:VSE mode is in effect (discussed in Section 15)
- The key stack that contains one seven-bit key for each 2K of processor storage installed. Each key consists of four access control (store protection) bits, one fetch protection bit, one reference bit, and one change bit. The key stack contains 1024 or 2048 entries for processor storage sizes of two or four megabytes, respectively.
- The ECC logic for processor storage (see Section 50)
- The eight-byte-wide input/output data register that is used to transfer data (1) among the components of the storage control function and (2) between processor storage and the instruction processing function

High-Speed Buffer Storage

The high internal performance of the 4341 Processor is achieved in part by the inclusion of high-speed buffer storage. The 8K buffer is a standard feature and provides high-speed data access for instruction processing function fetches. The instruction processing function can obtain a doubleword of instructions from the buffer in 150 nanoseconds. A doubleword of data can be read from or written to the buffer by the instruction processing function in 225 nanoseconds. If the buffer does not contain the instruction or data needed, the instruction or data must be obtained from processor storage.

Buffer storage control and use are handled entirely by buffer control function hardware and are transparent to the programmer, who need not adhere to any particular program structure in order to obtain close to optimum use of the buffer. Parity checking is used for data verification in the buffer.

When a fetch request is made by the instruction processing function, buffer storage control determines whether or not the requested data is in the high-speed buffer by interrogating the buffer directory, which indicates the current contents of the buffer. If the data requested is present in the buffer and is valid, it is sent directly to the instruction processing function without a processor storage reference.

If the requested data is not currently in the buffer, a processor storage fetch is made and the data obtained is sent to the instruction processing function. The data is also assigned a buffer location and stored in the buffer.

When data is stored by the instruction processing function, the buffer is updated if the contents of the processor storage location being altered is currently being maintained in the buffer. Processor storage is not modified, however, as the buffer in the 4341 Processor is a store-in, rather than a store-through, type of buffer, as discussed later. If the data is not currently being maintained in the buffer, a processor storage fetch is made to obtain the required block of data and load it in the buffer. The store is then made to the just loaded buffer location.

The channels read into and write from processor storage using the input/output data register in the storage control function. When a channel writes data, the data is placed only in processor storage and the buffer directory is interrogated. If data from the affected processor storage address is being maintained in the buffer, appropriate bits are set in the buffer directory to indicate that this buffer data is no longer valid.

When a channel reads data, the buffer directory is interrogated, and if the required data is in the buffer and valid, it is read from the buffer and presented to the channel. If the buffer does not contain the required data, the channel reads the data from processor storage but the data is not placed in the buffer.

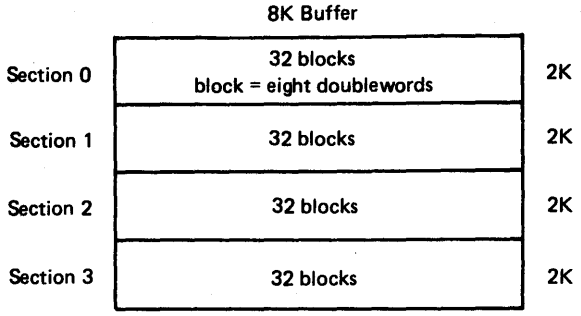
As shown in Figure 10.10.3, the 8K buffer is divided into four sections numbered 0 through 3, each of which is 2K bytes in size. Each 2K section contains 32 buffer block locations, each of which can contain eight doublewords (64 consecutive bytes from processor storage located on a 64-byte boundary). Thus, the buffer can contain 128 blocks of data from processor storage.

For buffer usage purposes, processor storage is also considered to be divided into 2K sections, each of which contains 32 blocks (eight doublewords) of data. The first block of data in each 2K section of processor storage can be placed in the first buffer block location in

any one of the four 2K buffer storage sections. The second block of data in each processor storage section can be placed in the second buffer block location in any of the four buffer storage sections, etc.

The buffer directory, also shown in Figure 10.10.3, is used to maintain knowledge of the current contents of the 128 block locations in the buffer. The directory contains 32 locations that correspond to the 32 buffer blocks in each 2K section of the buffer. Each location contains four directory entries to indicate the contents of the corresponding buffer block in each of the four sections of the buffer. The first directory entry in each location is associated with section 0, the second with section 1, etc.

Also associated with each of the 32 directory locations are six least recently used (LRU) bits that are used when a block of data from processor storage must be assigned a block location in the buffer. The LRU bits for a location are modified as appropriate each time one of the associated buffer blocks is referenced. The setting of these bits indicates which of the four entries is the least recently used.



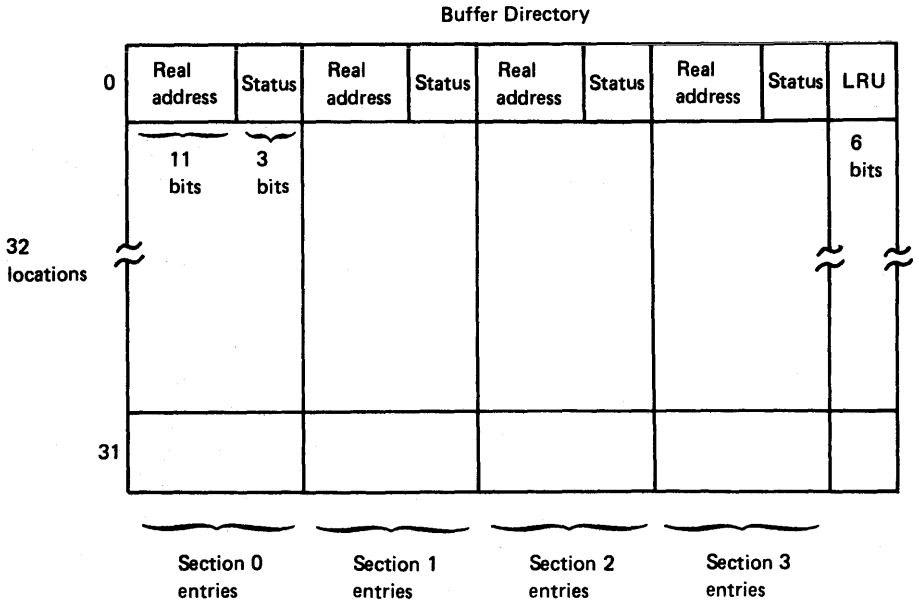


Figure 10.10.3. Buffer organization in the 4341 Processor

Each directory entry contains an eleven-bit real storage address field with an associated parity bit and a three-bit status field with an associated parity bit. The real storage address field contains bits 2 to 12 of the real storage address of the data located in the buffer block location with which the directory entry is associated.

The status field contains one invalid, one modification, and one malfunctioning bit. The invalid bit indicates whether the data in the corresponding buffer block is valid, while the malfunctioning bit indicates whether or not the buffer block is functioning correctly.

The modification bit for a directory entry is set when the instruction processing function stores data in the associated buffer block to indicate that this block has been changed. When the contents of a buffer block are to be replaced, the modification bit is inspected to determine whether the contents of the buffer block must be written to processor storage before the new data is loaded.

The store-in approach used for the high-speed buffer in the 4341 Processor contrasts with the store-through approach used in the high-speed buffers in System/370 processors in which processor storage is altered whenever data is stored in the buffer. The store-in approach increases performance in the situation in which the data in a buffer block is modified more than once or modified but never replaced. The data the block contains is written to processor storage only once and only if the data is to be replaced. The store-in approach becomes more and more advantageous as the difference between processor storage and high-speed buffer storage cycle times becomes greater.

Buffer operation is as follows. When the instruction processing function issues a fetch or store request to processor storage, bits 13 to 17 of the specified real storage address are used to address one of the 32 locations in the buffer directory. Bits 2 to 12 of the real storage address are then compared simultaneously with the real storage address bits in each of the four directory entries in the selected directory location to determine whether the contents of the addressed location are being maintained in the buffer.

If an equal comparison occurs for an entry and its valid bit is on, the requested doubleword is fetched from or stored in the buffer section associated with the entry. Bits 13 to 17 of the real storage address are used to select one of the 32 block locations. Bits 18 to 20 are used to select one of the doublewords within the 64-byte block. For a fetch, the doubleword is sent to the instruction processing function. For a store, the instruction processing function stores the doubleword in the buffer block and turns on the modification bit in the directory entry for the changed block. No processor storage reference is made.

If no directory entry produces an equal comparison, the 64-byte block containing the required data must be fetched from processor storage, assigned a buffer location, and stored in the buffer. To determine the block to assign, the buffer control function first inspects the invalid bit in each section entry in the addressed directory location. If any of the four buffer blocks that can be assigned contains invalid data, that block is assigned to receive the needed block of data from processor storage. If no entry has its invalid bit on, the LRU bits for the addressed directory location determine the buffer block to select.

The modification bit in the directory entry for the selected buffer block is then inspected to determine whether the data in the selected block has been modified. If the bit is off, the required 64-byte block of data is loaded from processor storage into the selected buffer block and the directory is updated (invalid bit is turned off in the directory entry for the selected block and LRU bits are modified as appropriate for the directory entry).

If the modification bit is on in the directory entry for the selected buffer block, the contents of the selected buffer block are written to processor storage before the required block of data from processor storage is loaded into the selected buffer block. The modification bit is turned off. Whether or not processor storage must be updated, the instruction processing function waits until buffer block loading is completed. That is, all 64 bytes are loaded in the assigned block before the required doubleword is fetched or stored.

Buffer block loading requires 2.36 microseconds when the contents of the selected block do not have to be written to processor storage. The time required to store the contents of a buffer block in processor storage and then load the required block of data from processor storage varies from 3.75 to 5.85 microseconds, depending on whether and how soon the instruction processing function makes a subsequent reference to the buffer and does not find the required data.

Note that when the CLEAR PAGE instruction for ECPS:VSE mode is issued, all the directory entries that reference the cleared processor storage page have their invalid bits turned on and no data is written to processor storage from the invalidated buffer blocks.

Operation of the entire buffer cannot be disabled. However, utilization of an individual buffer block can be disabled by turning on the malfunctioning bit in the associated directory entry. Using the operator console, the customer engineer can turn on a malfunctioning bit in the buffer directory.

10:15 THE SUPPORT PROCESSOR SUBSYSTEM

COMPONENTS AND FUNCTIONS

The support processor subsystem provides basic operational functions for the 4341 Processor and is the primary maintenance tool for diagnosing hardware malfunctions. It is designed to maximize total system availability and to provide rapid fault location and repair, where possible.

The components of the support processor subsystem are the support processor, support bus adapter, local channel adapter, console attachment adapters and attached devices, power controller adapter and power information panel, common communication adapter for the remote support facility, and system diskette drive and associated adapter.

The microcoded support processor controls the operation of the support processor subsystem. The support processor subsystem is responsible for the following:

- System initialization functions (IML and IPL), including microcode loading for the support processor and instruction processing function
- Control of the system diskette drive
- Control of the I/O devices that natively attach to the 4341 Processor via the console attachment adapters. These devices include the operator console display and up to three additional display consoles and/or printers.
- Analysis of logout data and the writing of processor logout data and analysis information (reference code) to the system diskette after a machine check occurs and retry of retryable instructions after an error occurs
- Diagnostic program loading and execution

- Microcode controlled power sequencing, power monitoring to detect under- and over-voltage conditions, and electrostatic discharge and temperature monitoring

- Control of the remote support facility

Operation of the support processor is independent of, and overlapped with, operation of the instruction processing function for certain of its functions. During system operation, while instruction execution occurs, the support processor controls the operation of the natively attached display consoles and printers. It also performs power, temperature, and electrostatic discharge monitoring under microcode control and, when necessary, logging to the system diskette of environmental conditions.

Whenever a machine check condition occurs, the support processor receives control to initiate an instruction retry operation or machine check interruption in the instruction processing function. While the instruction processing function is operating, the support processor logs the error to the system diskette and performs error diagnosis to generate a reference code. Details about the last five functions listed are covered in Section 50. The other functions of the support processor subsystem are discussed in the remainder of this subsection.

Details about the last five functions listed are covered in Section 50. The other functions of these support processor subsystem are discussed in the remainder of this subsection.

SYSTEM INITIALIZATION

When the power-on/IML pushbutton on the operator control panel (located on the 3278 Model 2A Display Console) is pressed, a hardwired sequence is activated to power on the support processor, system diskette drive, and adapters connected to the I/O bus of the support processor.

Diagnostics resident in storage of the support processor are executed to test the operation of the support processor and the system diskette drive and its adapter. If these tests execute successfully, the resident microcode for the support processor is loaded from the system diskette. Diagnostics that verify the correct operation of the console attachment adapters, operator console, and power controller adapter are then loaded into the support processor and executed. The power controller adapter is initialized if no errors occur.

When the bootstrap functions have completed successfully, the support processor loads its own control storage and reads the IML program for the instruction processing function from the system diskette. The path to the operator console is tested and finally the microcode-controlled power-on sequence for the balance of the 4341 Processor is initiated. The instruction processing function and channel hardware, Channel-to-Channel Adapter (if installed), and channel-attached I/O devices with their power control switch set to the remote position are powered on, in the sequence listed, by power sequencing microcode.

Powering of the natively attached 3278 Model 2A displays and 3287 printers must be done by the operator. The instruction processing function and the Channel-to-Channel Adapter can be powered off and on individually when CE mode is in effect. CE mode is established using a customer engineer panel that is located within the frames of the 4341 Processor.

If no errors occur during powering, instruction processing function hardware is initialized at the completion of the power-on sequence. The

microcode for the instruction processing function is then loaded into reloadable control storage from the system diskette drive if the installation has established that an automatic IML is to occur at the completion of a power-on (see discussion in Section 12). The local channel adapter is initialized and the general selection display is shown on the operator console.

SYSTEM DISKETTE DRIVE

The system diskette drive is a small read/write drive, that is located in the right-hand end of the 4341 Processor (as shown in Figure 05.15.1). It reads removable prerecorded disk cartridges (diskettes). Recording is done on both sides of the diskette.

A power-on of the 4341 Processor causes the system diskette drive to be turned on and made ready for I/O operations that are required by the initial microcode load of the support processor that follows a power-on.

The operation of the system diskette drive is controlled by command bytes that are interspersed within the data (microcode or diagnostics) contained on the tracks of the diskette. There are no I/O instructions or commands that a user program can execute to cause read or write operations to the system diskette drive.

Several diskettes are sent to each 4341 Processor installation. Two are identical system diskettes (one for backup) and the other four are the diagnostic diskettes. The system diskette contains all the microcode required for the configuration (instruction processing function and support processor microcode), areas for logout data from the 4341 Processor, diagnostic programs, and error analysis programs (see additional discussion of the system diskette in Section 50).

The microcode on a system diskette for the 4341 Processor is not customized. However, the system diskette does contain certain timing values that are specific to the particular 4341 Processor with which it is to be used. The service diskettes contain additional service programs that are to be used by customer engineers. Storage space for diskettes is provided within the frames of the 4341 Processor. Normally, the system diskette will stay mounted on the system diskette drive and diskette changing will occur only when diagnostics are to be performed.

When the system diskette is mounted on the system diskette drive, an IML of instruction processing function microcode can occur automatically after a power-on of the 4341 Processor is performed. The operator can establish this mode of operation using the program load display for the operator console. If an IML for the instruction processing function is required thereafter, it can be performed using the program load display for the operator console. Parity checking is used for reloadable control storage during processor operation.

A procedure exists that enables the customer engineer to temporarily patch the microcode in reloadable control storage or the support processor. Any patches made are also made to the mounted system diskette. Such patches are included in the microcode provided on the next level of system diskette sent to the installation.

Note that when processor power is turned off, the data in control storage for the support processor, processor storage, and control storage for the instruction processing function is lost, and an IML must be performed when power is turned on again.

A system diskette for a given 4341 Processor contains the processor serial number and is not portable from one 4341 Processor to another

(since the serial number on the diskette is checked against the processor serial number during any IML and a mismatch causes termination of the IML procedure). The system diskette for a given 4341 Processor also contains configuration information specific to that 4341 Processor that is written by the customer engineer (such as UCW assignments) and that precludes portability.

The system diskette drive is also used for loading and executing diagnostic routines, and it is a basic debugging tool for the system. A comprehensive set of fault-locating diagnostic routines is supplied to each 4341 Processor installation on service diskettes. These routines can be loaded directly from the system diskette drive into the 4341 Processor and executed (see Section 50:15).

NATIVELY ATTACHED DEVICES

Up to four devices can be natively attached to the 4341 Processor via the I/O bus of the support processor. The following devices can be natively attached:

- Required 3278 Model 2A Display Console (with operator control panel)
- Up to three additional devices, which can be any combination of 3278 Model 2A Display Consoles and 3287 Model 1 or 2 Printers.

The additional 3278 Model 2A displays can be used as alternate and/or additional consoles, as supported by the operating system utilized. The additional displays cannot have the operator control panel that is located on the primary operator console. The 3287 Printers can be used for hard-copy backup of the 3278 Model 2A displays. The 3287 Printer is a desktop printer with a print speed of 80 characters per second (Model 1) or 120 characters per second (Model 2).

The natively attached devices attach to channel 0 in the 4341 Processor via the local channel adapter and must have specific addresses assigned (see discussion in Section 10:20).

SUPPORT BUS ADAPTER

The support bus adapter provides an interface between the support processor and the instruction processing function and channel hardware. Via this direct path, the support processor can access maintenance hardware in the instruction processing function and the channels.

The support bus adapter also provides an interval timing facility for power monitoring microcode in the support processor by informing this module that it is time for it to execute. This adapter also controls the system and wait indicators and the lamp test switch on the operator control panel.

10:20 CHANNELS

GENERAL DESCRIPTION

While channel functions compatible with those available on Models 50 and 65 are provided, the 4341 Processor also offers additional facilities (such as block multiplexing), faster channel data rates, and attachment of faster direct access devices with larger capacities. These capabilities enable the user to tailor a 4341 Processor configuration to I/O processing needs, on an improved price performance basis, to increase channel throughput.

One standard and one optional channel group are provided for the 4341 Processor. The standard channel group consists of one byte multiplexer channel, addressed as channel 0, and two block multiplexer channels, addressed as channels 1 and 2. The optional channel group (Block Multiplexer Channels, Additional feature) provides (1) three additional block multiplexer channels, addressed as channels 3, 4, and 5 or (2) one byte multiplexer channel addressed as channel 4 and two block multiplexer channels addressed as channels 3 and 5.

The byte multiplexer channels for the 4341 Processor, like that for System/360 Models 50 and 65, can handle the concurrent operation of multiple slower speed devices when operating in byte interleave mode. The block multiplexer channels, not available for the Models 50 and 65, are designed to increase system throughput by increasing the amount of data entering and leaving the system in a given period of time (the effective data rate).

A single block multiplexer channel can support interleaved, concurrent execution of multiple high-speed channel programs. A block multiplexer channel can be shared by multiple high-speed I/O devices operating concurrently, just as the byte multiplexer channel can be shared by multiple low-speed devices.

Each installed channel can have up to eight control units attached. For byte multiplexer channel 0, one control unit position is used by the local channel adapter. This internal adapter provides attachment of support processor subsystem devices to this byte multiplexer channel.

Without the optional Channel Control Units, Additional feature installed, the 4341 Processor can have up to 23 external control units attached in addition to the internal local channel adapter (the maximum number attachable to the standard number of channels). When the optional channel group is installed, the Channel Control Units, Additional feature may be required to provide the ability to attach more than 23 external control units (up to 47 maximum). The optional channel group must be present in order to install the Channel Control Units, Additional feature.

Comprehensive error checking is incorporated in the basic design of the channel hardware. Checking is performed on the control logic in most areas, and standard parity checking is done on the data flow between the channels and instruction processing function. Improved error recovery data is provided by the channels (discussed fully in Section 50).

The standard instruction set for the 4341 Processor includes three I/O instructions not provided for System/360: HALT DEVICE, CLEAR I/O, and START I/O FAST RELEASE. HALT DEVICE is specifically designed to stop an I/O operation on a particular device on a byte or block multiplexer channel without interfering with other I/O operations in progress on the channel. HALT DEVICE, instead of HALT I/O, should always be used to stop an I/O operation on a multiplexer channel.

The CLEAR I/O instruction is provided to reset byte and block multiplexer subchannels when errors and control unit lockups occur that could cause processor termination. START I/O FAST RELEASE is implemented as a START I/O instruction in the 4341 Processor.

Optionally one Channel-to-Channel Adapter can be installed in a 4341 Processor and attached to any block multiplexer channel. The other channel to which the adapter is attached can be contained in a System/360, System/370, or 4341 processor. One control unit position and one nonshared UCW for each of the two channels interconnected via the adapter are required. The adapter operates in burst mode and

transfers data at the rate of the lower speed channel to which it is attached.

DEVICE ADDRESSES AND UNIT CONTROL WORDS

The byte multiplexer channel and each block multiplexer channel installed can have 256 device addresses (00 to FF). Any device addresses can be used for channels 1 to 5. For the byte multiplexer channel 0, addresses 0F0 to 0FF are reserved for support processor subsystem devices attached via the local channel adapter and any device addresses other than these can be used for the I/O devices attached to the byte multiplexer channel via external control units.

The 4341 Processor can have a minimum of 128 and a maximum of 1024 UCWs as a standard feature. UCWs are allocated by the customer engineer, using the display console. UCWs above 128 are allocated in groups of 32. Each UCW is 64 bytes in size and resides in auxiliary storage. Each group of 32 UCWs requires 2K bytes of storage.

The UCWs allocated are assigned a three-digit reference number 000 to N-1, where N is the number of UCWs allocated. UCWs with reference numbers 000 to 00F are reserved for internal functions (system diskette drive and support processor, for example) and support subsystem devices.

The UCWs defined are shared by all the channels actually present in the processor. A maximum of 256 can be assigned to any one channel. The customer engineer assigns UCWs to specific channel addresses using the console (display/alter display). Each UCW can be designated as shared or nonshared.

A shared UCW can be used by a set of devices, one device at a time. A shared UCW generally is assigned to a control unit that has multiple devices attached, only one of which can be in operation at a time. A nonshared UCW is one that is assigned to only one device. A nonshared UCW is designed for use with a control unit that has only one I/O device attached or that has multiple I/O devices attached that can operate concurrently.

A channel directory for each channel is allocated in auxiliary storage. Each directory has 256 entries, one for each of the possible device addresses for a channel. A directory entry indicates whether a UCW is assigned to the associated device address, characteristics of the assigned UCW, and characteristics of the device assigned the associated device address.

A channel directory entry contains the following:

- Valid bit to indicate whether or not a UCW is assigned to the associated device address
- Reference number of the UCW assigned, if any
- An indication of whether the UCW is shared or nonshared (shared bit)
- An indication of whether the device associated with this entry can operate in block multiplexer mode (disconnect during command-chained channel programs)--DCC bit

Devices attached to a block multiplexer channel that are capable of block multiplexing should have the shared bit off and DCC bit on in their channel directory entry to indicate allocation of a nonshared UCW that is capable of disconnection. Devices such as the 3272 Control Unit should be assigned a shared UCW that is capable of disconnecting (shared and DCC bits on in the appropriate channel directory entries).

The display/alter display for the operator console is used by the customer engineer to select functions associated with UCWs. The functions provided enable the customer to display the allocated UCW reference numbers and the device addresses they are assigned, and to display and alter the contents of a channel directory. The alter capability is used to assign device addresses and attributes to UCWs.

GENERAL OPERATION OF THE CHANNELS

The channels in the 4341 Processor are microcode- and hardware-controlled. They are integrated channels and, thus, share the use of certain hardware with the instruction processing function, such as the arithmetic logic unit, byte shifter, and control storage.

The general flow of data between I/O devices and processor storage via the channels is shown in Figure 10.20.1. Each installed channel has an interface controller that contains a data-in and a data-out register for transferring data between the standard I/O interface to I/O devices. All the interface controllers can be transferring data to I/O devices (one device per controller) at the same time.

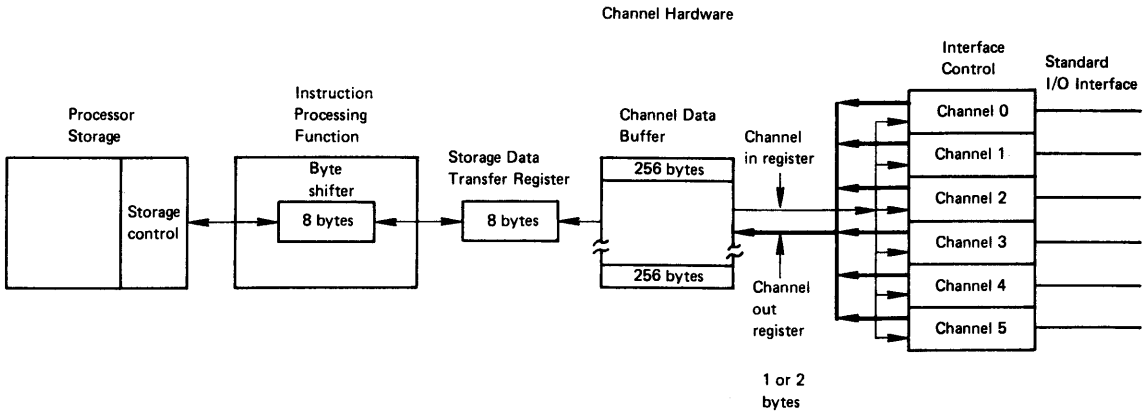


Figure 10.20.1. General flow of data between the channels and processor storage

Data is transferred between the individual interface controllers and the channel data buffer via a channel-in and channel-out register, each of which is two bytes in size. One or two bytes are transferred at a time. The channel data buffer contains one 256-byte buffer area for each channel. Only one channel can be transferring data to, or receiving data from, the channel data buffer at a time. A set sequence for handling channel requests is implemented in the channel control hardware.

Data is transferred between a buffer area in the channel data buffer and processor storage via the eight-byte data transfer register and the eight-byte shifter, which is in the instruction processing function. This data transfer is microcode-controlled. The shifter provides doubleword boundary alignment for data entering processor storage, when required, and any needed alignment for eight bytes of data entering the channel data buffer.

A data transfer between processor storage and the channel data buffer handles 64 bytes aligned on a 64-byte boundary, except for beginning and ending transfers for a processor storage buffer that is not located on a 64-byte boundary. A 64-byte data transfer requires four microseconds.

Channel control hardware determines the priority for servicing the channels according to predetermined priorities. When multiple channel trap requests (requests for microcode service) are outstanding, the lowest numbered channel with an outstanding request is serviced first. A trap request for this channel will not be serviced again until the other channels with a request outstanding have one trap request serviced. That is, each channel is guaranteed not to have to wait for the servicing of more than an average of two (if the optional channels are not installed) or five (if the optional channels are installed) other trap requests between the servicing of two successive trap requests of its own (each channel is guaranteed, on an average, every third or sixth trap service).

The channels are given priority over the instruction processing function for access to shared facilities. The channels interfere with instruction processing function operation when a channel trap request is serviced. Trap requests occur for such operations as data transfer between processor storage and the channel data buffer, processing of a UCW, command chaining, data chaining, and status handling.

The 4341 Processor generates less total interference with instruction execution than intermediate-scale System/360 and System/370 processors because the amount of time required to transfer a byte of data between processor storage and the channel data buffer during an I/O operation is much less (64 bytes are transferred in 4 microseconds in the 4341 Processor versus 4 bytes transferred to or from processor storage in .54 microseconds in the Model 148, for example).

The channels in the 4341 Processor do not prefetch CCWs for input operations. For output operations, one CCW and up to 128 bytes of associated data are prefetched.

BYTE MULTIPLEXER CHANNELS

The byte multiplexer channels for the 4341 Processor are functionally identical to the byte multiplexer channels for System/360 and System/370 processors. A byte multiplexer channel can operate in byte interleave mode to permit several slower speed I/O devices to operate concurrently or in burst mode to permit one buffered device to operate.

For input, a maximum of up to 1 MB/sec for channel 0 and up to 2 MB/sec for channel 4 is possible for a burst mode operation involving a buffered device. The effective burst mode data rate for an output operation involving a buffered output device must be calculated. This rate equals 1000 times the data rate of the device in KB/sec divided by 1000 plus the data rate of the device in KB/sec.

Table 10.20.1. Maximum byte mode data rates for the byte multiplexer channels in the 4341 Processor

Type of Transfer	Maximum data rate when only byte multiplexer channel 0 is active(KB/sec)	Maximum data rate when only byte multiplexer channel 4 is active (KB/sec)
Single-byte transfer	16	22
Two-byte transfer	32	44
Four-byte transfer	64	88

multiplexer channels are present, their maximum aggregate data rate is 9 MB/sec.

Like a byte multiplexer, a block multiplexer channel can have multiple subchannels, each of which can support one I/O operation. The setting of a channel mode bit (0) in control register 0 determines whether the addressed subchannel of a block multiplexer channel operates in block multiplexer (assuming it is capable of operating in block multiplexer mode) or selector mode when a START I/O instruction is issued. The mode bit is set to 0 (selector mode) at IPL and can be altered by programming at any time.

This page intentionally left blank.

The data rate for byte mode operation depends on other channel activity and the number of bytes transferred per burst. The maximum data rate can be achieved when there is no activity on any other channel or a console device. Table 10.20.1 gives the maximum byte multiplexer data rates for byte mode operations.

I/O devices in the support processor subsystem attach to byte multiplexer channel 0 via the local channel adapter, which occupies the last control unit position on this channel. Thus, a maximum of seven external control units can be attached to byte multiplexer channel 0.

The local channel adapter operates as a channel-to-channel adapter that connects the I/O bus of the support processor to channel 0. The local channel adapter provides a low-cost method of attaching support processor subsystem devices to byte multiplexer channel 0.

The local channel adapter appears as a shared control unit that can have multiple device addresses. It operates in multibyte mode in the 4341 Processor and has a maximum data rate of 24 KB/sec. Data is transferred from the local channel adapter to the byte multiplexer channel two bytes at a time.

I/O devices attached to the byte multiplexer channel must be assigned addresses in the range 000 to OEF. Addresses 0F0 to OFF are reserved for internal assignments and cannot be used for I/O devices attached to the byte multiplexer channel via control units or natively.

BLOCK MULTIPLEXER CHANNELS

Block multiplexer channels 1 to 5 in the 4341 Processor can operate in block multiplexer or selector mode. When operating in selector mode, a block multiplexer channel in the 4341 Processor is functionally equivalent to the selector channels for System/360 and System/370 processors. When a block multiplexer channel in the 4341 Processor operates in block multiplexer mode, it is functionally equivalent to a System/370 block multiplexer channel. A block multiplexer channel presents a standard I/O interface and can have a maximum of eight control units attached.

The maximum data rate for block multiplexer channels 1, 2, 3, and 4 is 2 MB/sec each. Block multiplexer channel 5 has a maximum data rate of 1 MB/sec. The maximum aggregate data rate of the block multiplexer channels installed is the sum of the individual maximum data rates. Thus, when the optional channels are not present, the maximum aggregate data rate of channels 1 and 2 is 4 MB/sec. When four block multiplexer channels are present (channel 4 is a byte multiplexer channel), their maximum aggregate data rate is 7 MB/sec. When all five block

Block Multiplexer Channel Operation

A block multiplexer channel functions differently from a selector channel in the way in which it handles command-chained channel programs. A selector channel or a block multiplexer channel operating in selector mode executing a command-chained channel program is busy during the entire time the channel program is in operation, whether or not data transfer is occurring. A block multiplexer channel operating in block multiplexer mode and executing a command-chained channel program has the ability to disconnect from the operational channel program during certain non-data-transfer operations. That is, a block multiplexer channel can be freed during a nonproductive activity, for example, during disk seeking and most record positioning, thereby allowing more data to be transferred per unit of channel busy time.

Block multiplexing operates as follows. Assume a block multiplexer channel is executing a channel program consisting of multiple command-chained CCWs. When channel end is presented without concurrent device end, the channel disconnects from the I/O device and becomes available for an I/O operation on another device--even though the channel program of the disconnected device is not complete. At channel disconnect time, the subchannel and the control unit for the device retain the information necessary to restart the disconnected channel program.

When the device signals that it is again ready for the channel (by presenting device end), its control unit attempts to regain use of the channel. If the channel is free at this time, the channel registers are reloaded with the information previously saved (in the UCW for the device), and the disconnected channel program is resumed at the appropriate CCW. If the channel is busy when reconnection is requested, the device must wait until the channel becomes available. Once multiple channel programs have been initiated on one channel, the interleaving of data transfer operations is controlled by block multiplexer channel hardware and the control units of the devices operating in block multiplexing mode.

To facilitate channel scheduling on block multiplexer channels, an interruption condition, called channel available, is defined. At disconnect time for a channel program, the block multiplexer channel is available for the resumption of an uncompleted channel program previously started, or another channel program can be initiated. A channel available interruption occurs at disconnect time to indicate channel availability if a START I/O, TEST I/O, TEST CHANNEL, or HALT DEVICE instruction was issued previously while the block multiplexer channel was busy.

Two additional facts should be noted about block multiplexer channel operations:

1. When multiple channel programs are operating concurrently in block multiplexing mode, a device can regain control of the channel only when the channel is not busy. Thus, only cyclic devices (such as direct access devices with rotational position sensing capability) or buffered devices (such as the 3505 Card Reader and the 3203 Printer) can disconnect during the execution of a command-chained channel program on a block multiplexer channel and resume operation later.

2. Data transfer operations for concurrently operating devices on a block multiplexer channel are interleaved on a first-come, first-served basis as the desired records become available. Thus, devices are serviced in the order in which their records become available, not necessarily in the order in which their channel programs are initiated.

Examples of devices that can block multiplex on the 4341 Processor when attached to a nonshared subchannel of a block multiplexer channel are:

- 3330-series, 3340/3344, and 3350 disk storage. One UCW per drive in each string is required.
- 3370 Direct Access Storage. One UCW per logical device address (two UCWs per 3370 drive) are required.
- 2305 disk storage. Eight UCWs per drive are required.
- 2540 Card Read Punch. One UCW for the reader and one for the punch are required.
- 3505 Card Reader and 3525 Card Punch. One UCW for each reader and for each punch is required.
- 3203 Model 5 Printers. One UCW per printer is required.
- 1403 Printers attached to a 2821 Control Unit. One UCW per printer is required.
- 3211 Printer. One UCW per printer is required.

When attached to the 4341 Processor, magnetic tape units and direct access devices without rotational position sensing capability, such as the 2311, 2314, and 2319, should be associated with a shared UCW of the block multiplexer channel. Selector mode (DCC bit off) should be assigned for these shared UCWs.

Each 3272 Control Unit attached to the block multiplexer channel in the 4341 Processor should be assigned a shared UCW that is set to operate in block multiplexer mode. While only one of the devices attached to a 3272 can operate at a time, the 3272 can disconnect from the channel during certain operations. Thus, when shared UCWs with block multiplexing capability are assigned to 3272 Control Units, multiple 3272 units can have a channel program executing concurrently.

The following summarizes how direct access devices without rotational position sensing capability and other I/O devices operate on a block multiplexer channel for the 4341 Processor when executing a command-chained channel program:

1. Direct access devices without rotational position sensing capability (2311, 2314, and 2319) assigned to a nonshared or shared UCW operate in the same way whether the channel is in block multiplexer or selector mode. That is, the channel and the disk control unit are busy during the entire time a command-chained disk channel program is in operation. Thus, there is no disconnection after a chained seek.
2. All tape drives attached to a shared or nonshared UCW operate exactly the same whether the channel is in block multiplexer or selector mode. That is, they do not block multiplex and the channel is busy during the entire time a command-chained channel program is in operation.
3. Buffered card and print devices (or devices operating with buffered control units), such as the 1442, 2501, 2520, 2540, 3505, 3525, 1403, 1443, 3203, 3211, and 3800, disconnect during the mechanical motion of the device when assigned to a nonshared UCW. Reconnection occurs later to fill or empty the associated buffer.

For example, a 3203 Model 5 Printer assigned to a nonshared UCW on a channel operating in block multiplexing mode disconnects from the channel during print time and carriage motion. Reconnection occurs when the channel is free to transfer the data for the next line to the buffer in the control function in burst mode.

4. The following control units and I/O devices are not capable of block multiplexing (do not disconnect during command-chained channel programs): 1419, 2250, and 2701 units.

Performance is degraded if a device that is capable of block multiplexing is not assigned a nonshared UCW of a block multiplexer channel. Table 10.20.2 indicates, for the most frequently used I/O devices, the (1) type of channel to which the device can be attached, (2) preferred type of channel to which the device should be attached, and (3) preferred UCW type when the device is attached to a block multiplexer channel operating in block multiplexer mode. Where selector is specified as the channel type, it means the block multiplexer channel operating in selector mode.

Table 10.20.2. Channel attachment and UCW mode for frequently used I/O devices for the 4341 Processor

I/O Device or Control Unit	Channel Attachment Capability	Recommended Channel Type	Recommended or Required UCW Mode When Attached to a Block Multiplexer Channel
Card Readers, Card Punches, and Printers			
1442/1443	Byte, Selector, Block	Byte	Nonshared
2501/2520	Byte, Selector, Block	Byte	Nonshared
3203 Model 5	Byte, Selector, Block	Block	Nonshared
2821/3811	Byte, Selector, Block	Block	Nonshared
3505/3525	Byte, Selector, Block	Block	Nonshared
Magnetic Character Readers			
1255/1259	Byte	Byte	-
1419	Byte	Byte	-
3890	Byte, Block	Block	Nonshared
Optical Character Readers			
1287/1288	Byte, Block	Byte	Nonshared
3881	Byte	Byte	-
3886	Byte, Selector, Block	Byte	Nonshared

Table 10.20.2 (continued)

I/O Device or Control Unit	Channel Attachment Capability	Recommended Channel Type	Recommended or Required UCW Mode When Attached to a Block Multiplexer Channel
Displays			
2250	Byte, Selector, Block	Byte	Shared in selector mode
3272	Byte, Selector, Block	Block	Shared in block multiplexer mode
Magnetic Tape			
2415	Selector, Block	Selector	Shared in selector mode
2803/3803	Selector, Block	Selector	Shared in selector mode
3411	Selector, Block	Selector	Shared in selector mode
Direct Access Storage			
2314/2319	Selector, Block	Selector	Shared in selector mode
2841	Selector, Block	Selector	Shared in selector mode
2835	Block	Block	Nonshared
3830	Block	Block	Nonshared
3880	Block	Block	Nonshared
Communications			
2701	Byte, Selector, Block	Byte	Nonshared
3704	Byte	Byte	-
3705-Channel Adapter Type 1	Byte	Byte	-
3705-Channel Adapter Type 2, 3, or 4	Byte, Selector, Block	Block	Nonshared

10:25 BLOCK MULTIPLEXING OPERATIONS WITH COUNT, KEY, DATA ROTATIONAL POSITION SENSING DEVICES

Rotational position sensing, multiple requesting, and block multiplexing are designed to increase system throughput by increasing channel throughput. The rotational position sensing function is provided for certain count, key, data format direct access devices (such as 3330-series, 3340/3344, 3350, and 2305) to enable them to utilize block multiplexing more effectively. This function is also implemented in fixed block architecture devices, such as the 3370.

This subsection discusses the operation and advantages of the rotational position sensing function for count, key, data devices while Section 10:30 discusses the operation and advantages of fixed block architecture disk devices.

The presence of the rotational position sensing (RPS) function in the control unit of a direct access device enables it to operate in block multiplexing mode. The use of rotational position sensing reduces the number of channel programs that have to be initiated for direct access devices that require an arm-positioning seek, frees a channel more often during direct access device operations--specifically, during most of the time required to position a head assembly to a desired record--and permits disk channel programs to be initiated sooner on a block multiplexer channel than is possible with a selector channel.

Multiple requesting is implemented in a direct access device control unit to enable it to handle concurrent execution of multiple RPS channel programs. The 3830 Model 2 Storage Control unit, for example, can simultaneously control 32 RPS channel programs, one on each of its attached disk drives.

In order to overlap seek operations for movable arm direct access devices without RPS, channel scheduling routines must initiate two channel programs for each record read or written. The first is a standalone seek, which frees the channel as soon as the control unit accepts the seek address. (The control unit is also free during arm movement.) At the completion of the seek, a device-end interruption is presented, and the data transfer channel program is subsequently initiated to search for the desired record and transfer the data.

A selector channel is busy during the entire search operation (execution of the SEARCH command by the control unit) that locates the desired disk record on the track. Search time can be significantly greater than data transfer time for disk records smaller than half a track in size. Search time averages one-half of a rotation for a read or write (8.3 ms for a 3330-series drive) and requires a full rotation, less record write time, for a write verification chained from a write.

Use of RPS reduces the time the channel is busy during the search for a disk record. It permits the SEARCH command to be initiated just before the desired record is to come under the read/write heads, that is, when the desired rotational position is reached. To accomplish this, a "sector" concept is employed. The tracks in each cylinder of a direct access device are considered to consist of equally spaced sectors (the number of sectors varies by device). Track formatting is unchanged but each record has a sector location as well as a record address.

A sector is not physically indicated on the disk tracks of count, key, data devices, but is the length of the track arc that passes under the read/write heads in one sector time. For 3330-series drives, for example, sector time is defined to be approximately 130 microseconds. Thus, there are 128 sectors per logical track.

A disk control unit with RPS and multiple requesting can determine the sector currently under the heads of each of its drives. A sector counter is contained in each drive. The counter is incremented once every sector time period and set to zero each time the index marker passes under the heads. The sector in which a record falls is a function of the length of all records that precede it and of its sequential position on the track. Therefore, sector location can be calculated for fixed-length records.

Two disk commands, SET SECTOR and READ SECTOR, are provided for use with rotational position sensing. If the sector address of a record is known or can be calculated, a SET SECTOR command can be included in the disk channel program to cause the control unit to look for the designated sector. Once the control unit accepts the sector number provided by a SET SECTOR command, both the block multiplexer channel and the disk control unit disconnect and are available for another I/O operation.

When SET SECTOR is used for positioning, the time the channel is busy during the search for a record is reduced from an average of 8.3 ms to an average of 260 microseconds for the 3330-series. (Allowing for the worst case of speed variation and for disk pack interchange, the search time for a record, from sector found to beginning of desired record, can vary from 120 to 380 microseconds for a 3330-series drive.)

The READ SECTOR command is useful for sequential disk processing and for write verification. When chained from a READ, WRITE, or SEARCH command, READ SECTOR provides the sector number required to access the record processed by the previous CCW. This sector number can be used to reposition the track in order to verify the record just written or in order to read or write the next sequential record. These two sector commands, used in conjunction with the block multiplexer channel, permit a single command-chained channel program, which frees the channel and disk control unit during seek and rotational positioning operations, to be initiated for each disk operation.

When the record ID is known, the two channel programs shown below illustrate direct retrieval of a record from a file/data set on a direct access device without RPS, such as the 2314 (key was not written). The seek operation can be overlapped with other seeks and one data transfer operation on the same selector channel. (Commands shown in the sample channel programs that follow are only those that illustrate the advantage of RPS. Thus, commands such as SEEK HEAD and SET FILE MASK, which are used by data management to ensure correct operation, are not shown.)

Channel program 1. Initiate the standalone seek to position the disk arm.

Command		Selector Channel and Disk Control Unit Status
SEEK	(Seek address)	Free as soon as the control unit accepts the seek address

Channel program 2. Initiate the data transfer operation after the seek is complete.

Command Chaining Flag	Command		Selector Channel and Disk Control Unit Status
CC	SEARCH ID EQ	(ID - sequential position on the track)	Busy (for 12.5 ms on the average for a 2314)
CC	TIC	(Back to search if ID not equal)	
	READ DATA	(Processor storage address of input area)	Busy

When the sector address is known or can be calculated, the channel program below illustrates direct retrieval of a record from the same file/data set on a 3330 drive with RPS installed that is attached to a block multiplexer channel. The records are fixed-length standard format, and sector numbers are calculated from record ID (by data management).

Channel program 1. Initiate the seek and data transfer operation.

Command Chaining Flag	Command		Block Multiplexer Channel and Disk Control Unit Status
CC	SEEK	(Seek address)	Free during arm motion
CC	SET SECTOR	(Sector number of sector preceding desired record)	Free until sector found
CC	SEARCH ID EQ	(ID - sequential position on track)	Busy (260 microseconds) average for a 3330)
CC	TIC	(Back to search if ID is not equal. With the logic shown, the first ID inspected normally is that of the desired record, and the TIC command is not executed.)	Busy
	READ DATA	(Processor storage address of input area)	Busy

The preceding example indicates the advantages of rotational position sensing and block multiplexing:

- Only one channel program is required to locate a disk record and transfer the data, thereby eliminating a standalone-seek I/O interruption and the I/O supervisor processing required to schedule a data transfer channel program. A channel-available interruption may occur, however, during channel program execution.
- The channel and disk control unit are free during arm motion and rotational positioning, allowing other I/O operations on that control unit and channel to be overlapped with the seek and rotational positioning operations in progress. Implementation of multiple requesting permits a disk control unit to control concurrent execution of multiple RPS channel programs in order to overlap seek and set sector operations for its drives.

Performance improvement gains achieved on a block multiplexer channel are not due entirely to the fact that direct access device rotational delays are overlapped. Also important is the ability to initiate seek commands a number of milliseconds earlier, because a block multiplexer channel is free. The initiation of standalone seeks on a selector channel is delayed during search and data transfer operations. On a block multiplexer channel, seeks can be initiated during rotational positioning, since the channel and disk control unit are not busy.

10:30 BLOCK MULTIPLEXING OPERATIONS WITH FIXED BLOCK ARCHITECTURE DEVICES

FIXED BLOCK ARCHITECTURE DESIGN

Fixed block architecture (FBA) devices are specifically designed to take advantage of the block multiplexing capability. They have rotational position sensing capability like 33XX count, key, data (CKD) format direct access devices and, therefore, offer all the advantages of the RPS capability, as discussed in Section 10:25.

However, FBA devices provide advantages over count, key, data devices. Specifically, they require a less costly storage control function to control their operation and they facilitate data mobility via their utilization of device-independent channel programs. These advantages are discussed in more detail at the end of this subsection.

TRACK FORMATTING

Data is recorded in fixed-length blocks of 512 bytes on the tracks of an FBA disk device instead of in the self-formatting record format used for CKD disk devices. Primary data blocks are addressed using a linear binary value (relative block address) that ranges from 0 to N-1, where N is the number of fixed-length blocks on the device. These relative block addresses are used in the channel programs for FBA architecture devices.

The basic unit of transfer between an FBA device and the channel is the addressable, fixed-length block, one or more of which can be read or written using a single read or write command. For 3370 disks, for example, an addressable block is physically recorded on a track as two fields separated by a small gap. The first field is a fixed-length block control field, which is an identification area for the fixed-length data block field of 512 bytes that follows.

The identification area provides control information for the data block (primary or alternate block, nondefective or defective block, alternate assigned, for example) and uniquely identifies the data block in terms of its physical location on the direct access device (contains the cylinder, head, and block number of the block). The identification field also contains error detection bytes while the data block field contains error detection and correction bytes.

A data set/file on an FBA device, whether contained on one or multiple extents or one or more disk volumes, is also addressed in channel programs via relative block addresses.

Alternate data blocks that can be assigned to replace defective primary blocks can be placed anywhere on the disk device and do not have to be located in the last cylinders. Alternates are assigned on a block basis, instead of on the track basis used for CKD devices.

COMMAND SET

While the command set for count, key, data format disks consists of over 70 commands, only 16 commands are defined to control the operation of FBA disk devices. Four of these commands (DEFINE EXTENT, LOCATE, READ, and WRITE) are used for all normal (nondiagnostic) read and write operations. Set file mask, seek, search, and sector commands, such as those used for CKD devices, are not defined for protection and record positioning in fixed block architecture. The DEFINE EXTENT and LOCATE control commands are provided in place of these commands.

As a result of the differences in the command sets utilized by CKD and FBA devices, channel programs for CKD and FBA disk devices are not compatible.

The following command-chained channel program is used to read or write one or more data blocks contained on an FBA device:

Command	Flag	Specification	Function
DEFINE EXTENT	CC	Command types that can be used, bounds of the extent (relative to the origin of the data set) that can be accessed in the remainder of the channel program, and location of the extent (relative to the beginning of the disk device)	Provides extent location and protection information used by the storage control function
LOCATE	CC	Location of the first data block to be accessed (relative to the beginning of the data set), number of data blocks to be processed, and the specific read/write operation to be performed when the following command is executed	Causes required seeking and rotational positioning to first data block to be accessed and orients the storage control function for a specific read or write operation. Storage control function is disconnected from channel during execution of LOCATE (seek and block locate) functions.
READ/WRITE		I/O buffer in processor storage and number of bytes to be transferred	Causes the transfer of data for the operation specified in the LOCATE command. One or more data blocks are transferred between the disk device and the processor via a channel.

The DEFINE EXTENT command transfers 16 bytes of data to the storage control function. DEFINE EXTENT specifies protection mask and extent information. The mask byte is used to inhibit or permit the use of certain commands in the balance of the channel program and to indicate that the data area or CE area is to be accessed. Execution of all write operations (formatting and nonformatting writes) and/or diagnostic commands (DIAGNOSTIC CONTROL and DIAGNOSTIC SENSE) can be inhibited or permitted. Format write subcommands (Format Defective Block operations) can also be inhibited separately from normal write commands.

A block size field is provided in the DEFINE EXTENT command for use by 3880 Storage Control in calculating the number of bytes to be transferred. Block size must be 512 or 0, where 0 is interpreted as a default for 512.

The extent information in a DEFINE EXTENT command consists of the range of data blocks that can be accessed by the subsequent chained commands in the channel program and the location of the first block of the extent on the disk device. The beginning and ending blocks of the

extent of data that can be accessed are defined relative to the data set itself.

For example, if a data set consists of 300 blocks contained in two extents of 200 blocks and 100 blocks and the second extent is to be accessed, the DEFINE EXTENT command would specify blocks 200 to 299. The actual location on the direct access device of the beginning of the second extent is specified in terms of a relative block address. Thus, if the second extent begins with the thousandth block on the disk device, the DEFINE EXTENT command would specify 999 as the location of the first block of the second extent.

If the parameters specified in the DEFINE EXTENT command are valid, they are saved by the storage control function for use in processing the balance of the channel program. The parameters are retained until the channel program completes. Only one DEFINE EXTENT command is permitted in a channel program. There is no disconnection from the channel during the execution of a DEFINE EXTENT command.

The DEFINE EXTENT command includes the functions of the SET FILE MASK command for CKD disk devices in that it permits or inhibits the operation of certain commands. However, DEFINE EXTENT also specifies data set/file protection information that is utilized by the storage control function during channel program execution. The entire range of data to be accessed can be specified regardless of internal device boundaries. For CKD devices, the protection specification is limited to a track or cylinder boundary.

The LOCATE command transfers eight bytes of data to the storage control function. It specifies the location and amount of data to be processed and the specific type of read or write operation (subcommand) that is to be performed when the following data transfer command is executed. LOCATE also establishes read or write orientation for the control function. It must be command-chained from a DEFINE EXTENT or READ IPL command. The latter command is provided to perform the initial program load function from disk.

The LOCATE command specifies the first or only data block to be processed by the subsequent commands in the channel program relative to the beginning of the data set and indicates the number of sequential data blocks to be processed. Access to a maximum of 65,535 consecutive blocks can be specified in a LOCATE command. LOCATE also specifies a subcommand: Read Data, Read Replicated Data, Write Data, Write and Check Data, or Format Defective Block.

For all subcommands except Format Defective Block, the subcommand specified in a LOCATE command is actually executed when the READ or WRITE command is processed. When Format Defective Block is specified, it is executed as part of LOCATE command processing and a read or write command does not follow the LOCATE command.

The storage control function checks the validity of the LOCATE parameters (all blocks to be accessed are within the defined extent, command type is not inhibited, etc.) and then performs the required seek and record positioning functions. The storage control function is also oriented for the specified read or write operation.

In order to position the access mechanism to the block specified in the LOCATE command, the storage control function calculates the actual location of the required block on the direct access device (cylinder, head, and block numbers) utilizing the relative block addresses provided in the DEFINE EXTENT and LOCATE commands and the physical characteristics of the disk device being accessed. The number of bytes to be transferred is also calculated by the storage control function using the block size of 512 and block count value.

Disconnection from the channel occurs as soon as the data specified by the LOCATE command is transferred to the storage control function. The channel then becomes available for the initiation of another channel program or the resumption of a previously initiated channel program.

Execution of the LOCATE command is completed as soon as the storage control function has completed the required seeking and record positioning. The storage control function tries to reconnect to the channel to execute the data transfer command after LOCATE. If the channel is not available, the storage control function will reposition to the required block and again request reconnection.

The LOCATE command combines the functions of the SEEK and SET SECTOR commands utilized with CKD devices that have the rotational position sensing feature. LOCATE provides the advantages of this feature and eliminates programmed seek address and sector address calculations. Since data blocks are fixed in size, sector location can always be calculated for an FBA device.

The READ or WRITE command in an FBA channel program causes the actual execution of the read or write subcommand specified in the LOCATE command. A READ or WRITE command must be command chained from a LOCATE command. READ or WRITE commands cannot be command chained from other READ or WRITE commands, respectively, but READ/WRITE commands can be data-chained when the chaining occurs between data blocks. If data chaining of READ or WRITE commands is attempted within a data block, an overrun condition occurs.

Commands that enable a program to determine the device type and its physical characteristics are also defined for FBA disk devices. The SENSE I/O command enables a program to obtain the storage control function type and model number and the disk device type and model number. The READ DEVICE CHARACTERISTICS command causes the storage control function to present 32 bytes of device characteristics to a program (number of blocks on the device, number of blocks on a track, etc.).

As for CKD devices attached to a 3830 Storage Control, 24 bytes of sense data are provided for FBA devices and their storage control function. These bytes provide information required by the FBA error recovery routine and statistical usage information. The READ AND RESET BUFFERED LOG command, also provided for 33XX CKD devices that attach to 3830 Storage Control, is provided to enable the contents of the statistical usage counters to be placed in processor storage at a time other than when a counter overflows (at the end of processing for the day, for example).

Two diagnostic commands, DIAGNOSTIC CONTROL and DIAGNOSTIC SENSE, are defined. The exact functions defined for the diagnostic commands are device-dependent.

Format Defective Block

The Format Defective Block subcommand is used to cause the storage control function to locate an available alternate block on the device and assign it to the specified defective primary block. The storage control function automatically does all the positioning required to seek to the alternate cylinder location, locate a nondefective alternate block, and establish forward and backward pointers (within the block identification fields) between the defective primary block and its assigned alternate.

The LOCATE command terminates with an operation incomplete unit check condition if no alternate block is available for assignment. When a

defective primary with an alternate assigned block is encountered during a read or write operation, the storage control function automatically seeks and positions to the alternate block. If additional blocks after the defective primary blocks are to be processed by the executing channel program, the storage control function automatically repositions the access mechanism to the next block after the defective primary block when processing of the alternate block is completed.

READ AND WRITE COMMAND EXECUTION

The following indicates how a READ or WRITE command executes, depending on the subcommand specified in the LOCATE command that precedes it.

Read Data and Write Data

The Read Data and Write Data subcommands function in a similar manner. Read Data prepares the control function to transfer one or more data blocks from disk to processor storage while Write Data prepares the control function to transfer one or more data blocks to disk from processor storage. When a READ/WRITE command is received after a LOCATE command that specifies Read Data or Write Data, the control function reads the next ID field encountered to verify that the positioning done by the LOCATE command is correct. If so, reading or writing of data block fields begins.

Reading or writing continues until the block count reaches zero. If the count in a READ or WRITE command is less than the count calculated from the LOCATE command block count specification, actual data transfer between processor storage and the disk device terminates when the count in the command reaches zero. Command execution completes when the block count reaches zero. For a Write Data operation, the storage control function writes zeros to any unfilled and/or remaining blocks until the block count reaches zero. If the count in the READ or WRITE command is greater than the calculated block count, data transfer terminates when the calculated block count reaches zero.

The data blocks read or written during execution of a READ or WRITE command can span cylinder as well as track boundaries. If the end of a cylinder is reached before the block count reaches zero, the storage control function calculates the block identification (cylinder, head, and block numbers) of the first block of the next cylinder, repositions the access mechanism to the next required block, and continues the data transfer operation. For CKD devices, a cylinder boundary cannot be crossed during a single data transfer operation and the crossing of cylinder boundaries must be programmed.

Write and Check Data

A WRITE command that follows a LOCATE command with the Write and Check Data subcommand specified functions like a WRITE command for which the Write Data subcommand was specified, with one exception. After the data is written, the storage control function automatically verifies that the data has been written correctly. That is, when the data transfer completes, the storage control function reinitializes the block count, and initiates repositioning to the first data block written.

When the correct position is reached, the storage control function reads the data blocks written but does not transfer any data to the channel. Checking is accomplished using the error detection and correction bytes that are at the end of each data block. The checking operation terminates when the block count reaches zero.

Read Replicated Data

The Read Replicated Data subcommand is provided to read one or more blocks of data from a range of replicated data, that is, from a range of data that contains the same data duplicated some number of times (such as a sequence set index record of the index for a VSAM data set).

The LOCATE command specifies the number of blocks to read (number of blocks required to contain the data that is replicated), the replication count, and the location of the first unit of replicated data relative to the beginning of the data set. The replication count indicates the number of blocks contained in the range of data. It is the product of the block count and the number of times the data unit is replicated.

When a LOCATE command specifies the Read Replicated Data subcommand, the storage control function can position the access mechanism to the first block of any unit of the replicated data. The unit chosen is the one that will minimize total access time, that is, the unit closest to the current position of the access mechanism. When the READ command for which Read Replicated Data was specified is executed, the unit of replicated data is read just as for a READ command for which the Read Data subcommand was specified.

DIFFERENCES BETWEEN FBA AND CKD CHANNEL PROGRAMS

The following summarizes the basic differences between the content and operation of channel programs for FBA devices and those for CKD devices with the rotational position sensing feature attached to 3830 Storage Control:

- Disk channel programs for FBA devices are simplified. Any normal read/write operation involving multiple (up to 65,535) data blocks can be accomplished using three commands, whereas to utilize rotational position sensing for CKD devices, a basic read or write operation involving only one physical record with file protection specified requires a minimum of six commands (a SET FILE MASK plus the five commands shown in Section 10:25). A write with verification for CKD devices requires additional commands, and the reading or writing of multiple physical records requires one additional read or write command per record if the Read Multiple Count-Key-Data command is not available. This command, however, is limited to reading multiple records on only one track of a CKD device.

- Programming is simplified for FBA devices because more functions are performed by the storage control function than for count, key, data devices. Specifically, seek and sector addresses are calculated by the storage control function, which also handles cylinder switching during channel program execution (eliminating the need to switch cylinders by programming for sequential processing).

- The protection checking specification for FBA devices is oriented to logical boundaries (extents of data sets) and is not limited by device (track and cylinder) boundaries, as it is for CKD devices.

- Alternate blocks can be located anywhere on a disk volume for FBA devices. Disk designers can place alternate blocks for a specific device in locations that minimize access mechanism movement. For CKD devices, by programming convention, alternate tracks are located at the end of the volume. (Positioning to, and returning from, an alternate block/track location during processing is automatically performed by the storage control function for both FBA and 33XX CKD devices.) The storage control function performs alternate block

assignment functions for FBA devices. For CKD devices, locating and assigning an alternate track is a programmed procedure.

ADVANTAGES

FBA disk devices provide the following two major advantages over CKD disk devices:

- FBA disk devices can be controlled by a storage control function that is significantly less costly than that required to control CKD devices. A storage control function that handles only disk devices with fixed-size data blocks does not have to be as complex as one that handles disks with self-formatting records.

  For example, during channel program execution, the storage control function for a CKD device must be able to obtain a read or write CCW from processor storage, determine the exact operation to be performed and the location of the required microcode, and orient itself for reading or writing in the time interval available for crossing the gap between a count or key area and the data area. By contrast, the storage control function for an FBA device determines the operation to be performed, locates the required microcode, and establishes read or write orientation during processing of the LOCATE command before the time-dependent connection with the device is established.

- The programming required to support FBA devices is independent of device type and thus facilitates data mobility. Only one FBA access method is required to handle any number of different FBA devices, because channel programs contain only relative block addresses and are independent of device type characteristics (number of cylinders in the volume, number of tracks per cylinder, number of blocks per track, etc.). For CKD devices, disk access methods are device-type-dependent and device independence may be achieved within a processing program, in an OS/VS environment, for example, only by dynamically loading the specific disk access method required when processing of the disk device begins.

  The device independence provided by FBA devices permits an installation to move from one FBA direct access device type to another (for example, from a 4331 Processor with 3310 Direct Access Storage to a 4341 Processor with 3370 Direct Access Storage) without changing the operating system being used, processing programs that access the disks, or job control statements for these programs solely for the purpose of handling the new disk device type.

  Device independence reduces maintenance costs for operating system control programs, program products, and user-written disk access methods and error recovery procedures. Once support of FBA devices is added to a program, it supports all FBA devices without the need for modifications.

10:35 STANDARD AND OPTIONAL FEATURES

STANDARD FEATURES

The following are standard features of the 4341 Processor and are operative for both System/370 and ECPS:VSE modes:

- Instruction set that includes binary, decimal, and floating-point (including extended precision) arithmetic instructions, the new general purpose instructions, conditional swapping instructions, PSW

key-handling instructions, certain new control-program-type instructions, and the instructions required to handle new standard features. Standard instructions for the 4341 Processor (that are not available for Models 50 and 65) are the ECPS:VSE-mode-only instuctions and the following:

COMPARE AND SWAP
COMPARE DOUBLE AND SWAP
COMPARE LOGICAL CHARACTERS UNDER MASK
COMPARE LOGICAL LONG
*HALT DEVICE
INSERT CHARACTERS UNDER MASK
*INSERT PSW KEY
*LOAD CONTROL
MONITOR CALL
MOVE INVERSE
MOVE LONG
*RESET REFERENCE BIT
*SET CLOCK
*SET CLOCK COMPARATOR
*SET CPU TIMER
*SET PSW KEY FROM ADDRESS
SHIFT AND ROUND DECIMAL
*STORE CHANNEL ID
STORE CHARACTERS UNDER MASK
STORE CLOCK
*STORE CLOCK COMPARATOR
*STORE CONTROL
*STORE CPU ID
*STORE CPU TIMER
*STORE THEN AND SYSTEM MASK
*STORE THEN OR SYSTEM MASK
(START I/O FAST RELEASE, a privileged instruction functionally implemented in certain System/370 processors, is executed as a START I/O in the 4341 Processor because it provides no advantage as a result of the utilization of the instruction processing function for I/O initialization.)

- BC and EC modes of operation and control registers
- Monitoring feature
- Program event recording
- Interval timer (3.3-ms resolution)
- Time-of-day clock
- CPU timer and clock comparator
- Reference and change recording
- Expanded machine check interruption class
- Interruption for SSM instruction
- Reloadable control storage
- ECC on processor storage
- Instruction retry
- Byte-oriented operands
- Store and fetch protection
- External signals
- ECPS:VS1
- ECPS:VM/370
- High-speed buffer storage (8K)
- Support processor subsystem
- Channel group 1 (which includes one byte multiplexer and two block multiplexer channels)
- 128 to 1024 UCWs in increments of 32
- Channel retry data in a limited channel logout area

*Privileged instruction

The following are standard features that operate only when System/370 mode is in effect:

- Dynamic address translation to perform address translation for instruction addresses but not CCW addresses (including the PURGE TLB and LOAD REAL ADDRESS instructions)
- Channel Indirect Data Addressing
- Store Status

The following are standard features that operate only when ECPS:VSE mode is in effect:

- Internal mapping function to perform address translation for instruction and channel program addresses
- Page control functions that include page control exceptions, page description bits, and capacity counts
- Page control privileged instructions:

  CLEAR PAGE
  CONNECT PAGE
  DECONFIGURE PAGE
  DISCONNECT PAGE
  INSERT PAGE BITS
  LOAD FRAME INDEX
  MAKE ADDRESSABLE
  MAKE UNADDRESSABLE
  SET PAGE BITS
  STORE CAPACITY COUNTS

- Machine save function (including the RETRIEVE STATUS AND BLOCK privileged instruction)

OPTIONAL FEATURES

Optional features for the 4341 Processor, all of which can be field-installed and can operate with System/370 or ECPS:VSE mode in effect, are:

- 3278 Model 2A Display Console with operator control panel (includes display and printer-keyboard modes)--required feature
- Block Multiplexer Channels, Additional (channel group 2, which includes three block channels that can be configured as one byte and two block multiplexers or three block multiplexers)
- Channel Control Unit Positions, Additional (provides for attachment and automatic powering of 24 additional control units for a total of 48)
- Channel-to-Channel Adapter (one maximum)
- Remote Support Facility (specify feature)-no charge

12:05 GENERAL DESCRIPTION

A display console for system control and operator/operating system communication is required for the 4341 Processor. The 1052 Model 7 Printer-Keyboard (for System/360 models) and 3210/3215 Console Printer-Keyboards (for System/370 models) cannot be attached to the 4341 Processor. A display console provides faster display than a typewriter keyboard and the console for the 4341 Processor offers functions not available for the typewriter consoles for System/360 and System/370.

The operator console is used to (1) manually control operation of the 4341 Processor when the console is in manual mode, (2) communicate with the operating system when the console is in system mode, and (3) perform diagnostic operations when CE mode is in effect. CE mode is made effective when the CE switch on the CE panel located within the frames of the 4341 Processor is turned on. The cable connecting the operator console to the 4341 Processor can be a maximum of 6 meters (20 feet) in length.

The operator console for the 4341 Processor is the 3278 Model 2A Display Console (cathode ray tube for displaying data and keyboard for entering data). The 3278 Model 2A is a 3278 Model 2 with modifications to the function of several of its keys to support console-type operations. The primary 3278 Model 2A has an operator control panel mounted on it. This control panel cannot be mounted on the other optional 3278 Model 2A displays that can be natively attached to the 4341 Processor.

The 3278 Model 2A Display Console has twelve program function keys. A light pen, such as that for the System/370 Model 158 display console, is not provided for the 3278 Model 2A console. The 3278 Model 2A is also used as the operator console for the 4331 Processor.

The audible alarm, which is sounded under program control, is standard on the 3278 Model 2A. The security keylock feature is optional. When the security key is in the locked position, the console becomes inoperative, with the keyboard locked and the screen blank. A keyboard-locked message appears on line 25 of the display.

The screen of the operator console can simultaneously display 25 lines of 80 characters each. Lines 21 to 25 on the screen cannot be used for operator-to-operating system communication. Predefined displays are provided to enable the operator to select and execute manual functions (such as resets, IPLs, address compares, etc.) that for System/360 processors are performed using a control panel on the processor.

OPERATOR/OPERATING SYSTEM COMMUNICATION MODES

The 3278 Model 2A Display Console for the 4341 Processor has two standard modes of operation for operator/operating system communication: display and printer-keyboard. Printer-keyboard mode is provided to enable the operator console to emulate 1052, 3210, and 3215 printer-keyboards. Printer-keyboard mode must be used when an operating system that uses a 1052 or 3210/3215 as the operator console executes in a 4341 Processor.

Display Mode

For display mode, the operator console appears to be a 3277 display attached to a 3272 Model 2 Control Unit. The keyboard is used for input and the cathode ray tube for output. The first 20 lines of the screen are used by the operator and operating system. Optionally, a natively attached 3287 Model 1 Printer (80 characters per second) or 3287 Model 2 Printer (120 characters per second) can be used for hard-copy output.

The display/keyboard combination and console printer, if present, are addressed separately when display mode is in effect. While addresses in the range of 000 to 0EF can be utilized, for compatibility with System/370 processors the preferred addresses for the display/keyboard are X'01E' and X'009' and for the 3287 Printer, X'012' and X'015'.

DOS/VS, DOS/VSE, OS/VS1, and VM/370 support display mode operations for the 3278 Model 2A. The 3287 Printer is supported for hard-copy during display mode operations by DOS/VSE, OS/VS1, and VM/370. The operator can also use the copy key, when the console is in manual mode, to write the contents of the display console to the hard-copy printer. This copy capability must be established using the program load display.

For display mode, the first 20 lines of the screen are the system area while the next 4 lines are the system status area. Line 25 is used as a console indicator area. The system area is used for communication between the operator and the operating system and for displays associated with manual operations performed by the operator or customer engineer. The system status area is used to display certain status information about the processor and messages for the customer engineer. The console indicator area displays status data regarding the screen and keyboard.

Printer-Keyboard Mode

For printer-keyboard mode, the display console appears to the processor as a 1052 Printer-Keyboard if a System/360 operating system is being used or as a 3210/3215 Console Printer-Keyboard if a System/370 operating system is being used. The keyboard is used for input and the cathode ray tube is used for output. A natively attached 3287 Model 1 or 2 Printer is optional for hard-copy output. Device address X'01F' or X'009' would normally be used for the display/keyboard.

For printer-keyboard mode, the message area for operator-to-operating system communication consists of lines 1 to 18. Lines 19 and 20 are the operator input area that displays the data the operator keys in (up to 126 characters). Lines 21 to 25 are used for the same functions as when display mode is in effect.

In this mode, the screen is treated like a printer-keyboard device. Messages appear on the screen in successive lines until the screen becomes full. Then the top six lines are deleted automatically by the hardware and the remaining lines are moved up to leave six blank lines in positions 13 to 18. Since the operator cannot control the contents of the screen, as with display mode, the 3287 Printer is recommended for hard-copy output.

In printer-keyboard mode, the 3278 Model 2A Display Console is controlled using 3210/3215 commands. The display/keyboard and optional 3287 Printer have the same address and the same data is automatically printed on the 3287 as is displayed on the screen. A maximum of two of the natively attached 3287 Model 2A Display Consoles can be operating in printer-keyboard mode. Each can have an associated hard-copy 3287 Printer.

System Status Area

Lines 21 to 24 of the operator console screen are used as the system status area. This area appears on all operator and maintenance displays. Line 21 shows certain status information about the processor, line 22 shows modes in effect that the operator established via a manual operation, and lines 23 and 24 are reserved for customer engineer use.

The following is indicated on lines 21 and 22 of the operator console:

- Running when the instruction processing function is operating or there is I/O activity
- Wait state when the instruction processing function is in the wait state
- Manual state when the instruction processing function is in the stopped state
- Test state when a control function is not operating in its normal state or when a maintenance function is being performed
- Load state during a program (new PSW) load
- Machine save function successfully completed
- BC or EC mode of operation in effect when System/370 mode is operative
- Mode in effect: System/370 or ECPS:VSE
- Whether the interval timer is enabled or disabled
- The condition that will cause the processor to stop execution
- A machine check occurred that caused the processor to stop operating
- When the processor is in the stopped state, the contents of the related address. During processor operation, this indicator does not change.
- Addressing state for the mode in effect. For System/370 mode, the addressing state is virtual (V) when DAT is enabled and real (R) when DAT is disabled. For ECPS:VSE mode, the addressing state is always virtual.
- The setting of the rate control: process, instruction step, and, for CE mode, word step or clock step
- The check control setting in effect
- Time-of-day clock state for setting (enabled or secure)
- The setting of the address compare control: normal or stop for normal (non-CE mode) operations
- The area in which address compares are to be made (any reference, I/O reference, processor storage, instruction counter)
- The address or the data value currently set for an address compare stop

Lines 23 and 24 are designed to be used primarily by customer engineers. For example, they contain such items as the reference code

that is generated when a hardware error occurs (see Section 50), messages from the support processor, and the identifier of the last module loaded into the support processor.

Console Indicator Area

Information can be displayed on line 25 of the screen only by the support processor. It can contain one of the following indications:

- MANUAL CONTROL to indicate the display is currently operating in manual mode and is not available to the operating system
- DISPLAY MODE or PTR/KYBD MODE to indicate the display console is operating in display or printer-keyboard mode, respectively.
- INSERT MODE when the INS key is pressed to place the console in insert mode. This indication is removed when the reset key is pressed.
- INHIBIT when all keys except RESET, START, STOP, and INTR are disabled because the operator tried to perform a certain invalid function, such as enter data in a protected area of the screen. The operator can press the RESET key to clear the condition or it can be cleared via the programming.
- PRT-BUSY when the printer for which a hard-copy operation was requested via the copy key is busy. When the printer becomes available or the RESET key is pressed, the indicator is removed.
- PRT-CHECK when the printer for which a hard-copy operation was requested is out of paper, out of order, or not attached. The RESET key is used to remove the indication.
- PRT-INV REQ when operator intervention is required for the hard-copy printer (such as when end of forms occurs). The indicator is removed when the RESET key is pressed.
- DISCONNECTED to indicate the display console is not logically connected to the 4341 Processor (no unit address has been assigned to the console via the program load display).
- USAGE CONFLICT to indicate a function is not currently allowed. This indicator is lit if the start, stop, interrupt, mode select, change display, or diagnose key is pressed when another display console is already in manual mode or if a previous request has not been completed.

When printer-keyboard mode is in effect, the following indicators can be displayed in position 47 to 80 of line 25:

- PROCEED when a READ command is in process. This indicator informs the operator that data can be keyed in to the input area.
- REQUEST when the REQ key is pressed and the attention status is stacked because the console is busy executing a command. The indicator is cleared when the current command completes and the attention status is presented to the channel.
- ALARM when the audible alarm is sounded by the execution of a command
- INTV-REQD when the 3287 Printer being used for hard-copy output becomes not ready (power is turned off, paper runs out, error condition exists, etc.)

OPERATOR CONTROL PANEL

The operator control panel is located on the keyboard of the primary 3278 Model 2A Display Console. It contains the following pushbutton controls and indicators:

- Power on/IML pushbutton. When this pushbutton is pressed, a power on of the support processor subsystem and IML of the support processor occur. At the successful completion of these operations, the balance of the 4341 Processor is powered on as are all I/O devices that are set to be powered on/off with the processor. An IML of instruction processing function microcode occurs automatically if the operator has specified an automatic IML after power on using the program load display (see discussion under "Program Load Display" in Section 12:10).

  Then a power-on reset of the 4341 Processor (clear reset and time-of-day clock reset) is performed. The processor is placed in the stopped state and the program load display is shown on the operator console. When a power on is not successful, an IML and reset do not occur. This pushbutton is also used to perform an IML of the support processor only when power is already on.

  The 3278 Model 2A Display and 3287 Printers installed must be powered on individually. The Channel-to-Channel Adapter must be enabled using the channel-to-channel switch.

- Power off pushbutton. This pushbutton is used to remove power from the 4341 Processor only (not the 3278 displays or 3287 under control of the power-off sequencing microcode that is resident in the support processor.

- Channel-to-Channel switch. When the optional Channel-to-Channel Adapter is installed, this switch is used to enable and disable the logical interface to the other processor to which the adapter is attached.

- Chan-Chan Disabled indicator. When the I/O interface control switch is in the disable position, this indicator is lit to indicate the Channel-to-Channel Adapter logical interface to the other processor is not enabled. When this indicator is lit, the operator can power down the 4341 Processor without loss of data or interruption of the other processor. The power off pushbutton should not be pressed unless the disable indicator is on.

- Power in process indicator. This light turns on as soon as the power on/IML pushbutton is pressed and stays on until power-on sequencing of all system components is successfully completed, at which time it is turned off.

- Power complete indicator. When lit, this light indicates power is on. It is turned on at the successful completion of a power-on sequence when the power in process indicator is turned off.

- Basic check indicator. When this indicator is lit, a hardware malfunction exists in the hardwired power sequence or the display console.

- System indicator. This indicator is lit whenever instruction processing is taking place.

- Wait indicator. This indicator is lit when instruction execution is not occurring because the current PSW has the wait bit on.

- Lamp test pushbutton. When pressed, this pushbutton causes all functional indicator bulbs on the operator's control panel to be lit and is used for testing purposes.

KEYBOARD

There are 75 keys on the keyboard. Certain keys have a normal and an alternate function. The alternate function is selected by holding the ALT key down and pressing the desired functional key.

In addition to alphabetic, numeric, cursor control, and keyboard control, the following keys are provided:

- MODE SEL/DIAG. This key is used to initiate use of the display screen for manual operations instead of operator-to-operating system communication (switch from system to manual mode). Activation of the mode select function invokes a general selection display that lists the manual functions the operator can perform. The specific mode selection display shown depends on whether System/370 or ECPS:VSE mode is in effect. Activation of the DIAG function causes a diagnostic program (the test case monitor discussed in Section 50:15) to be loaded into the support processor and executed.

- CHG DPLY. This key causes a switch between system and manual modes and a switch in the display currently being shown.

- CNCL (PA2). When system mode is in effect, this key causes an attention interruption (PA2 type) to be generated for display mode operations or a unit exception when printer-keyboard mode is in effect. The key is inactive when manual mode is in effect.

- INTR/LINE DISC. When the INTR function is selected, an external interruption occurs. The LINE DISC function is used to terminate operation of the Remote Support Facility.

- REQ (PA1)/COMM REQ. When the REQ function is selected, an attention interruption (PA1 type) is generated for display mode. An attention interruption without the PA1 indication is generated when printer-keyboard mode is in effect. The COMM REQ function is used to request communication between a local and remote customer engineer when the Remote Support Facility is active.

- COPY. When this key is pressed, the contents of lines 1 through 24 of the current display are written to the locally attached 3287 that has been designated to receive copy-key data. This key is functional only when manual mode is in effect for either display or printer-keyboard mode.

- START and STOP keys. These keys are use to start and stop instruction processing.

- Page up and page down keys. These keys increase and decrease addresses during manual operations. The increment/decrement depends on the operation being performed.

- Program function keys 1 to 12. These keys are effective only when the ALT key is pressed. The function performed by each of these keys is defined via programming.

- SP/MO key (alternate function for the ERASE EOF key). This support processor manual operations key is active only when the CE mode is active and is used to invoke SPIL manual operations (read, display, modify, instruction step, etc.) for support processor microcode for debugging purposes.

DISPLAY CONTROLS AND INDICATORS

Certain control switches and indicators are contained on the display console (to the left and right of the screen). These include the following:

- A power on/off switch, and a power on/off indicator to control power to the display and indicate the current power condition
- A switch to cause both upper- and lowercase characters or only uppercase characters to be displayed on the screen
- A normal/test mode switch that establishes normal operating mode for controlling system processing operations or a test mode that can be used by the operator to execute problem determination procedures when the display console malfunctions. Three different tests that check for correct operation of various display console hardware can be executed in test mode.
- An audible alarm control that is used to raise or lower the volume of the sound produced by the program-controlled audible alarm
- The screen contrast and screen brightness control knobs that permit the operator to adjust the character display for comfortable viewing

CONFIGURATION DISPLAY

The configuration display is designed to be used by the customer engineer and can be selected from the general selection display only when CE mode is in effect. The configuration display is used to select the system configuration display, module transfer display, and three displays associated with the Remote Support Facility (discussed in Section 50:15).

Configuration data about the 4341 Processor is displayed when the system configuration display is selected. The customer engineer uses this display to change configuration data. The entered data is stored on the system diskette in a configuration record. An IML must be performed whenever the configuration record is changed.

The following configuration data can be entered using the system configuration display:

- The part number of the system diskette and the date and number of its engineering change level
- Processor storage size
- Console language
- Number of channels installed
- Addresses of the natively attached 3278 Model 2A Display Consoles and 3287 Model 1 or 2 Printers and the printers that are to be used for hard copy
- Channel-to-Channel Adapter is installed
- I/O power-up timeout (length of time for all I/O devices with their processor switch in the remote position to power on)

The system configuration display also lists the model and serial number of the 4341 Processor and indicates which natively attached 3278

Model 2A consoles and 3287 Printers are actually operational. This data cannot be changed by the customer engineer.

The module transfer display is used when a new system diskette is sent to a 4341 Processor installation. This display enables the customer engineer to copy the processor-dependent data, such as configuration data, UCW directories, logout records, clock skew data, and reference codes, from the existing system diskette to the new diskette.

12:10 OPERATOR DISPLAYS

Several displays are provided that enable the operator to perform manual operations. The functions the operator can perform are listed in the general selection display. Each function has a single or multiple-character identification associated with it. The operator selects the function to be performed by keying in the associated identification. Certain functions have their own display associated with them while others do not.

GENERAL SELECTION DISPLAY

The general selection display for operator use is shown in Figure 12.10.1. When CE mode is in effect, additional functions that can be utilized only in CE mode are listed as well. The general selection display is automatically displayed when manual mode operations are initiated (via pressing of the CHG DPLY key, for example) and can be selected by pressing the mode select key. This display can also be selected from other operator displays.

The functions listed on the general selection display are the same for System/370 and ECPS:VSE modes with one exception. When System/370 mode is in effect, the store status function is listed. When ECPS:VSE mode is in effect, the machine save function is listed instead of store status.

```
                          GENERAL SELECTION


      Y   TIME-OF-DAY ENABLE        L   PROGRAM LOAD
          J   INTERVAL TIMER SWITCH       A   COMPARE/TRACE
      PROGR   SYSTEM RESET PROGRAM        K   CHECK CONTROL
      CLEAR   SYSTEM RESET CLEAR          O   OPERATION RATE
        RES   RESTART                     D   DISPLAY/ALTER
   SAVE   STORE STATUS

      Z   RETURN TO PROG SYS

SELECTION:
```

Figure 12.10.1. The general selection display

The functions listed on the left-hand side of the general selection display do not have a display associated with them. Each of the functions listed on the right of the display does have an associated display. To perform a function, the operator first must display the

general selection display on the screen by pressing the MODE SEL key on the operator console keyboard. The general selection display can also be invoked by entering Q in the selection area of another display.

Before the general selection display is shown on the screen, the current contents of the screen are saved. The current state of the processor (running or in the stopped state) is not changed when the MODE SEL key is pressed.

To perform the desired function, the operator enters its associated identification and presses the enter key. For functions that do not have a display, execution of the function occurs as soon as the enter key is pressed. For functions that have a display, the associated display replaces the general selection display on the screen when the enter key is pressed.

A fast selection capability is provided that enables (1) any manual mode display to be selected from any other manual mode display without a return to the general selection screen first and (2) many manual functions to be performed without first displaying the display that controls the function. The two functions can also be combined.

Thus, the operator can enter Q (to select the general selection display) and the code for the desired display (for fast screen selection) and for certain functions the code for the the desired function of the requested display (for fast function selection). When the enter key is pressed, the associated display is shown and the function is executed.

A validity check is performed on the identification and any supplied data as soon as the enter key is pressed. The operator is notified if the specification is incorrect.

Functions

The TIME-OF-DAY ENABLE function is used to enable time-of-day clock setting via the SET CLOCK instruction. TOD:ENBL is indicated on line 22 of the screen when this function is selected. After 6 seconds, the ability to set the clock is automatically disabled. The INTERVAL TIMER SWITCH function is used to enable/disable the interval timer.

The program reset function causes a reset of the instruction processing function and channels. Pending I/O interruptions are cleared and the processor is placed in the stopped state. The clear reset function causes a program reset to be performed and clears the current PSW, CPU timer, clock comparator, general registers, floating-point registers, and processor storage to zero. In addition, control registers are initialized to their reset values. The time-of-day clock value is not altered.

The message "Selection Complete" appears on the general selection display as soon as a program reset or clear reset function completes. The operator can select another function (such as program load to re-IPL the operating system) to resume processor operation. The CNCL or CHG DPLY key must also be pressed to return control of the display screen to the operating system.

When the restart function is selected, program execution is resumed using the PSW located at program processor storage location 0 if the processor is in the stopped state. When the processor is in the operating state, the current operation is completed, PSW switching occurs (current PSW is stored at processor storage location 8, PSW at location 0 becomes current PSW) after all enabled interruptions are taken, and program execution continues, using the PSW located in

processor storage location 0. The operating system regains control of the display screen. A restart cannot be performed if the processor is in a check stopped state.

The store status function, not available for System/360 processors and identical to the System/370 store status function, can be performed only when System/370 mode is in effect and the processor is in the stopped state. It is designed to be used after the processor enters a disabled wait state after an uncorrectable error occurs. After a store status is completed, the message SAVE appears on line 21 and the processor is in the stopped state.

The contents of the following are placed in program processor storage during a store status operation:

CPU timer - locations 216-223
Clock comparator - locations 224-231
Current PSW - locations 256-263
Floating-point registers - locations 352-383
General registers - locations 384-447
Control registers - locations 448-511

The operator should perform the store status function to preserve processor status after an error causes a processor halt and before resetting the processor to load a standalone processor storage dump program. Otherwise, the contents of these fields and registers at the time the halt occurred are lost during the reset that is performed to IPL the dump program. The standalone dump programs provided are modified to obtain the system status information indicated. The store status function should not be performed before SEREP is executed.

The machine save function, which is not available for System/360 or System/370 processors, can be performed only when ECPS:VSE mode is in effect and the processor is in the stopped state. It is designed to be used instead of the store status function. When machine save is invoked, 256 bytes of processor status and the contents of page 0 (addresses 0 to 2047 of processor storage) are saved in auxiliary storage.

The processor status information saved is the following: CPU timer and clock comparator values, current PSW contents, time-of-day clock value, floating-point register contents, general register contents, control register contents, processor identification (as for a STORE CPU ID instruction), page capacity count, existing frame capacity count, available frame capacity count, free frame capacity count, page bits for page 0, reference and change bits for page 0, access control key for page 0, fetch protection bit for page 0, and frame index for page 0.

A reset, IPL, or power off that occurs during the machine save procedure halts the procedure. When the machine save successfully completes, the message SAVE appears on line 21.

The RETRIEVE STATUS AND PAGE instruction can be used to place the saved processor status and page 0 contents in two specified locations in processor storage (as long as the machine save completed successfully). This instruction can be issued, for example, by a dump program that is loaded after the machine save function is performed. Thus, the dump program can obtain the processor status, contents of addresses 0 to 2047, and contents of any other processor storage locations desired at the time of a hard failure. The processor should not be cleared before the dump program is loaded.

The RETURN TO PROG SYS function switches the display to system mode and causes the previously saved contents of the screen to be displayed.

PROGRAM LOAD DISPLAY

The program load display, shown in Figure 12.10.2, is used to display certain parameters that are to be used either during IML or IPL, initiate an IML, initiate an IPL, and set up certain operator console parameters. The parameters controlled by this display are stored in the configuration record on the system diskette.

When the program load display appears on the screen, it indicates the parameters selected previously using the program load-mode/assist selection display. These parameters are the mode of operation (System/370 or ECPS:VSE) and, for System/370 mode, the assist (ECPS:VS1 or ECPS:VM/370) to be enabled.

```
                    * PROGRAM LOAD *

PERFORM IML:M                       PERFORM IPL:  P

MODE_ASSIST MENU:I                        IPL UNIT:  U

      *MODE*                          RESET TYPE: R1 LOAD CLEAR
                                                  R2 LOAD NORMAL
      *ASSISTS*
                                    CONSOLE MODE: N1 PTR/KYBD
                                                  N2 DISPLAY

IML AT POWER_ON:  X1                CONSOLE ADDR:  T0 T2
                  X2                               T1 T3
STATUS                              COPY KEY DEV:  K
                                                Q GENERAL SELECTION
SELECTION                                       RETURN TO PROG SYS
```

Figure 12.10.2.The program load display

The program load display always appears automatically after a power on when CE mode is not in effect and can be selected from the general selection display. Any time a parameter is selected using the program load display, it is written to the configuration record and becomes effective the next time an IML or IPL is performed, as appropriate, unless changed by the operator.

The program load display is used to control and/or display the following parameters for IML:

- Whether an IML of instruction processing function microcode is to occur automatically after a power on is performed (power on/IML pushbutton is pressed). If an automatic IML is not performed, only microcode for the support processor is loaded during a power on and the operator must select the program load display and execute the perform IML function to cause an IML of instruction processing function microcode. The first IML that occurs after a power on resets the entire 4341 Processor. Subsequent IMLs will reset everything except the time-of-day clock.

- The mode, System/370 or ECPS:VSE, that is to be made effective. When System/370 mode is selected, either ECPS:VS1 or ECPS:VM/370 can be selected also.

- Whether or not microcode patches contained on the system diskette are to be loaded. The customer engineer establishes the patches

that can be loaded. A RUN PATCHES message appears after a successful IML to permit the operator to load patches or not.

The following IPL parameters can be specified using the program load display:

- Address of the IPL device
- The type of clear to be performed: load normal or load clear

The following parameters associated with the natively attached consoles and printers are specified using the program load display:

- The mode, display or printer-keyboard, to be made effective for the console(s)
- The addresses and types (console or hard-copy printer) of the natively attached devices. T indicates a 3278 Model 2A console while H indicates a 3287 Printer. When DSC is specified as an address, the device is logically disconnected from the operating system even though it may be physically installed.
- The 3287 Printer(s) to be used for hard copy. For printer-keyboard mode, a 3287 Printer is designated as a hard-copy device by assigning it to the same address as a 3278 Model 2A console. Only two unique addresses can be specified for printer-keyboard mode. For display mode, no two addresses can be the same. A hard-copy printer is designated by specifying H for its type.
- The 3287 Printer that will receive the data contained on the operator console display when the copy key is pressed. The copy key function can also be made inoperative.

When the program load display appears on the screen, the device type (display or printer) and assigned device address for each natively attached device is displayed. The operator can use the display to logically disconnect one of these devices, when it is malfunctioning, for example, and change the address of a device. Thus, if the primary operator console malfunctions during system operation, the operator can stop the processor, select the program load display, logically disconnect the malfunctioning operator console, and assign the address of the malfunctioning console to another attached 3278 Model 2A display. System operation can then be continued without a re-IPL.

Note also that during IML of the support processor, if the primary operator console device is not functional, the microcode determines whether there are any other natively attached 3278 Model 2A displays. The next functional display found, if any, is assigned as the primary operator console and the operator is notified that the display normally used as the primary operator console is not functional.

If an error occurs during IML of the support processor or the instruction processing function, the message IML ERROR appears on the program load display and a reference code is also displayed. There is no automatic re-IML after an error occurs. When an IML is successfully completed, the message IML COMP appears on the program load display. If the operator selects the perform IPL function and an IML has not been performed, the message NO IML appears on the program load display.

DISPLAY/ALTER DISPLAY

The display/alter display enables the operator and customer engineer to display and/or alter the contents of several items. Figure 12.10.3 shows the System/370 mode display/alter display for the operator. The

display/alter display for ECPS:VSE mode is the same except for the absence of the real storage entry and the presence of the page description entry in place of the storage entry.

The operator selects the item to be altered/displayed by keying in its associated one-character identification. When the operator requests a display, it is shown on the right-hand portion of the screen while the selection list remains on the left-hand side.

```
DISPLAY/ALTER

G  GENERAL REGISTERS
C  CONTROL REGISTERS
F  FLOATING POINT REGISTERS
P  CURRENT PSW
K  KEY STORAGE
V  VIRTUAL STORAGE
M  REAL STORAGE
T  TRACE AREA


= HEX CALCULATOR
Q  GENERAL SELECTION
Z  RETURN TO PROG SYS SELECTION
```

Figure 12.10.3. The display/alter display

The processor is stopped before any alter or display operation is actually executed. To resume processor operations, the operator must press the START key and the CHG DPLY key must be pressed to return control of the screen to the operating system.

When the virtual storage or real storage item is selected, 128 bytes of storage can be displayed in hexadecimal. The page up and page down keys enable the operator to display the next 128 bytes after the current display or the 128 bytes preceding the current display, respectively. The hex calculator function provides the ability to add and substract hexadecimal numbers to calculate storage addresses. For display operations, the translate function can be requested (by including a T with the selected function) to cause hexadecimal characters to be displayed as EBCDIC characters if they have such a translation.

The trace area function can be selected to access the contents of the instruction trace area that is located in auxiliary storage. This area contains the addresses of the last 168 instructions that were executed when instruction tracing was activated using the address compare display.

CHECK CONTROL DISPLAY

The check control display, shown in Figure 12.10.4, is used to specify the action to be taken when a machine check condition occurs. The normal, hardstop, no retry, disable, and stop after log-switch settings are mutually exclusive. Section 50 discusses the actions taken for the various settings.

```
                    * CHECK CONTROL *


                    N   NORMAL
                    S   HARDSTOP
                    R   NO RETRY
                    D   DISABLE
                    C   STOP AFTER LOG_SWITCH
                    1   MS SINGLE-BIT SWITCH
                    2   MS DOUBLE-BIT SWITCH
                    Q   GENERAL SELECTION
                    Z   RETURN TO PROG SYSTEM

SELECTION:
```

Figure 12.10.4. The check control display

OPERATION RATE CONTROL DISPLAY

The operation rate control display is used to select the rate at which instructions are executed. Certain rate control functions are designed for customer engineer use only and can be selected only when CE mode is in effect.

One of the following rates can be selected by the operator:

- Normal - After the START pushbutton has been pressed, the instruction processing function executes instructions at the normal speed. This function also clears any other rate selection.
- Instruction step - The instruction processing function executes one instruction each time the START key is pressed. The processor accepts all pending enabled interruptions and then returns to the stopped state. If another function is selected, such as display/alter, it will be accepted and instruction step remains in effect. Changes in data can be seen on the system console as each instruction step is completed. Line 21 shows the address and the data at this address.

COMPARE/TRACE DISPLAY

The compare/trace display can be used to perform the following:

- Establish an address comparison on a virtual or real storage address within a specific location (any reference, instruction counter, processor storage, I/O instruction reference, microword, or local storage). The action to be taken when the address match occurs (normal stop or turn off compare and continue execution, for example) can be specified also.
- Establish a data comparison between the data in a specific processor storage location and data in another location (any reference, instruction counter, processor storage, or I/O instruction reference). The processor stops when an equal comparison occurs.
- Establish a trace function that inspects the instruction addresses utilized by the instruction counter to search for a specified address. When the trace function is activated, the instruction address utilized by the instruction counter is saved in a trace area

that is located in auxiliary storage. This area can hold 168 addresses. When the area becomes full, the action specified when tracing was initiated is taken. If trace stop was specified, no more tracing occurs. If trace wrap was specified, tracing continues and each new address overlaps the oldest saved address. The contents of the trace area can be inspected using the trace area function of the display/alter display.

12:15 MAINTENANCE

The IBM 3278 Model 2A Display Console Problem Determination Guide (GA23-0020) is provided with each 4341 Processor. This guide is located under an access panel below the keys on the 3278 Model 2A display. It contains problem determination and abbreviated operating and reference information. The IBM 4341 Processor Operator's Guide (GA24-3669) also contains problem determination procedures. When the operator console malfunctions, the operator can take the steps outlined in the guides before calling the customer engineer.

This page intentionally left blank.

SECTION 15: VIRTUAL STORAGE AND ADDRESS TRANSLATION

The first subsection, 15:05, discusses the needs that virtual storage and address translation are designed to meet. No previous understanding of these facilities is assumed. In this discussion, an address space is defined as a consecutive set of addresses that can be used in programs to reference data and instructions. System operation in IBM-supplied virtual storage environments is explained conceptually, without use of all the terminology new to such an environment.

The general advantages of IBM-supplied virtual storage operating systems are presented in Section 15:05. Included in this subsection are those that apply to DOS/VSE and OS/VS1.

The last portion of subsection 15:05 defines the terminology associated with virtual storage and address translation hardware. The terminology included is that common to the IBM-supplied programming systems that support a virtual storage environment for 4300 Processors.

Subsection 15:10 describes in detail the implementation of address translation for System/370 mode. The operation of dynamic address translation and channel indirect data addressing hardware in the 4341 Processor are discussed. Other hardware items associated with dynamic address translation, such as reference and change recording, are discussed also.

Subsection 15:15 describes in detail the implementation of address translation for ECPS:VSE mode. The internal mapping function and address translation table are described together with the page control instructions. Address translation for instruction addresses and channel programs is discussed.

The last subsection, 15:20, discusses the new factors that affect system performance in a virtual storage environment. The information presented is related to efficient installation and utilization of an IBM-supplied virtual storage operating system.

15:05 VIRTUAL STORAGE CONCEPTS, ADVANTAGES, AND TERMINOLOGY

THE NEED FOR LARGER ADDRESS SPACE

The past and present rapid growth in the number and types of data processing applications being installed has led to an increasing demand for more freedom to design applications without being concerned about, or functionally constrained by, the physical characteristics of a particular computer system--system architecture, I/O device types, and processor storage size. As program design and implementation become easier, they can enable more rapid installation of applications so that the benefits of data processing can be achieved sooner.

The design of System/360 and OS allowed programmers to be less concerned than before about specific processor architecture and I/O device types when designing and implementing applications by (1) providing a compatible set of processor models ranging in size from small to large scale; (2) providing a variety of high-level languages with greatly expanded capabilities, including a new language (PL/I); (3) providing comprehensive data management functions, including support of I/O device independence where data organization and the physical characteristics of devices permitted; and (4) supporting dynamic allocation of system resources (channels, I/O devices, direct access space, and processor storage). System/360 users who installed DOS Version 3 (Release 26) also experienced more system configuration

independence than was previously available, although to a lesser degree than OS users.

While System/360 and its primary operating systems represented major steps toward giving programmers a larger measure of system configuration independence, constraints imposed by the need to design applications to fit within the available processor storage still existed. In addition, although System/360 processors provided more, less-costly processor storage than was previously available, increasingly larger amounts of processor storage began to be required as the use of high-level languages increased, the usage and level of multiprogramming increased, the functions supported by operating system control programs expanded, and applications that require relatively larger amounts of processor storage (such as telecommunications and data base) were designed and installed more frequently.

The requirement for more processor storage is still growing. The new applications being developed and installed tend to have larger and larger storage design points in order to provide the functions desired. More processor storage is also required for I/O buffer areas to achieve maximum capacity and performance for sequential operations using new direct access devices with significantly larger track capacities. Larger blocking of tape records, which requires larger I/O buffers, also results in increased tape reel capacity and decreased tape processing time. As a result, System/370 processors provide significantly more processor storage than their predecessor System/360 processors and offer it for a lower cost. This trend is continued with 4300 Processors.

The availability of more processor storage, however, has not relieved all the constraints associated with it. Applications still must be tailored to the amount of processor storage actually available in a given system even though storage design points (partition sizes in DOS/VSE and OS/VS1, for example) can be larger than they were previously.

Consider the following situations that can occur in installations:

1. An application is designed to operate in a 50K processor storage area that is adequate to handle current processing needs and that provides room for some expansion. Some time after the application is installed, however, maintenance changes and the addition of new functions cause one of the programs in the application to require 51K and another to require 52K. Installation of the next processor storage increment cannot be justified on the basis of these two programs, so time must be spent restructuring and retesting the programs to fit within 50K.

2. An existing application has programs with a planned overlay structure. The volume of transactions processed by these programs has doubled, and better performance is now required. Additional processor storage is installed. However, the overlay programs cannot automatically use the additional storage. Therefore, reworking of the overlay programs is required to take them out of planned overlay structure and, thereby, achieve the better performance desired.

3. A low-volume, terminal-oriented, simple inquiry program that will operate for three hours a day is to be installed. If the program is written without any type of overlay structure, it will require 60K of processor storage to handle all the various types of inquiries. However, because of a low inquiry rate, only 8K to 12K of the total program will be active at any given time. In order to justify its operational cost, considerable additional program development time is spent designing the inquiry program

to operate with a dynamic overlay structure so that only 12K of processor storage is required for its execution.

4. A multiprogramming installation has a daily workload consisting primarily of long-running jobs. There are also certain jobs that require a relatively small amount of time to execute. The times at which these jobs must be executed is unpredictable; however, when they are to be run, they have a high completion priority. While it is desirable to be able to initiate these high-priority jobs as soon as the request to execute them is received, this cannot be done because long-running jobs are usually in operation. Hence, a certain time of day is established for initiating high-priority jobs and the turnaround time for these jobs is considerably longer than is desired.

5. A series of new applications are to be installed that require additional computing speed and twice the amount of processor storage available in the existing system. The new application programs have been designed and are being tested on the currently installed system until the new one is delivered. However, because many of the new application programs have storage design points that are much larger than those of existing applications, testing has to be limited to those times when the required amount of processor storage can be made available.

   Although another smaller scale processor is also installed that has time available for program testing, it cannot be used because it does not have the amount of processor storage required by the new application programs. In addition, although the smaller scale processor now provides backup for the currently installed larger scale processor, the smaller scale processor cannot be used to back up the new system because of processor storage size limitations.

6. A large terminal-oriented application is to be operative during one entire shift. During times of peak activity, four times more processor storage is required than during low-activity periods. Peak activity is experienced about 20 percent of the time and low activity about 40 percent. The rest of the time, activity ranges from low to peak. Allocation of the peak activity processor storage requirement for the entire shift cannot be justified, and a significantly smaller storage design point is chosen. As a result, a dynamic program structure must be used, certain desired functions are not included in the program, and response times during peak and near-peak activity periods are increased above that originally planned.

   In this installation, most of the batched jobs are processed during the second shift. However, there is also a need to operate the large terminal-oriented application for a few hours during second shift. This cannot be done because the system does not have the amount of processor storage required for concurrent operation of the batched jobs and the terminal program (which must have its storage design point amount allocated even though that amount of processor storage would not be required during second-shift operations). The large amount of additional processor storage required to operate the terminal program for only a portion of the second shift cannot be justified.

7. An application program with a very large storage design point is executed only once a day as a batched job. A significant benefit would result from putting the program online to a few terminals during the morning hours. However, the program continues to be run as a batched job because it is very large and would be made larger by putting it online. The large amount of additional

processor storage required to operate the program concurrently with the existing morning workload cannot be justified.

8. A terminal-based application has been installed on a full production basis for several months. During this period, the benefits accrued from the online application have encouraged the gradual installation of several additional terminals, and peak activity is considerably higher than it was initially. Because growth has been gradual, much additional programming time (significantly more than is required to maintain batch-oriented applications) has to be spent periodically restructuring the terminal-based application program to handle the increasing volume of activity.

9. An online application is currently active during an entire shift and operates concurrently with batched jobs. It would be advantageous to install a second terminal-oriented application that would operate concurrently with the existing workload during the entire shift. However, the amount of processor storage that would have to be dedicated to each online application for the entire shift in order to handle its peak activity is very large, and times of peak activity for the two applications do not completely overlap. Because so much processor storage would be unused during a large portion of the shift if both online applications were always active, installation of the second online application is difficult to justify.

In the situations described, processor storage is a constraining factor in one way or another and the constraints highlighted can apply in some degree to all systems regardless of their scale (small, intermediate, large) or processor storage size. The availability of larger, less expensive processor storage does not remove these constraints for two major reasons.

First, once a storage design point has been chosen for an application, whether the design point is relatively large or small, the application is dependent on that processor storage size for its operation. The application cannot execute in less than its design point storage amount, nor can it take advantage of additional available processor storage without being modified (unless it has been specifically structured to use additional storage as, for example, are most IBM-supplied language translators).

Second, although processor storage has become less costly, it still is a resource that should be used efficiently because of its importance in the total system operation. Thus, when storage design points are chosen, tradeoffs among processor storage cost, application function, and system performance are often made. Making applications fit within the storage design points selected becomes the responsibility of application designers and programmers. This situation is made more difficult by the fact that for many applications an optimum storage design point cannot be determined until the application is written and tested using expected transaction volumes.

The significance of processor storage restraints should be evaluated in light of the following trends evidenced by new types of applications: (1) the total amount of storage required to support their new facilities continues to grow larger, (2) the storage they actually require for operation during their execution is tending to become more variable, and (3) it is becoming as desirable to install many of these new applications on smaller scale systems with relatively small maximum processor storage sizes and low volume requirements as it is to install them on larger scale systems. Reduction of the constraining factors currently imposed by processor storage is, therefore, a necessary step

in making new applications easier and less costly to install and available to a wider range of data processing installations.

Given the described processor storage restraints on application design and development and the storage requirements that are becoming increasingly more characteristic of many of the new types of applications, it becomes advantageous to allow programmers to design and code applications for a larger address space than they currently have. That is, programmers should be able to use as much address space as an application requires so that special program structures and techniques are not required to fit the application into a given storage size. Programmers can then concentrate more on the application and less on the techniques of programming. In addition, the size of the address space provided should not be determined by processor storage size, as it is in System/360 operating systems, such as DOS Versions 3 and 4 and OS MFT and MVT, so that the address space can be larger than the processor storage available.

A larger address space should be provided, therefore, by a means other than making processor storage as large as the address space desired. This requirement can be satisfied by providing programmers with an address space (called virtual storage) that is supported using online direct access storage and address translation hardware. This approach also offers the advantage of supporting a larger address space for a lower cost than if larger processor storage is used, since direct access storage continues to be significantly less expensive per bit than processor storage. In addition, address translation hardware offers functional capabilities that large processor storage alone cannot provide.

VIRTUAL STORAGE AND ADDRESS TRANSLATION CONCEPTS

Virtual storage is an address space the maximum size of which is determined by the addressing scheme of the computing system that supports it rather than by the actual number of physical processor storage locations present in the computing system. In the 4341 Processor, for example, which uses a 24-bit binary address, a virtual storage of 16,777,216 bytes is supported. When virtual storage is implemented, the storage that can be directly accessed by the processor, normally called processor storage, is referred to as real storage.

The concept of virtual storage is made possible by distinguishing between the names of data and instructions and their physical location. In a virtual storage environment, there is a distinction between address space and real storage space. Address space (virtual storage) is a set of identifiers or names (virtual storage addresses) that can be used in a program to refer to data and instructions. Real storage space is a set of physical storage locations in the computer system in which instructions and data can be placed for processing by the processor. The number of addresses in the two spaces need not be the same, although both spaces begin with address zero and have consecutive addresses. The programmer refers to data and instructions by name (virtual storage address) without knowing their physical (real storage) location.

When virtual storage is not implemented, there is, in effect, no differentiation between address space and real storage space. The address space that can be used in programs is identical in size to the real storage space available and the address in an instruction represents both the name and the location of the information it references.

In a virtual storage environment, therefore, the address space available to programmers is that provided by the virtual storage size implemented by a given system--not the address space provided by the

real storage available in the given system configuration. In DOS/VSE and OS/VS1, virtual storage rather than real storage, is divided into consecutively addressed partitions for allocation to problem programs. The fact that storage addresses in executable programs are virtual rather than real does not affect the way in which the programmer handles addressing. For the 4341 Processor, for example, an Assembler Language programmer assigns and loads base registers and manipulates virtual storage addresses in a program just as if they were real storage addresses.

Virtual storage is so named because it represents an "image of storage" rather than physical processor storage. Since virtual storage does not actually exist as a physical entity, the instructions and data to which its virtual storage addresses refer, which are the contents of virtual storage, must be contained in some physical location.

In a virtual storage operating system environment, the contents of virtual storage are divided into a portion that is always present in real storage, namely, all or part of the control program, and another portion that is not always present in real storage. The instructions and data that are not always present in real storage must be placed in locations from which they can be brought into real storage for processing by the processor during system operation. This requirement is met by using direct access storage to contain this portion of the contents of virtual storage (see Figure 15.05.1). The amount of direct access storage required to support a given amount of virtual storage varies by operating system, depending on how direct access storage is organized and allocated.

In addition, a mechanism is required for associating the virtual storage addresses of instructions and data contained in direct access storage with their actual locations in real storage when instructions and data are being processed by the processor. This requirement is met by using address translation hardware in the processor to associate virtual storage addresses with appropriate real storage addresses.

With this design, a processor can support an address space that is larger than the actual size of the real storage present in the processor. This is accomplished by bringing instructions and data from direct access storage into real storage only when they are actually required by an executing program, and by returning altered instructions and data to direct access storage when the real storage they occupy is needed and they are no longer being used. At any given time, real storage contains only a portion of the total contents of virtual storage.

Such a design is made practical by the fact that the logical flow of processing within the majority of programs is such that the entire program need not be resident in real storage at all times during execution of the program. For example, initialization and termination routines are executed only once during the operation of a program. Any exception-handling procedure, such as an error routine, is required only if the exception condition occurs. A program that handles a variety of transaction types (whether batch or online oriented) need have resident at any given time only the transaction routine required to process the current transaction type.

It is this property of programs that has enabled planned overlay and other dynamic program structures to be used successfully in nonvirtual storage environments when the amount of processor storage available was not large enough. As indicated previously, this variable storage requirement characteristic of programs tends to be even more pronounced in new types of applications and in online environments in which processing is event-driven.

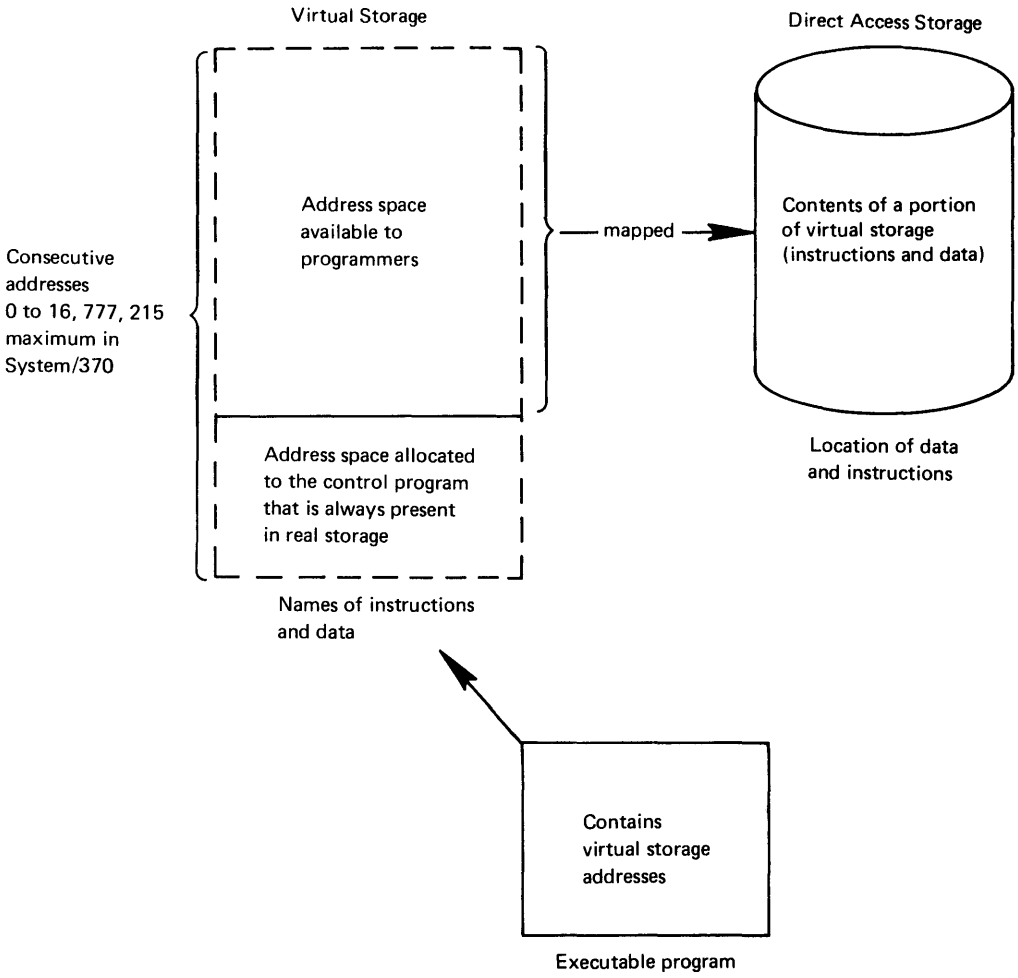


Figure 15.05.1. Names and location of instructions and data in a virtual storage environment

For the purpose of resource management in a virtual storage environment, virtual storage and its contents, direct access storage used to contain a portion of the contents of virtual storage, and real storage are divided into contiguous, fixed-length sections of equal size. Once a program has been fetched from a program library and initiated, instructions and data within a program are transferred between real storage and direct access storage, a section at a time, during program execution. A section of an executing program is brought into a real storage section only when it is required, that is, only when a virtual storage address in the section is referenced by the executing program. A program section that is present in real storage is written back in a direct access storage section only when the real storage assigned to it is required by another program section and only if it has been changed.

A virtual storage operating system control program monitors the activity of the sections of all executing programs and attempts to keep the most active sections in real storage, leaving the least active sections in direct access storage. Figure 15.05.2 illustrates the relationship of virtual storage, direct access storage, and real storage

without regard to a specific virtual storage operating system implementation.

The division of a program and its data into sections and the transfer of these sections between direct access storage and real storage during program execution is handled entirely by the virtual storage operating system without any effort by the programmer. When a planned overlay or dynamic overlay program structure is used, the programmer is responsible for dividing the program and its data into phases, determining which phases can be present at the same time in the amount of real storage available (partition), and indicating when phases are to be loaded into real storage during processing.

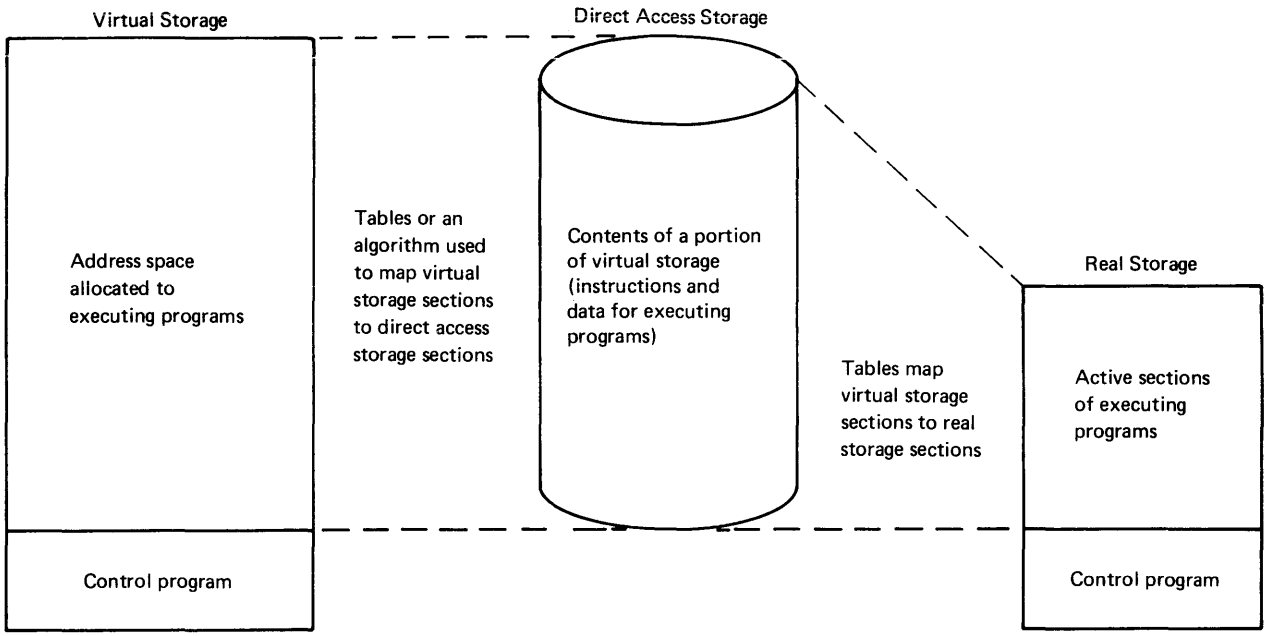


Figure 15.05.2. Relationship of virtual storage, direct access storage, and real storage

While a virtual storage 16 million bytes in size can be addressed by any 4341 Processor, the virtual storage size that can be effectively implemented by a given system is affected by (1) the amount of real storage present, (2) the amount of direct access storage space made available to contain the contents of virtual storage, (3) the speed of the direct access storage devices containing virtual storage contents and contention for these devices or the channels to which they are attached, (4) the speed of the processor, and (5) the characteristics of the programs operating concurrently. Hence, the amount of real storage required to effectively implement a specific amount of virtual storage can vary by system, depending on the characteristics of the applications in the workload and the performance desired, as is discussed in Section 15:20.

Once a program section has been loaded into real storage, its virtual storage addresses can be translated when they are referenced. Dynamic address translation hardware (for System/370 mode) or an internal mapping function (for ECPS:VSE mode) is the mechanism in 4300 Processors that translates the virtual storage addresses contained in instructions into actual physical storage addresses during instruction execution.

In 4300 Processors, the dynamic address translation facility of System/370 mode provides address translation using a hardware-implemented, two-level table lookup procedure that accesses tables contained in real storage. These tables, which are maintained by control program routines, (1) define the amount of virtual storage supported and allocated, (2) indicate whether or not any given program section is currently present in real storage, and (3) contain the addresses of real storage sections allocated to the program sections that are currently present in real storage.

The address translation mechanism for the ECPS:VSE mode of 4300 Processors consists of an internal mapping function that utilizes one table, called the address translation table in the 4341 Processor. This table contains the same type of information that is present in the tables used by DAT hardware but the table is maintained entirely by hardware instead of by the control program.

During the execution of each instruction in either System/370 or ECPS:VSE mode, address translation is performed on any virtual storage address in the instruction that refers to data or to an instruction. Translation occurs after the 24-bit effective virtual storage address has been computed by adding base, displacement, and, if any, index values together, as usual. The result of the address translation is a 24-bit real storage address designating the location containing the data or instruction referenced by the virtual storage address in the instruction.

For System/370 mode, the virtual storage addresses in channel programs (CCW lists) are not translated to real addresses by channel hardware during channel program execution, and programmed translation before initiation of a channel program is required. For ECPS:VSE mode, virtual addresses in channel programs are translated during channel program execution, eliminating the necessity for programmed translation.

In reality, address translation hardware (both DAT and the internal mapping function) provides dynamic relocation of the sections of a program during its execution. This capability is not provided by DOS Version 3 or 4 and OS MFT and MVT. DOS Version 3 and 4 support program relocation only at link-edit time. OS MFT and MVT support program relocation at program load time as well as at link-edit time.

Once a program has been loaded into an area of real storage by the program fetch routine, the DOS Version 3 or 4 and OS MFT or MVT operating systems cannot relocate the program to another area of real storage during its execution. Thus, an entire program or a portion of a program cannot be written in direct access storage during execution and later reloaded into different real storage locations to continue execution. Once loaded, therefore, a program is bound during its execution to its initially allocated real storage addresses. In a virtual storage environment a program is bound only to the virtual storage addresses it was assigned during loading.

The dynamic relocation provided by address translation hardware eliminates, for most programs, the need for allocating and dedicating a contiguous area of real storage to an entire program for the duration of its execution, a requirement for all programs in DOS Version 3 or 4 and OS MFT or MVT. (As discussed later in this subsection, some programs cannot operate correctly in the manner being described, that is, with sections transferred only as required between direct access storage and real storage.) In DOS/VSE and OS/VS1 environments, real storage is no longer divided into contiguously addressed partitions that can contain one executing job step (program) at a time.

Further, when real storage is allocated to a section of an executing program, the real storage is not dedicated to that program section for

the duration of program execution. Concurrently executing programs can dynamically share the same real storage sections. That is, in general, the real storage available for allocation to executing programs can be allocated to any program section as needed. When a section of an executing program must be loaded, any available section of real storage can be assigned (subject to certain restrictions imposed by operating-system-dependent real storage organizations). When the program section is no longer required, it can be written to direct access storage, if it has been altered, and the real storage assigned to it can be made available for allocation to another section of the same program or to a section of another program.

The assignment of real storage sections is handled entirely by the operating system, which keeps account of which sections of concurrently operating programs are the most active. The problem programmer has no explicit control over when and how much real storage is allocated to an executing program. The operating system does not attempt to allocate a given amount of real storage to each executing program. It merely allocates real storage to those sections it determines are the most active, without taking into account the particular program to which the active section belongs.

Address translation hardware, therefore, provides more than translation from address space (virtual storage) to real storage space. It provides the capability of implementing dynamic real storage management that requires no effort on the part of the programmer and significantly less processor time than programmed address translation during program execution. (The large amount of processor time required to translate addresses during program execution using programmed means has precluded implementation by IBM of an operating system that supports such programmed address translation.)

Much of the real storage utilization preplanning required for OS MFT or MVT and DOS Version 3 or 4 environments in order to use real storage can effectively be eliminated in a virtual storage environment. Dynamic real storage management capability is another advantage the technique of using address translation hardware and direct access storage to support a large address space has over using larger real storage.

Another capability made available by the implementation of large address space using direct access storage and dynamic address translation (DAT) hardware in System/370 mode is that of supporting more than one virtual storage with one processor. (The internal mapping function used in ECPS:VSE mode supports only one virtual storage.) Multiple virtual storages can be used to support multiple virtual machines. The concepts and general advantages of virtual machines are discussed in Section 18.

The use of virtual storage and address translation hardware to enable programs to operate in less real storage than the total storage requirement of the programs can also offer better performance potential than the technique of using a planned overlay program structure. When a planned overlay program executes in MFT or MVT, considerable time can be spent executing the overlay supervisor in order to perform programmed address translation (relocation) when a program phase is loaded.

In addition, more efficient real storage utilization may be achieved in a virtual storage environment, since the control program reacts to changing processing needs and only portions of the program that are actually required are loaded (all phases of an overlay program may not be the same size and all code within a phase may not be used when the phase is loaded). Once a planned overlay program has been structured to handle the currently required set of program phases efficiently, it cannot automatically adapt to a change in the set of program phases required or to a change in the activity of the required set of phases.

In a virtual storage environment, the performance of the system can be directly affected by the amount of time spent transferring program sections between direct access storage and real storage. Satisfactory system performance is achieved when each of the concurrently executing programs has enough real storage dynamically allocated to it so that the need for transferring program sections into and out of real storage is kept at an acceptable level.

As previously mentioned, most programs can be structured in such a way that processing is localized in one area or another of the program during time intervals rather than equally spread over the entire program. In other words, at any given time period during execution of the program, only a subset of the entire program need be referenced. This is sometimes called the "locality of reference" characteristic of programs. A program achieves satisfactory performance when its most frequently referenced sections in any given time interval remain in real storage and there is a limited amount of program section transfer activity.

Most programs require a certain minimum amount of real storage in which to execute in order to achieve satisfactory performance. If such programs operate with less than their minimum requirement dynamically allocated, program section transfer activity increases and performance degradation can occur. The minimum real storage requirement of a program is related to the amount of real storage required by the most active sections of the program. Because of the locality of reference characteristic of most programs, the minimum real storage requirement of a program for satisfactory operation frequently can be less than its total storage requirement. This fact enables an operating system to efficiently support a virtual storage that is larger than the real storage actually present in the computing system.

A virtual storage environment, therefore, enables most programs to be independent of real storage size to a large degree. A program can execute using varying amounts of dynamically available real storage without being modified. The amount of real storage dynamically available to a program during its execution primarily affects its performance, to the extent that program section transfer activity is affected, rather than its capability to be executed.

For example, while a given 100K language translator might be able to operate with an average of 40K of real storage dynamically available to it during its operation, the time required to compile a program on a smaller scale processor under these conditions might be unacceptable.

Alternatively, the performance desired on the smaller scale processor might be achieved if an average of 60K is dynamically available to the language translator while it operates. Without a virtual storage operating system, the 100K language translator probably could not be used at all on the smaller-scale processor because of its relatively large design point size.

The availability of lower-cost processor storage for 4300 Processors and the storage independence that a virtual storage environment offers provide new flexibility in tradeoffs among processor storage cost, function, and individual program or total system performance.

GENERAL ADVANTAGES OFFERED BY IBM OPERATING SYSTEMS THAT SUPPORT A VIRTUAL STORAGE ENVIRONMENT

Each of the IBM operating systems that supports a virtual storage environment for the 4341 Processor using address translation hardware offers the capability of using address space that is larger than that provided by the processor storage actually available, and each supports

dynamic processor storage management that is transparent to the user. As a result, these operating systems offer certain general potential advantages that do not depend on their unique features. The implementation of virtual storage also provides benefits that are specific to each of these operating systems because of their design and the particular functions they support. The following discusses the potential advantages of virtual storage and address translation hardware that are common to DOS/VSE and OS/VS1 environments.

The general advantages of virtual storage operating systems are the potential they offer for:

- Increased application development
- Expanded operational flexibility
- System performance improvement

A virtual storage operating system can facilitate more rapid development of new applications because, by removing most existing real storage restraints on application design, it can help improve the productivity of programmers. Specifically, a virtual storage operating system has characteristics that can be used to reduce the effort, time, and cost associated with application design, coding, testing, and maintenance. This makes the installation of new applications more readily justifiable and encourages the addition of new functions to existing applications.

The potential advantage of improved operational flexibility is made possible by the greater independence of applications from real storage size. Enhanced system performance can result from improved real storage utilization. While these latter two benefits have their own individual value, they too, either indirectly or directly, ease the installation of new applications.

Potential for Increased New Application Development

The following capabilities are characteristic of a virtual storage operating system environment.

- Greater flexibility in the design of applications is possible.

  Larger programs can be written without the necessity of using planned overlay techniques or other dynamic program structures designed to fit programs into the amount of real storage available. The need for intermediate (or working) data sets is reduced or eliminated because tables, relatively small data groups, etc., that are placed on direct access storage because of real storage limitations can become part of the program and will be brought into real storage automatically, as required. Program planning, coding, and testing time can be reduced by elimination of the use of these programming techniques and other real storage management facilities, which also require additional programming knowledge and skill. Also avoided is the restructuring of application programs after they have been written, because they are larger than the real storage available for their execution. Hence, applications can become operational more quickly.

  Open-ended, straightforward application design is possible and more comprehensive programs can be written. An application can be segmented into a series of programs according to its logical flow instead of according to the functions that can be performed in the specific amount of real storage available to each step in the application. Programming and processing duplication inherent in the

approach of using two or more job steps to perform one logical process is thereby avoided.

Additional programming facilities can become available that otherwise could not be used because of real storage limitations. Specifically, full-function high-level language translators, which offer more capabilities than their subset versions (such as additional debugging facilities and performance options) but which also have larger storage design points, can be used because they can operate in a virtual storage environment using less real storage than their design point requirement.

- Preproduction testing of larger than average application programs can be increased if enough virtual storage can be made available to enable them to run during normal testing periods. Turnaround time during testing can be reduced.

  In a nonvirtual storage environment such programs are usually grouped together and executed only at certain times when their larger design point storage requirements can be made available.

- Fine tuning of application programs to achieve performance improvements, when necessary, can be delayed until after the application is in production. This capability enables an application to become operative sooner.

- Startup costs for new applications may be reduced.

  A new application can be developed and tested on the existing system, assuming the required I/O devices are present in the configuration, before the additional real storage the application requires for performance on a production basis is actually installed. When the application is ready for production, the additional real storage required can be added to the system. In some cases it may be possible to operate the application on a production basis on the existing system without adding real storage initially, because during the startup period, transaction volume is very low. As the volume grows, real storage can be added to achieve better performance.

- Growth of existing applications and the maintenance of operational programs is simplified.

  Because of the removal of most real storage restraints, new functions can be more easily and more rapidly added to most existing applications. Program expansion because of added functions or maintenance changes does not require the use of overlay techniques, multiple job steps, etc., when the size of the extended program exceeds the original storage design point size.

  In general, alteration and debugging of nonoverlay programs are also easier than alteration and debugging of programs with planned overlay or dynamic structures.

- Application programs whose real storage requirements, based on transaction volume and complexity, vary widely during their execution may be justified, designed, and installed more easily.

  Design, coding, and testing time can be reduced because dynamic storage management is automatically provided by the operating system. Time and effort need not be spent structuring such programs to use available real storage dynamically to support the functions and/or response times required.

- Design and installation of one-time, low-usage, or low-volume programs of very large storage size are more easily justified. Existing applications in these categories that currently operate in a batch environment can also more easily be altered to operate online, a growth step that might not be justifiable in a nonvirtual storage environment.

- Applications can be installed on a trial basis for the purpose of observing and evaluating their functions and their operation.

  Most IBM-supplied application program products can be temporarily installed on an existing system, assuming the required I/O devices are present. The additional hardware resources that may be required to operate the application on a production basis can be added later, when the application is permanently installed.

- The benefits of the functions provided by many IBM-supplied application program products with larger storage design points can be realized using smaller 4300 Processors with relatively smaller amounts of available real storage.

  Currently, it may be difficult to justify the real storage required to install a relatively large storage design point application on a smaller scale system to handle a low volume of transactions, even though the functions provided by the application are very desirable. In a virtual storage environment, such an application can execute using that amount of dynamically available real storage required to satisfy the desired performance requirements for the low volume of activity.

Potential for Additional Operational Flexibility

The reduction of real storage restraints makes most applications more independent of the real storage size of a system configuration and permits most applications to be processed on systems with varying amounts of available real storage without program modification. Dynamic real storage management reduces the amount of jobstream and operations preplanning that is normally done to use real storage as efficiently as possible in a multiprogramming environment. The following benefits can result:

- A system can back up another system even though it has less real storage than the system it backs up.

  A smaller-scale system with the appropriate I/O configuration can provide backup for a larger-scale system if necessary. (Performance experienced on the backup system may vary from that normally achieved, depending on the two system configurations involved.)

- A single design and one operating procedure can be used for an application that is to operate on multiple systems with varying amounts of real storage, as long as the virtual storage required is supported by all the systems.

  When data processing is decentralized among multiple installations with systems that have different amounts of real storage, one location can design, implement, and maintain an application that can be used by other installations. Duplication of this type of effort can be minimized or eliminated.

- Most applications can be tested on systems with less real storage than the one on which they will run in a production environment, as long as the required amount of virtual storage is supported.

- Growth to a larger real storage configuration can be easier.

Real storage can be added to an existing system to improve system performance (by the reduction of program section transfer activity) without the necessity of modifying existing application programs so that they can take advantage of additional real storage. Additional real storage (up to a maximum of their design point size) is automatically used by programs that operate in a virtual storage environment.

- Operators need not perform certain procedures that are solely related to efficiently managing real storage.

  The operator is concerned primarily with the division of virtual storage and therefore need not change partition sizes at various times (in DOS/VSE or OS/VS1, for example) for the purpose of making storage available for larger than average jobs. (An installation can define virtual storage partitions that are larger than those currently defined in the DOS Version 3 or 4 or OS MFT environment, and the partitions can be made big enough to contain the largest existing or currently planned storage design point programs.)

- Priority jobs whose need to be processed cannot be predicted can be scheduled when required.

  A nonvirtual storage environment does not provide the capability of effectively handling the scheduling of high-priority jobs on a random basis. Hence, this type of job is not permitted to exist in an installation, or such jobs must be scheduled to operate only at certain times. In a virtual storage environment, a high-priority virtual partition (in DOS/VSE and OS/VS1) can be defined and reserved for the purpose of processing only high-priority jobs. Except for that required for certain tables, real storage is not required for this partition until a job is actually scheduled.

Potential for Performance Improvement

The improved real storage utilization made possible by the use of address translation hardware can have a positive effect on the performance of a system that handles a job mix whose use of real storage varies considerably while it is being processed. The extent of the performance improvement depends on the types of applications involved and the current utilization of system resources. Therefore, the amount of performance gain, if any, that may be achieved is highly variable by installation. Environments with the greatest potential for improved performance are as follows:

- Batch-oriented multiprogramming environments with application programs of widely varying real storage requirements.

  Real storage may not be most efficiently used in such an environment, because in DOS Version 3 or 4 and OS MFT environments, it is difficult to divide real storage into a set of partitions that is optimum for all programs. (Consider the inefficient use of real storage in a 54K partition allocated for assemble, link-edit, and test jobs in which a 54K language translator, a 10K linkage editor, and problem programs no larger than 40K execute.) In addition, real storage is not efficiently used when the real storage requirement of a given program, based on transaction mix or volume, varies widely, and the amount of real storage that is allocated is designed to handle the peak requirement. (This is typically true of graphics applications, for example.)

  Further, real storage assigned to a program is not productively used during the time the program is waiting for a human response, such as for the operator to locate and/or mount a volume or to make a

decision and enter a message on the console, or during the time required to guiesce the system in order to change partition definitions or start a high-priority job.

In a virtual storage environment, in which all concurrently executing job steps share real storage dynamically and use real storage only when it is actually required for program execution, real storage is more efficiently used. Hence, if real storage currently is the restraint, a given real storage size might be capable of supporting a higher level of multiprogramming than can be achieved without the use of dynamic storage management (assuming other required resources, such as processor time, I/O devices, and channels, are available). For example, installation of a large storage design point, terminal-oriented application to handle only a few terminals might be possible. Alternatively, a higher level of multiprogramming might be supported by the addition of a smaller real storage increment than would otherwise be required.

System performance may also be improved if more efficient use of available real storage enables additional heavily used functions to be made resident instead of transient or allows the incorporation of performance-oriented options in the control program. This improvement can apply to environments with batch and online operations, as well as to batch-only multiprogramming environments.

- Multiprogramming environments with a mixture of batch-oriented and terminal-based applications.

  While the real storage required for the communication control portion of a teleprocessing application remains constant, terminal-based processing programs are typically subject to wide variations in the amount of real storage they require during their execution, because the transaction mix being handled concurrently varies, the activity of each terminal online varies, or the number of terminals operating concurrently changes. In order to provide the functions desired, ensure the capability of handling peak activity periods and maximum transaction type mixes that can occur concurrently, and guarantee a given response during times of peak activity, a certain amount of real storage is required.

  This peak requirement is generally significantly more than is needed during periods of medium and low activity. Allocation of the maximum storage requirement results in inefficient use of real storage, since unused real storage dedicated to any terminal program cannot be used by other concurrently operating batched or terminal-oriented jobs in a nonvirtual storage environment. In addition, it is usually difficult, and sometimes impossible, to effectively preplan real storage usage for an online application.

  Dynamic real storage management in a virtual storage environment automatically provides a more efficient method of allocating real storage in such an environment. Real storage is not divided into that which can be used only by the terminal-based program(s) and that which can be used only by batched jobs. During times of peak terminal activity, the active sections of terminal-oriented processing programs with a higher priority are automatically allocated more real storage, making less real storage available to the lower-priority batched jobs in execution at that time. During periods when terminal activity is relatively low, real storage not used by any terminal program is available for assignment to the active sections of executing batched jobs. Such an environment is represented conceptually in Figure 15.05.3.

  In existing mixed batch and online-oriented installations, dynamic real storage management allows programming techniques that can

improve the performance of the online application. This improvement can be in the form of better response for existing terminals or the ability to support more terminals. A given online application may also be able to support a higher level of multiprogramming, as a result of better real storage utilization, without any additional programming effort. A virtual storage environment can also make the concurrent operation of multiple terminal-based applications more practical, because real storage equal to the design point storage amount of each online application need not be dedicated to applications the entire time the applications are active.

Figure 15.05.3 shows sample allocations of real storage to two batched jobs and two terminal-oriented jobs in a multiprogramming environment during low, medium, and peak activity points in time. Job priority from high to low is TP2, TP1, BJ2, BJ1. For simplicity, virtual and real storage are shown to be totally allocated at all times. No particular virtual storage operating system (DOS/VSE or OS/VS1) is assumed, since the concepts illustrated apply to DOS/VSE and OS/VS1 telecommunications environments.

Real storage is shown to be contiguously allocated to each job in high-to-low priority sequence. This is done only to illustrate the relative amount of real storage the control program has dynamically allocated to each program during the instant shown. In reality, the total amount of real storage allocated to an executing program at any given time is usually not contiguous in a virtual storage environment. In addition, during times of low terminal program activity, it may be possible to support a higher level of batched job multiprogramming, which is not shown in the figure.

Summary

As the preceding discussion indicates, a virtual storage environment is designed primarily to provide new functional capabilities for the installation as a whole, although performance gains are possible for installations with particular environmental characteristics. The general functional aims of IBM-supplied virtual storage operating systems are (1) to use new hardware features and additional control program processing to support certain facilities that are not possible in a nonvirtual storage environment because of real storage restraints and (2) to handle other functions that must be performed by installation personnel (programmers, operators, and system designers) when virtual storage and address translation are not used.

It is also important to note that, while a virtual storage operating system permits an installation to be independent of real storage restraints to a large degree and enables real storage to be utilized more efficiently, the performance of the system and specific advantages that can be achieved still depend largely on the amount of real storage present in the system and on the computing speed of the processor, among other things. Hence, virtual storage and an address translation capability are not a substitute for real storage. Rather, they provide an installation with greater flexibility in the tradeoff between real storage size and function or performance.

The degree to which a particular installation experiences the potential benefits of a virtual storage/address translation environment is highly system configuration dependent and application dependent (number, type, complexity of applications installed or to be installed). In addition, consideration must be given to the system resources that are specifically required to support a virtual storage environment (discussed in Section 15:20).

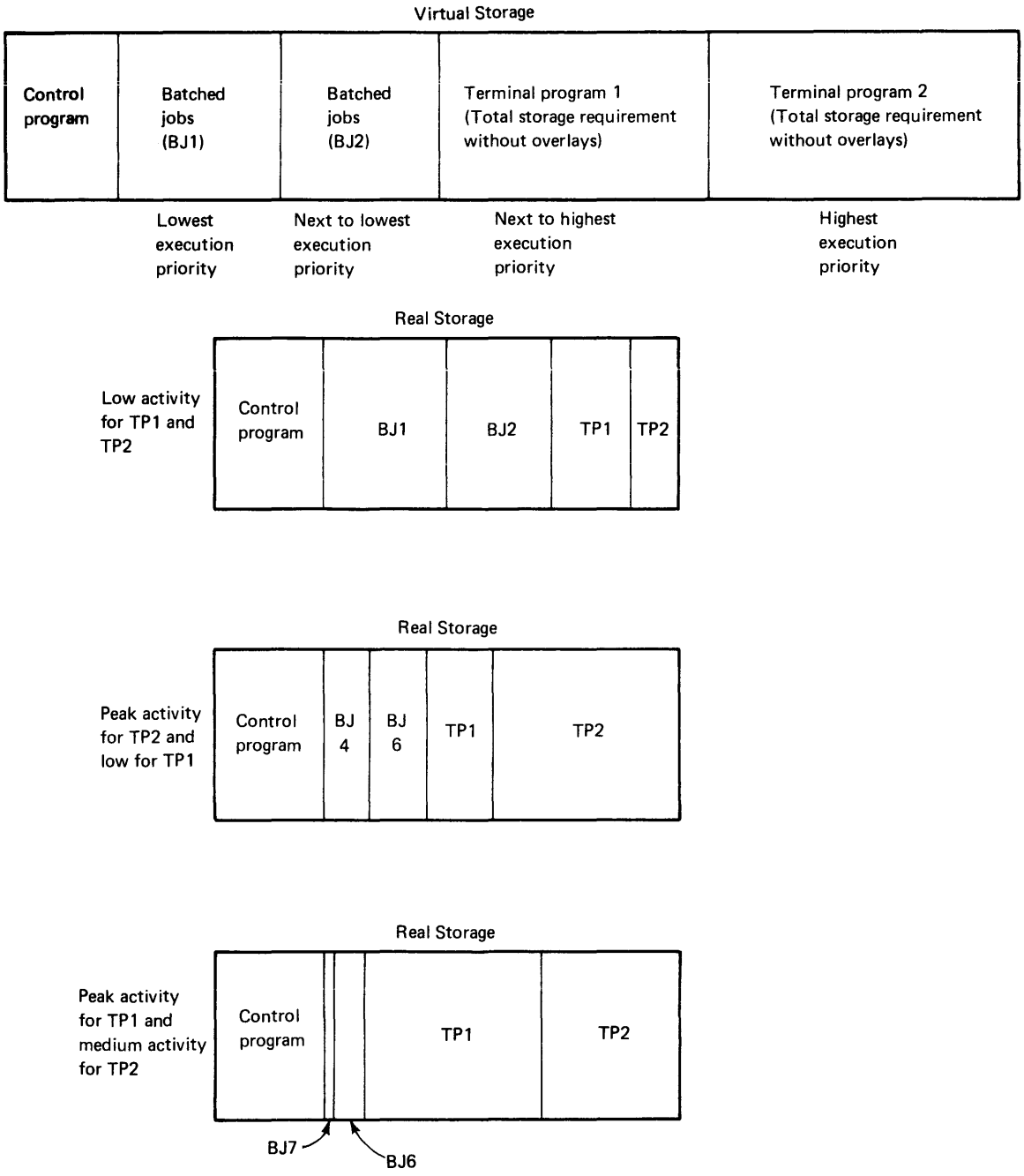


Figure 15.05.3. Conceptual illustration of real storage utilization in a mixed batch and online virtual storage environment

Some of the potential advantages, such as those associated with application maintenance and operational flexibility and those that result from better management of real storage, can be experienced as soon as a virtual storage operating system is installed. Others may be achieved in the future, when new applications are installed and the less restrictive program design techniques available in a virtual storage environment are more fully utilized. In any case, installation of a virtual storage operating system can make a 4341 Processor easier to use

and can be a major step toward more rapid installation of applications. Such an operating system can be of greatest benefit to installations desiring to move to, or extend, online operations and thereby attain the advantages such an environment offers.

VIRTUAL STORAGE AND ADDRESS TRANSLATION TERMINOLOGY

For the purpose of presenting the concepts of virtual storage and address translation in the previous discussion, virtual storage, programs and data, direct access storage, and real storage were described as being divided into areas called sections. In reality, a unique term is used to describe each of the various sections, namely, virtual storage page, page, slot, and page frame, respectively. In addition, virtual storage, as implemented for System/370 mode, has two levels of subdivision. The following defines the new terminology used by the IBM-supplied virtual storage operating systems.

The virtual storage supported for System/370 mode is divided into contiguous segments, which contain virtual storage pages. A virtual storage segment, as implemented in the 4341 Processor, is a fixed-length, consecutive set of addresses for either 64K or 1024K bytes that begins on a 64K or 1024K boundary, respectively, in virtual storage. A virtual storage is divided into segments all of one size or the other. In general, in an OS/VS1 environment, a segment is the unit of virtual storage allocation.

Each segment of virtual storage is divided into contiguous, fixed-length, consecutive sets of addresses called virtual storage pages. Each segment in the virtual storage contains the same number of virtual storage pages, each of which is the same size. A virtual storage page, as implemented for System/370 mode, can be either 2K or 4K bytes and is located on a 2K or 4K virtual storage boundary, respectively, within a segment.

For ECPS:VSE mode, virtual storage is divided into contiguous, fixed-length, consecutive sets of addresses called virtual storage pages. There are no segments for this mode. A virtual storage page is always 2K bytes in size and located on a 2K boundary.

The contents of virtual storage--instructions and data--are divided (by the operating system) into fixed-length contiguous areas called pages. For System/370 mode, a page corresponds in size to the virtual storage page size chosen, either 2K or 4K bytes. For ECPS:VSE mode, a page is always 2K bytes. The addresses associated with a virtual storage page refer to the contents of a page.

The direct access storage used to contain the portion of the total contents of virtual storage that is not always present in real storage is called external page storage. Direct access space within external page storage is divided into physical records called slots, which are of page size, either 2K or 4K bytes (always 2K for ECPS:VSE mode). Hence, a slot can contain one page at a time. A virtual storage page that is allocated and that actually has contents usually has a slot in external page storage associated with it to contain these contents (depending on the nature of the contents and how external page storage is managed by the operating system).

Instructions and data are transferred between external page storage and real storage, as needed, on a page basis. This transfer process is called paging, and a direct access device that contains external page storage is called a paging device. A slot in external page storage is associated with a particular virtual storage page by means of an algorithm or via tables that are maintained by the control program.

Real storage also is divided into fixed-length, consecutively addressed areas called page frames, which are always the same size as the page being used, either 2K or 4K bytes (always 2K for ECPS:VSE mode). Page frames are located on 2K or 4K real storage boundaries. A page frame is a block of real storage that can contain one page. Hence, a page of data and/or instructions occupies a slot when it is in external page storage and a page frame when it is in real storage. Whether or not a page is present in real storage, a program addresses the contents of the page using virtual storage addresses.

The act of transferring a page from external page storage into real storage is called a page-in. This action may also be described as the loading of a page. The reverse act, transferral of a page contained in real storage to a slot in external page storage, is called a page-out. Figure 15.05.4 illustrates the relationship of virtual storage, external page storage, and real storage that was conceptually shown in Figure 15.05.2.

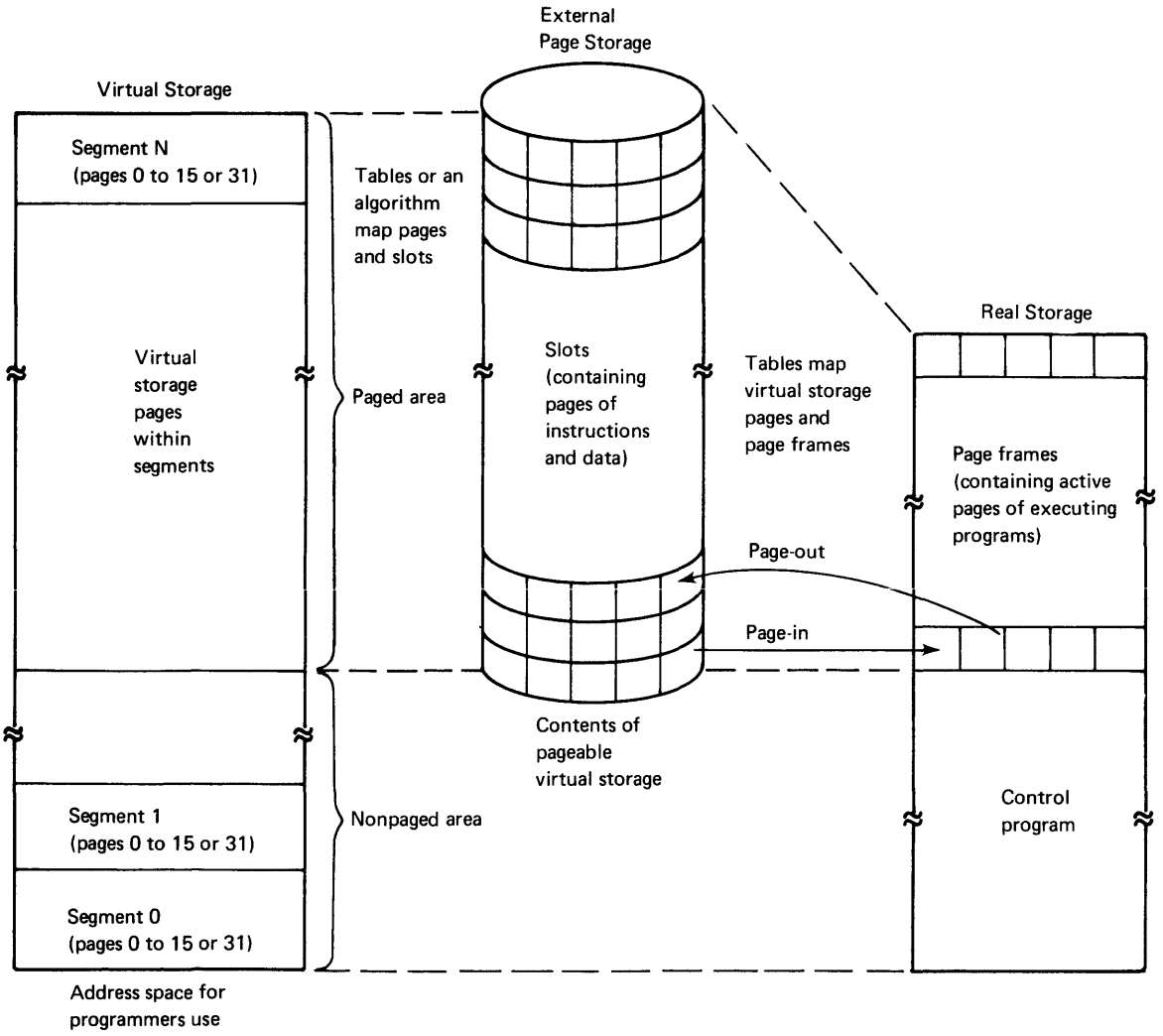


Figure 15.05.4. Layout of virtual storage, external page storage, and real storage

As previously indicated, DAT hardware for System/370 mode uses tables to perform address translation. These tables are the segment table and

page tables and are located in the processor storage available for programming (program processor storage). One segment table and a set of page tables are required to perform address translation for one virtual storage.

The segment table defines the virtual storage size, indicates allocated virtual storage, and points to the real storage location of the page tables. The page tables indicate which pages are currently in real storage and contain the real storage addresses of these pages. As pages are paged in and out, the control program makes changes to the page tables as required.

The internal mapping function provided for ECPS:VSE mode uses one table, called the address translation table, to perform virtual storage address to real storage address translation. This table indicates which pages are currently in real storage and the real storage addresses of these pages. The address translation table is located in auxiliary storage. A program cannot directly address the address translation table. However, instructions provided for page handling in ECPS:VSE mode cause the address translation table to be updated as necessary.

Basic to the implementation of virtual storage using direct access storage and address translation hardware is the method of determining when pages are to be brought into real storage and, therefore, when real storage is allocated to pages. The method supported by IBM-supplied virtual storage operating systems, that of bringing a page into real storage only when it is needed by an executing program, is called a demand paging technique. Since programs execute on a priority basis in DOS/VSE and OS/VS1 environments, as they do in OS MFT and MVT and DOS (Versions 3 and 4) environments, real storage is, in effect, still allocated on a priority basis.

When System/370 mode is in effect, a request for a page-in is generated by the occurrence of a page exception or a page translation exception, a condition that is also called a page fault. An interruption occurs during the execution of an instruction when dynamic address translation hardware attempts to translate a virtual storage address into a real storage address and the appropriate page table indicates that the page is not currently present in real storage. A page fault condition causes an interruption in order to alert the control program to the fact that a page frame must be allocated. Usually, a page-in is also required to bring in the referenced instruction or data.

When ECPS:VSE mode is in effect, a request for a page-in is generated when a page access exception (or page fault condition) occurs as a result of explicitly or implicitly addressing a virtual storage page that does not have real storage assigned (a page that is not addressable). The address translation table entry for the virtual storage page indicates its state.

While page-ins usually are initiated as a result of a page translation/page access exception, OS/VS1 and DOS/VSE provide an Assembler Language macro that can be used to cause one or more pages to be brought into real storage before they are referenced. Such requests are sometimes referred to as page-ahead requests. A page-ahead is required if, for reasons of proper system operation, a routine must operate without incurring any page translation/page access exceptions. However, unlimited use of this facility can defeat the objective of demand paging.

When a page translation/page access exception occurs and the control program determines that a page frame is not currently available for allocation, a choice must be made as to which allocated page frame will be taken away from the page to which it is currently assigned. The rule

governing this choice is called the page replacement algorithm. If the page replacement algorithm is designed to choose from among only those page frames currently allocated to the program that caused the page fault, it is said to operate locally. If a page frame can be chosen from among all those available for allocation to all executing programs, the algorithm is said to operate globally. DOS/VSE and OS/VS1 implement a global page replacement algorithm. VM/370 implements a global page replacement algorithm and supports a local page replacement algorithm as an option.

The algorithms used attempt to keep the most active pages of executing programs present in real storage. Hardware is included in the 4341 Processor that indicates whether or not a page has been referenced or changed. Hence, when a page frame is required, a page determined by the algorithm to be relatively inactive is chosen for replacement.

Before loading a new page into the page frame chosen, the existing contents of the page frame must be saved if they were modified during processing. If modification occurred, a page-out operation is required; otherwise, an exact copy of the page already exists in external page storage. Code that is not modified during its execution, therefore, has an additional advantage in a virtual storage environment in that it need never be paged out once it has been written in external page storage. A program requiring a page-in is placed in the wait state until the page it requires has been loaded, during which time processor control is given to another ready task, if one is available.

For various reasons, it is necessary to prevent a page-out of certain pages that are in real storage. One reason is for better operation of the system. This reason applies to a portion of the control program, some routines that operate with the processor in a disabled state (masked for I/O and external interruptions), most system tables, and most system control blocks. Integrity of system operation is another reason. Pages associated with certain types of operations must not be paged out while the operation is in progress, in order for the operation to proceed correctly.

For example, pages that contain I/O buffer areas must remain in real storage while the buffers are being referenced during an I/O operation, after which a page-out can take place, if necessary. Another reason is the existence of time dependency. A page should not be written out if the program to which the page belongs must complete a logical operation that requires the page in less time than it takes to perform a page-in. Programs that handle I/O device testing operations, such as online tests (OLTs), can have such a time dependency.

A page that is identified as one that cannot be paged out (or that is nonpageable) is called a fixed page. IBM-supplied virtual storage operating systems support both long-term fixing and short-term fixing, which are called permanent fixing and temporary fixing, respectively, in DOS/VSE. In VM/370, a nonpageable page is called a locked page. Pages that should never be paged out when they are present in real storage are marked permanently (long-term) fixed. The resident portion of an operating system control program is never paged and, therefore, its pages are marked long-term fixed.

Pages that must be fixed for only a portion of the time they are present in real storage are marked temporarily (short-term) fixed. For example, a page containing an I/O buffer is marked temporarily (short-term) fixed before the initiation of the I/O operation that references the buffer. After the I/O operation completes, the page is unfixed and it becomes eligible for a page-out. Pages should be marked fixed only when necessary, since page fixing reduces the amount of real storage that can be shared by concurrently executing paged programs (that real

storage available to be allocated to the nonfixed pages) and can, therefore, affect system performance.

As indicated previously, a portion of the control program is resident in real storage. That is, its pages are marked fixed and they are not placed in external page storage (because they are not paged) even though they are allocated space in virtual storage. In both DOS/VSE and OS/VS1, certain other portions of the control program are pageable and are made resident in virtual storage, which means they are contained in external page storage during system operation. During system initialization, these pageable control program routines are allocated virtual storage and loaded into real storage from system libraries by the program fetch routine. These routines will be written in external page storage as a result of normal paging activity. Control program routines that are resident in virtual storage are brought into real storage from external page storage, instead of from a system library, when they are required during system operation.

Just as control program routines can be fixed or pageable, problem programs operate in one of two modes in a DOS/VSE environment: virtual mode or real mode. For an OS/VS1 environment, these are paged mode or nonpaged mode, respectively. The latter is also sometimes called virtual equals real (V=R) mode.

When a problem program operates in virtual (paged) mode, it is resident in virtual storage and pageable. A pageable program operates in a contiguous area of virtual storage (partition) and is assigned available real storage on a demand paged basis. Hence, virtual storage addresses must be translated into real storage addresses. The real storage dynamically allocated to programs operating in paged mode need not be contiguous, and such programs normally can operate with less real storage than the design point (virtual storage) amount dynamically allocated to them. This is the mode of operation described in this subsection.

Virtual (paged) mode is the normal mode of operation of programs in a paging environment. However, certain programs cannot operate correctly in this mode and must run in real (nonpaged) mode. In general, a program must operate in real (nonpaged) mode if it:

- Contains a channel program that is modified while the channel program is active (for System/370 mode operations only). Section 15:10 discusses the reason.
- Is highly time-dependent (involves certain testing operations on I/O devices, for example)
- Must have all of its pages in real storage when it is executing (for performance reasons, for example)

In a DOS/VSE environment with System/370 mode in effect, one or more contiguously addressed real storage partitions must be defined if any programs are to operate in real mode. For ECPS:VSE mode, real partitions are not defined and real mode programs execute in virtual partitions that have permanently fixed page frames assigned. Real mode programs are not paged and do not occupy external page storage. The entire program (except for dynamically loaded phases) is loaded when the program is initiated and must operate in a real/virtual partition that is equal to or larger than its design point size.

In OS/VS1, a program that operates in nonpaged mode is dynamically allocated a contiguous virtual storage area and a contiguous real storage area with addresses identical to those of the allocated virtual storage area. (That is, virtual and real storage addresses of the allocated area are equal.) As in a DOS/VSE environment, programs

operating in nonpaged (V=R) mode are not paged and do not occupy external page storage. The entire program (except for dynamically loaded modules) is loaded into real storage when it is initiated, and all its pages are fixed. The amount of real storage allocated to a program that runs in nonpaged mode must be a multiple of the page size used.

15:10 ADDRESS TRANSLATION FACILITY FOR THE 4341 PROCESSOR OPERATING IN SYSTEM/370 MODE

When the 4341 Processor is operating in System/370 and EC modes, dynamic address translation hardware is made operative by turning on the translation mode bit in the current PSW. When DAT is operative, virtual storage addresses in programs referring to instructions and data are translated into real storage addresses after instructions are fetched during program execution. The address in the instruction counter is translated also. When DAT is not operative and System/370 mode is in effect, storage addresses in programs are used as real storage addresses.

When DAT is operative, the storage addresses in CCW lists are not translated by channel hardware during channel program operation. The channel indirect data addressing feature, also standard in the 4341 Processor, and programmed channel program translation are discussed later in this subsection under "Channel Indirect Data Addressing".

The following instructions are associated with the dynamic address translation facility: LOAD REAL ADDRESS (LRA), RESET REFERENCE BIT (RRB), and PURGE TLB (PTLB). The LRA and PTLB instructions are valid only for System/370 mode, with either EC or BC mode in effect. All three instructions operate in the same way regardless of which mode (EC or BC) is in effect and all are privileged.

VIRTUAL STORAGE ORGANIZATION

The 4341 Processor supports a virtual storage segment size of either 64K or 1024K bytes, as determined by bits 11 and 12 of control register 0. With either segment size, the page size can be 2K or 4K, as determined by bits 8 and 9 of control register 0. A segment size of 1024K bytes is not supported by DOS/VSE, OS/VS1, or VM/370. Table 15.10.1 summarizes the virtual storage organization provided by 4300 Processors for System/370 mode, which is identical to the virtual storage organization provided by System/370 processors.

Table 15.10.1. Number and size of segments and pages for a 16-million-byte virtual storage

CR 0 Bits 11,12	8,9	Segment Size in Bytes	Number of Segments in the Virtual Storage	Page Size in Bytes	Number of Pages in a Segment
10	01	1,048,576	16	2048	512
10	10	1,048,576	16	4096	256
00	01	65,536	256	2048	32
00	10	65,536	256	4096	16

While a 16-megabyte virtual storage is always available for System/370 mode operations as far as the DAT hardware is concerned, the actual amount of virtual storage to be supported for a given 4341 Processor is defined by the installation when the operating system to be used (DOS/VSE or OS/VS1) is generated. The size of the real storage in the given configuration is equal to the amount of program processor storage available (that is, the amount of processor storage available after the auxiliary storage requirement is subtracted).

As already described, the addresses supplied in programs directly address a location in the virtual storage that is supported by the virtual storage operating system. In this sense, program-supplied addresses can be viewed as virtual storage addresses that specify a byte within a particular virtual storage page and segment. The logic of the translation process is described in this subsection in these terms.

The architectural definition of dynamic address translation found in System/370 Principles of Operation (GA22-7000) assumes that the addresses in programs consist of three fields, two of which are used to index tables during the translation process. Under these conditions the addresses supplied by a program are considered to be logical addresses instead of virtual storage addresses.

For the purpose of translation, a virtual storage address is divided into three fields: (1) a segment field, which identifies a segment within the virtual storage; (2) a page field, which identifies a page within the segment addressed; and (3) a byte displacement field, which identifies a byte within the page addressed. The number of bits in each field varies depending on the segment and page sizes used. Virtual storage address fields for a segment size of 64K and a specific example of how the fields are used to address a location in virtual storage are shown in Figure 15.10.1.

OPERATION OF DYNAMIC ADDRESS TRANSLATION HARDWARE

Address Translation Tables

The page and segment tables used for the DAT facility in the 4341 Processor reside in real storage and are identical to the tables used by the DAT facility in System/370 processors without the System/370 Extended Facility/Feature (that is, there is no common segment capability for 4300 Processors). The address translation process using these tables in the 4341 Processor is identical to the process used in System/370. The segment and page table formats and entries used for address translation are shown in Figure 15.10.2.

One segment table is required to describe one virtual storage. If more than one virtual storage is supported by a single processor, there is a segment table for each virtual storage. A segment table contains one four-byte entry for each segment in the virtual storage the table describes, up to a maximum of 256 entries for the maximum-size virtual storage of 16 million bytes (using 64K segments).

The real storage address of the segment table (or of the currently active segment table if multiple virtual storages are implemented) is contained in control register 1. The current length of the segment table is also indicated in control register 1. The length value is used by the hardware during translation to ensure that the segment entry being referenced falls within the segment table. The segment table entries contain the real storage addresses of the page tables.

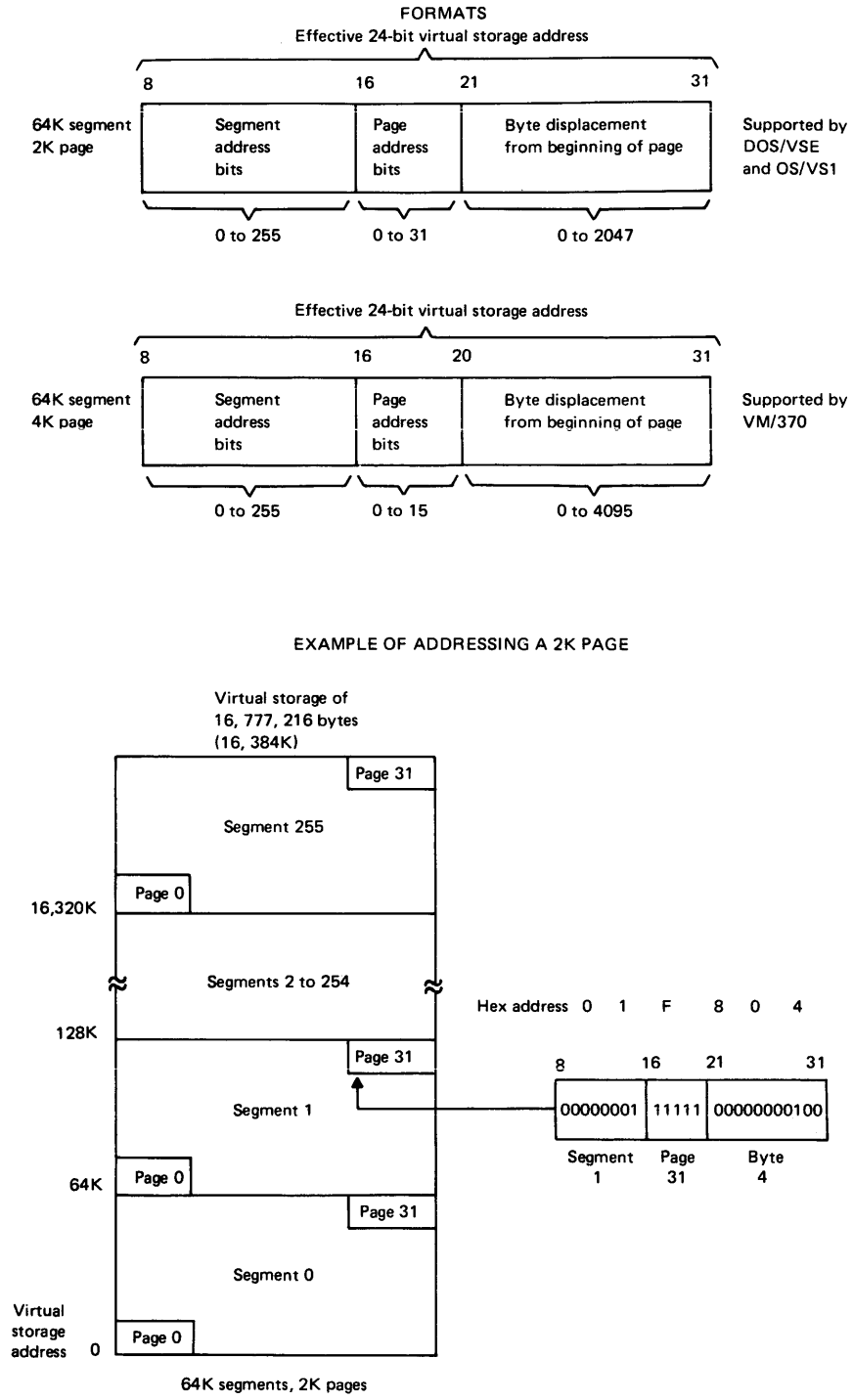


Figure 15.10.1. Virtual storage address fields for a 64K segment

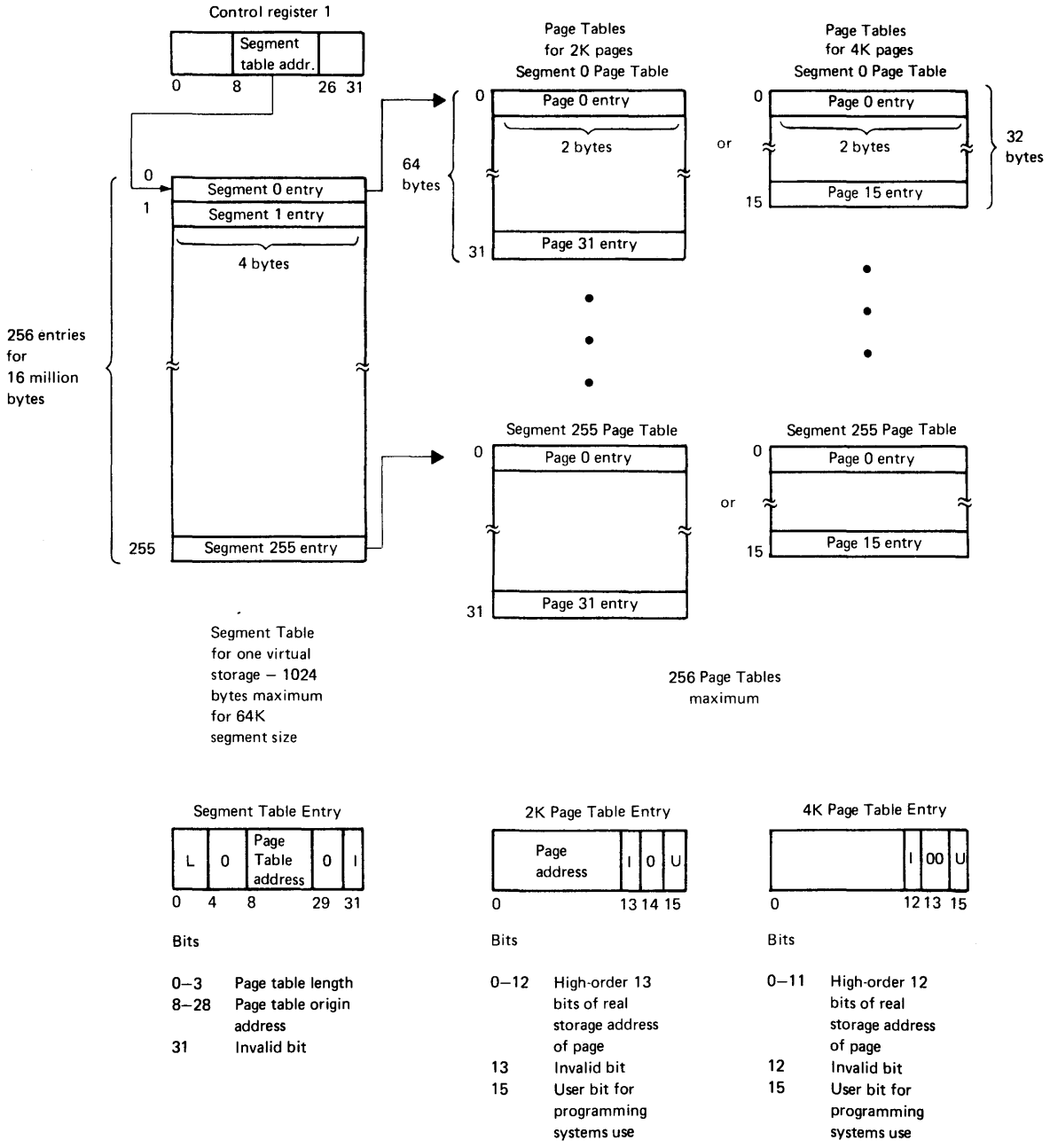


Figure 15.10.2. Segment table and page tables used for dynamic address translation

There is one page table for each segment in the virtual storage defined in a DOS/VSE or OS/VS1 environment, up to a maximum of 256 page tables for a 16-million-byte virtual storage with 64K segments. A segment table entry contains an indication of the length of its associated page table, the high-order 21 bits of the real storage address of the beginning of the page table, and an indication of whether or not the entry itself is valid and can be used for translation purposes (invalid bit). If the invalid bit is on in a segment table entry, a translation exception occurs during the translation process.

A page table has one entry for each page in the particular segment the page table describes. For a 64K segment, there are 32 or 16 entries in a page table depending on whether a 2K or a 4K page is used, respectively. A page table entry is two bytes in size. It contains the 12 (for a 4K page) or 13 (for a 2K page) high-order bits of the real storage address of the page frame that is currently allocated (if any) to the virtual storage page that the page table entry describes.

Each page table entry also contains an invalid bit to indicate whether the entry can be used for translation. The invalid bit is on when a virtual storage page does not have real storage currently allocated to it. A page translation exception occurs during the translation procedure if this invalid bit is on.

For System/370 mode, the control program maintains knowledge of the page frames available for allocation. When a translation exception occurs, the control program receives control and tries to allocate an available page frame. If none are free, the page replacement routine is executed to make a page frame available.

In effect, the segment and page tables define the relationship between virtual and real storage at any given time. The segment table reflects the current size of virtual storage and the location of required page tables. The segment table also indicates, by means of its invalid bits, which segments of virtual storage are currently allocated and have a page table available. The page tables indicate, via their invalid bits, which virtual storage pages currently have a page frame allocated and the location (real storage address) of these page frames.

In DOS/VSE and OS/VS1 environments, segment and page tables are defined by the control program at system initialization. Page tables are modified during system operation by control program routines to reflect the current allocation of real storage to virtual storage so that address translation can take place.

Address Translation Process

A translation request is either explicit or implicit. Explicit translation in System/370 mode is invoked via execution of the LOAD REAL ADDRESS instruction. Implicit translation is invoked to translate all instruction addresses and data addresses contained in other instructions. Implicit address translation takes place during instruction execution.

The logical flow of the translation process for System/370 mode is given in Figure 15.10.3. The procedure consists of a two-level, direct address table lookup operation that produces a real storage address. Any type of translation exception that occurs during the address translation process causes a program interruption and termination of the translation process. The processor cannot be disabled for translation exception interruptions. Segment and page translation exceptions that occur during explicit translation requests (LOAD REAL ADDRESS instruction) are indicated via the condition code setting instead of via an interruption.

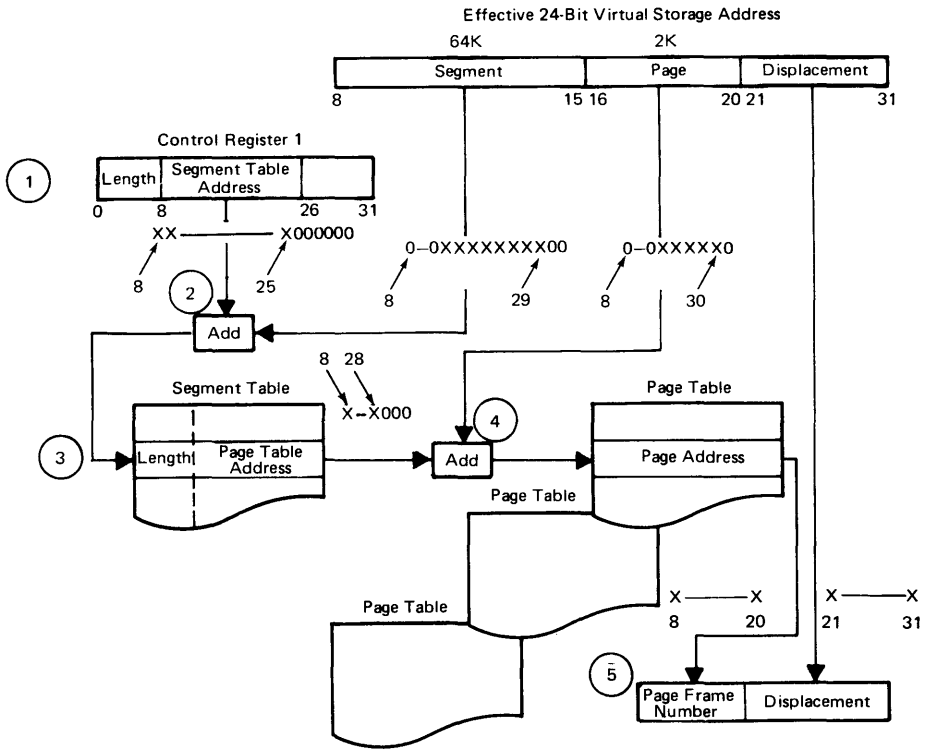


Figure 15.10.3. Dynamic address translation procedure for System/370 mode

The details of the translation process are as follows (refer to Figure 15.10.3):

1. Bits 8, 9, 10, and 11 in control register 0 are checked for validity. A translation specification interruption occurs if an invalid setting is present. Segment address bits from the virtual storage address are checked using length bits in control register 1. If the segment entry addressed is outside the segment table, a segment translation exception is indicated.

2. Six low-order zeros are appended to the segment table address in control register 1. Two low-order zeros are appended to the segment bits from the virtual storage address. The two values are added to obtain the real storage address of a segment table entry. If the invalid bit is on in the selected entry, a segment translation exception is indicated.

3. Page address bits from the virtual storage address are checked using page table length bits contained in the segment table entry. A page translation exception is indicated if the entry addressed is outside the page table.

4. Three low-order zeros are appended to the page table address contained in the segment entry. One low-order zero is appended to the page address from the virtual storage address. The two values are added to obtain the real storage address of a page table entry. If the invalid bit is on in this entry, a page translation exception is indicated.

5. The 24-bit real storage address is formed using the 12 or 13 high-order bits from the page table entry and the 12 or 11 low-order bits from the virtual storage address, depending on page size (2K or 4K). The resulting real storage address is used to access program processor storage.

The time required for address translation utilizing the DAT hardware in System/370 mode is 5.8 microseconds when 1024K-byte segments are used and 5.95 microseconds when 64K-byte segments are used. This translation time is eliminated if the processor translation lookaside buffer can be used for the translation. (See discussion below.)

Processor Translation Lookaside Buffer

When the 4341 Processor operates in System/370 mode with DAT specified, additional processor time is required to perform address translation using the segment and page tables. Thus, a processor translation lookaside buffer (TLB) is implemented primarily to minimize the amount of time required to perform address translation when DAT mode is enabled.

However, in the 4341 Processor, the TLB is also utilized when System/370 mode is in effect for BC mode operations and when EC mode is in effect without DAT enabled. The TLB is used for all System/370 mode operations so that the same microcode can be used, regardless of the other modes in effect, and because accessing the protect key in the TLB is faster than accessing the key stack.

As shown in Figure 15.10.4, the TLB contains 32 rows and 2 columns. Each row contains two entries (one per column). Each entry contains one address translation, three status bits, and the seven-bit storage key. A set of 32 least recently used (LRU) bits (one for each row) is provided for determining which column to assign when a translation is stored in the TLB. The LRU bits are set to indicate the left-hand column during a TLB reset.

An address translation in an entry consists of a virtual address and a real address when DAT mode is enabled or the same two real addresses when DAT mode is not enabled. The status bits are valid, virtual/real, and ACB check, while the storage key contains the store protect, fetch protect, reference, and change bits.

The valid bit indicates whether or not the entry is valid and can be used for address translation. When the TLB is reset during a processor reset, all the valid bits are turned off. In addition, when a SET STORAGE KEY instruction is issued, the entire TLB is searched by hardware. All virtual entries are invalidated as is any real entry that has the same real address as the one for which the key is being set.

The virtual/real bit indicates the entry contains a virtual/real address pair (virtual entry) or a real/real address pair (real entry). The TLB can contain a mixture of virtual and real entries. The bit is set when the entry is placed in the TLB based on the setting of the DAT bit. A real entry is indicated when the virtual/real bit is on.

The ACB check bit is used to indicate the real address in a virtual entry is invalid because it is equal to or higher than the address in the ACB register. When a virtual entry is placed in the TLB, the real address is checked against the ACB register and the ACB check bit is set, as appropriate.

When DAT is enabled, every time a virtual storage address is translated during instruction execution using the segment and page tables, the real storage address resulting from the translation

procedure, its associated access control key, and the virtual storage address are placed in the TLB. The validity bit for the entry is turned on. The virtual/real bit is set to zero to indicate a virtual entry. The ACB check bit is set to zero when the real address is less than the ACB register value.

Bits 8 to 12 of the virtual storage address are used to select the row to assign. The LRU bit for the row determines which column (left or right) is assigned. Thereafter, each time a TLB entry is loaded or its translation is used, the associated LRU bit is set to indicate the column not containing the referenced entry.

Note that the TLB is oriented toward a 2K-byte page size. When the page size is 4K bytes and a page is referenced that does not have a translation in the TLB, a translation is placed in the TLB for the 2K-byte portion of the 4K page that was referenced. A translation for the other 2K bytes of the page is not placed in the TLB unless that 2K bytes is referenced.

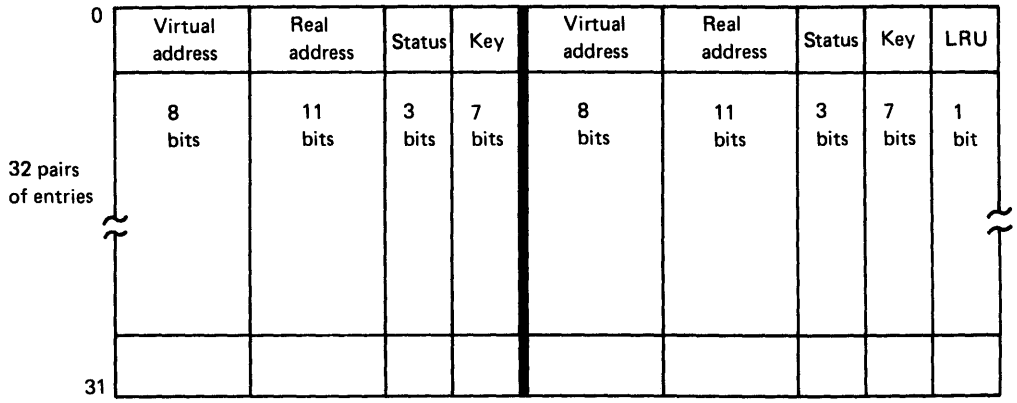


Figure 15.10.4. Processor TLB for the 4341 Processor

During System/370 mode operations with DAT enabled, when a virtual storage address must be translated, the TLB is inspected before the translation procedure is performed. Bits 8 to 12 of the virtual address in the instruction are used to select one of the 32 rows.

The virtual address in the instruction is compared simultaneously with the first address in each entry in the selected row. If there is an equal comparison with either address, the status bits of the associated entry are checked. If the address is virtual (virtual/real bit is zero), the entry is valid, and the ACB check bit is off, the real address bits in the entry are combined with bits 13 to 31 of the virtual address in the instruction to form the real storage address. The LRU bit is set appropriately. When the required real storage address can be obtained from the TLB, no address translation time is added to the time required for execution of the instruction.

If the valid bit is off, the translation is not used and the full translation procedure is performed. If the ACB check bit is on, an addressing exception program check occurs and no further address translation processing occurs. If an equal comparison does not occur when the instruction address is compared with the two addresses in the addressed TLB row, the full translation process, using segment and page tables, is performed, as previously described, to obtain a real storage address. The appropriate bits of the virtual storage address in the instruction and the appropriate real storage address bits obtained from

the translation process are placed in the TLB. The LRU bit setting for the selected row determines the column assigned.

When DAT is not enabled during System/370 mode operations, the real addresses in instructions are used to address the TLB. When the TLB contains the required real address pair, no address translation time is added to instruction processing time. If the TLB does not contain the required real storage address, the real storage address is placed in the two address locations in the selected TLB entry (as determined by the LRU bit setting). When the real address is not in the TLB during System/370 mode operations with DAT disabled, 1.43 microseconds are added to instruction processing time.

All entries in the TLB are automatically invalidated when the page size in effect changes. The PURGE TLB instruction provides the capability of invalidating all the TLB entries by programming, when required. For example, in general, this instruction must be issued whenever an entry in a page table is invalidated, since the processor storage address associated with the real storage address bits being invalidated could be contained in the TLB. (The control program purges the TLB as required.)

Operation of the TLB cannot be disabled. If an error occurs in the TLB, the instruction in execution when the error occurred is retried if it is a retryable instruction. For unretryable errors, an uncorrectable error condition (instruction processing damage) exists (see Section 50).

Addresses Translated

All storage addresses that are explicitly designated by a program and that are used by the instruction processing function to refer to instructions or data in processor storage are virtual storage addresses when System/370 and DAT modes are in effect and are subject to address translation. Thus, when DAT is operative, the starting and ending storage addresses used with the program event recording feature are virtual, as are the storage addresses stored in PSWs during interruptions.

Address translation is not applied to addresses that explicitly designate access control key storage locations or to quantities that are formed as storage addresses from the values designated in the base and displacement fields of an instruction that are not used to address processor storage (shift instructions, for example). In addition, address translation is not applied to the storage addresses in CCW lists used for I/O operations during System/370 mode operations.

Some of the storage addresses supplied to a program by the instruction processing function are virtual and some are real. Table 15.10.2 lists, for the 4341 Processor operating in System/370 mode with DAT enabled, those storage addresses designated by a program, either explicitly or implicitly, that are virtual. The virtual addresses are subject to translation using the TLB or segment and page tables. Table 15.10.2 also indicates those storage addresses that are real or not used to reference processor storage. The table also indicates which storage addresses supplied to a program are virtual and which are real.

Table 15.10.2. Virtual and real storage addresses used by, and supplied to, programs in the 4341 Processor operating in System/370 mode with DAT enabled

Virtual Storage Addresses Explicitly Designated by the Program (translated using the segment table and page tables)

- Instruction address in the PSW
- Branch addresses
- Addresses of operands in real storage
- Operand address in LOAD REAL ADDRESS instruction
- PER starting address in control register 10 and PER ending address in control register 11

Real Storage Addresses Explicitly Designated by the Program (not translated)

- Operand addresses in SET STORAGE KEY, INSERT STORAGE KEY, and RESET REFERENCE BIT instructions
- Segment-table-origin address in control register 1
- Page-table-origin address in a segment table entry
- Page frame address in a page table entry
- CCW address in the channel address word (CAW)
- Address in a CCW specifying a data area or the location of another CCW
- Data address in channel indirect data address lists

Addresses Not Used To Address Storage (not translated)

- Operand addresses specifying the amount of shift in fixed-point, logical, or decimal shift instructions
- Operand address in LOAD ADDRESS and MONITOR CALL instructions
- I/O addresses in I/O instructions and the fixed processor storage area

Real Storage Addresses Used Implicitly (not translated)

- Addresses of PSWs used during an interruption and in executing the programmed or manually initiated restart function
- Address used by processor to update the timer at location 80
- Address of the CAW, the CSW, and the I/O address within the fixed processor storage area used during an I/O interruption or during execution of an I/O instruction, including execution of STORE CHANNEL ID
- Addresses used for the store status function

Virtual Storage Addresses Provided to the Program

- Address stored in the instruction address field of the old PSW during an interruption
- Address stored by a BRANCH AND LINK instruction
- Address stored in register 1 by TRANSLATE AND TEST and EDIT AND MARK instructions
- Address stored in location 144 on a program interruption for a page translation or segment translation exception
- Address stored in location 152 on a PER interruption

Real Storage Addresses Provided to the Program

- The translated address generated by a LOAD REAL ADDRESS instruction
- Address of a segment table entry or page table entry provided by the LOAD REAL ADDRESS instruction
- Failing storage address at location 248
- CCW address in the CSW

Reference and Change Recording Facility for Processor Storage Blocks

A hardware recording facility is standard in the 4341 Processor. This facility provides continuous recording of the activity of all 2K processor storage blocks in the program processor storage sections via reference and change bits. The settings of these recording bits can be used by control program routines to support a demand paging environment. This hardware facility is always active in both System/370 and ECPS:VSE modes.

The seven-bit storage key associated with a 2K processor storage block in the 4341 Processor has four access control bits (for store protection), one fetch protection bit, one reference bit, and one change bit. In the 4341 Processor, the storage keys are located in the key stack and a copy of the storage key settings is maintained in the address translation table.

During processor operation, the activity of each 2K block in program processor storage is monitored by hardware. Whenever a fetch is made by either the instruction processing function or a channel to a real storage address during System/370 mode operations, the reference bit in the storage key associated with the 2K processor storage block that contains that real storage address is turned on by the hardware. A store into any real storage address causes the hardware to turn on both the change bit and the reference bit for the affected 2K block.

Alter/display operations initiated from the operator console also cause appropriate changing of reference and change bits. The RESET REFERENCE BIT instruction is provided to allow the reference bit of any 2K storage block to be reset to zero by programming without altering the contents of the other six bits in the storage key.

The hardware reference and change recording facility is used by the page replacement algorithm of a virtual storage operating system. When a page is loaded into a page frame, the reference and change bits for that page frame are set to zero. Thereafter, the reference bit is used to determine the activity of a page. The change bit is inspected to determine whether a page must be paged out when its page frame is reassigned. The SET STORAGE KEY instruction must be used to reset a change bit.

Instruction Nullification

When a page fault occurs in a demand paging environment, execution of the instruction that caused the page fault stops and the control program gains control to initiate a page-in operation. When the contents of the missing page have been loaded (and the appropriate page table entry has been updated), the instruction that caused the page fault is reissued. In order for the instruction to operate correctly the second time, execution of the instruction must have been stopped in such a way that reexecution gives the same results as would have occurred if the instruction had been executed only once. Therefore, the contents of processor storage, the general and floating-point registers, and the PSW must not be altered.

The execution of an instruction is said to be nullified when it is stopped so that no operation is performed, no fields are changed, and the PSW indicates the address of the instruction that was stopped. Interruptible instructions, such as MOVE LONG, are divided into execution units. One or more execution units may have completed before a page fault is detected. In this case, only the current execution unit is nullified.

Various methods are used, depending on the type of instruction, to determine the need for nullification. In some cases, execution of the instruction is attempted where hardware detection of page faults permits nullification. In other cases, pretesting is required to determine whether the virtual storage pages to be referenced have page frames allocated. Nullification testing is required only for instructions that reference virtual storage.

CHANNEL INDIRECT DATA ADDRESSING

Since address translation is not performed by the channels for programs that operate in paged (virtual) mode when System/370 mode is in effect and DAT is enabled, address translation must be performed on CCW lists by programming before the initiation of I/O operations. Such address translation need not be performed on the CCW lists of programs that operate in nonpaged (real) mode.

In addition, a contiguously addressed I/O area in virtual storage can span a set of noncontiguous page frames. Hence, a method of handling a noncontiguously addressed I/O area in real storage during the operation of a CCW list is required. The standard channel indirect data addressing feature is used to provide this capability. It applies to the byte multiplexer channel and all block multiplexer channels. As shown in Figure 15.10.5, the use of channel indirect data addressing allows the channel program logic used in the CCW list with virtual storage addresses to be maintained in the new CCW list that contains real storage addresses.

When channel indirect data addressing is present in a processor, bit 37 of a CCW is designated as the indirect data address (IDA) flag. The IDA flag applies to read, read backward, write, control, and sense commands and is valid in both BC and EC modes. When the IDA flag in a CCW is zero, bits 8 to 31 of the CCW specify the real storage address of the beginning of the I/O area as usual.

When the I/O area referenced by a CCW is completely contained in one page, an indirect data address list (IDAL) is not required and the IDA flag is set to zero. When the IDA flag is one, CCW bits 8 to 31 specify the real storage address of an IDAL instead of an I/O area. When the I/O area referenced by a CCW spans two or more pages, an IDAL is required and the IDA flag is set to one.

An IDAL consists of two or more contiguous indirect data address words (IDAWs) of four bytes each aligned on a fullword boundary. There is one IDAW in an IDAL for each 2K storage block spanned by the I/O area. An IDAW, which must be aligned on a fullword boundary, contains a real storage I/O area address in bits 8 to 31. Bits 0 to 7 must be zero. The first IDAW in the list points to the beginning of the I/O area to be used by the CCW and is obtained by translating the virtual storage address contained in the original CCW.

Any valid real storage address can be specified in the first IDAW of a list. All IDAWs after the first must address the beginning (or end for a read backward operation) of a 2048-byte block located on a 2048-byte boundary, or a program check occurs. That is, bits 21-31 of the address in the IDAW must be zeros (or ones for a read backward).

Figure 15.10.5 shows an example of the IDALs required for a command-chained CCW list when 2K pages are used. The IBM-supplied virtual storage operating systems construct a new CCW list with translated (real storage) addresses that is used to control the I/O operation. The new CCW list points to any required IDALs.

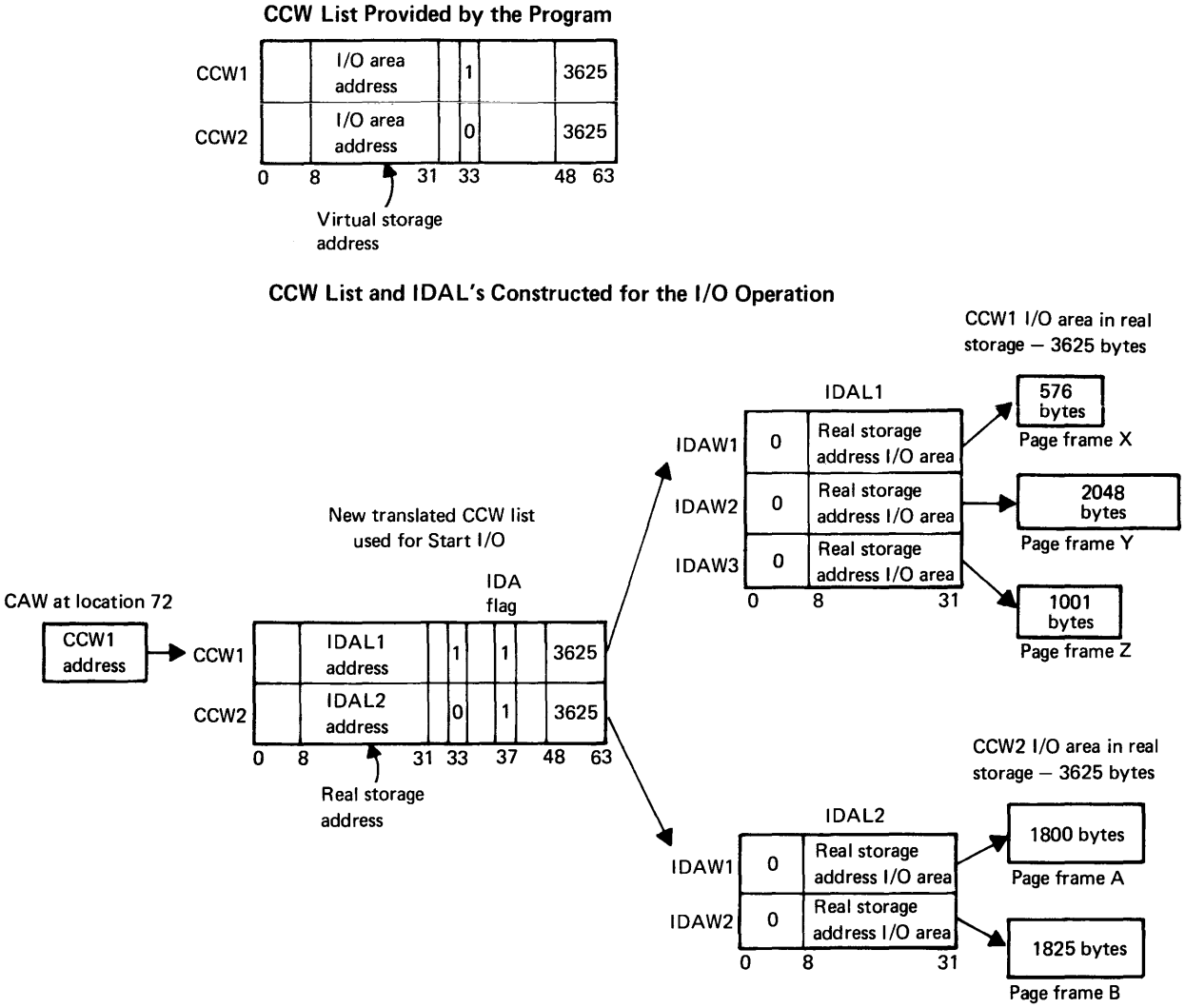


Figure 15.10.5. Example of IDALs required for a CCW list when page size is 2K

When a START I/O instruction is executed, the channel fetches the first CCW in the list, pointed to by the channel address word (CAW), and inspects bit 37. If it is zero, the I/O operation is started to the real storage address specified in the CCW.

If bit 37 in the first CCW is a one, the first IDAW is fetched from real storage address specified in the CCW. The I/O operation is begun using the real storage address in the first IDAW. Assuming that the I/O operation is not a read backward, ascending real storage addresses in the I/O area are used by the channel until a 2048-byte boundary is reached.

The channel detects a 2K boundary by monitoring I/O area address bits 21-31. When these bits change from all ones to all zeros, which causes a carry from bit 21 when the address is incremented by one, the first byte of the next 2K real storage block is indicated. At this point, the channel accesses the second IDAW in the list to obtain the next real storage I/O area address to be used and the data transfer operation continues. The channel continues using the IDAL until the operation indicated by the CCW completes (CCW count reaches zero, IBG on tape is

reached, etc.). The next CCW is accessed if command or data chaining is indicated. Bit 37 is inspected and the I/O operation continues as described until the CCW list is exhausted.

When a program operates in paged mode, the CCW list for each I/O operation must be inspected, a new CCW list with real addresses must be built, and the appropriate IDALs must be constructed before a START I/O instruction is issued. At the completion of an I/O operation, some retranslation is also required. In general, the following steps must be taken for each CCW in a given list:

1. Determine whether the I/O area specified in the CCW spans pages or is contained in only one. If a single page is involved, translate the virtual storage address to a real storage address and store it in the CCW. Ensure that a page frame is allocated to the page containing the buffer and that the page frame is marked fixed.

2. If two or more pages are involved, set up the required number of IDAWs, place a pointer to the IDAL in the CCW, and turn on CCW bit 37.

3. While setting up IDAWs, determine whether all pages in the I/O area have real storage assigned. If not, ensure that page frames are allocated and fixed.

At the completion of the I/O operation, the real storage address in the channel status word must be translated to a virtual storage address, and the pages that were short-term fixed before the initiation of the I/O operation must be unfixed. Channel program translation and page fixing are performed by the I/O control portion of the control program in IBM-supplied virtual storage operating system support.

A program containing a CCW list that is dynamically modified during its execution in System/370 mode with DAT enabled cannot operate correctly in paged mode, since the modification is made to the CCW list with virtual storage addresses rather than to the translated CCW list that is actually controlling the I/O operation on the channel.

15:15 ADDRESS TRANSLATION FACILITY FOR THE 4341 PROCESSOR OPERATING IN ECPS:VSE MODE

The address translation facility for ECPS:VSE mode is an internal mapping function that utilizes the address translation table. This internal mapping function is always active when ECPS:VSE mode is in effect. That is, it cannot be disabled and is used for both BC and EC modes. All addresses in programs are assumed to be virtual when ECPS:VSE mode is in effect for both BC and EC modes.

The internal mapping function is used to translate virtual storage addresses in instructions and the instruction counter to real storage addresses during program execution. Virtual storage addresses in CCW lists are translated to real storage addresses during channel program operation using a set of channel TLBs. Reference and change recording and instruction nullification are performed in ECPS:VSE mode in the same manner as described for System/370 mode.

Several privileged instructions are provided for page control during ECPS:VSE mode operations. These instructions, which are valid only when ECPS:VSE mode is in effect (for both BC and EC modes), are:

- CLEAR PAGE
- CONNECT PAGE

- DECONFIGURE PAGE
- DISCONNECT PAGE
- INSERT PAGE BITS
- LOAD FRAME INDEX
- MAKE ADDRESSABLE
- MAKE UNADDRESSABLE
- SET PAGE BITS
- STORE CAPACITY COUNTS

The CLEAR PAGE instruction enables 2K bytes of program processor storage located on a 2K boundary to be cleared quickly with a single instruction and validated. CLEAR PAGE should be used in ECPS:VSE mode (instead of MOVE LONG, for example) to validate processor storage. The other page control instructions are provided to enable the control program to support address translation.

VIRTUAL STORAGE ORGANIZATION

In the 4341 Processor, a virtual storage size of 16,777,216 bytes is always supported for ECPS:VSE mode. Virtual storage size cannot be set by the operator, as for a 4331 Processor.

Virtual storage is divided into 2K-byte virtual storage pages that are located on 2K-byte address boundaries. These pages are addressed 0 to 8191. The page control instructions listed previously can address pages in virtual storage using the address of any byte within the virtual storage page. When the entire page is being addressed, only the 13 page address bits (8 to 20 in the virtual address) are utilized.

In order for a virtual storage page to be accessed by the instruction processing function or channel programs, it must have processor storage assigned. For ECPS:VSE mode, as for System/370 mode, the processor storage available to programs is divided into 2K-byte page frames that are located on 2K-byte address boundaries. A page frame can be assigned to only one virtual storage page at a time. The allocation and deallocation of page frames to virtual storage pages is accomplished using page control instructions.

Each page frame has a unique 16-bit binary integer associated with it that is called its frame index. The first page frame in program processor storage has frame index 0, the next has frame index 1, etc. The maximum value of the frame index is the total number of page frames in program processor storage less one. The frame index is the method by which the instruction processing function keeps account of individual page frames for assignment.

Virtual Storage Page States

Each virtual storage page is considered to be in one of three states: disconnected, connected, or addressable. The state of a virtual storage page determines its accessibility and is checked whenever the virtual storage page is addressed explicitly or implicitly by the instruction processing function or the channels.

A virtual storage page is in the disconnected state when it does not have a page frame assigned. A disconnected virtual storage page cannot

be accessed and causes a page access or page state exception if the instruction processing function attempts to access the disconnected page. A page access exception occurs when the referenced virtual storage page does not have a page frame assigned (as indicated by the address translation table). A page transition exception can occur only when a page control instruction that causes a page state change is issued and an invalid page state transition is attempted. An I/O interruption that indicates protection check occurs when a channel attempts to access a disconnected page.

A connected virtual storage page has a page frame assigned. It can be accessed by the channels but not by the instruction processing function except via the CLEAR PAGE instruction. A page access exception occurs if the instruction processing function attempts to access a connected page other than by a CLEAR PAGE instruction. In effect, a connected virtual storage page is not disconnected and not yet addressable by the instruction processing function.

An addressable virtual storage page has a page frame assigned and is addressable by the instruction processing function and the channels. Normally, a virtual storage page is placed in the addressable state after a page-in is performed. Page zero is always addressable. It cannot be placed in the connected or disconnected state.

The connected state is defined to provide protection against accessing a page frame during a page-in or page-out operation. In System/370 and when System/370 mode is in effect in the 4341 Processor, a page frame cannot be accessed by the instruction processing function during a page-in or page-out, since the invalid bit in the page table entry for the associated virtual storage page is on during the paging I/O operation. However, a channel can access the page frame during an I/O operation, since the CCWs contain real addresses.

The states of virtual storage pages are managed using page control instructions. A disconnected virtual storage page must enter the connected state before it can enter the addressable state. Similarly, an addressable virtual storage page must enter the connected state before it can enter the disconnected state.

The CONNECT PAGE instruction is used to assign a free page frame to a disconnected virtual storage page. It specifies a general register and the address of the virtual storage page to be placed in the connected state. If a page frame is free, it is assigned to the virtual storage page, the page is placed in the connected state, the frame index of the assigned page frame is placed in the specified general register, and the condition code is set to indicate a successful connection. See the discussion under "Page Frame Assignment" in this subsection for an explanation of how the page frame allocated by a CONNECT PAGE instruction is chosen by the instruction processing function.

When the CONNECT PAGE instruction is issued for a virtual storage page that is already connected, the frame index is returned in the specified register and the condition code indicates the page was already connected. If the virtual storage page is in the addressable state when the CONNECT PAGE is issued, a page transition exception exists and the instruction is suppressed. A program interruption occurs.

If no page frames are free when the CONNECT PAGE instruction is issued, no frame index is stored in the specified register and the condition code indicates an unsuccessful connection.

The MAKE ADDRESSABLE instruction is used to place a connected virtual storage page in the addressable state. This instruction specifies the address of the virtual storage page that is to be made addressable. If the page is in the connected state, it is made addressable and the

condition code is set to indicate the page was in the connected state. If the MAKE ADDRESSABLE instruction is issued to a virtual storage page that is already in the addressable state, it remains addressable and the condition code is set to indicate the page was already addressable. If the page was in the disconnected state, a page transition exception exists, the instruction is suppressed, and a program interruption occurs.

The MAKE UNADDRESSABLE instruction is used to place an addressable virtual storage page in the connected state. It specifies only the address of a virtual storage page. If the specified page is in the addressable state, it is placed in the connected state and the condition code is set to indicate the page was in the addressable state. The condition code indicates already connected when the MAKE UNADDRESSABLE instruction is issued to a connected page, and a page transition exception program interruption occurs if the page was in the disconnected state.

The DISCONNECT PAGE instruction is used to place a connected virtual storage page in the disconnected state. It specifies the address of a virtual storage page. If the page is in the connected state, it is placed in the disconnected state and the condition code is set to indicate the page was in the connected state. When the DISCONNECT PAGE instruction is issued to a page that is already disconnected, the page remains in that state and the condition code indicates the page was already disconnected. A page transition exception program interruption occurs if the page was in the addressable state.

The LOAD FRAME INDEX instruction can be used to determine whether a page frame is assigned to a virtual storage page. The returned condition code indicates the state of the specified virtual storage page (addressable, connected, or disconnected). The frame index of the page frame assigned is returned in the specified register if the page is addressable or connected.

Virtual Storage Page Description

A page description is associated with each virtual storage page in the virtual storage defined. This page description is located in the address translation table and consists of the following:

- A seven-bit storage key that consists of a four-bit access control key, a fetch protection bit, one reference bit, and one change bit. The access control key and fetch protection bit are used to provide store and fetch protection that is functionally equivalent to the same facility in System/360 and System/370. The reference and change bits provide the same reference and change recording function as described for System/370 mode. Reference and change recording in the 4341 Processor is functionally equivalent to the same facility in System/370.

  Note that the following instructions (while they address a page) do not cause implicit setting of the reference or change bit: CONNECT PAGE, INSERT PAGE BITS, INSERT STORAGE KEY, LOAD FRAME INDEX, MAKE ADDRESSABLE, and MAKE UNADDRESSABLE. The DECONFIGURE PAGE and DISCONNECT PAGE instructions cause the reference and change bits to be turned off while RESET REFERENCE BIT turns off the reference bit. The SET PAGE BITS and SET STORAGE KEY instructions cause the reference and change bits to be set, as indicated in the instruction.

- Three programmable bits that are provided for use by the page supervisor of virtual storage operating systems. For example, one of these bits could be utilized like the user bit (15) in the page

table entry. IBM-supplied operating systems that operate in System/370 mode use bit 15 to indicate whether a page-in is required when a page frame is assigned.

- Page state bits (two) to indicate the state of the page (disconnected or connected and unaddressable or addressable)

- The address of the page frame currently assigned to the virtual storage page, if any. This value is bits 8 to 20 of the real storage address of the assigned page frame.

The page control instructions, except for CLEAR PAGE, operate on the page description for the addressed virtual storage page rather than on the page itself.

Virtual Storage Page and Page Frame Capacity Counts

Four capacity counts are defined for the management of virtual storage pages and page frames during ECPS:VSE mode operations. Each count is a 16-bit unsigned binary integer that is initialized during IML and updated by the processor during processing. These four counts, which are not established for System/370 mode of operation, are located in the K-addressable area of auxiliary storage in the 4341 Processor. The four counts can be placed in real storage (each as a 32-bit unsigned binary integer with 16 high-order zeros) using the STORE CAPACITY COUNTS instruction.

The four capacity counts are:

- Page capacity count (PCC), which is the number of virtual storage pages in the virtual storage supported. This count is set during IML and for the 4341 Processor is always 8192.

- Existing frame capacity count (EFCC), which is the number of page frames in program processor storage. This count is established whenever a processor clear reset occurs.

- Available frame capacity count (AFCC), which is the existing frame capacity count less any page frames that are made unavailable for use by programming via execution of the DECONFIGURE PAGE instruction during processor operation (because they are malfunctioning, for example). This is the number of page frames available for allocation to virtual storage pages during processor operation. During a processor clear reset operation in the 4341 Processor, the AFCC is set to the same value as the EFCC.

- Free frame capacity count (FFCC), which is the number of page frames that are currently not allocated to a virtual storage page and thus are available for assignment. The FFCC can range from 0 to the AFCC minus one. (Since virtual storage page 0 must always be addressable, one page frame can never be free.) The FFCC is changed as required during processor operation as each CONNECT PAGE and DISCONNECT PAGE instruction is executed.

  During a processor clear reset operation, the FFCC is set to zero whenever the PCC is equal to or greater than the AFCC because reset processing causes the allocation of all available page frames to the lowest addressed virtual storage pages.

Page Frame Assignment

In ECPS:VSE mode, as in System/370 mode, the allocation and deallocation of page frames to virtual storage pages during processor

operation is initiated by programming. However, in ECPS:VSE mode, the actual page frame that is assigned to a virtual storage page is selected by the hardware instead of by programming, as in System/370 mode.

In ECPS:VSE mode, the hardware maintains a list of the addresses of the page frames that are free for allocation to virtual storage pages. This list is contained in the page management area of auxiliary storage. This area is 2K bytes for a processor storage size of two megabytes and 4K bytes for a processor storage size of four megabytes.

After processor initialization, the page management area is empty, since all program processor storage is allocated to the first N virtual storage pages. Page frame addresses are added to and deleted from the page management area as DISCONNECT PAGE and CONNECT PAGE instructions, respectively, are issued during processing.

A last-in, first-out queuing technique is used for the page frame addresses in the page management area. A pointer is maintained in K-addressable storage that indicates the next address location in the page management area to be used. It points to the next address location in the area after the last address location containing the frame index of an available page frame. The pointer is initialized to indicate the first address location in the area.

When a page frame address is added to the area, as a result of the execution of a DISCONNECT PAGE instruction, it is placed in the address location indicated by the pointer and the pointer value is incremented. When a CONNECT PAGE instruction is issued, the pointer value is decremented. The page frame address in the location indicated by the decremented pointer value is used for the connection and, in effect, is removed from the page management area.

A page frame must be assigned to a virtual storage page during processor operation in ECPS:VSE mode when a page access exception occurs. This exception occurs for an instruction when the address translation table indicates the referenced page is not in the addressable state. If the page is in the connected state, a MAKE ADDRESSABLE instruction should be issued. Otherwise, the CONNECT PAGE instruction should be issued to cause the assignment of free page frame.

If the page management area is not empty (FFCC is not zero), the CONNECT PAGE instruction causes the instruction processing function to decrement the page management pointer and assign the free page frame indicated by the pointer. The contents of the assigned page frame are not cleared. The FFCC is reduced by one. The virtual storage page is placed in the connected state (connected bit in the associated address translation table entry is turned on). After a page-in of the required page is performed (if necessary), the MAKE ADDRESSABLE instruction must be issued to place the connected page in the addressable state so that it can be accessed.

If the FFCC is zero when a CONNECT PAGE instruction is issued to a disconnected virtual storage page, the control program must then execute its page replacement algorithm to make a page frame free for assignment. When the control program determines the virtual storage page whose page frame is to be taken, it first must issue a MAKE UNADDRESSABLE instruction to place the virtual storage page in the connected state.

While the virtual storage page is in the connected state, a page-out can be performed if the page was changed. The DISCONNECT PAGE instruction should be issued after the page-out. The address of the page frame the virtual storage page was assigned is placed in the page management area and the FFCC is increased by one when the instruction is issued. The page frame is not cleared by the processor.

The preceding procedure makes a page frame free for assignment to the virtual storage page that caused the page access exception while the FFCC was zero. A CONNECT PAGE instruction for that virtual storage page can then be issued again and connection will occur. The MAKE ADDRESSABLE instruction should then be issued to place the virtual storage page in the addressable state so that address translation can be performed.

When the DECONFIGURE PAGE instruction is issued, the AFCC is decremented and the address of the specified page frame is not placed in the page management area. In effect, the page frame becomes unavailable for use.

While programmed translation of the virtual storage addresses in channel programs is not required for ECPS:VSE mode operations, all virtual storage pages within the buffers addressed by a given channel must have page frames assigned and the page frames must be fixed before the channel program is started. The LOAD FRAME INDEX instruction can be used to determine whether page frames are assigned to buffer areas.

OPERATION OF ADDRESS TRANSLATION

Address Translation Table

The address translation table is used in ECPS:VSE mode for the translation of virtual storage addresses in instructions and channel programs when the required translation is not in the processor TLB or channel TLB. There is one four-byte entry in the address translation table for each virtual storage page. Thus, the size of the address translation table is 32K bytes for a 16-megabyte virtual storage.

A 32-bit address translation table entry contains the following:

- Connected bit (0). This bit indicates whether the virtual storage page associated with this entry is connected (bit is 1) or disconnected (bit is 0). This bit is set by microcode when CONNECT PAGE and DISCONNECT PAGE instructions are issued. When a page is addressable, its connected bit is one to indicate it is also connected.

- Addressable bit (1). This bit indicates whether the virtual storage page associated with this entry is addressable (bit is 1) or unaddressable (bit is 0). The setting of this bit indicates whether the entry can be used for translation purposes. This bit is set by microcode when MAKE ADDRESSABLE and MAKE UNADDRESSABLE instructions are issued.

- Programmable bits (5, 6, and 7). These bits are set by programming using the SET PAGE BITS instruction. Their contents can be placed in program processor storage using the INSERT PAGE BITS instruction.

- Page frame address bits (8 to 20). This entry is the address of the page frame currently assigned to the associated virtual storage page when it is in the addressable or connected state. This address is set by the microcode when it assigns a page frame as a result of a CONNECT PAGE instruction.

- Storage key bits (21 to 27). These bits are the reference, change, fetch protection, and four access control bits (in the sequence listed) that are used for reference and change recording and store and fetch protection. The entire storage key (all seven bits) is set using the SET STORAGE KEY instruction and inspected using the INSERT STORAGE KEY instruction. The reference and change bits also can be set and reset using the SET PAGE BITS instruction and the

INSERT PAGE BITS stores their value in program processor storage. The RESET REFERENCE bit is used to set the reference bit to zero and, via the condition code setting, determine the setting of the reference and change bits before the instruction was executed. The reference and change bits are altered by the microcode as pages are referenced and changed.

During a processor clear reset operation for ECPS:VSE mode, the microcode assigns all existing page frames in program processor storage to the first N virtual storage pages and initializes the address translation table as follows:

- All addressable bits and connected bits are set to one in the first N entries, which are for the first N virtual storage pages with a page frame assigned. The virtual storage pages are cleared to zeros. For the balance of the entries these bits are set to zero.
- All programmable bits in all entries are set to zero.
- The first N entries with page frames assigned have a valid page frame address that indicates the page frame assigned to the associated virtual storage page. The balance of the entries have zeros in this field.
- All storage key bits in all entries are set to zero.

This initialization enables a control program that operates in BC mode (the default established during IPL) and does not support ECPS:VSE mode (such as a System/360 operating system) to execute with this mode in effect. No page access exceptions will occur during translation operations as long as no address higher than the program processor storage size is addressed.

Translation Process for Instructions

Address translation for instructions occurs as follows during instruction execution with ECPS:VSE mode in effect. As each instruction is processed, the microcode uses any virtual storage address in the instruction to directly address the address translation table (if the translation is not in the processor TLB). Bits 8 to 20 of the virtual storage address are the page address and are used to select the appropriate four-byte table entry. The addressable bit is inspected. If it is on (page is addressable), the 13 bits in the page frame address field are combined with bits 21 to 31 of the virtual storage address to form a program processor storage address. This translation process requires 1.43 microseconds.

If the addressable bit is off, a page access translation program interruption is generated and instruction execution is suppressed. The control program receives control and must perform the procedure previously described to make the referenced virtual storage page addressable. Once this has been completed, the instruction causing the interruption is reissued when PSWs are switched. Address translation is then performed.

Processor Translation Lookaside Buffer

The processor translation lookaside buffer is also utilized to speed up address translation during ECPS:VSE mode operations. However, it is

used only for the translation of virtual storage addresses in instructions and not for the virtual storage addresses in channel programs. A separate set of channel TLBs is used for channel programs.

When an instruction address must be translated, the TLB is inspected first to determine whether the translation can be taken from the TLB. If so, no translation time is required, as for System/370 mode. If the TLB does not contain the translation, the address translation table is accessed to perform the translation, as previously described. The method of addressing a TLB entry and use of the LRU bits are the same for ECPS:VSE and System/370 modes.

The TLB is managed entirely by microcode in ECPS:VSE mode. When a MAKE ADDRESSABLE instruction is issued, the specified virtual storage page address and its associated processor storage address are placed in the TLB when the page is referenced. Similarly, when a MAKE UNADDRESSABLE instruction is issued, the TLB is inspected to determine whether the address of the specified page is currently in the TLB. If so, the entry containing the page address is invalidated.

The full TLB is automatically purged during system reset. Thereafter, the full TLB is never purged during ECPS:VSE mode operations, since address translation is always operative and both virtual and real entries can be contained in the processor TLB.

As a result of the technique used for address translation in ECPS:VSE mode, the PURGE TLB instruction is not required and is not valid for this mode. The PURGE TLB instruction is utilized in System/370 mode, for example, when multiple virtual storages are being supported and a switch from one virtual storage to another is made or when page replacement occurs and a page table entry is invalidated (since the invalidated entry could be in the TLB).

Channel Program Translation

One channel TLB consisting of four entries for each channel is provided in the 4341 Processor to handle address translation during operation of a channel program for the channel. For each channel, the channel TLB contains one even and one odd entry for translating virtual storage addresses in CCWs and one even and one odd entry for translating data (I/O buffer) addresses in CCWs.

A CCW or data entry contains a 12-bit virtual storage address, an 11-bit real storage address (the address of the assigned page frame), and a validity bit to indicate whether the entry is valid.

A pair of entries is provided for CCW address translation to handle the situation in which the SEARCH and TIC commands in a disk channel program are located in two separate virtual storage pages and have noncontiguous page frames assigned. If only one CCW entry were available, a TLB miss would occur each time the SEARCH and TIC commands were executed during a searching operation. Two data entries are utilized for consistency in the way in which addresses are translated using the channel TLB.

When a START I/O instruction is processed, a READ CCW microinstruction is issued. This microinstruction causes a reference to the channel TLB to translate the virtual storage address of the first CCW in the channel program. This virtual storage address is in the channel address word. The three channel address bits specified in the instruction are used to locate the channel TLB CCW entries for the specified channel. Bit 12 in the virtual storage address of the channel program to be started is used to select the odd or even CCW entry for the channel.

The virtual address in the selected CCW entry is compared to the virtual address in the channel address word. If they are equal and the entry is valid, the real address bits in the CCW entry are combined with the real address bits from the channel program address and the first CCW is fetched from this real address. The CCW is then processed. The channel TLB reference procedure is performed by hardware and does not add any time to I/O processing when the required virtual address is in the channel TLB.

When the required translation is not in the channel TLB (unequal virtual address comparison occurs), bits 8 to 20 in the virtual address in the channel address word are used to select the required entry in the address translation table. The real storage address and the setting of the connected bit are taken from the addressed entry. The assumption is the page containing the channel program is connected to a page frame.

The real storage address from the address translation table and its associated virtual storage address are placed in the appropriate CCW entry in the channel TLB and the setting of the connected bit is inspected. If the bit indicates the page was connected (and thus the real address is valid), the invalid bit is turned off in the CCW entry. If the page was not connected, the invalid bit is turned on in the CCW entry. This translation procedure requires 1.95 microseconds.

After channel TLB updating is completed, the READ CCW microinstruction is reexecuted. The appropriate CCW entry in the channel TLB is again addressed and the virtual storage address in the entry is compared with the address in the channel address word. If the entry is valid, the CCW is fetched and processed. If the entry is invalid, a channel error condition exists.

When the CCW is fetched, a READ DATA or WRITE DATA microinstruction, as appropriate for the CCW, is issued to cause processing of the first 64 bytes to be transferred. The READ/WRITE DATA microinstruction specifies the virtual storage address of the data to be processed. This address is taken from the CCW.

The READ/WRITE DATA microinstruction causes the appropriate odd or even data entry in the channel TLB to be referenced to determine whether the translation for the data address from the CCW is in the channel TLB and is valid. Bit 12 in the virtual storage address determines which one of the two data entries for the specified channel is used.

If the channel TLB does not contain the required translation, the translation procedure described for translating a CCW address, using the address translation table, is executed to translate the address in the CCW and place it in the channel TLB. The READ/WRITE DATA microinstruction is then reexecuted.

The READ/WRITE DATA microinstruction causes the processing of up to 64 bytes of data. If the CCW count is not zero after 64 bytes have been read from or written to processor storage, another READ/WRITE DATA microinstruction with its virtual data address incremented by 64 is issued. This causes another reference to the channel TLB to translate the address. Translations for successive READ/WRITE DATA microinstructions can be taken from the channel TLB at least until the end of the assigned page frame is reached.

When an I/O buffer spans two or more noncontiguous page frames, the first READ/WRITE DATA microinstruction whose virtual data address crosses a 2K boundary will cause an unequal comparison between that virtual address and the one in the appropriate channel TLB entry. This TLB miss causes the translation procedure to be performed and the appropriate channel TLB entry for the channel to be updated. The new

translation can then be used for accessing the next 2K bytes of data, if necessary.

All channel TLB entries are invalidated during processor resets. A channel TLB entry is updated only when a translation is performed during channel program operation. An entire channel TLB is never purged during processor operation. However, when an entry in the address translation table is invalidated, all the odd or even channel TLB entries (as appropriate) are inspected. Any entry containing a virtual storage address that is the same as the one for which invalidation occurred in the translation table is also invalidated.

15:20 SYSTEM PERFORMANCE IN A VIRTUAL STORAGE ENVIRONMENT

A virtual storage environment is designed to provide new data processing capabilities. As is true of any other capability offered by an operating system, support of a new function requires control program use of a certain amount of the hardware resources of the system. In this respect, virtual storage is no different from multiprogramming and the many other new capabilities that were added to DOS and OS after their initial release.

The characteristic that makes virtual storage different from most other features is that virtual storage is not primarily designed to improve system performance, as are many other control program facilities. Virtual storage is first a functional tool and, in certain cases, can also be a performance tool. The objectives of DOS and OS virtual storage operating systems are to (1) provide new functions, (2) maintain upward compatibility with DOS and OS non-virtual-storage environments, and (3) provide performance equal to or better than that achieved with a non-virtual-storage operating system using the same system hardware configuration. Attainment of the last objective may not be possible for all 4341 Processor configurations.

In addition, some of the new functions a virtual storage environment provides cannot be achieved in a non-virtual-storage environment or are not practical. In these cases, performance is not the primary consideration when using the facility virtual storage offers. As the cost of hardware resources continues to decline on a unit cost basis (cost per processor storage bit, cost per direct access bit, etc.), it becomes increasingly more economical to use system resources to perform functions that otherwise are handled by installation personnel.

The other new characteristic of virtual storage is that it enables a given system configuration to provide a wider range of performance, as well as function, as a result of the new factors that affect operation of a system with virtual storage support. Thus, a slightly different approach must be taken in planning for and in evaluating system performance in a virtual storage environment.

Many of the same factors that affect system performance in a DOS/VSE or OS/VS1 environment apply to DOS Version 3 or 4 and OS MFT, respectively. First, the system configuration must include the hardware resources (processor speed, channels, I/O devices, storage) required for the control program and job mix. This subsection identifies the system resources specifically required to support a virtual storage environment. Second, the system should be designed to balance resource usage to achieve optimum throughput, and to use applicable performance and control program design options the particular operating system offers, taking into account the characteristics of the installation jobstream.

The performance of a system in a virtual storage environment is also affected by certain new factors that do not apply to systems without

virtual storage support. This subsection identifies these new factors, explains how they generally affect system performance, and indicates the steps that can be taken to increase and maximize system performance when a virtual storage operating system is used.

This discussion applies to DOS/VSE and OS/VS1, and is restricted to performance factors that are common to the virtual storage environments they support. The virtual storage operating systems also offer new performance-oriented enhancements that are not related to the implementation of virtual storage.

The performance information in this subsection is designed to present concepts and considerations for a virtual storage environment. Figures and graphs are used for illustrative purposes. They do not represent any particular installation or measured results. Their purpose is to illustrate the interrelated factors of multiprogramming performance in a virtual storage environment. The performance information presented is conceptual. It is based on the experience and judgment of IBM individuals with performance knowledge and on performance measurements made during development of OS/VS. Therefore, it may not apply to all installations.

SYSTEM RESOURCES REQUIRED TO SUPPORT A VIRTUAL STORAGE ENVIRONMENT

In order to support a demand paged virtual storage environment in a 4341 Processor, in which programs are operating in paged mode, additional system resources are used by the IBM-supplied virtual storage operating systems, as follows:

- Address translation hardware (dynamic address translation facility or the internal mapping function) requires processor time to perform virtual storage to processor storage address translation. The amount of time required for a 4341 Processor is affected by which address translation hardware is used and the number of times the address translation hardware procedure must be performed. The 4341 Processor has a processor translation lookaside buffer that is designed to minimize use of the address translation hardware. The processor time required for translation is also affected by program structure (which is discussed later).

  A small amount of additional processor time is also required to pretest certain instructions that reference storage, as discussed under "Instruction Nullification" in Section 15:10. Studies have shown that a relatively small percentage of the total processor time specifically required to support a virtual storage environment is devoted to address translation by hardware. In the 4341 Processor, hardware address translation time can be minimized by the use of ECPS:VSE mode, which provides significantly faster translation for virtual storage addresses in instructions than System/370 mode when the required address is not in the processor TLB.

- For System/370 mode operations in the 4341 Processor, processor time is required to translate the virtual storage addresses in channel programs (CCW lists) into real storage addresses, build indirect data address lists (when necessary), and temporarily (short-term) fix pages that will be referenced during I/O initiation, execution, and interruption handling. Channel program translation and page fixing are performed before the initiation of each I/O operation with a channel program that contains virtual storage addresses. Channel status word retranslation and page unfixing are performed at the completion of these I/O operations.

  The amount of processor time this function requires per data set is affected by the number of I/O requests (EXCP macros) issued, the

number of CCWs in the channel programs started, the number of pages that must be fixed, and whether or not indirect data address lists have to be constructed. Studies have shown that a large portion of the total processor time specifically required to support a virtual storage environment is used to perform channel program translation and page fixing.

When the 4341 Processor operates in ECPS:VSE mode, programmed address translation for CCW lists and construction of indirect data address lists are not required. The I/O supervisor must only ensure that each page referenced in a channel program has a page frame assigned and that each page is marked temporarily (short-term) fixed. Page unfixing must be performed at the completion of each I/O operation. The elimination of channel program translation for CCW lists reduces the total processor time required to support a virtual storage environment for ECPS:VSE mode operations.

- Processor time is required to process page translation/access exceptions and for the execution of other control program code that is specifically required to support a virtual storage environment. Processor time is required for such things as servicing additional program interruptions, managing and allocating real and external page storage, maintaining segment and page tables used by DAT hardware (System/370 mode only), and testing for paged or nonpaged mode of program operation. The processor time required for this support during ECPS:VSE mode operations is less than for System/370 mode operations, since the updating of the table used for address translation (address translation table) and available page frame maintenance are handled by hardware rather than the control program.

- I/O time is required for paging operations. The amount of paging I/O time required is related to the number of page faults that occur and the speed of the paging I/O device(s) used.

- Direct access storage is required for external page storage. The amount required depends on the amount of virtual storage that is to be supported and the way in which the particular operating system organizes and manages external page storage.

- The amount of processor storage required by the resident (fixed) control program is increased by the amount of processor storage needed for additional routines and code that are included specifically to support a demand paged virtual storage environment.

The effect this additional use of hardware resources has on the performance of a given system configuration, when a change from a non-virtual-storage to a virtual storage operating system is made without expanding the system configuration, depends on the resource requirements of the jobstream and current utilization of system resources. To the degree that the additional required processor and I/O time can be overlapped with existing processor and I/O time that currently is unoverlapped, system throughput is not affected. System throughput will be affected by the increase in processor and I/O time that cannot be overlapped.

When a virtual storage operating system is used with an existing system configuration (say DOS/VSE replaces DOS Version 3 in a 4341 Processor configuration), for example, and the same jobstream is processed, performance is affected by the use of any new performance enhancements these operating systems provide as well as by an increase in resource utilization that is required to support a virtual storage environment.

NEW FACTORS THAT AFFECT SYSTEM PERFORMANCE

In addition to the factors that affect system performance in a non-virtual-storage environment, the performance of a system in a virtual storage environment is affected by the relationship of the following factors: the speed and number of paging devices, the speed of the processor, the size of real storage, the structure of the programs in the jobstream, and the way in which real storage is organized and allocated by the virtual storage operating system. The interrelationship of each of these factors and their individual effect on performance, except for the last factor listed, are as follows (page replacement algorithms are not discussed).

Speed and number of paging devices. A certain amount of I/O time is required to read in (or write out) a page using a given direct access device type. This time is a function of device type characteristics--seek time, rotation time, and data transfer rate. Assuming one page-in is performed at a time, no page-outs, and no contention for the paging device or its channel/adapter, a maximum paging rate, in terms of the number of page faults that can be serviced per time interval, could be calculated for a given device type. This rate could be improved by certain programming techniques, such as use of rotational position sensing when it is present, and initiation of multiple page-in and page-out requests with a single channel program. The maximum paging capability of a given system can be increased by various means, such as using a faster paging device or using more than one paging device.

The paging characteristic of a virtual storage environment is the feature that permits an operating system to support virtual storage that is larger than real storage. The paging activity of a system begins to adversely affect system performance, however, once the processor is in the position of frequently having to wait for paging I/O operations to complete. When requests for paging operations are permitted to occur faster than the paging rate the system can sustain, so that the processor can do little or no processing except that related to paging, the system is in a paging-I/O-bound situation and is said to be thrashing. When a thrashing condition exists, little or no productive work can be accomplished unless paging activity is reduced.

In order to prevent thrashing, DOS/VSE and OS/VS1 monitor the activity of the system to determine when paging activity becomes excessive. At this point, task deactivation is performed. This involves placing a partition in deactivated status. When the page frames associated with a deactivated partition become available, they can be allocated to other tasks to reduce paging activity. Later, when paging activity becomes sufficiently low, the deactivated partition is reactivated.

Processor speed. An improperly balanced relationship between processor speed and paging device speed can also cause the system to become I/O-bound as a result of paging. A 4341 Processor can execute a certain number of instructions during the time required to service a page-in request using a given direct access device type. As long as there is useful work for the processor to perform while paging operations occur, the system is not kept waiting for paging I/O. However, if the concurrently operating programs are constantly executing instructions faster than the pages they require can be brought into real storage, an excessively high paging rate could develop if task deactivation were not invoked. Therefore, the direct access device type selected for paging operations should be selected for its ability to handle the particular page fault rate of the given configuration.

Real storage size. The amount of real storage present in a processor (that is, program processor storage in a 4341 Processor) affects the number of page faults that occur when a given jobstream is processed.

If the amount of real storage present in the system is equal to the total amount of virtual storage being used by the concurrently executing tasks, no page faults occur for programs that have been fetched and initiated. When the amount of real storage present is less than the amount of virtual storage being used, page faults occur. The total number of page faults that occur for a given jobstream is affected by the ratio of virtual storage used to real storage available.

Assuming the amount of virtual storage used in a given system remains the same, the virtual-to-real storage ratio can vary. This occurs while a given system experiences variations in the amount of real storage actually available for paging as the amount of fixed real storage changes during jobstream processing. The real storage available for paging at any point in time is the difference between the amount of real storage in the system and the total amount of long- and short-term fixed real storage. For IBM-supplied virtual storage operating systems, the total amount of fixed real storage at any given time is the sum of the:

- Resident (fixed) control program size, which does not vary after IPL
- Amount of long-term fixed real storage required for control blocks, which can change as the level of multiprogramming changes (OS/VS1 only)
- Amount of short-term fixed real storage required for outstanding I/O operations that have virtual channel programs, which fluctuates with the I/O activity of the system
- Amount of long-term fixed real storage required by the job steps executing in nonpaged (real) mode, if any
- Amount of long-term fixed real storage required by programs that operate in paged mode but that have a portion of their partition always fixed (VTAM, for example)

As the virtual-to-real storage ratio of a jobstream increases, so usually does the page fault rate. In general, the page fault rate increases slowly for a while. At some point, the increase in page faults begins rising rapidly as the virtual-to-real storage ratio continues to increase. Figure 15.20.1, shown later, illustrates the general relationship between the number of page faults and the virtual-to-real storage ratio.

The amount of real storage available to process a given jobstream also varies when a given jobstream is processed on systems with various amounts of real storage, such as when a smaller-scale system is used to back up a larger-scale system.

The degree to which reducing the real storage available for paging affects the paging fault rate depends on the paging activity pattern of the programs in a jobstream. Therefore, the virtual-to-real storage ratio at the point at which a given number of page faults occurs will usually vary by jobstream. The point can also be different for systems with similar paging activity patterns and the same amount of real storage installed, but with different amounts of long-term fixed real storage.

As the virtual-to-real storage ratio increases, because of a reduction in the real storage available (or an increase in the amount of virtual storage used) and the page fault rate increases, more demand is placed on the paging devices. If too small an amount of real storage is present in a system, this situation can cause the page fault rate to exceed the permissible rate and task deactivation will occur. In general, therefore, in order to obtain a certain level of performance, a configuration that supports a given jobstream and virtual storage size

requires more real storage when a relatively slower paging device is used than if a faster paging device is used.

Program structure. The total amount of virtual storage a program uses is not nearly as significant a factor in system performance as the way in which virtual storage is used. That is, the pattern and frequency of reference to pages in a program have more effect on the number of page faults that occur than does the total size of the program.

For example, assume a case in which a program has a 100K virtual storage design point. If the program can be structured to execute as a series of logical phases of four or five pages each and the pages of each logical phase reference only each other, no more than four or five page frames (8K to 10K or 16K to 20K of real storage, depending on page size) need be dynamically available to the program at one time, and paging activity occurs only as the program progresses from one logical phase to the next.

However, assume the program is structured such that during its execution each page of instructions constantly references a large number of different pages of instructions and data for short durations on a highly random basis. An excessively high paging rate could occur if only four or five page frames were dynamically available to such a program at any time.

As indicated previously, most types of programs naturally have a locality-of-reference characteristic so that they can be structured to operate as a series of logical phases. In the simplest case, for example, a program can logically consist of an initialization phase, a main phase, one or more exception-handling phases, and a termination phase. The total amount of virtual storage referenced in each logical phase usually varies but, generally, the amount is less than the total size of the program. In addition, the pages that are part of (referenced in) a given logical phase can usually be described as active or passive.

For the purpose of the discussion in this subsection, an active page is defined as one with a high probability of being referenced multiple times during execution of the logical phase, while a passive page has a low probability of being referenced more than once during execution of the phase. A logical phase experiences the least amount of paging activity as it executes when its active pages remain in real storage during its execution and its passive pages are paged in when required. A program uses real storage most efficiently when the active instructions and data in each logical phase are contained within the fewest number of pages possible.

The locality-of-reference characteristic does not apply to certain types of programs. For example, it does not apply to any program that is designed to optimize its performance at execution time by using the total amount of storage it has been allocated. This characteristic is usually true of sort/merge programs that initialize themselves to use all the storage made available to them in their partition during the sorting passes. The reference pattern for such a sort/merge is random and encompasses all the storage (and, therefore, all the pages) the program is assigned.

RELATIONSHIP BETWEEN VIRTUAL STORAGE SIZE AND SYSTEM PERFORMANCE

Assuming other required system resources are available, a given configuration can support a given virtual storage size and provide satisfactory performance when paging activity is kept at an acceptable level. Minimal paging activity occurs when enough real storage is

present in the system to contain most or all of those pages of concurrently executing programs that are active at any given time. Paging activity is then required primarily for passive pages. Active pages are paged in (and later paged out as required) as the set of active pages for each program changes from one logical phase to another. The paging device(s) present must be capable of handling the demand for pages that results from the range of paging activity of the system.

As the amount of virtual storage used in a given system increases, the number of active and passive pages that the system must handle increases also. The ratio of active to passive pages will vary for a given increase in virtual storage, depending on how the additional virtual storage is used. As long as enough real storage is present to contain all or most of the increased number of active pages, the increase in paging activity required to support the additional virtual storage will be needed primarily for passive pages and should be relatively small. As soon as the use of more virtual storage causes the number of concurrently active pages to constantly exceed the capacity of real storage, the paging activity increase required to support the additional virtual storage becomes relatively large. As more and more active pages must be handled, paging activity could exceed the maximum paging capability of the system if task deactivation did not occur.

Figure 15.20.1 illustrates the increase in page faults that generally occurs as more virtual storage is used in a given system configuration. The curve begins at the point at which the amount of virtual storage used is equal to the amount of real storage present (virtual-to-real storage ratio is 1). Paging activity begins as soon as the amount of virtual storage used exceeds the real storage present. As the virtual-to-real storage ratio increases, so does paging activity. The system moves from passive paging activity (primarily paging of passive pages) into active paging (paging active pages in and out more of the time) and approaches the maximum paging capability of the system. As indicated previously, Figure 15.20.1 also illustrates the increase in page faults that generally occurs as less real storage is made available to support a given virtual storage size. The increase in page faults also causes the virtual-to-real storage ratio to increase.

Figure 15.20.2 illustrates the general effect on system performance of the paging factor only. Figure 15.20.4, shown later, illustrates system performance, taking into account all factors. The curve shows the performance of the system when passive and active paging are occurring, relative to the virtual-to-real storage ratio. The use of virtual storage can be increased with little or no adverse effect on performance as long as paging remains in the passive area. This is true because in the passive paging area there is a relatively small amount of paging and a high probability that all or most paging processing (processor and I/O time) can be overlapped with other processing. As paging activity increases, there is a higher probability that processor processing will be held up waiting for a paging operation to complete. As the processor enters the wait state more frequently to wait for paging I/O and less paging I/O is overlapped, the paging factor causes performance to degrade more rapidly.

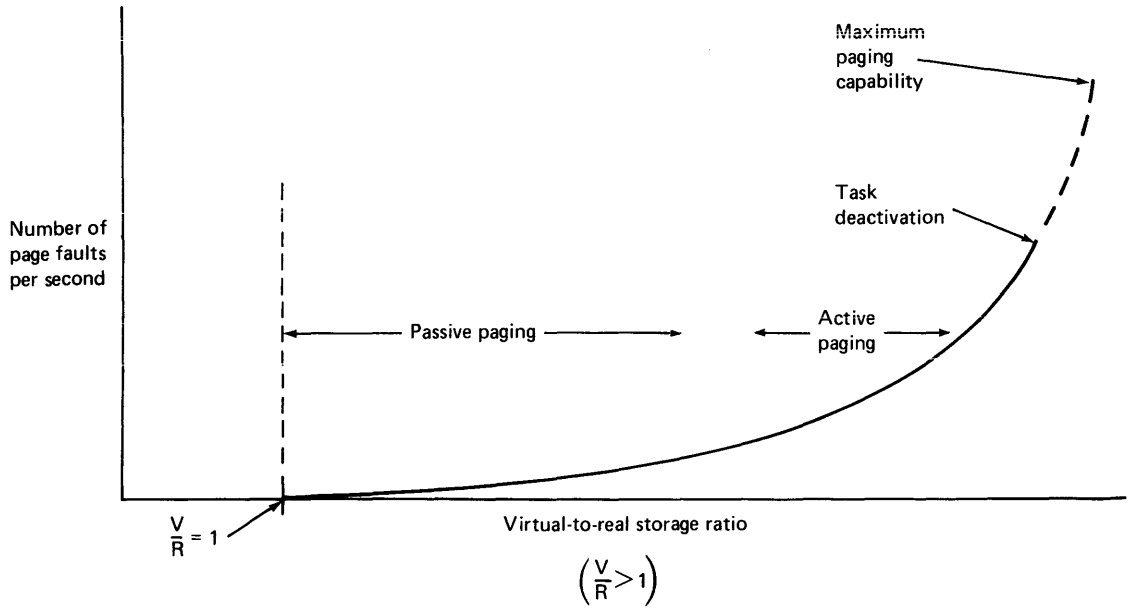


Figure 15.20.1. General effect on page faults of increasing the ratio of the virtual storage used to the real storage present in the system

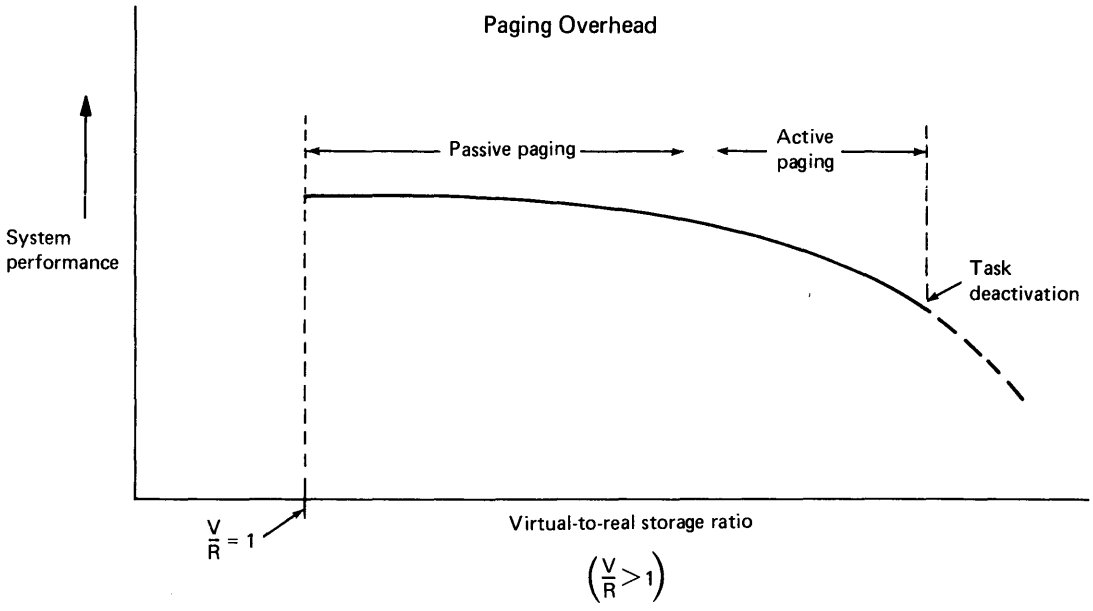


Figure 15.20.2. General effect on system performance of the paging factor only

The actual virtual-to-real storage ratio at the time active paging begins in Figures 15.20.1 and 15.20.2 is a variable and depends on the

way in which virtual storage is used, that is, active-to-passive page ratio of concurrently executing tasks.

Figure 15.20.3 illustrates the way in which the paging factor alone can affect system performance in a given configuration, based on the active-to-passive page ratio. If the ratio of active to passive pages for executing tasks is relatively high most of the time, as shown in curve 1, the virtual-to-real storage ratio at the point at which active paging begins will be relatively low. Performance drops very rapidly in this case as more virtual storage is used. This happens because the increased paging processing (I/O and processor time) cannot be overlapped with other processing. This situation may apply initially to an installation when a switch from a non-virtual-storage to a virtual storage environment is made and more virtual storage is used, since existing programs were structured for optimum performance in a given partition size rather than for optimum performance in a virtual storage environment.

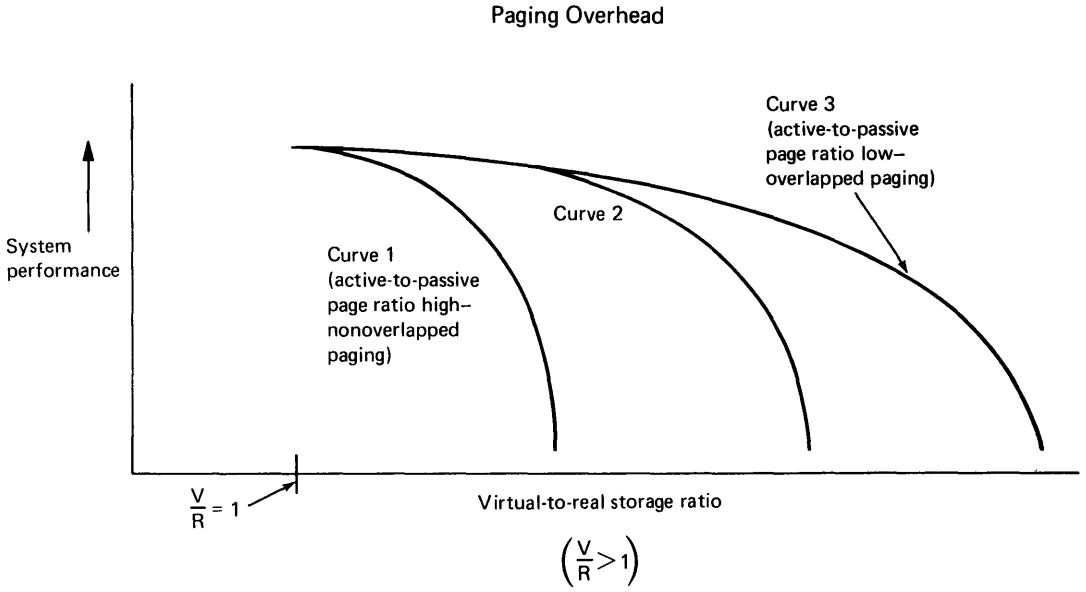


Figure 15.20.3. General effect of the paging factor on system performance for various active-to-passive page ratios

If the active-to-passive page ratio for the system is low, as shown in curve 3, the virtual-to-real storage ratio can be relatively high when active paging begins. The performance curve stays flatter longer as virtual storage is increased when the active-to-passive page ratio is low. This situation can apply to an installation in which all executing programs are structured in such a way that real storage requirements and page faults are minimized. An installation that continues executing all or most existing programs as they are presently designed and that structures new applications for most efficient operation (low active-to-passive ratio) may be more common. Such installations may experience a virtual-to-real storage ratio somewhere between the low and the high extremes possible for a given jobstream, as shown in curve 2.

The amount of virtual storage used in a system can be increased in several ways. First, the size of existing application programs can be increased by the addition of new functions. Second, the level of multiprogramming and/or multitasking can be increased, assuming other required resources, such as processor time and I/O devices, are available. Third, the size of existing application programs can be

expanded by (1) restructuring programs with a planned overlay or a dynamic structure to take them out of these structures and (2) combining two or more job steps within a job into one logical job step. The active-to-passive ratio of the additional pages the system must handle will usually be higher when the level of multiprogramming is increased than when existing jobs are restructured.

The way in which an installation should view the amount of virtual storage used and system performance for a given configuration, taking all performance factors into account, is illustrated in Figure 15.20.4. The increased use of virtual storage is beneficial to system performance up to a point. Thereafter, additional virtual storage can be used to handle additional functions at a variable cost in system performance. In reality, the virtual-to-real storage ratio and the page fault rate vary during system processing as the amount of virtual storage used (out of the total amount supported) and the amount of real storage available for paging vary. Best overall system performance is achieved when paging activity falls most of the time in the area identified on the curve as the operating range. More significant performance reduction begins when active paging is experienced.

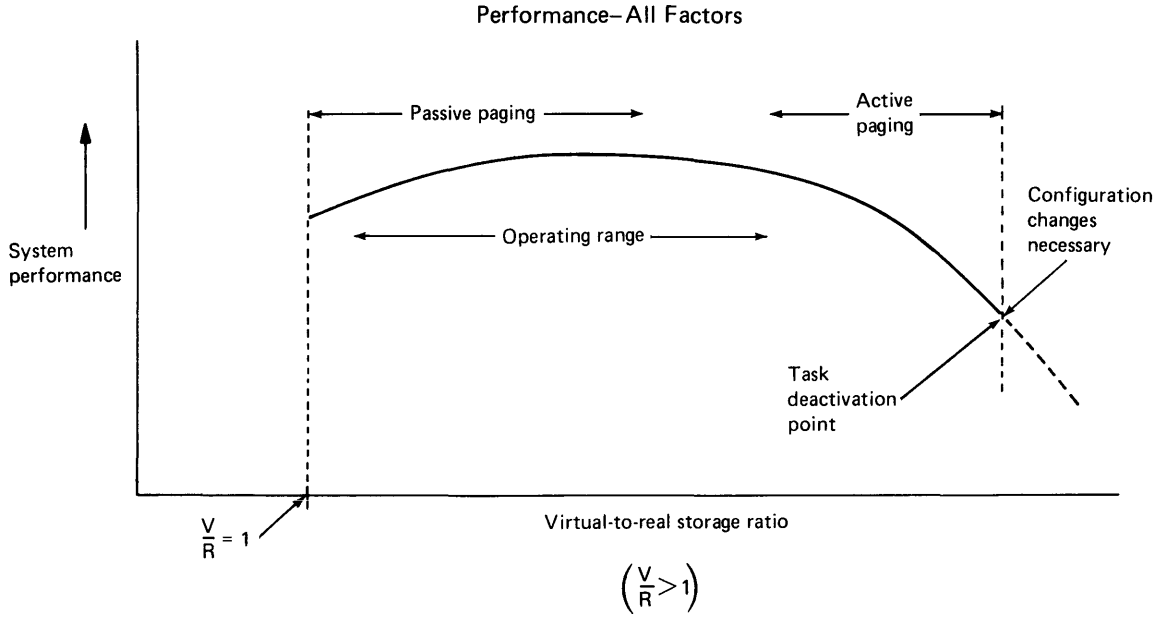


Figure 15.20.4. General system performance curve for a virtual storage environment

Occasional active paging on an exception basis should be acceptable. More frequent active paging can be performed to achieve a desired function that does not justify changing the system configuration. However, when paging activity in a system is constantly at the point at which task deactivation occurs, system configuration changes should be made to improve system performance. Such changes might be the addition of more real storage, the addition of more (in OS/VS1 environments) or faster paging devices, or installation of a faster processor. A history of the paging activity of the system can be maintained by recording the paging statistics provided by the virtual storage operating systems. OS/VS1 provides page-in and page-out statistics, while DOS/VSE provides a page fault trace capability.

INCREASING SYSTEM PERFORMANCE IN A VIRTUAL STORAGE ENVIRONMENT

The IBM-supplied virtual storage operating systems are designed to provide an acceptable level of performance when existing problem programs are run without modification. However, given the additional resource requirements of virtual storage support and the new factors that affect system performance in a virtual storage environment, once a virtual storage operating system is installed (either on an existing configuration or a larger configuration), certain steps can be taken to improve performance or to achieve optimum performance. The benefit to be achieved by taking any one of the steps outlined must be evaluated on an installation basis, taking the specific configuration and operating environment into account. Some steps, for example, are more practical for large configurations than for small configurations. The following can be done:

- Use larger I/O buffers. This step is practical primarily for sequential data sets and can be used most effectively when previous real storage limitations prevented the use of larger buffer sizes in general and optimum buffer sizes for disk data sets. In addition to reducing the total I/O time required to process a data set, as would occur in a non-virtual-storage environment, increasing buffer size reduces the number of I/O requests required to process the data set and, thereby, reduces the processor time required for channel program translation and page fixing.

  This technique should be used taking into account the amount of real storage present in the system. If the buffer size of several data sets that are being processed concurrently is increased considerably or made large initially, the amount of real storage that must be short-term fixed increases considerably also and potentially increases the number of active pages. This may adversely affect system performance in systems with a relatively limited amount of real storage available for paging.

- Increase the page-fault-handling capability of the system when paging activity constantly causes task deactivation. This can be accomplished by (1) using a direct access device for paging that is faster than the currently used paging device, (2) allocating more direct access devices for paging to enable more overlap of paging activity, or (3) reducing or eliminating contention for the existing paging device(s). Contention for the paging device can be relieved by using dedicated paging devices or reducing the amount of other I/O activity on the channel/adapter to which the paging device is attached. Alternatively, the same paging device configuration can be maintained while page fault occurrence is decreased by the addition of real storage.

- Use code that does not modify itself. Use of this type of code can reduce the amount of page-out activity required. Such code can be produced using the Assembler Language and PL/I language translators.

- Execute programs in nonpaged (real) mode only when actually required. Use of nonpaged mode should be limited because the amount of real storage available for paging operations during the operation of a nonpaged program is reduced by the size of the program and can affect system performance. If a nonpageable program is to be present in a system for an extended period of time or at all times, it should be considered part of the fixed real storage requirement so that the amount of real storage actually available for paging can be more accurately determined.

- Structure new application programs to operate efficiently in a paging environment. This is done by structuring programs to achieve a reasonable balance between page faults and real storage

requirements. The extent to which this is done can vary widely by installation. The benefits that can be obtained should be evaluated in light of the additional programmer effort required. In this respect, deciding on the degree to which programs should be structured for efficient operation in a paging environment is similar to deciding how a high-level language should be used. The emphasis can be on most efficient program execution, which can require more programmer effort, or on most efficient use of programmer time, which can result in less efficient programs. Alternatively, there can be a tradeoff between programmer time and efficient programs (only the most frequently or heavily used programs are optimized, for example).

Many of the general program structure techniques discussed do not require a large amount of additional effort or knowledge on the part of programmers--only that they adopt a particular programming style. All of the suggested techniques can be used by Assembler Language programmers. Some can be used with certain high-level languages and not with others. More of the suggested techniques can be used in PL/I programs than in other high-level language programs.

Two major steps can be taken to structure programs to use real storage most efficiently and to incur the smallest possible number of page faults. The first is to adopt a certain programming style, one aspect of which is similar to the style that has been encouraged with System/360 and System/370, namely, that of modular programming. The second is to take page boundaries into account and to package program code and data into page groups.

The objective of improving programming style is to construct a program that consists of a series of logical processing phases, each of which contains a relatively small number of active pages. The objective of packaging code within pages is to group active code together to avoid crossing page boundaries in such a way that more real storage than is really necessary is required to contain the active pages of a logical phase.

In order to cause references to active instructions and data to be localized, the following general rules should be applied to programs:

1. A program should consist of a series of sequentially executed logical phases or--in System/370 and 4300 Processor programming terminology--a series of subroutines or subprograms. The mainline of the program should contain the most frequently used subroutines in the sequence of most probable use, so that processing proceeds sequentially, with calls being made to the infrequently used subroutines, such as exception and error routines. This structure contrasts with one in which the mainline consists of a series of calls to subroutines. Frequently used subroutines should be located near each other. Infrequently used subroutines that tend to be used at the same time whenever they are executed should be located near each other also.

2. The data most frequently used by a subroutine should be defined together so that it is placed within the same page, or group of pages, instead of scattered among several pages. If possible, the data should be placed next to the subroutine so that part or all of the data is contained within a page that contains active subroutine instructions (unless the routine is to be written so that it is not modified during its execution). This eliminates references to more pages than are actually required to contain the data and tends to keep the pages with frequently referenced data in real storage.

3. Data that is to be used by several subroutines of a program (either in series or in parallel by concurrently executing subtasks) should be defined together in an area that can be referenced by each subroutine.

4. A data field should be initialized as close as possible to the time it will be used to avoid a page-out and a page-in between initialization and first use of the data field.

5. Structures of data, such as arrays, should be defined in virtual storage in the sequence in which they will be referenced, or referenced by the program in the sequence in which a high-level language stores them (by row or by column for arrays, for example).

6. Subroutines should be packaged within pages when possible. For example, avoid starting a 1500-byte subroutine in the middle of a 2K page so that it crosses a page boundary and requires two page frames instead of one when it is active. Subroutines that are smaller than page size should be packaged together to require the fewest number of pages, with frequently used subroutines placed in the same page when possible. This applies to large groups of data as well.

   The linkage editor supplied with OS/VS1 has new control statements that can be used to cause CSECTs and COMMON areas to be aligned on page boundaries, and to indicate the order in which CSECTs are placed in the load module. This linkage editor facility can be used with certain high-level language programs that contain multiple CSECTs (such as PL/I and ANS COBOL) as well as with Assembler Language programs.

- Use the PL/I Optimizing Compiler available for DOS/VSE and OS/VS1 instead of OS PL/I F or DOS PL/I D. The code produced by these language translators has characteristics that makes it more suited to a virtual storage environment than the code produced by the Type I PL/I language translators. First, generated code is grouped into functionally related segments, by PROCEDURE and DO group, for example, which can help reduce paging. When PL/I allocates buffers and I/O control blocks, they are packed together and potentially can require fewer pages than if no attempt was made to define them together. Reentrant code can be produced by the OS PL/I Optimizing Compiler, and its library routines are reentrant. This reduces page-out requirements. User-written reentrant OS PL/I routines that are required by concurrently executing problem programs can be made resident in virtual storage and shared to reduce real storage and paging requirements for active pages of these routines.

- Use the shared library feature of the OS Optimizing Compiler and the COBOL Library Management Facility of the OS ANS COBOL language translators to make library modules resident in virtual storage so they can be shared by concurrently executing problem programs. Pages containing active library modules will tend to remain in real storage and thereby reduce paging and real storage requirements for these modules.

- Restructure existing application programs to incur as few page faults as possible, to use the least amount of real storage, and to take advantage of the program structure facilities that a virtual storage environment offers. This can be accomplished by (1) using the techniques described above, (2) taking planned overlay and dynamic structure programs out of these structures, and (3) combining into one logical job step two or more steps of a job that would have been one job step if the required real storage were available. The last of these techniques can eliminate redundant I/O

time that is currently used for such things as reading the same sequential input data into two or more job steps and writing intermediate results from one job step in one or more sequential data sets for input to the next job step.

- Increase the level of multiprogramming in the system. This can be accomplished by (1) performing more peripheral I/O operations concurrently (more readers and writers in OS/VS1, use of VSE/POWER in DOS/VSE), (2) operating more partitions concurrently, or (3) increasing the use of multitasking (structuring a VSE/VTAM message processing program to use multitasking to enable several different types of transactions to be processed concurrently, for example).

  System throughput can be improved in a virtual storage environment if a higher level of multiprogramming causes more processor and I/O time to be overlapped and results in more effective utilization of system resources. The larger the number of tasks in the system under these conditions, the less chance there is for the processor to enter the wait state because no task is ready to execute. Better utilization of real storage in a virtual storage environment can enable more tasks to be present in the system.

  In order to achieve performance gains by increasing the level of multiprogramming, the potential for more overlap of processor and I/O time must exist in a system, and/or the potential must exist for a reduction of I/O time via increased overlapping of channel operations and reductions in unoverlapped seek time (that can result from new system performance enhancements). The required hardware resources, such as processor time, real storage, I/O devices, and direct access storage, must be available as well. The most critical resource in this situation is available processor time. As the percentage of processor utilization gets higher, there is less potential for increasing throughput via increasing the level of multiprogramming.

The information presented in this subsection is designed to enable the reader to more fully understand the way a system operates in a virtual storage environment and the factors that influence system performance. Understanding the environment and knowing the actions that can be taken to increase system performance will enable each installation to quantify the amount of effort that is to be devoted to optimizing the performance of a virtual storage operating system.

SECTION 18: VIRTUAL MACHINES

This section discusses the basic concepts, general operation, and advantages of virtual machines, as defined and implemented in Virtual Machine Facility/370. No previous knowledge of virtual machines is assumed.

The virtual machine concept is a logical extension of the virtual storage concept and requires support of multiple virtual storages. Therefore, VM/370 can execute only with System/370 mode in effect in 4300 Processors. Comprehension of dynamic address translation hardware and virtual storage concepts, terminology, and advantages, as discussed in Sections 15:05 and 15:10, is assumed.

VM/370 consists of the Control Program (CP), Conversational Monitor System (CMS), Remote Spooling Communications Subsystem (RSCS), and Interactive Problem Control System (IPCS) components. CP supports the concurrent operation of multiple virtual machines. CMS, operating in a virtual machine under CP control, provides conversational time-sharing facilities to a single user. RSCS, operating in a virtual machine under CP control, provides for the transmission of data between remote users and virtual machines via binary synchronous communications lines. IPCS, operating in a CMS virtual machine, provides interactive problem management, problem determination, and problem isolation.

VM/370 is the successor to CP-67/CMS. Virtual machine support was first provided by IBM in CP/67. In the CMS time-sharing environment in which CP-67/CMS was primarily used, the major advantage of the virtual machine facility was that it enabled each CMS user to appear to have a complete System/360 (Model 22 to 75) at his disposal and to be isolated from all other CMS users. Each CMS user had access only to his own virtual machine and, therefore, could not inadvertently interfere with the operation of other CMS virtual machines. VM/370 also provides these facilities and can be used in nondedicated time-sharing environments to provide other advantages as well.

18:05 DEFINITION AND GENERAL OPERATION

A virtual machine is a functional simulation of a complete computer system, including a virtual processor, virtual storage, virtual channels, virtual I/O devices, and a virtual operator's console, that appears to the user to be a real machine. In a VM/370 environment, a virtual machine is the functional equivalent of a 4300 Processor or System/370 processor (Models 135 to 168 and 3033, 3032, and 3031 Processors) and its associated I/O devices.

The control program (CP) component of VM/370, executing in a real machine (4300 Processor, System/370 Models 135 through 168 with dynamic address translation hardware, and 3033, 3032, and 3031 Processors), supports concurrent operation of multiple virtual machines using multiprogramming techniques that enable real machine resources to be shared by multiple virtual machines. Each virtual machine is dedicated to a single user and isolated from other virtual machines. None of the components of one virtual machine can be accessed by a program that is executing in another virtual machine except via the controlled sharing facilities that are provided by CP.

The operation of a virtual machine and scheduling of the work it performs are handled by an operating system rather than by CP. That is, each virtual machine has an operating system executing in it that allocates machine resources and schedules the execution of problem

programs just as if the operating system were executing in a real machine.

In order to initiate operations in a virtual machine, the user must log on the virtual machine and IPL an operating system in it. The logon procedure establishes a connection with CP and the existence of a specific virtual machine for this user. A logon is performed using a console or terminal device of the type that CP supports as a virtual operator's console.

The virtual operator's console is the means by which the user controls the operation of his virtual machine and communicates with the operating system executing in it. CP provides a set of commands that (1) simulate the system control panel or operator console of the virtual machine, (2) provide for alteration of a virtual machine configuration, (3) request various services from CP for a virtual machine, and (4) control operation of the real machine. When a CP command is entered via the virtual operator's console, CP receives control and performs the required functions.

Communication between the user and the operating system is accomplished using the operating system command language and the virtual operator's console. CP performs any simulation required to make the real I/O device the operator is using as a virtual operator's console appear to be the primary console device type that is defined for the operating system.

In a VM/370 environment, a virtual operator's console is frequently called a remote terminal because, in most cases, a terminal device type is actually used as the virtual operator's console device. However, the real I/O device that is used as the virtual operator's console can be a console device for the specific processor as well as a local or remote terminal.

VM/370 supports execution of any one of the following System/360 and System/370 operating systems in a virtual machine:

- CMS component of VM/370
- RSCS component of VM/370
- DOS Version 3, DOS Version 4, DOS/VS, or DOS/VSE (DOS/VSE operating in a virtual machine must be generated to support System/370 mode when VM/370 is executing in a 4341 Processor)
- APL 360-DOS
- OS PCP, MFT, or MVT
- OS ASP Version 3
- OS/VS1
- OS/VS2 SVS (Releases 1, 1.6, and 1.7)
- OS/VS2 MVS (Releases 2 and up) operating in uniprocessor mode only
- PS44
- VM/370

Any number and combination of the above operating systems can execute concurrently in a VM/370 environment, subject to the availability of the required real machine resources, including multiple copies of the same operating system (DOS/VSE executing in more than one virtual machine,

for example). With a few exceptions, all the facilities that are supported by these operating systems when they execute in a real machine can be used when the operating system executes in a virtual machine in a VM/370 environment. Figure 18.05.1 conceptually illustrates the real and virtual machine environment that is supported by VM/370.

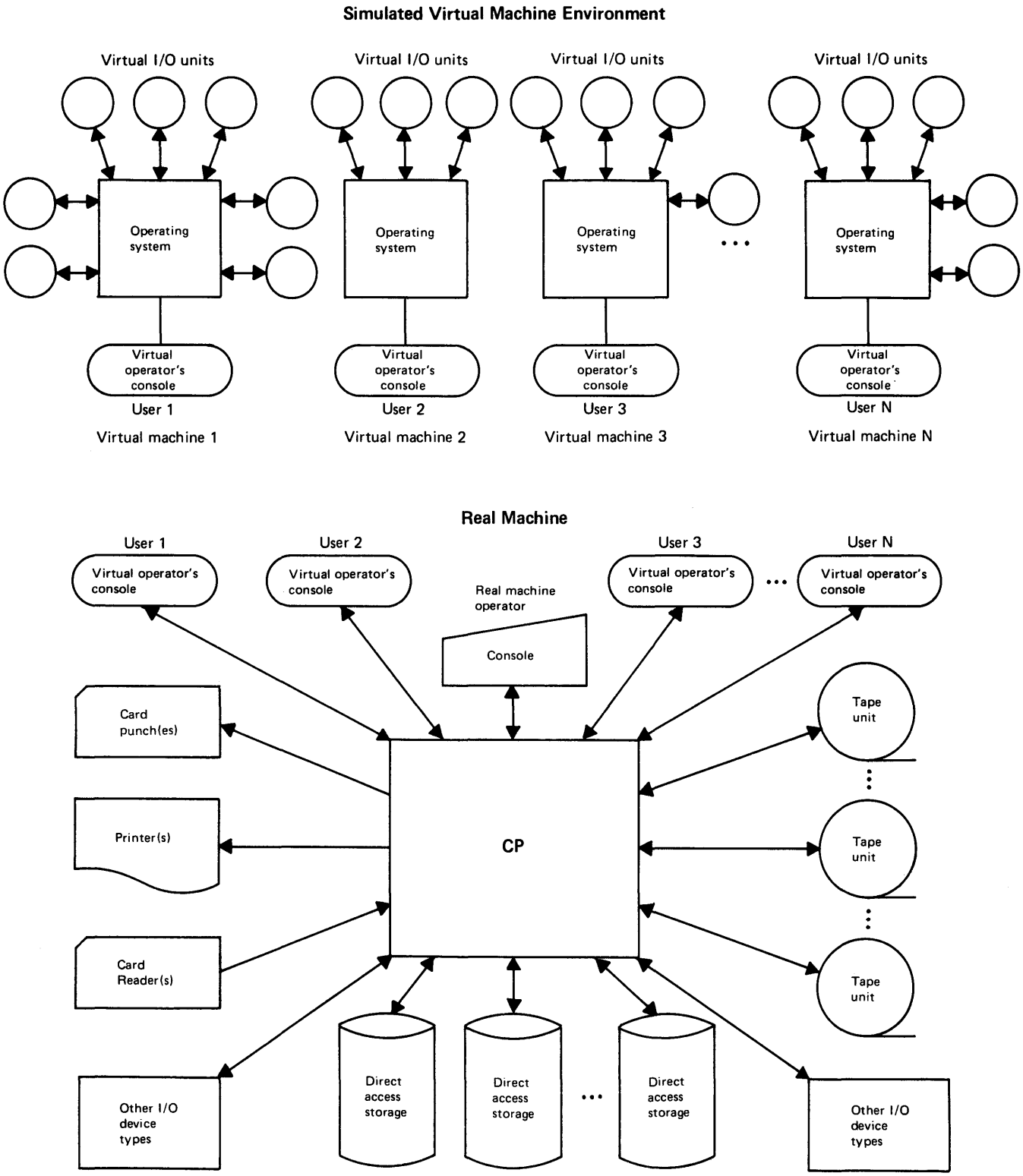


Figure 18.05.1. Conceptual illustration of the real and virtual machine environment that is supported by VM/370

Each virtual machine that is to be supported by CP must be user-defined and the definition stored in a VM/370 directory. The size of virtual storage, virtual I/O devices to be used, options to be used, and a virtual console are usually specified. Virtual machine configurations can be different from each other and, within certain limitations, different from that of the real machine in terms of these specifications.

Virtual Instruction Processing Function Simulation

CP is resident in processor storage during operation of the real machine. It controls the operation of the real machine, schedules the execution of virtual machines, and simulates virtual machine hardware components using the hardware components of the real machine. In order to be able to perform its functions and isolate virtual machines from each other, CP must have exclusive control over the status and modes of operation of the real machine, as does the control program of an operating system. Hence, CP always executes with the real machine in supervisor state and receives control after all real machine interruptions.

Virtual machines always operate with the real machine in problem state. Therefore, any time any program that is executing in a virtual machine issues a privileged instruction, an interruption occurs in the real machine. CP receives real machine control and takes the required action. This may involve simulating execution of the privileged instruction for the virtual machine or returning real machine control to the control program in the virtual machine for which the interruption occurred so that the interruption can be processed by that control program. In this manner, CP maintains control of the real machine. In addition, CP simulates the existence of both a supervisor state and a problem state in the virtual machine while, in reality, the virtual machine operates only in problem state.

CP gives control of the real machine to operating virtual machines on a time-shared basis to simulate the existence of multiple processors. A virtual machine can execute any instruction for the 4341 Processor except SET CLOCK, which is treated as an NOP because CP controls the setting of the time-of-day clock, and the instructions that are valid only when ECPS:VSE mode is in effect. In addition, the DIAGNOSE instruction is reserved for communication between executing operating systems and CP.

The processor features that are used by the control and problem programs executing in a virtual machine must be present in the real machine in which CP executes. CP does not simulate the existence of processor hardware features that are not present in the real machine. A virtual machine can appear to be executing in System/370 mode with either BC mode or EC and DAT modes specified, depending on the mode required by the operating system executing in it. However, EC and DAT modes are always specified in the real processor when a virtual machine is executing since dynamic address translation hardware is required to support the existence of virtual storage for the virtual machine. A virtual machine cannot appear to be executing in ECPS:VSE mode.

Virtual Storage Simulation

The implementation of virtual storage in a virtual machine environment is conceptually illustrated in Figure 18.05.2. Each virtual machine can have up to 16,777,216 bytes of virtual storage, which is the virtual storage size for the 4341 Processor. The existence of virtual storage for a virtual machine is simulated by CP using DAT hardware and

external page storage, as for a virtual storage environment that is supported in System/370 mode (discussed in Section 15).

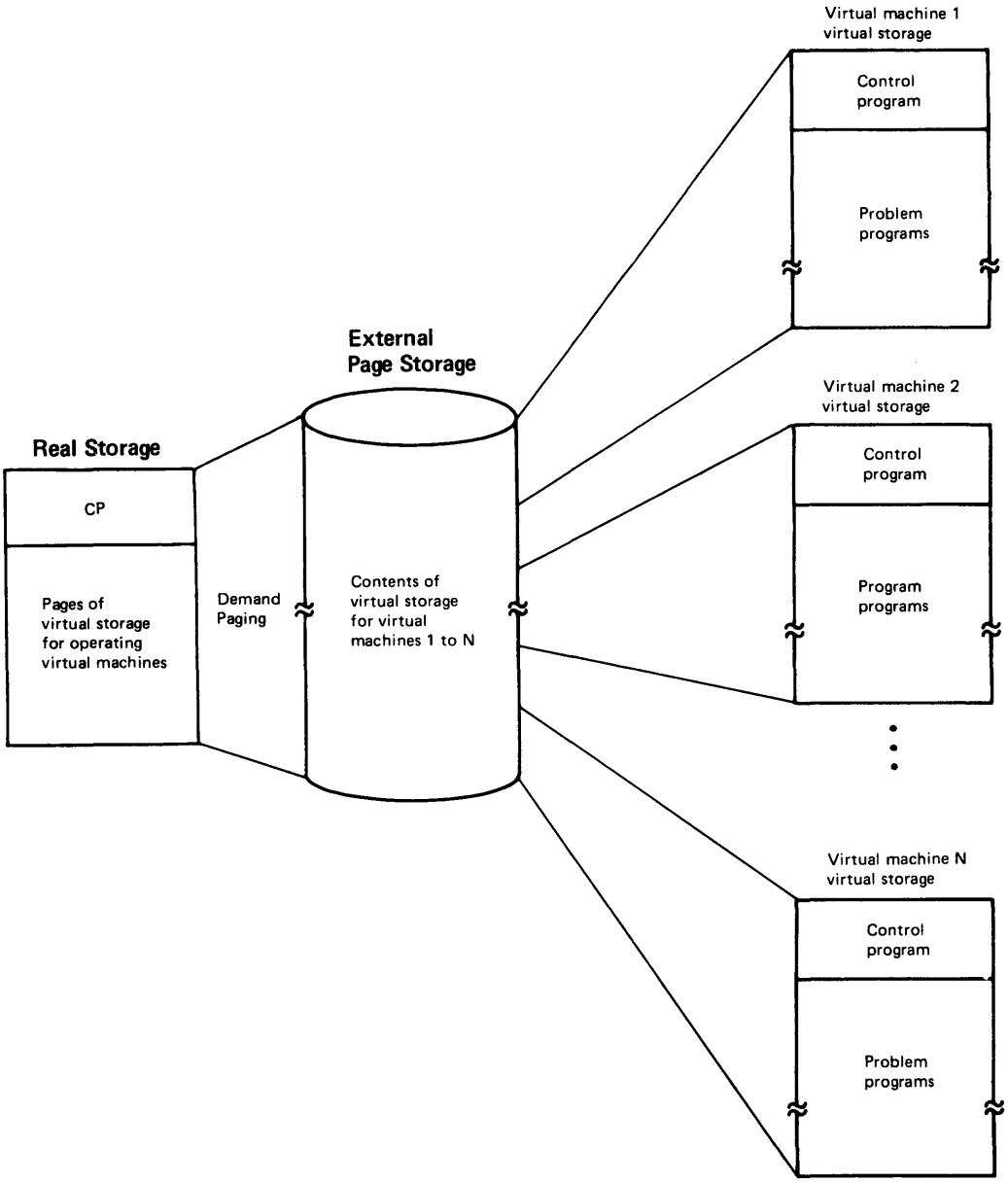


Figure 18.05.2. Conceptual illustration of the implementation of virtual storage in a virtual machine environment

Operating system programs that are executing in a virtual machine (both control and problem programs) are paged in and out of processor storage in the real machine on a demand paged basis as they execute. Processor storage allocation, external page storage allocation, and paging operations are handled entirely by CP and are transparent to the control and problem programs that are executing in the virtual machines. In this manner, CP provides one virtual storage for each virtual machine, and processor storage in the real machine is shared by concurrently operating virtual machines.

The virtual storage defined for a virtual machine always appears to be real storage to the operating system that is executing in the virtual machine. In effect, an operating system that does not support virtual storage, such as DOS (Version 3 or 4) or OS MFT and MVT, has virtual storage support provided by CP when such an operating system executes in a virtual machine and, therefore, offers the functional advantages of a virtual storage operating system.

When executing in a virtual machine, an operating system that does support virtual storage uses the virtual storage defined for the virtual machine as real storage in order to simulate the existence of the virtual storage it is designed to support. As shown in Figure 18.05.3, the virtual storage operating system builds a segment table and page tables to translate addresses in the virtual storage it supports to addresses in the virtual storage defined for the virtual machine, which the operating system assumes is real storage. CP always builds and maintains a segment table and page tables for each virtual machine. These tables are used to translate addresses in the virtual storage of the virtual machine to addresses in real storage in the real machine.

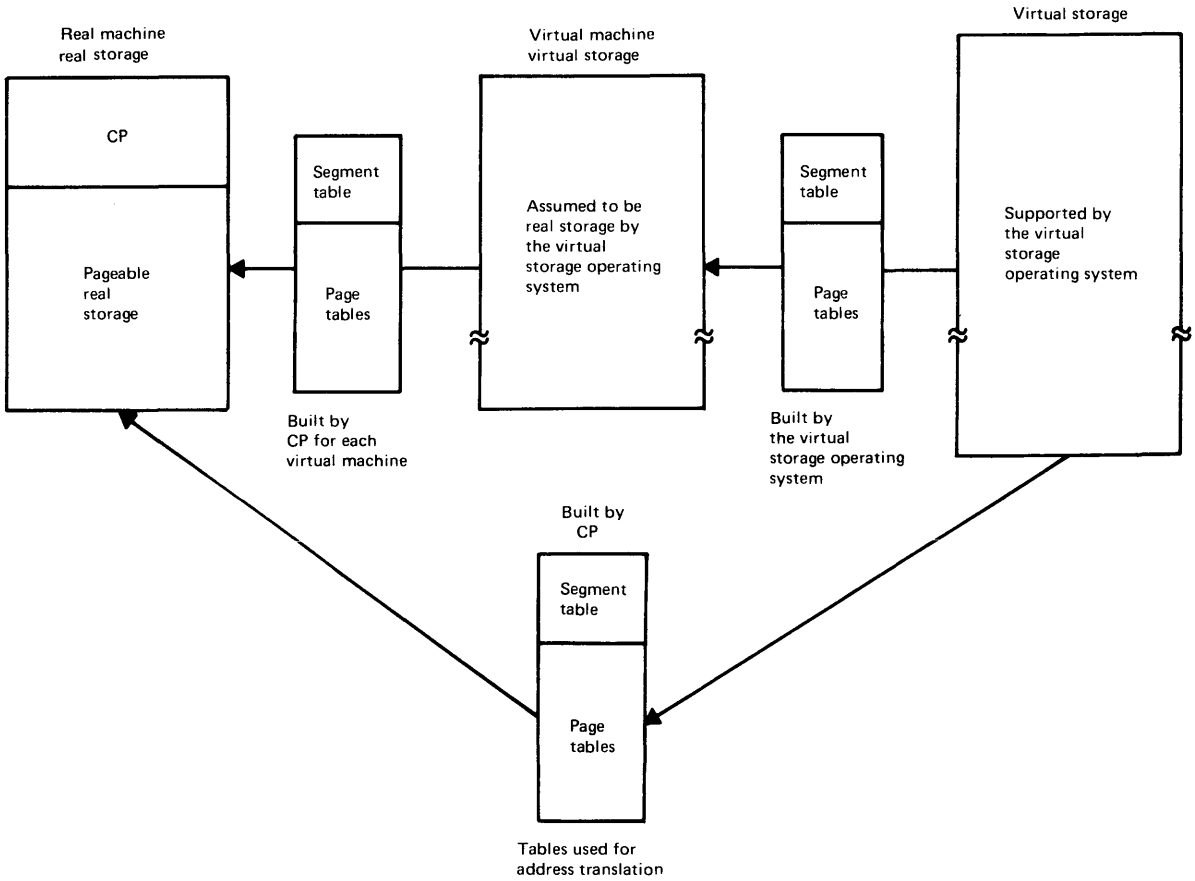


Figure 18.05.3. Segment tables and page tables built when a virtual storage operating system executes in a virtual machine

When a virtual storage operating system is executing in a virtual machine, CP constructs and maintains a third set of tables, using the contents of the other two sets of tables. The third set of tables, a shadow segment table and shadow page tables, are the tables that are actually used for address translation when the virtual machine operates.

The shadow tables are used to translate addresses in the virtual storage the operating system supports to addresses in real storage.

Virtual I/O Component Simulation

The virtual channels, control units, and I/O devices defined in each virtual machine configuration are simulated by CP using real channels, control units, and I/O devices that are of the same type. While each virtual I/O device defined must have a real I/O device counterpart in the real machine configuration, there does not necessarily have to be a one-to-one correspondence. In addition, the I/O device addresses assigned to virtual I/O devices need not be the same as the addresses of their real I/O device counterparts.

CP also allows a virtual direct access device to be simulated by only a portion of a real direct access device volume. Such a virtual direct access device is called a minidisk. Support of a minidisk facility enables one real direct access device to simulate the existence of several virtual direct access devices of the same type and thus provides more efficient use of available direct access storage.

Virtual I/O devices are always simulated on a real I/O device of the same device type unless the spooling facility of CP is used. (CP also allows 2311 disk storage to be simulated using 2314/2319 disk storage and the minidisk facility.) The local spooling capability of CP provides data transcription between unit record devices and direct access storage devices and is functionally similar to DOS and DOS/VS POWER, DOS/VS POWER/VS, VSE/POWER, OS readers and writers, OS HASP, and OS/VS JES. In effect, the CP spooling facility enables virtual unit record devices (card readers, card punches, and printers) to be simulated using direct access storage. CP also provides console spooling and a remote spooling facility.

The virtual I/O devices in a virtual machine configuration are logically controlled by the operating system that is executing in the virtual machine rather than by CP. That is, all the data management routines of the operating system (physical record processing, logical record processing, and error recovery routines) execute as usual. Therefore, a virtual machine I/O configuration can include any I/O device types that are supported by the operating systems that will execute in the virtual machine, as long as real I/O device counterparts exist in the real machine I/O configuration as required.

CP controls only the scheduling and actual initiation of virtual machine I/O operations in the real machine. When a START I/O instruction is issued by an operating system control program that is executing in a virtual machine, a privileged operation exception interruption occurs and CP receives real processor control. CP translates the virtual I/O device address to its counterpart real I/O device address and, for minidisks, converts virtual cylinder addresses to corresponding real cylinder addresses, as required. CP also performs the necessary channel program translation and page-locking operations and queues the I/O request if it cannot be started.

After the I/O operation is started, CP returns the condition code to the operating system control program that initiated the I/O request so that appropriate action can be taken. When the I/O operation completes and causes an I/O interruption, CP receives processor control, gathers I/O status information, and attempts to restart the available real I/O components. CP presents the status data to the operating system control program via a simulated I/O interruption for the virtual machine in which the operating system is executing.

CP completely controls operation of the real I/O devices that are required for its own execution, such as paging and spooling devices. This includes determining the need for I/O operations, scheduling and initiating I/O requests, handling I/O interruption processing, and performing error recovery procedures.

ECPS:VM/370

General Description

ECPS:VM/370, a portion of the standard Extended Control Program Support feature for the 4341 Processor, is designed to improve VM/370 performance by providing for the execution of certain CP routines and functions in hardware to reduce the amount of processor time used by CP. A reduction of up to 84 percent of processor busy time for the VM/370 CP has been measured when compared to the same version of VM/370 running without ECPS:VM/370 enabled.

The hardware assist is provided for the following functional areas:

- Virtual machine and control block dispatching
- I/O handling and CCW translation
- Virtual interval timer simulation
- Privileged instruction simulation
- SVC handling
- Page management functions
- Storage management functions

ECPS:VM/370 as implemented in the 4341 Processor is like the VM/370 hardware assist function for System/370 Models 135 Model 3, 138, 145 Model 3, and 148. Thus, ECPS:VM/370 for the 4341 Processor includes as one of its components the Virtual Machine Assist feature that is available for the 3031 Processor and Models 135 (Model 0), 145 (Models 0 and 2), and 158 (and as an RPQ for Models 165 II and 168 and the 3032 and 3033 Processors).

ECPS:VM/370 consists of the Virtual Machine Assist, Control Program Assist, Expanded Virtual Machine Assist, and Virtual Interval Timer Assist components. The Virtual Machine Assist and Expanded Virtual Machine Assist components consist of microcode routines that perform the function of certain frequently used virtual machine instruction simulation routines of CP. The Control Program Assist component consists of microcode routines that perform certain frequently used general CP routines and functions that are required to support a virtual machine environment. The Virtual Interval Timer Assist component simulates a virtual interval timer for a virtual machine.

Either ECPS:VM/370 or ECPS:VS1 can be activated in a 4341 Processor during IML when System/370 mode is in effect. However, the actual operation of the four components of ECPS:VM/370 is enabled and disabled using three bits (0, 6, and 7) in control register 6. The Virtual Machine Assist component can be enabled separately from the other components. Other bits in control register 6 enable and disable specific functions within the Virtual Machine and Expanded Virtual Machine Assist components.

Shown below are the settings for control register 6 bits 0, 6, and 7 and the assist components they enable.

Control Register 6			Assist Component Enabled
Bit 0	Bit 6	Bit 7	
0	0	0	None. (This is the processor reset setting.)
0	1	0	Control Program Assist only
1	0	0	Virtual Machine Assist only
1	0	1	Virtual Machine Assist and Virtual Interval Timer Assist
1	1	0	Control Program, Virtual Machine, and Expanded Virtual Machine Assists
1	1	1	Control Program, Virtual Machine, Expanded Virtual Machine, and Virtual Interval Timer Assists

Virtual Machine Assist Component

The Virtual Machine Assist component is entered when one of the following occurs:

- A privileged instruction program exception occurs that is caused when a virtual machine issues an INSERT PSW KEY, INSERT STORAGE KEY, LOAD PSW*, LOAD REAL ADDRESS, RESET REFERENCE BIT, SET PSW KEY FROM ADDRESS, SET STORAGE KEY, SET SYSTEM MASK*, STORE CONTROL, STORE THEN AND SYSTEM MASK*, or STORE THEN OR SYSTEM MASK* instruction. For most of these instructions, the Virtual Machine Assist component simulates execution of the privileged instruction, and operation of the virtual machine continues with execution of the instruction after the privileged instruction. CP code is not entered.

  For the instructions identified by an asterisk, certain conditions prevent simulation of the instruction by the Virtual Machine Assist component. In these cases, control is passed to the Expanded Virtual Machine Assist component (without entry into CP coding) and simulation of the instruction is attempted again.

  Bit 3 in control register 6 determines whether all or only some of the instructions listed above are handled by the Virtual Machine Assist component. When bit 3 is a zero, all the instructions are handled. This setting would be used when an operating system that operates in EC mode, such as DOS/VS, DOS/VSE, OS/VS, or VM/370, is executing in a virtual machine. When bit 3 is a one, only instructions that are valid for System/360 and System/370 as well as 4300 Processors (LPSW, ISK, SSK, and SSM) are handled. This setting would be used when an operating system that operates in BC mode, such as OS MFT or MVT or DOS Version 3 or 4, is executing in the virtual machine.

- An SVC instruction, except SVC 76, is issued by a virtual machine. PSW switching for the virtual machine is simulated by the Virtual Machine Assist component. Handling of SVC instructions other than SVC 76 by the Virtual Machine Assist component can be disabled by turning on bit 4 in control register 6.

- A page translation program exception occurs in a virtual machine in which a virtual storage operating system is executing. The Virtual Machine Assist component updates the appropriate shadow page table, if possible. Handling of page translation exceptions by the Virtual Machine Assist component must be enabled by turning on bit 5 in control register 6.

Expanded Virtual Machine Assist Component

The Expanded Virtual Machine Assist component is designed to perform some or all of the CP simulation required for the following privileged instructions when they are issued by a virtual machine:

- LPSW - Load PSW
- SCKC - Set Clock Comparator
- SIO - Start I/O and Start I/O Fast Release
- SPT - Set CPU Timer
- SSM - Set System Mask
- STNSM - Store Then And System Mask
- STOSM - Store Then Or System Mask
- PTLB - Purge Translation Lookaside Buffer
- STPT - Store CPU Timer
- TCH - Test Channel

When an LPSW, SSM, STNSM, or STOSM instruction is issued by a virtual machine, the Virtual Machine Assist component is entered first to determine whether it can simulate the instruction. When the Virtual Machine Assist component can perform the simulation, the Expanded Virtual Machine Assist component is not invoked. When certain conditions exist, the Virtual Machine Assist component cannot perform the simulation and control is passed to the Expanded Virtual Machine Assist component, which makes further tests to determine whether it can simulate the instruction. The Expanded Virtual Machine Assist component handles certain conditions that are not handled by the Virtual Machine Assist component. If the Expanded Virtual Machine Assist component cannot perform the simulation, the appropriate CP simulation routine is entered via an interruption (a privileged operation exception occurs).

For PTLB, STPT, and TCH instructions, the Expanded Virtual Machine Assist component completely simulates the instruction and no entry into CP is made. For all the other instructions listed above, simulation by the Expanded Virtual Machine Assist component is only partial and a certain amount of CP code is also executed to perform the simulation.

Note that while the Control Program Assist and the Virtual Machine Assist components execute independently from the other components of ECPS:VM/370, the Expanded Virtual Machine Assist component uses portions of the Control Program Assist component in its execution.

Control Program Assist Component

Privileged instructions are defined for the CP functions and routines supported by the Control Program Assist component. These new instructions are designed to be used only in the VM/370 programming system and are not provided for general use in installations. Assembler Language mnemonics are not provided for these new instructions. These Control Program Assist instructions are automatically included in the CP of a VM/370 system that is generated using a Release 3 PLC 8 or later starter system.

When one of the Control Program Assist instructions is executed during VM/370 operation, the required routine or function is performed by the Control Program Assist function instead of by the appropriate CP

routine if the Control Program Assist function is enabled and supervisor state is in effect.

A Control Program Assist instruction is executed as an NOP instruction if the Control Program Assist function is present in a system but disabled (bit 6 in control register 6 is a zero) and supervisor state is in effect. This enables a VM/370 system that includes ECPS:VM/370 support to execute correctly on a system that has a VM/370 assist feature when the feature is not enabled (because it is malfunctioning, for example). If a Control Program Assist instruction is issued in a processor that does not have the Control Program Assist component, an operation exception program interruption condition occurs.

All the new instructions defined for the Control Program Assist function are six bytes in length and have a storage-to-storage instruction format and the same operand code to identify them as Control Program Assist instructions. An extended operand code byte in each instruction uniquely identifies each individual instruction.

The following are the Control Program Assist instructions and the CP function they perform:

- FREE - Obtain space from the free storage area
- FRET - Return space to the free storage area
- PTRLK - Lock a page
- PTRUL - Unlock a page
- DESCCW - Decode subsequent CCW commands in a list
- UNTFR - Free CCW storage
- SCNVU - Locate virtual I/O control blocks
- DSPBLOK0 - Dispatch a control block or a virtual machine
- TRANBRNG - Determine whether a page is present in real storage and, if not, load it
- TRANLOCK - Lock or load and lock a page in real storage
- ZAPSEGS - Invalidate a segment table
- ZAPPAGE - Invalidate a page table
- DEFRCCW - Decode the first CCW command in a list
- DSPCH - Main entry into the virtual machine dispatch function
- SCNRU - Locate real I/O control blocks
- CCWGENRL - Common CCW command processing
- UNTRN - Untranslate a CSW
- DSPBLOK1 - Dispatch a control block or a virtual machine without testing for the need to call the scheduler to process scheduling and dispatching queues
- STECPSVM - Store VM/370 assist level identification. This instruction stores a value to indicate the architecture level of the VM/370 assist. It is used by CP during IPL to determine whether the ECPS:VM/370 or VM/370 hardware assist function is installed in the processor and, if so, the level of the support.

In addition to the above, the SVC instructions with codes X'08' and X'0C' specified that are issued by CP are defined as the LINK and RETURN Control Program Assist instructions, respectively. When the Control Program Assist function is enabled, these two SVC instructions are also executed by the function.

Virtual Interval Timer Assist Component

The Virtual Interval Timer Assist component is active only when the Virtual Machine Assist component is also enabled (bits 0 and 7 in control register 6 are both one). However, this component operates independently from the other three components (does not reference any other component during its operation).

The Virtual Interval Timer Assist component maintains a virtual interval timer for the currently operating virtual machine in location 80 of page 0 of the virtual machine in a manner that is similar to the way the real interval timer in location 80 of the real machine is maintained. This component also presents virtual interval timer interruptions to virtual machines and the real machine. Virtual machines operating in BC or EC mode are supported.

When the Virtual Interval Timer Assist component is enabled and a virtual machine is executing (indicated by problem state set in the current PSW in the real machine), bit 23 in the virtual interval timer for the virtual machine is also decremented when the real interval timer is decremented. If page 0 of the virtual machine is not currently resident in real storage, the interval timer word is maintained in the virtual machine control block (VMBLOK) for the virtual machine until such time as page 0 is loaded. When the real machine is in the wait state, the virtual interval timer may or may not be decremented and, if it is, may or may not be decremented at the same rate as the real interval timer.

When a virtual interval timer being maintained by the Virtual Interval Timer Assist component decrements from a positive to a negative value, a virtual interval timer interruption condition is generated. The assist component attempts to present the interruption to the virtual machine. The interruption is presented if the virtual machine is enabled for interval timer interruptions, page 0 of the virtual machine is resident in real storage, the virtual external new PSW for the virtual machine has a valid format, and no illegal state changes for the virtual machine would occur if the interruption were presented (current virtual PSW and external new PSW for the virtual machine are compared).

When an interval timer interruption cannot be presented to the virtual machine, a virtual interval timer interruption is presented to the real machine if the real machine is enabled for interval timer interruptions. A unique external interruption code is presented to differentiate between virtual interval timer and real interval timer interruptions.

18:10 GENERAL ADVANTAGES OF A VIRTUAL MACHINE ENVIRONMENT

The advantages of VM/370 complement those of virtual storage operating systems. Like a virtual storage environment, a virtual machine environment is designed primarily to support new functions rather than increase system performance.

The new functions provided by virtual machines can (1) increase the rate of new application development and (2) expand operational capabilities over those provided by virtual storage. The CMS component of VM/370 supplements these two major advantage areas of a virtual machine environment by supporting time-sharing facilities such as online program development, conversational program execution and problem solving, and interactive text processing.

The following indicates the way in which the virtual machine environment that is supported by the CP component of VM/370 aids the installation of new applications and identifies the new operational features such an environment supports.

Increasing New Application Development

Since virtual machine support includes support of a virtual storage environment for each virtual machine, all the capabilities virtual storage provides that aid new application development are present in a virtual machine environment as well. (These capabilities are discussed at the end of Section 15:05.) By enabling multiple operating systems to execute concurrently in one real machine, the virtual machine environment supported by CP also provides the following new capabilities:

- Testing of new programs can be more extensive and completed sooner through the elimination of dedicated testing periods. While a virtual storage environment can eliminate most program testing restrictions that result from processor storage size limitations, the isolation that is provided by executing a program in a virtual machine eliminates the need to test programs that can cause total system termination in a dedicated environment.

  For example, system-oriented routines written by system programmers and teleprocessing programs, which usually are tested only during scheduled dedicated testing periods, can be tested while production work is in progress. This can eliminate the need to establish testing periods during second or third shift and, by reducing individual test turnaround time, enables more of this type of testing to be accomplished in a given time period.

- Testing of new programs can be completed sooner through the use of console debugging, when necessary. Using the CP commands that simulate system control panel functions, the programmer can use any console debugging facility that is available on a real machine, such as setting address stops, examining and altering general registers, displaying and altering virtual storage, etc., without interfering with production work. CP also provides other debugging services, such as an extensive set of traces, that can be invoked by CP commands.

  Console debugging, which can enable difficult-to-locate program errors to be detected more quickly than with desk debugging, is usually not permitted in a nonvirtual machine environment, except as a last resort, or is scheduled for nonproduction periods. Program testing turnaround time can be significantly reduced through the use of console debugging.

- Transition from one release of an operating system to another release or from one operating system to another can be accomplished more quickly because of the capability of executing multiple operating systems concurrently. A new release of an operating system can be generated and tested in one virtual machine while production work continues in another virtual machine using the existing release. Existing application programs and system-oriented programs that must be modified or newly written (to use a new facility or new language translator, for example) can be tested during production processing as well.

  The multiple virtual machine facility also enables an installation to execute programs that are dependent on a back release (because the release is user-modified, for example) concurrently with each new release of that operating system or with an entirely new operating system (such as a back release of a DOS/VS version operating concurrently with OS/VS1).

- CMS can be used to perform online program development concurrently with the processing of production work using DOS, DOS/VS, or DOS/VSE. Significant gains in programmer output can be realized

through writing, compiling, and testing programs using an online terminal in a conversational manner. This enables new applications to become operational sooner. When CMS is used, each programmer has his own virtual machine with CMS executing in it. Therefore, the occurrence of a programming or operational error in one virtual machine can cause termination of that virtual machine only. Other programmers and production work are not affected.

Expanded Operational Capabilities

In addition to the new operational facilities a virtual storage environment provides (discussed in Section 15:05), a multiple virtual machine environment offers the following capabilities:

- Operating system maintenance can be performed concurrently with production work. PTFs can be applied and tested using one virtual machine without causing the abnormal termination of another virtual machine that is processing production work.

- Operator training can be done using a virtual machine, which eliminates the need to dedicate the entire real machine to this function. Multiple operators can be trained while production work is in process without terminating real system operations through an operator error.

- A system can be backed up by another system that not only has less processor storage but that has real I/O devices with different addresses, fewer direct access devices, and fewer channels, as long as sufficient I/O devices of the required type are available.

- New channel and direct access device configurations can be simulated using a virtual machine for the purpose of evaluating the load on the new I/O configuration before it is installed on the real machine.

As the above indicates, a virtual machine environment, as supported by VM/370, offers several unique capabilities that can be of benefit to small, intermediate, and large system users. In most cases, VM/370 can be used to best advantage as complementary programming system support in installations in which a version of DOS, DOS/VS, DOS/VSE, OS/VS, or OS is used as the primary programming system.

SECTION 20: I/O DEVICES

20:05 I/O DEVICE SUPPORT

Most I/O devices that attach to System/360 and System/370 can be attached to the 4341 Processor. This subsection discusses the operation and advantages of certain disk, printer, and tape devices. The following I/O devices are discussed in this section:

- 3330-series disk storage
- 3350 Direct Access Storage
- 3340 Direct Access Storage Facility
- 3344 Direct Access Storage
- The 2305 Fixed Head Storage Facility Model 2
- The 3203 Model 5 Printer
- The 3803/3420 Magnetic Tape Subsystem

The 3330-series, 3340/3344, 3350, and 2305 represent significant advances in direct access device technology. They provide larger online data capacity and faster data rates and access than direct access devices for System/360, as well as expanded error correction features. All have rotational position sensing and multiple requesting capabilities. The 3340, 3344, and 3350 offer advances in reliability as a result of their design.

The 3203 Model 5 Printer offers several advantages over the 1403-N1 Printer. The 3803/3420 Magnetic Tape Subsystem incorporates advances in tape speed, density, design, and technology. It offers new features and enhanced reliability, availability, and serviceability to 2400-series magnetic tape users.

The major new characteristics of these I/O devices are discussed in the following subsections.

20:10 3330-SERIES DISK STORAGE

3330 DISK STORAGE AND 3333 DISK STORAGE AND CONTROL

The 3330-series is large-capacity, high-performance direct access storage that consists of 3330 Disk Storage Models 1, 2, and 11 and 3333 Disk Storage and Control Models 1 and 11. A 3330-series string can consist of from one to four modules (from one to eight drives), and connects to a block multiplexer channel in the 4341 Processor via 3830 Storage Control Model 1 or 2 and 3880 Storage Control. The removable 3336 Model 1 or 11 Disk Pack is used for data storage. Track overflow, rotational position sensing, and multiple requesting are standard features of 3830 Storage Control.

A 3330-series string can consist of 3333 Disk Storage and Control and/or 3330 Disk Storage modules. A 3333 Model 1 or 11 Disk Storage and Control module consists of two independent drives, device-oriented control functions, and power for itself and the drives that can be attached to it. It must be the first unit in a 3330-series string attached to a block multiplexer channel via 3830 Storage Control Model 2 or 3880 Storage Control. The 3333 does not attach to 3830 Storage Control Model 1. Models 1 and 11 of 3330 Disk Storage consist of two drives and do not contain the device-oriented control functions that are part of a 3333 module. The 3330 Disk Storage Model 2 module contains one drive and no control functions.

A 3330 module connects directly to 3830 Storage Control Model 1 and to 3333 modules. Up to four 3330 modules, in any combination of Models 1 and 2, can be attached to the 3830 Model 1. Up to three 3330 modules, in any combination of Models 1, 2, and 11, can be attached to a 3333 Model 1 or 11 module.

With one exception, Model 11 3330-series drives are functionally like Model 1 and 2 drives. The drives in 3330 and 3333 Model 11 modules have a standard write format release feature that is not provided for 3330 Model 1 and 2 and 3333 Model 1 drives. This feature enables a Model 11 drive to disconnect from a 3333 Model 11 or 3830 Model 2/3880 while the drive is erasing to the end of the track after a record has been written with a formatting write command. This facility frees the control unit and channel for the initiation of another I/O operation.

The 3336 Model 1 Disk Pack is used with 3330 Model 1 and 2 and 3333 Model 1 drives. It has ten recording disks that contain 19 recording surfaces. The bottommost surface is the servo surface. It contains information for the servo system that is used to control seek operations, positioning of the read/write heads over tracks, data clocking, index generation, and signal generation required by the RPS feature. The required servo information is prerecorded on the servo surface of each 3336 Model 1 Disk Pack at the plant of manufacture and is read by a servo head at the bottom of the access mechanism. The servo system is designed to provide fast, precise access mechanism positioning.

The 3336 Model 1 Disk Pack has a maximum capacity of 100 million bytes when full-track records are used. A 3330-series string can consist of a maximum of eight drives for a total of 800 million bytes per string when only Model 1 drives are used. The removable 3336 Model 1 Disk Pack is interchangeable across all 3330 Model 1 and 2 and 3333 Model 1 drives. Model 1 3336 Disk Packs are not interchangeable with the 2316 Disk Packs used on 2314 disk drives.

The removable 3336 Model 11 Disk Pack is used with 3333 and 3330 Model 11 drives. Like a 3336 Model 1, a 3336 Model 11 has 19 recording surfaces. However, the Model 11 disk pack has 808 data cylinders, instead of 404, for a maximum capacity of 200 million bytes. The Model 11 disk pack also has seven alternate cylinders, like a Model 1. Hence, the maximum capacity of a 3330-series string of all Model 11 drives is 1600 million bytes.

Model 11 3336 Disk Packs are interchangeable across all 3330 Model 11 and 3333 Model 11 drives but cannot be used with Model 1 and 2 3330-series drives. The 3336 Model 11 Disk Pack has a physical interlock so that it cannot be mounted on a 3330 Model 1 or 2 drive or a 3333 Model 1 drive. The 3336 Model 1 Disk Pack has a physical interlock so that it cannot be mounted on a Model 11 drive. The 3336 Model 1 Disk Pack can be converted to a Model 11.

Self-formatting records are written on 3336 packs the same as on 2316 packs. However, each physical area written (count, key, and data) has a field of error correction code appended to it for data validity checking by the 3830 instead of the cyclic check area used on the 2314. The correction code used detects single-error bursts of 22 bits or less and corrects single-error bursts of 11 bits or less.

Table 20.10.1 compares Model 1, 2, and 11 drive characteristics. Table 20.10.2 compares 3336 Model 1 and 11 and 2316 Disk Pack characteristics.

Table 20.10.1. Capacity and timing characteristics of 3330-series drives

Characteristic	3330-series Model 1 or 2 drive	3330-series Model 11 drive
Capacity in thousands of bytes (full-track records)	100,018	200,036
Seek time (ms)		
Maximum	55	55
Average	30	30
Average cylinder-to-cylinder	10	10
Time channel busy searching when SET SECTOR is used (ms)		
Minimum	.120	.120
Maximum	.380	.380
Rotation time (ms)	16.7	16.7
Rotation speed (rpm)	3600	3600
Data transfer rate (KB/sec)	806	806

Table 20.10.2 3336 Model 1 and 11 and 2316 Disk Pack characteristics

Characteristic	3336 Model 1	3336 Model 11	2316
Number of disks per pack	12	12	13
Number of recording disks	10	10	11
Number of recording surfaces	19	19	20
Disk diameter in centimeters (inches)	35.6 (14)	35.6 (14)	35.6 (14)
Disk pack weight in kilograms (pounds)	9(20)	9(20)	6.8(15)
Disk pack maximum capacity in millions of bytes	100	200	29.1
Full-track capacity in bytes	13,030	13,030	7294
Cylinders per pack	404 plus 7 alternates	808 plus 7 alternates	200 plus 3 alternates
Tracks per cylinder	19	19	20
Tracks per pack	7676	15,352	4000

Each drive in a 3330 or 3333 module is mounted in a powered drawer that is opened and closed by a switch on the operator panel contained in each module. Each panel provides the switches and indicators associated with individual drives in the module. Included on each operator panel are write-inhibit switches, one for each drive. A switch can be individually set to permit both reading and writing or reading only on its associated drive.

When an attempt is made to write on a drive that is set for read operations only, an interruption occurs and IBM-supplied programming support terminates the program that attempted to write on the protected drive. Each operator panel also contains a removable logical address plug, similar to that of the 2314, for each drive in the module. In addition, a CE service plug is supplied that is to be used when IBM-supplied diagnostic programs are executed.

Functionally, the 3330-series provides more capabilities than the 2314, particularly in the areas of performance, availability, and configurability. The 3330-series supports all the standard 2314 commands (except the file scan commands) in addition to several new operations, including RPS and diagnostic commands. (Table 20.10.4, at the end of this subsection, compares 3330-series and 2314 features.)

The 3330-series also is an attractive growth device for the 2321 Data Cell Drive. Table 20.10.3 compares the capacity and timing characteristics of the 3330-series with those of the 2314 and the 2321 Data Cell Drive. The increase in capacity achieved by replacing a 2314 or a 2321 with a 3330-series string depends upon the block size chosen for the data on the 3330.

Table 20.10.3. Capacity and timing characteristics of 3330-series and 2314 disk storage and the 2321 Data Cell Drive

Characteristic	3330-series (Model 11 drives only)*	3330-series (Model 1 and 2 drive) combinations	2314	2321
Capacity in thousands of bytes (full-track records)				
Pack or cell	200,036	100,018	29,176	39,200
String or Data Cell Drive				
2 drives/cells	400,072	200,036	58,352	78,400
3 drives/cells	-	300,054	87,528	117,600
4 drives/cells	800,144	400,073	116,704	156,800
5 drives/cells	-	500,091	145,880	196,000
6 drives/cells	1,200,216	600,109	175,056	235,200
7 drives/cells	-	700,127	204,232	274,400
8 drives/cells	1,600,288	800,146	233,408	313,600
10 cells	-	-	-	392,000
Access time (ms)				
Maximum	55	55	130	600 (for strip select and load)
Average	30	30	60	175 (minimum for strip select and load)
Average cylinder-to-cylinder	10	10	25	95 (on a strip)
Time channel busy searching when SET SECTOR is used (ms)				
Minimum	.120	.120	-	-
Maximum	.380	.320	-	-
Rotation time (ms)	16.7	16.7	25	50 (strip on drum)
Rotation speed (rpm)	3600	3600	2400	1200
Data transfer rate (KB/sec)	806	806	312	55

*Permissible combinations of Model 11 drives and Model 1 and 2 drives are not listed.

For example, if the 2314 full-track block size of 7294 bytes is maintained for a given data set on the 3330 to avoid programming changes, the 3330 yields a 91% increase in full-pack capacity (almost twice the capacity) when a 3336 Model 1 is used. However, reblocking to a full track on the 3330, 13,030 bytes, yields a 242% full-pack capacity increase using a 3336 Model 1. If enough processor storage is not available to allocate I/O areas of 13,030 bytes, lowering the 3330 block

size to one-half of a 3330 track yields a 239% increase in full 3336 Model 1 pack capacity.

If a 2321 is replaced by a 3330-series string, six full-track blocks of data from the 2321 (2000 bytes/2321 track) can be placed on each 3330 track, if full-track blocking is used, for a total of 92,112,000 bytes per 3336 Model 1 pack (12,000 bytes times 7676 tracks). Thus, slightly over four 3336 Model 1 packs provide the capacity equivalent of ten data cells, or a full 2321 drive, if full tracks are used. Ten full data cells, blocked full track, also can be contained in slightly more than four 3336 packs if half-track blocking is used on the 3336 Model 1.

3830 STORAGE CONTROL

Model 1

The 3830 Storage Control Model 1 unit contains the power and the control functions required to operate one 3330-series string consisting of from one to eight drives in one-drive increments. Only 3330 Disk Storage Models 1 and 2 can be attached to 3830 Storage Control Model 1. When multiple requesting is used, the 3830 Model 1 can control concurrent operation of eight channel programs, one on each drive. Only one of the eight drives can be transferring data at any given time.

The 3830 Model 1 control unit is microcode controlled. Read/write monolithic storage contained in the control unit is used for microcode residence. The control unit also contains a diskette device that reads interchangeable disk cartridges. This device is used for microcode backup storage and for storage of nonresident diagnostics for the 3330-series string. During a 3830 power-on sequence, the functional microcode is loaded from the device into control storage within the control unit. Therefore, microcode engineering changes can be installed merely by replacing the current disk cartridge with another cartridge that contains the new microcode.

The Two-Channel Switch feature, identical to the same feature for the 2314 facility, can be installed on a 3830 Model 1 control unit to allow the 3830 to be attached to two channels. The Two-Channel Switch, Additional feature can be added to this configuration to permit the 3830 to be attached to four channels. A maximum of two of the channels can be in the same processor.

The channels to which a 3830 Model 1 with one or both of these features are connected must each have one control unit position and, if block multiplexing is to be used, eight nonshared subchannels available. The enable/disable switch can be set to dedicate the 3830 Model 1 to any subset of the four channels. The string switching capability that is available for the strings attached to the 3830 Model 2 is not available for the string attached to the 3830 Model 1.

The 3830 Model 1 also incorporates error detection, correction, and logging features, designed to improve its availability and serviceability. The following features implemented in the 3830 Model 1 are not provided for System/360 direct access devices:

- I/O error routine correction of recoverable data errors on read operations with data supplied by the control unit in sense bytes
- Command retry initiated by the control unit to attempt hardware correction of certain errors without programming assistance
- Error logging by the control unit in its control storage of successful command retry operations

- Inline diagnostic tests contained on disk cartridges, which can be run on a single drive to diagnose hardware malfunctions while other drives in the string continue normal operations. (Inline diagnostics are provided only for 2314 facilities.)

Recovery of correctable data area errors. When the control unit detects a correctable data error during the reading of the data portion of a record, it generates the information necessary to correct the erroneous bytes. The sense bytes presented by the control unit contain a pattern of corrective bits and a displacement value to indicate which of the bytes transferred to processor storage contain the errors. The disk error recovery program need only EXCLUSIVE OR (logical operation) the corrective bit pattern with the error bytes in the input area in processor storage to correct the errors.

Command retry. Error correction (without programming assistance) is performed by a channel/control unit command retry procedure without an intervening I/O interruption in the following situations:

1. When a correctable data error occurs during a search or read operation on home address, record count, or record key.

   During a search or read operation, the home address, count, or key read from the disk track is placed in a buffer in control storage within the control unit. When a correctable data error occurs, the control unit corrects the data in the buffer and reissues the command that caused the error. During reorientation to the record, the control unit disconnects and frees the block multiplexer channel. When the failing search or read command is reexecuted, the corrected data in the buffer is used instead of the data actually on the track.

2. When an uncorrectable data error is detected on any portion of the record during a read or a search operation.

   The failing CCW is reissued twice by the control unit. If one of the two retries is successful, the channel program continues.

3. When a seek malfunction is detected.

   The control unit retries the command ten times in an attempt to position the access mechanism correctly.

4. When an alternate or defective track condition is recognized before data transfer begins.

   The control unit determines the location of the alternate or defective track (from R0 on the track), initiates a seek to this track, and orients to the index point. When this sequence completes, the original command is reissued by the control unit. This is a programmed procedure for announced System/360 direct access devices.

5. When a command overrun (or late command-chaining) condition occurs during execution of a channel program because of interference from another channel or the processor.

   The control unit initiates a retry of the command that was late.

6. When a data overrun occurs, except for

   a. A data overrun that occurs during a track overflow operation in the second or subsequent segments

   b. A data overrun that occurs during a formatting write

Error logging. Usage and error counters for each drive in the string are maintained continuously in the control unit. The usage counters are used to accumulate the number of bytes read and seeks issued. The error counters are used to accumulate the number of seek, correctable data, and uncorrectable data errors that were retried successfully by a command retry procedure, as already described. Also accumulated is the total number of command and data overrun conditions that were retried by the storage control unit.

When a counter reaches its threshold or when a pack is removed from a drive, the control unit indicates the condition via a unit check when the next I/O operation is initiated to the drive. Counter data can be obtained and counters can be reset by issuing a READ AND RESET BUFFERED LOG command. These statistics can then be logged in the system error data set for later diagnosis.

Inline diagnostic tests. A 3830 Model 1 control unit can execute diagnostic tests on a malfunctioning drive while normal operations take place on the remaining drives in the string. After the service address plug is inserted in the malfunctioning drive, diagnostic tests can be loaded into a transient area of the control storage of the 3830 Model 1 and executed on the malfunctioning drive. This can be done in an online environment using OLTEP or the CE panel on the 3830 Model 1. OLTSEP can be used in a standalone environment.

Inline testing allows CE diagnosis and repair of most 3330-series drive failures without the necessity of taking the entire 3330-series string out of the system configuration.

Model 2

The 3830 Storage Control Model 2 unit is functionally equivalent to the 3830 Storage Control Model 1 unit, except for the following:

- Up to four strings of from two to eight drives, in one-drive increments, can be attached to Model 2 of the 3830. When multiple requesting is used, the 3830 Model 2 can control up to 32 channel programs concurrently, one on each of its drives. Only one of the 2 to 32 drives attached to the 3830 Model 2 can be transferring data at any given time.
- The 3830 Model 2 does not contain power for any of the drives that can be attached to it.
- The 3830 Model 2 does not contain the device-oriented hardware that is present in Model 1 of the 3830.
- A string switching capability is available for the strings attached to a 3830 Model 2.
- The 3830 Model 2 can have 3340, 3340/3344, or 3350 strings attached and 3330-series, 3340, and 3350 strings can be mixed on one 3830 Model 2.

Model 2 of the 3830 can have from one to four strings of 3330-series drives attached to it. The 32 Drive Expansion and Control Store Expansion optional features must be installed on a 3830 Model 2 in order to attach more than two strings to it. Both of these features are field installable. Each string must include 3333 Disk Storage and Control Model 1 or 11 as the first module attached to Model 2 of 3830 Storage Control. Up to three 3330 Disk Storage modules, in any combination of Models 1, 2, 11, can then be added to each string for a total of eight drives per string. Only one 3333 module can be present in each string.

A 3333 module is connected to a 3830 Model 2 via cables, which can be a maximum of 150 feet in length. The 3830 Model 2 attaches to a block multiplexer channel via a cable up to 150 feet in length. Model 1 of 3830 Storage Control can be field-converted to Model 2. Field conversion of Model 2 of 3830 Storage Control to Model 1 is not recommended. The 3830 Model 2 provides lower cost attachment of two, three, or four 3330-series strings to a channel than the 3830 Model 1, since only one 3830 Model 2 unit is required.

The Two-Channel Switch and the Two-Channel Switch, Additional features can be installed on a 3830 Model 2 unit to enable the strings it controls to be accessed by two or four channels, as discussed for the 3830 Model 1. The 3333 String Switch optional feature can be installed on a 3333 attached to the 3830 Model 2. This field-installable feature enables the 3333 and all its attached 3330s (a 3330-series string) to be connected to two control-unit-type attachments instead of only one. The attachments can be any combination of two of the following:

- 3830 Storage Control Model 2 attached to a block multiplexer channel in a 4341 Processor or System/370 processor (Models 135 and up)
- 3880 Storage Control attached to a block multiplexer channel in a 4341 Processor or System/370 processor (Models 145 and up)
- Integrated Storage Controls for System/370 Models 158 and 168
- Integrated Storage Control for the System/370 Model 145 or 148
- 3345 Storage and Control Frame Models 3, 4, and 5 for the System/370 Model 145 Model 0
- 3330/3340-series Integrated File Adapter for the System/370 Model 135 or 138

The two attachments to which a 3333 with the 3333 String Switch feature is attached can be attached to the same or different channels in the same processor, or to channels in two different processors. In addition, channel switching features can be installed on one or both of these attachments.

The 3333 String Switch is functionally similar in its operation to the Two-Channel Switch. A switch on the 3333 can be set to allow the 3330-series string to be accessed via both attachments, one at a time. In effect, this setting provides two control unit paths to the string. String switching is accomplished dynamically under program control. Alternatively, the switch can be set to dedicate the string to one attachment or the other so that the string can be accessed only via that attachment.

Figure 20.10.1 illustrates 3333 string switching for two 3330-series strings. In the configuration shown, both strings can be accessed via two channels and two control units. Figure 20.10.1 also summarizes the 3330-series string configurations that are possible for a 4341 Processor. Channel switching, string switching, and 32 Drive Expansion features can be used to enhance the availability of 3330-series disk storage and to extend backup capabilities when two 4341 Processors or a 4341 Processor and a System/370 processor that can have 3330-series disk storage attached are present in an installation.

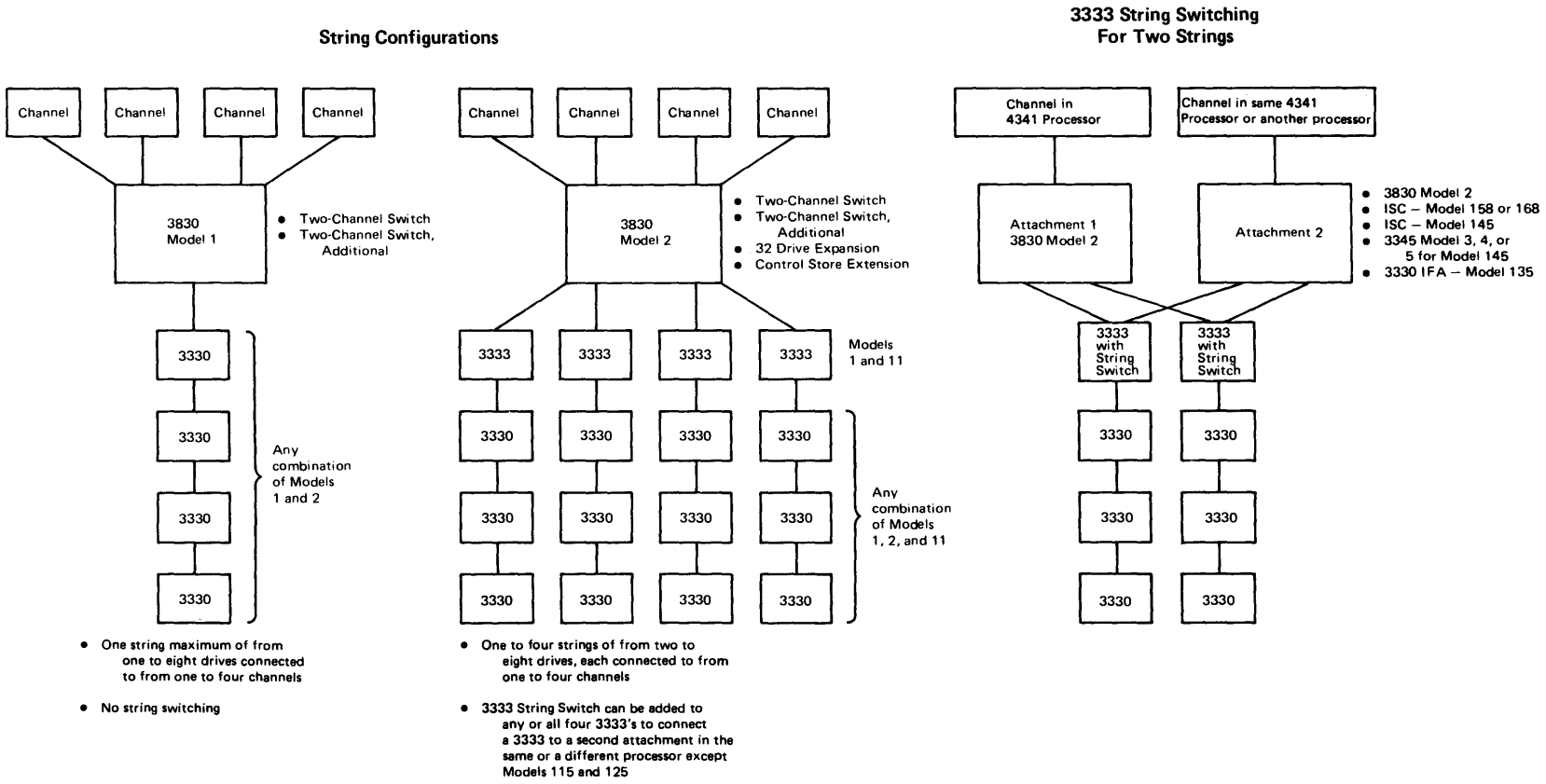


Figure 20.10.1. Summary of 3330-series string configurations for the 4341 Processor

SUMMARY

The 3330-series offers more than large capacity, faster access, and attractive price performance. A 3330-series string and its control unit are actually a subsystem. A 3830 control unit can control the concurrent execution of one RPS channel program on each of its drives and can handle certain error correction and logging functions, which normally must be programmed, thereby relieving the control program of these activities.

In addition, the availability and serviceability of the 3330-series are improved by the implementation of error correction features in hardware, by use of inline diagnostics, and by the speed and ease of engineering change installation. These factors, together with the extended channel switching and string switching features available, add to the improvement of total system availability.

Table 20.10.4. Hardware features of 3330-series and 2314 disk storage

Feature	3330-Series	2314 (A-Series)
Number of drives per string or facility	1 to 8 or 2 to 8 in one-drive increments	1 to 8 in one-drive increments. (A ninth can be included as a spare only.)
Number of strings or facilities per control unit	From 1 to 4	1 maximum
Removable interchangeable disk packs	Yes	Yes
Removable address plugs	Yes	Yes
Record Overflow feature	Standard	Standard
File Scan feature	Not available	Standard
Multiple track operations	Standard	Standard
Two-Channel Switch	Optional	Optional
Attachment of the control unit to four channels	Yes, using the Two-Channel Switch and Two-Channel Switch, Additional features	Yes, using Two-Channel Switch and 2844 Auxiliary Storage Control
Attachment of a string or facility to two control units in the same or different processors	Yes via 3333 String Switch feature. Only one data transfer operation permitted per string.	Yes via 2844 Auxiliary Storage Control. Two concurrent data transfer operations per facility permitted.
Rotational position sensing	Standard (128 sectors/track)	Not available
Multiple requesting	Standard	Not available
Read-only switch on each drive to prevent all write operations	Standard	Not implemented
Command retry by control unit and channel	Standard	Not implemented
Error correction data presented by control unit	Yes	No
Writable storage in control unit loaded from a disk cartridge	Yes	No
Inline diagnostic tests initiated via OLTEP or the CE panel in the string or facility	Standard	Standard

20:15 3350 DIRECT ACCESS STORAGE

MODELS, FEATURES, AND STRING CONFIGURATIONS

The 3350 Direct Access Storage is very large capacity, high-performance, modular, fixed- media disk storage. It has a significantly faster data transfer rate than 3330-series disk storage and much larger capacity. Two 3330-compatible modes of operation, in addition to 3350 native mode, are provided to aid in converting from 3330-series to 3350 direct access storage.

A 3350 does not have a storage medium that is removable by the operator. The read/write head mechanism and recording disks of the 3350 are assembled as units and fixed in the 3350 drive. The head/disk assembly can be removed only by the customer engineer if it is necessary to send the assembly to the plant of manufacture for repair or data retrieval.

The 3350 is available in Models A2, A2F, B2, B2F, C2, and C2F. The Model A2 consists of a controller and two drives. The Model B2 consists of two drives and no controller. The Model C2 has a controller function and two drives. It can be utilized to provide an alternate controller for a 3350 string.

The A2, B2, and C2 models contain storage that is accessible only by movable read/write heads and have a maximum capacity (with full-track records) of 317.5 million bytes per drive. The A2F, B2F, and C2F models are identical to the A2, B2, C2 models, respectively, except that each of the two drives in a unit contains up to 1,144,140 bytes of fixed head storage with a seek time of zero, in addition to movable head storage.

A 3350 string consists of one to four 3350 units. Movable head and fixed head models can be mixed within a string. The first unit in the string must be a Model A2/A2F. The remaining three units can be all Model B2/B2F units or two Model B2/B2F units and one Model C2/C2F unit. Only one Model C2/C2F unit can be included in a 3350 string and it must be the last unit in the string. The maximum capacity of a 3350 string is more than 2.5 billion bytes.

When a Model C2/C2F is present in a 3350 string, the Primary Controller Adapter feature is required on the Model A2/A2F and a switch on the control panel of the Model C2/C2F must be set to determine whether the controller in the Model A2/A2F or Model C2/C2F is online as the controller for the string. The unit whose controller is offline operates as a Model B2/B2F. Manual switching from one controller to the other must be preceded by a power-off of both the Model A2/A2F and C2/C2F units.

A 3350 string attaches to a block multiplexer channel of the 4341 Processor via 3830 Storage Control Model 2 and 3880 Storage Control. The 3830 Model 2 must have the Control Store Extension, Expanded Control Store, and Register Expansion features installed in order to attach a 3350 string. Up to four 3350 strings can be attached to a 3830 Model 2 and to each storage director in a 3880. The four strings attached to a 3830 Model 2 can be any mixture of 3350, 3340, and 3330-series strings, while only 3330-series and 3350 strings can be attached to the same storage director in a 3880. A 3340/3344 string (a 3340 string with at least one 3344 unit) cannot be intermixed with 3350 or 3330-series strings attached to the same 3830 Model 2.

All models of the 3350 also can be attached to 3830 Storage Control Model 3 to be used as real (but not staging) devices for the 3850 Mass Storage System. Only 3350 native mode of operation can be used when a 3350 drive is attached to the 3830 Model 3 as a real device.

An operator panel is located on the top of each 3350 drive. This panel contains a start/stop switch to start and stop rotation of the disks and a switch to provide read-only protection for the disks. When the latter switch is in the read position, no write or erase commands can be executed on the drive. When the switch is in the R/W position, all commands can be executed.

The standard selective format feature of the 3350 permits any model drive to operate in one of three modes: 3350 native mode, 3330 Model 11 compatibility mode, or 3330 Model 1 compatibility mode. The mode is specified for each drive at the time of manufacture and the tracks in each drive are formatted for the specified mode. The mode can be changed in the field by the customer engineer. The 3350 drives within a given string can have any mixture of the three formats.

When 3350 native mode is in effect, a track in a 3350 has a maximum capacity of 19,069 bytes (for full-track records) and maximum drive capacity is 317.5 million bytes. When 3330 Model 11 compatibility mode is in effect, maximum track capacity is that of the 3330 Model 11 (13,030 bytes) and the maximum capacity of the 3350 drive is 200 million bytes. The 3330 Model 1 compatibility mode provides the equivalent of two logical 3330 Model 1 volumes per 3350 drive and, thus, a maximum capacity of 200 million bytes (100 million bytes per logical 3330 volume).

When a fixed head 3350 model operates in a 3330 compatibility mode, a maximum of 742,710 bytes of fixed head storage are available. They are associated with the first logical volume only for 3330 Model 1 compatibility mode. Programs written to access 3330-series drives can be used without modification to access 3350 drives operating in a 3330 compatibility mode.

When a 3350 string contains one or more drives formatted in 3330 Model 1 compatibility mode, logical device addresses are used to address the two logical volumes within a drive. A maximum of 64 logical addresses (00 through 3F) are possible for the maximum of four 3350 strings (32 drives) that attach to the 3830 Model 2. The bits in the I/O device address byte in a START I/O instruction (byte 3) are coded to perform the required addressing for 3350 strings.

When a drive in 3330 Model 1 compatibility mode is in a 3350 string, bits 0 and 1 of the I/O address byte are the storage control address bits. (Bits 0, 1, and 2 are used to address storage control when there is no 3330 Model 1 compatibility mode drive in the string.) Bits 3 and 4 address one of the four strings and bits 5, 6, and 7 address one of the eight drives in the selected string. Bit 2 is then used to indicate which logical device within the selected drive is addressed. The first logical volume is selected when bit 2 is off, while the second logical volume is selected when bit 2 is on. (See 3350/3344 Installation and Conversion Guide, GC20-1780, for the permissible device addresses for 3350 string configurations.)

The timing characteristics of a 3350 drive are 1198 KB/sec data transfer rate, 10 milliseconds cylinder-to-cylinder seek time, 25 milliseconds average seek time, 50 milliseconds maximum seek time, and 8.4 milliseconds average rotational delay. These figures compare with 806 KB/sec, 10 milliseconds, 30 milliseconds, 55 milliseconds, and 8.4 milliseconds for 3330-series drives.

The 3350 timing characteristics are the same regardless of the mode in which the 3350 drive is operating. Thus, the 3330 compatibility modes of the 3350 offer improved seek and data transfer times as well as easier conversion from 3330-series drives. They also offer fixed head storage, which is not available for 3330-series drives. Table 20.15.1

compares capacity characteristics of the 3350 drive for its three operating modes.

Table 20.15.1. 3350 capacity characteristics by mode

Characteristic	3350 Native Mode	3330 Model 1 Compatibility Mode	3330 Model 11 Compatibility Mode
Cylinders per drive	555 plus 5 alternates	404 plus 7 alternates (per logical volume)	808 plus 7 alternates
Tracks per cylinder	30	19	19
Tracks per drive	16,650 plus 150 alternates	7676 plus 133 alternates (per logical volume)	15,352 plus 133 alternates
Full-track capacity in bytes	19,069	13,030	13,030
Cylinder capacity in bytes	572,070	247,570	247,570
Drive capacity in millions of bytes	317.5	100 per logical volume (200 total)	200
Fixed head capacity in bytes	1,144,140	742,710 in first logical volume	742,710

Rotational Position Sensing and Record Overflow are standard on all 3350 drives, which have no optional features. String switching can be installed on 3350 Models A2, A2F, C2, and C2F to enable the string to be accessed by two control-unit-type attachments, as for 3330-series strings. When string switching is installed on a Model A2/A2F, it is recommended that it also be installed on the Model C2/C2F, if one is present in the string. In addition, a 3830 Model 2 to which 3350 strings are attached can have channel-switching features installed (Two-Channel Switch and Two-Channel Switch, Additional).

The following 3350 model changes are field installable: A2 to A2F, B2 to B2F, C2 to C2F and the reverse conversions (A2F to A2, etc.). The following model changes are available at the time of manufacture only: A2 to B2 or B2F; B2 to any other model except B2F; C2 to B2 or B2F; A2F to B2 or B2F; B2F to any other model except B2; and C2F to B2 or B2F. The following conversions are not recommended for field installation: A2 to C2; A2 to C2F; C2 to A2F; A2F to C2F; and the reverse conversions (C2 to A2, etc.).

The physical, capacity, and timing characteristics of 2314, 3350, 3330-series, 3340, and 3344 drives are compared in Table 20.25.1 in Section 20:25.

TRACK, CYLINDER, AND READ/WRITE HEAD LAYOUT

The layout of physical tracks on a data surface in a 3350 drive and the relative position of the read/write heads for a surface are shown in Figure 20.15.1. A data surface contains 1122 physical tracks. There

are two read/write heads associated with each data surface that are positioned 561 physical tracks apart, as shown in Figure 20.15.1.

The access mechanism can be placed at any one of 561 access positions on the data surface. Therefore, an outermost head on the access mechanism can access physical tracks 0 to 560 on its associated data surface while an innermost head can access physical tracks 561 to 1121. At any of the 561 possible access mechanism positions, two physical tracks can be accessed on a data surface. However, only one read/write head in a 3350 drive can be active at a time.

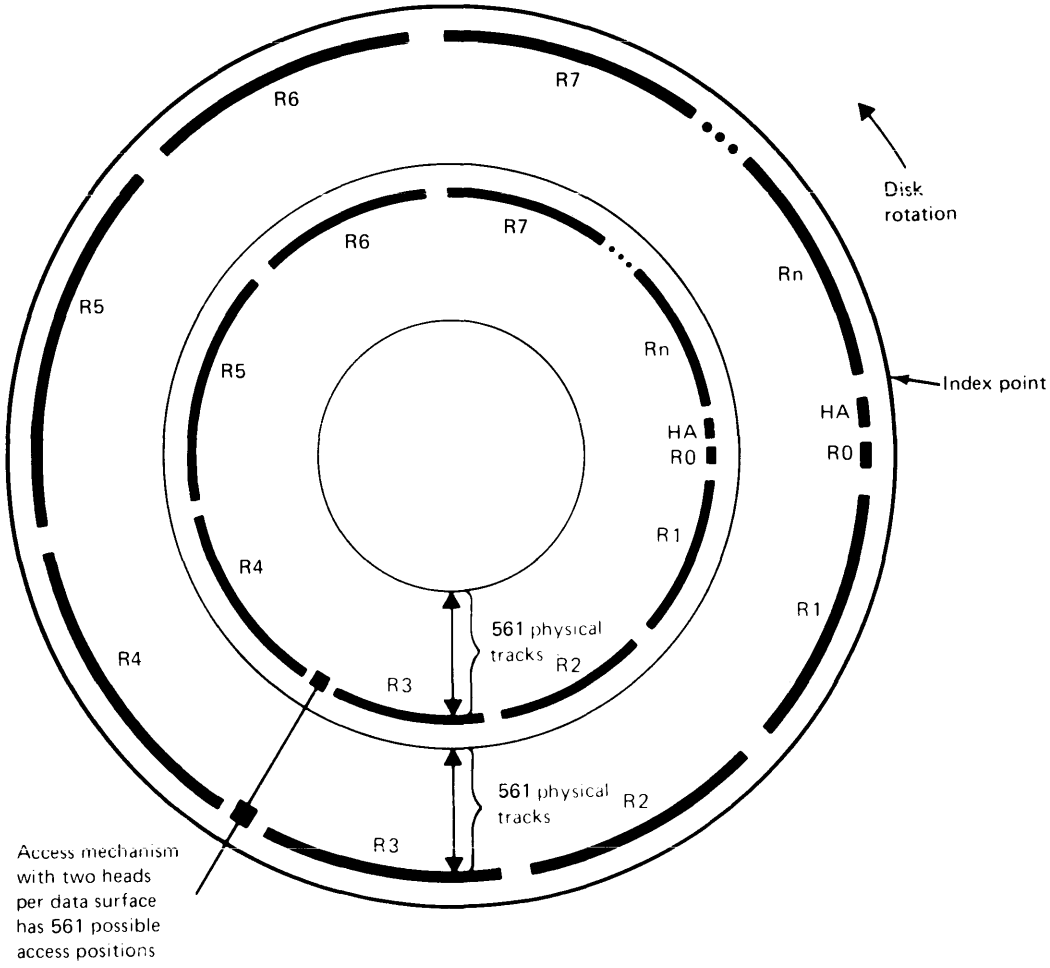
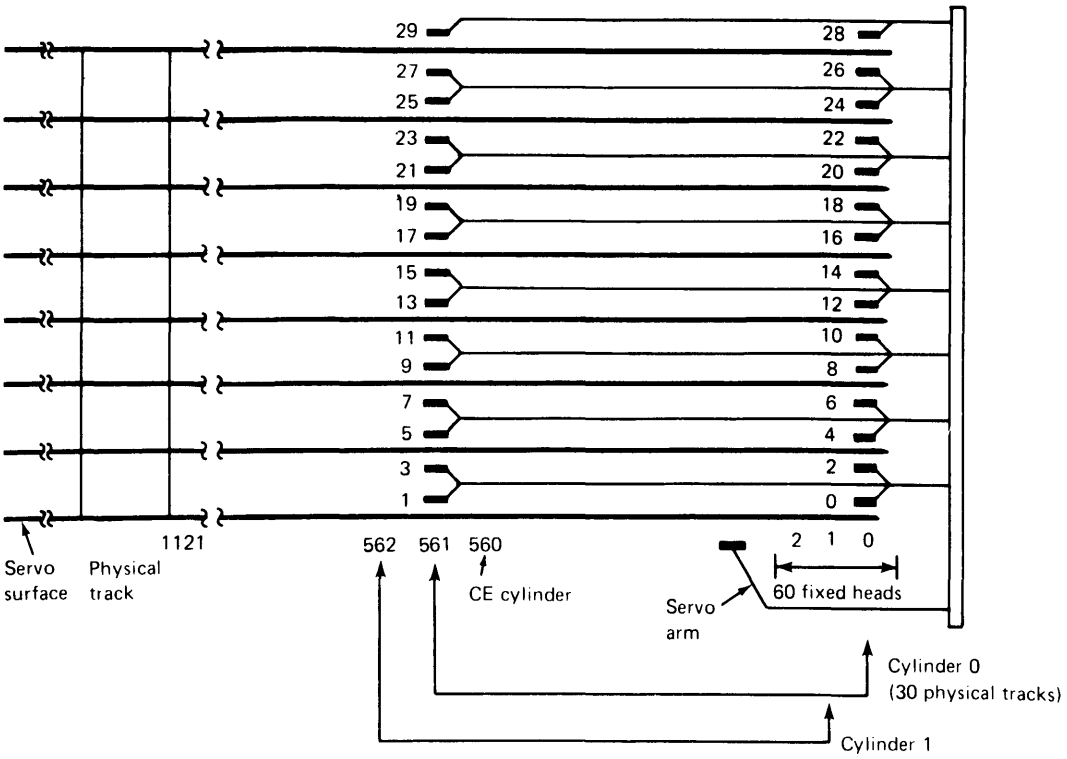


Figure 20.15.1. Location of physical tracks and read/write heads on a data surface in 3350 Direct Access Storage

Figure 20.15.2 shows the layout of cylinders and read/write heads in 3350 Direct Access Storage for 3350 native mode. A 3350 drive always contains eight recording disks. Fifteen of the surfaces are used for data recording. They are accessed by 30 read/write heads addressed 0 to 29. The 30 physical tracks that can be accessed at each physical position of the access mechanism constitute a cylinder in 3350 native mode. There are 555 data cylinders (addressed 0 to 554) and 5 alternate cylinders (addressed 555 to 559).

Cylinder 560 is the CE cylinder. It is designed to be used only by the customer engineer for testing the read/write capability of the 3350 drive. It contains a prewritten area for read testing and an area in which write tests can be performed.



Number of recording disks	8
Number of recording surfaces	15
Number of logical tracks per physical track	1
Number of movable read/write heads	30
Number of fixed read/write heads (Models A2F, B2F, and C2F only)	60
Number of tracks per cylinder	30
Number of data cylinders per drive	555
Number of data tracks per drive	16,650
Number of alternate tracks per drive	150
Number of CE cylinders per drive	2

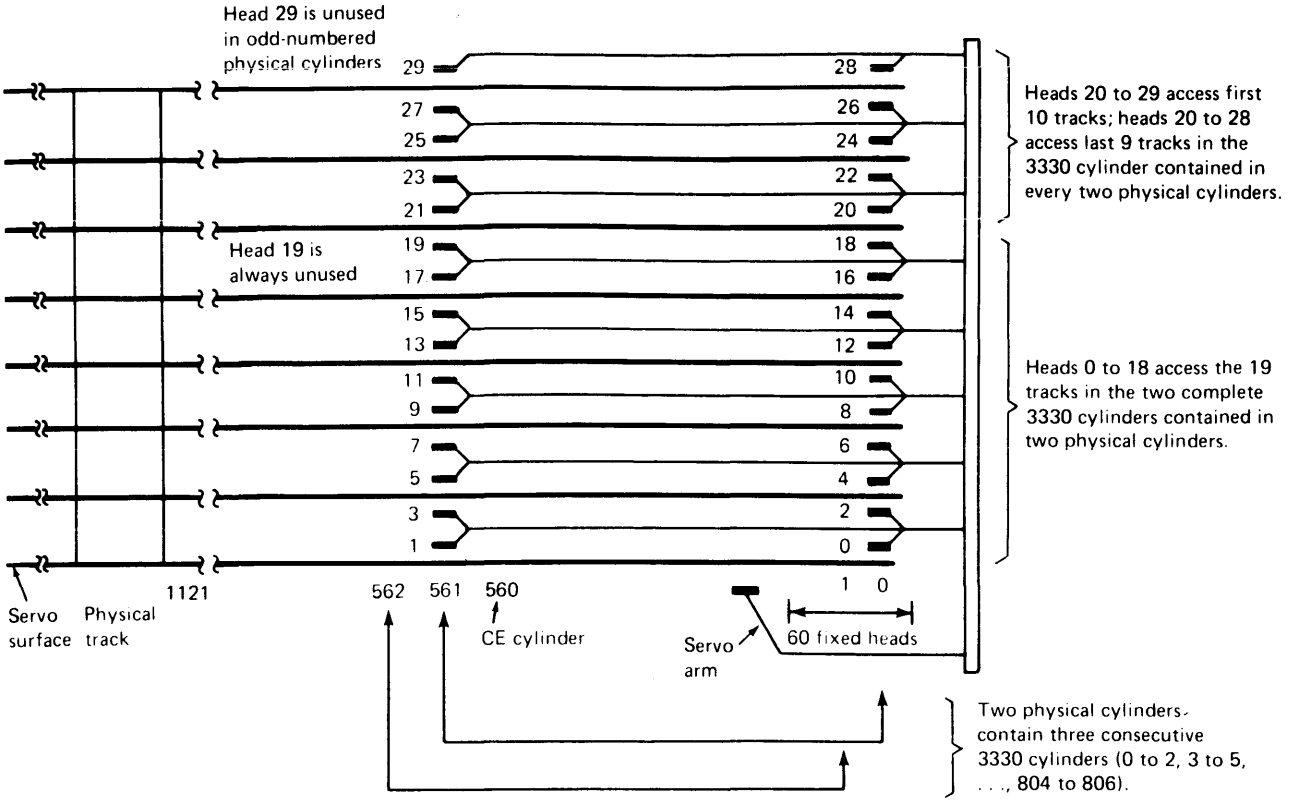
Figure 20.15.2. Cylinder and read/write head layout for a 3350 drive formatted in 3350 native mode

The bottommost surface in a 3350 drive is used as the servo surface, as it is in a 3330-series drive. This surface contains information for the servo system that is used to control seek operations, positioning of the heads over tracks, data clocking (the synchronization of data with rotational speed during writing operations), index generation, and signal generation required by the RPS feature. Functionally, the 3350 servo system is like that used in 3330-series drives. However, design improvements, such as elimination of the electromechanical tachometer, have been made.

The required servo information is prerecorded on the servo surface at the plant of manufacture and is read by a servo read head at the bottom of the access mechanism. The servo information on this surface cannot be read or written using 3350 commands. The servo surface for a fixed head 3350 model also contains the physical tracks that are read by the fixed heads.

Figure 20.15.3 shows the cylinder and read/write head layout for a 3350 operating in 3330 Model 11 compatibility mode. Three consecutively addressed 3330 cylinders are mapped in every two physical cylinders.

The first physical cylinder of a pair contains the 19 tracks of the first 3330 cylinder (accessed by heads 0 through 18) and the first 10 tracks of the second 3330 cylinder (accessed by heads 20 through 29). The second physical cylinder of a pair contains all 19 tracks of the third 3330 cylinder (accessed by heads 0 through 18) and the last 9 tracks of the second 3330 cylinder (accessed by heads 20 through 28). Head 19 is unused. Head 29 is unused for the second physical cylinder of a pair.



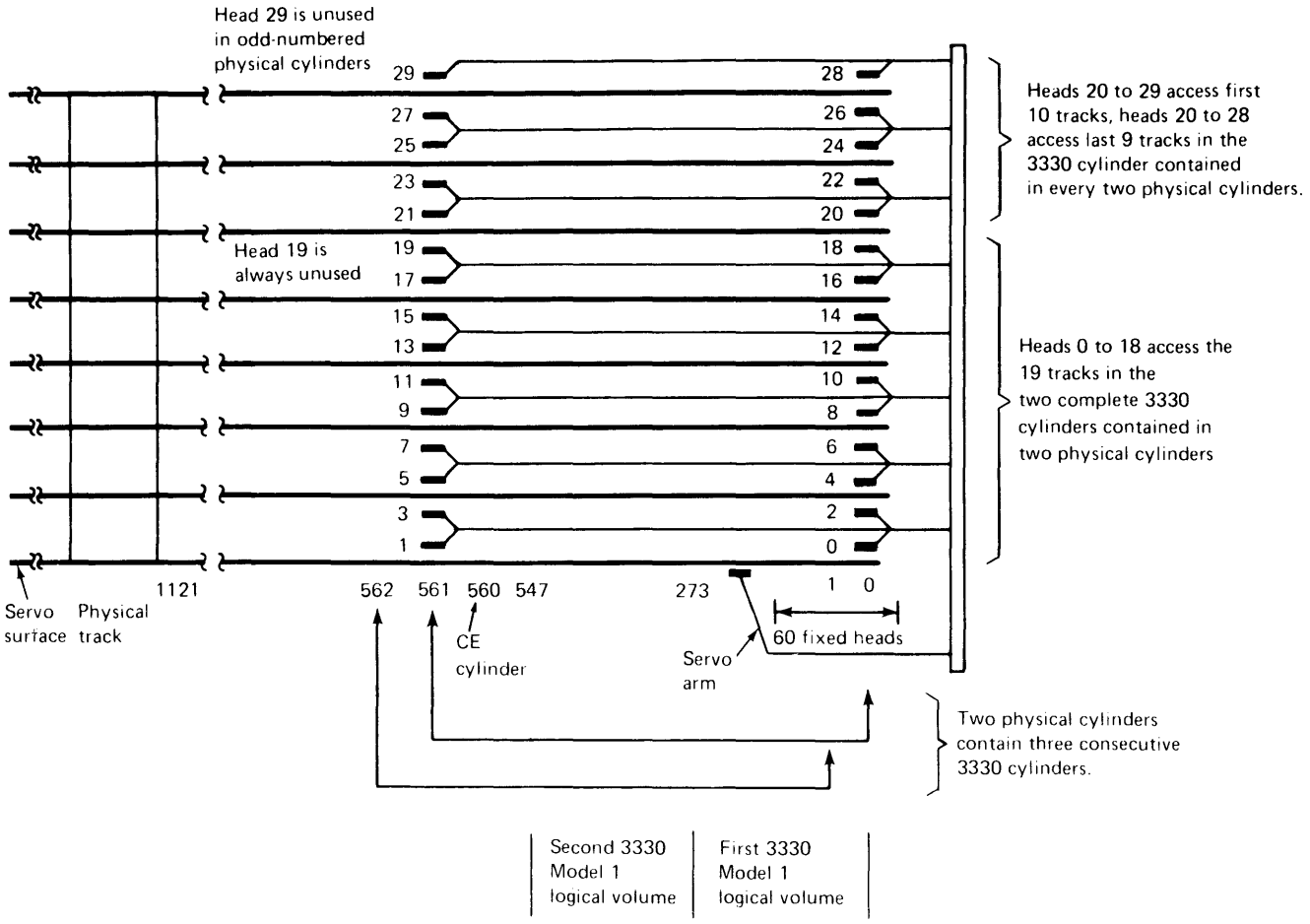
Two physical cylinders contain three consecutive 3330 cylinders
Physical cylinders 0 through 537 contain 3330 cylinders
0 through 806, while 3330 cylinder 807 is in physical cylinder 538
Alternate 3330 cylinders 808 through 814 are contained in
physical cylinders 538 through 543
Physical cylinders 544 through 559 and the balance of 543 are used
Physical cylinder 560 is the CE cylinder

Figure 20.15.3. Cylinder and read/write head layout for a 3350 operating in 3330 Model 11 compatibility mode

Physical cylinders 0 through 537 are used to map the 3330 cylinders 0 to 806 while the first 19 tracks in physical cylinder 538 are used to map 3330 cylinder 807. The balance of physical cylinder 538 and cylinders 539 through 542 plus the top nine tracks in cylinder 543 are used as alternate 3330 cylinders 808 through 814. Cylinder 560 is the CE cylinder while the remaining cylinders (544 through 559 and a portion of 543) are unused.

Figure 20.15.4 shows the cylinder and read/write head layout for a 3350 operating in 3330 Model 1 compatibility mode. As for 3330 Model 11 compatibility mode, each three 3330 cylinders are mapped in two physical cylinders. However, the first logical 3330 volume (404 3330 data cylinders plus 7 alternate cylinders) is mapped in physical cylinders 0

through 273. The second logical 3330 volume is mapped in physical cylinders 274 through 547. The CE cylinder is in 560 and cylinders 548 through 559 are unused.



Two physical cylinders contain three consecutive 3330 cylinders
Physical cylinders 0 through 273 contain the first
3330 Model 1 logical volume
3330 cylinders 0 through 403 are in physical cylinders 0 through 269
3330 cylinders 404 through 410 (alternates) are in physical cylinder 269 through 273
Physical cylinders 274 through 547 contain the second
3330 Model 1 logical volume
3330 cylinders 0 through 403 are in physical cylinders 274 through 543
3330 cylinders 404 through 410 (alternates) are in physical cylinders 543 through 547
Physical cylinders 548 through 559 are unused
Physical cylinder 560 is the CE cylinder

Figure 20.15.4. Cylinder and read/write head layout for a 3350 operating in 3330 Model 1 compatibility mode

A four-byte CCHH field is used in seek commands for the 3350. The seek limits depend on the mode being used. For native mode, valid CC values are 0 through 559 and valid HH values are 0 through 29. For 3330 Model 11 compatibility mode, 0 through 814 and 0 through 18 are valid CC and HH values, respectively. Valid CC and HH values for 3330 Model 1

compatibility mode are 0 through 410 and 0 through 18, respectively. When the second logical 3330 Model 1 volume in a 3350 drive is addressed, a value of 274 is added to the converted 3330 seek address to obtain the correct physical cylinder.

The fixed head 3350 models have 60 fixed read/write elements located on a plate under the servo surface, as shown in Figures 20.15.2 through 20.15.4. They access cylinders 1 and 2 for a drive in native mode and 3330 cylinders 1 through 3 for a drive in 3330 compatibility mode. When a command that addresses a fixed head cylinder is received, the 3350 automatically selects the associated fixed read/write element instead of the movable head. Therefore, the same channel programs used for movable head models can be used to access fixed head models.

The tracks in physical cylinders 1 and 2 are not used in 3350 Model A2F, B2F, and C2F drives and cannot be accessed by a user or customer engineer because of the way head selection is performed. Thus, the data capacity of a fixed head 3350 model is the same as that of a movable head model. Seek time for the cylinders accessed by the fixed heads is zero. Seek times for the cylinders accessed by movable heads are the same as for movable head models.

A data set or file can be contained in both fixed head cylinders and cylinders that are accessed by movable heads. A 3350 drive, however, can perform only one operation at a time. Therefore, a seek, search, or data transfer operation involving a fixed head in a Model A2F, B2F, or C2F cannot be performed at the same time a movable head is involved in a seek, search, or data transfer operation.

The best performance gains can be achieved for fixed head 3350 drives by assigning the fixed head tracks to small active system data sets/files (such as the DOS/VS or DOS/VSE page data set, system catalog, TCAM message queue, VTOC, OS/VS pageable link pack area, JES2 spool file), small active user data sets/files (containing tables, indexes, work areas, data, etc.), large active data sets/files that can be segmented (OS/VS1 page data set, partitioned data sets, ISAM index levels, for example), and data sets/files with major activity concentrated at the beginning of the data set/file (such as the OS/VS job queue).

The assignment of such data sets/files to the fixed head tracks in a Model A2F, B2F, or C2F is a user responsibility. DOS/VS or DOS/VSE EXTENT and OS/VS DD statements for these files and data sets must specifically request, by actual address, locations within the fixed head logical cylinders. Note also that the device type code in the device table that is generated in the control program during a system generation (DOS/VS or DOS/VSE PUB table, OS/VS UCB table) does not differentiate between 3350 drives with and without fixed heads. Therefore, if generic device type assignment by device type (3350) is used in a configuration that contains 3350 drives with and without fixed heads, either type drive can be selected by the operating system.

The assignment of a 3350 drive with fixed heads can be assured in an OS/VS environment by specifying a user-defined device class name for such 3350 drives at system generation and using this name (instead of UNIT=3350) in the appropriate DD statements. DOS/VS and DOS/VSE users utilizing the generic I/O device assignment capability can specify an address list of the 3350 drives with fixed heads in the ASSGN statements for files that are to be located on a 3350 Model A2F, B2F, or C2F.

Cylinders that provide alternate tracks for the physical tracks accessed by the movable heads can also provide alternate tracks for the fixed head tracks. This approach is taken because the probability a fixed head track will develop a defect is lower than that for movable

head tracks and the possibility of a defect occurring in a movable head track is very low.

The low probability of defects occurring in fixed head cylinders of a 3350 results in part from the fact that these cylinders are recorded on the servo surface, which is a specially manufactured surface because of its primary function. The width of a fixed head physical track is six times greater than that of a movable head track on a data surface.

If an uncorrectable error does occur on a fixed head track in a 3350, the customer engineer will normally replace the head/disk assembly instead of assigning an alternate movable head track to avoid loss of performance. If an uncorrectable error occurs on a movable head track, the track should be flagged and an alternate track should be assigned. This can be done using the IEHTLAS, IEHDASDR, or IBCDASDI utility of OS/VS. IEHDASDR or IBCDASDI should then be used to test the flagged track to determine whether the track is really defective. If the track is found not to be defective, the flag is removed and the assigned alternate track is released.

Note that the defective track testing capability of OS/VS IEHDASDR and IBCDASDI is not provided by any DOS/VS or DOS/VSE utility. DOS/VS users can obtain IBCDASDI, the standalone utility, by ordering the OS/VS1 system.

The Analysis Program-1 (AP-1) utility is provided as a problem determination aid for 3350 drives. When errors occur on a 3350, AP-1 can be used to determine whether the drive is failing or a problem exists on a recording disk. The AP-1 program operates as a problem program under DOS/VS, DOS/VSE, or OS/VS.

TRACK FORMATTING AND INITIALIZATION

Self-formatting records consisting of count, key, and data or count and data areas are written on the tracks in a 3350 drive just as on the tracks in a 3330-series drive. However, each home address, count, and key area written on a 3350 track has a detection code field appended to it for data validity checking by the 3830 Model 2. The detection code can detect a single error of ten bits span or less.

A correction code field is appended to each data area written on a 3350 track. The correction code used has the same detection capability as the detection code and the ability to correct single-error bursts of four bits span or less. The actual error correction procedure must be performed by programming (error recovery routines) using corrective bits that are supplied by the controller.

That is, when the controller detects a correctable data error during the reading of the data portion of a record, it generates the information necessary to correct the erroneous bytes. The sense bytes presented by the controller contain a pattern of corrective bits and a displacement value to indicate which of the bytes transferred to processor storage contain the errors. The disk error recovery program need only EXCLUSIVE OR (logical operation) the corrective bit pattern with the error bytes in the input area in processor storage to correct the errors.

The home address and count areas written on a track in a 3350 contain two new fields in addition to the same fields as are written in home address and count areas on 3330 tracks. The home address and each count area on a 3350 track contain a two-byte skip defect field and a two-byte physical address field in front of the flag byte. The automatic surface defect skipping capability of the 3350 allows valid data to be written before and after a surface defect on a track.

The skip-defect (SD) bytes are used to indicate the location of the center of the surface defect relative to the index point of the track. Bits in the flag byte field indicate whether the surface defect is located in the next count, key, or data area.

The error detection and correction code capabilities of the 3350 controller permit successful recovery from an error within the data portion of a physical record even when it contains a surface defect gap.

Partial initialization of the disks in all 3350 drives is performed at the plant of manufacture. A home address record and track descriptor (R0) record are written on each track in the drive. If a single skippable defect is found during the analysis of the surface of a track, the appropriate SD bytes and flag byte are written in the home address to indicate this fact. If no surface defect is found, the SD bytes are written as zeros.

The SD bytes and flag byte are supplied in the count area field in virtual storage only for a WRITE HOME ADDRESS command. When R0 is written during disk drive initialization and thereafter whenever a formatting write is performed, the SD and flag bytes for the count area to be written on disk are supplied by the controller, which reads them from the record immediately preceding the record to be written.

If data cannot be read from a 3350 track during its use in an installation and recovery of this data is critical, the head/disk assembly unit can be returned to the plant of manufacture where recovery will be attempted.

The two physical address bytes in home address and count areas on a 3350 track contain the physical cylinder and track address of the track on which they are written. When a seek command is issued to a 3350 drive operating in a 3330 compatibility mode, the controller converts the cylinder and track address specified by the seek command to a physical cylinder and track address that is actually used by the drive in the seek operation. This physical address is saved in the controller for later use in seek verification. For a drive operating in 3350 native mode, the seek address is saved as specified.

The physical address bytes are automatically written and read by the controller and are not processed by programming. That is, when a home address or count area is written, the physical address bytes are automatically supplied by the controller and are not contained in the home address or count area field in processor storage that is indicated by the write command. Similarly, when a home address or count area is read, the controller reads the physical address bytes but they are not placed in the home address or count field area in virtual storage.

The physical address bytes are used by the controller for seek verification during normal operations and by the 3350 microdiagnostic routines. When a home address or count area is processed during a read, search, or clock operation, the physical address bytes read are compared with the most recent seek address (physical cylinder and track address) that was saved in the controller when the last seek command was issued. If the two physical addresses are not equal, the command is terminated and a unit check condition results. Seek check is indicated in the sense bytes.

The commands for the 3350 are the same as those for the 3330, except that one additional command, READ MULTIPLE COUNT-KEY-DATA, is implemented for the 3350. This command causes reading to begin with the count field of the next record that passes under the read/write heads (unless it is R0). Reading continues until all remaining records on the track are read. This command enables all the records on a physical track to be read using one command. The command retry capability

implemented in the 3830 for 3330-series drives is also implemented for 3350 drives (see discussion in Section 20:10 under "3830 Storage Control").

The write format release feature that is implemented for 3330 Model 11 drives is also standard for 3350 drives. This feature enables a 3350 drive to disconnect from the 3830 Model 2 while the drive is erasing to the end of the track after a record has been written with a formatting write command. This facility frees the controller and channel for the initiation of another I/O operation.

ADVANTAGES

The 3350 is a growth direct access device for medium and large 3330-series installations. It offers significantly larger capacity at a lower cost per bit than the 3330 series; improved performance; enhanced physical characteristics; reliability, availability, and serviceability improvements; and minimal conversion effort. The 3350 can be used to increase online direct access capacity where data portability is not required.

Improved Performance

The 3350 has a higher data transfer rate than the 3330-series drive (almost 50 percent faster) as well as faster average and maximum seek times (approximately 16 percent and 9 percent faster, respectively). Improved seek time also results from the fact that more data is covered per minimum, average, and maximum seek time on a 3350 drive than on a 3330-series drive, and from the availability of fixed head 3350 models that have a zero seek time for a portion of the data.

Enhanced Physical Characteristics

A 3350 drive (which is the same size as a 3340 and 3344 drive) is smaller and contains significantly more data than a 3330-series drive. For a given capacity, a 3350 configuration requires approximately one-fourth the amount of space as a 3330 Model 1 configuration and approximately one-half the space required by a 3330 Model 11 configuration.

A 3350 unit requires 44 percent less power than a 3330-series Model 1 or 11 unit, which also results in reduced air conditioning requirements. In addition, the perceived noise level of a 3350 drive is approximately 40 percent less than that of a 3330-series drive.

Improved Reliability

Reliability is improved by the removal of head-to-disk alignment problems. Each read/write head within a 3350 drive is dedicated to certain tracks on one data surface. Therefore, each head reads only the data it wrote previously. Since common head alignment across 3350 drives is not required, the critical alignment tolerances that are normally necessary for a disk pack are not needed for the 3350. There is little chance of a misaligned head causing an error, because the 3350 drive servo system actually uses information from each data head to position itself. The 3350 provides accurate tracking and data transfer with minimal errors for far greater data densities than previously announced direct access devices.

Reliability is improved because the chance of damaging read/write heads through mishandling and exposure of disk data to outside contamination are eliminated by the fixed media design of the 3350.

In addition, the possibility of head crashes is minimized by the improved flying characteristics of the read/write heads in a 3350 drive. The low mass of the read/write heads and the low loading force used enable the heads to fly over the rotating disks at a very low height. This near contact (or proximity) recording capability of the read/write heads in the 3350 permits a bit with a weaker than normal signal to be read correctly.

The recording density in bits per inch of a track in a 3350 drive is approximately 1.5 times greater than the recording density of a track in 3330-series drives. The number of tracks per inch in a 3350 drive is approximately 2.5 times greater than the number of tracks per inch in a 3330-series Model 1 or 2 drive and almost 1.3 times greater than for a 3330-series Model 11 drive. The advanced head design used for the 3350 enables greater density to be achieved together with improved reliability.

Reliability of 3350 direct access storage is also improved because many critical mechanical parts have been eliminated, such as a complex head load/unload mechanism. In other cases, electronic functions have replaced mechanical functions. While the 3350 drive contains more electronics than the 3330, higher density logic cards are used in the 3350, which results in significantly fewer logic cards.

The improved power design of the 3350 further aids reliability as it results in fewer components, fewer power regulators, improved power isolation, and paralleling of critical circuits. As a result of the reliability features of the 3350, no preventive maintenance is scheduled for 3350 drives.

Improved Availability

The alternate controller function provided in Model C2 and C2F units enables a 3350 subsystem to remain operational in the event of a failure in the primary controller. This is accomplished without customer engineer intervention.

Improved Serviceability

Serviceability improvements for the 3350 are designed to help the customer engineer find and correct failures more quickly. The following are the major serviceability improvements:

- An enhanced fault symptom index has been added to provide faster problem isolation. It contains a list of "possible causes", an action/checklist, and cross references.
- Maintenance information Manuals for the customer engineer have been improved by the addition of summaries of adjustments and items to check. Additional theory and recovery actions have been developed, especially in the "no trouble found" sections.
- Microdiagnostics have been enhanced and are more powerful in fault isolation capability. In addition, they are much faster and execute with less CE intervention.
- System-level online test programs provide better read/write tests for the CE than have been available for 3330-series drives.

Serviceability is also improved because in a string that contains an alternate controller, the malfunctioning controller can be serviced while normal processing continues using the other controller.

20:20 3340 DIRECT ACCESS STORAGE

3340 DISK STORAGE DRIVES AND THE 3348 DATA MODULE

The 3340 Direct Access Storage unit is intermediate-capacity, modular, high-performance direct access storage that consists of 3340 Disk Storage and Control Model A2 and 3340 Disk Storage Models B1 and B2. A 3340 string can consist of from one to four units and is connected to a block multiplexer channel in a 4341 Processor via 3830 Storage Control Model 2 or 3880 Storage Control. Up to four 3340 and/or 3340/3344 strings can be attached to each storage director in 3880 Storage Control. A 3340 or 3340/3344 string cannot be intermixed with 3330-series or 3350 direct access storage attached to the same 3880 storage director.

A 3340 string can consist of from two to eight drives. A 3340 Disk Storage and Control Model A2 must be the first unit in a 3340 string. The 3340 Model A2 consists of two drives, drive-oriented control functions, and power for itself and the 3340 drives attached to it.

Up to three 3340 units, any combination of 3340 Disk Storage Models B1 and B2, can be attached to a 3340 Model A2. The 3344 (Models B2 and B2F) can also be attached to the 3340 Model A2 (see Section 20:25). The 3340 Model B2 consists of two drives and does not contain the power and device-oriented control functions that are part of the 3340 Model A2. The 3340 Model B1 contains one drive and no control functions. Functionally, all 3340 drives are alike regardless of whether they are part of a Model A2, B2, or B1 unit.

Figure 20.20.1 shows a 3340 string of five drives that includes one 3340 Model A2, one 3340 Model B2, and one 3340 Model B1. An operator control panel is located on the top of each 3340 drive. This panel contains the three-digit hexadecimal address of the drive, the switches required to operate the drive, and status indicator lights. The address of a 3340 drive is wired on a logic board in the 3340 unit.

The removable 3348 Data Module is used for data storage. Unlike the removable 2316 and 3336 Disk Packs that are the storage medium for 2314 and 3330-series disk storage, respectively, the 3348 Data Module is a sealed cartridge that contains a spindle, access mechanism, and read/write heads in addition to disks on which data is written and read. The cover of the data module, which is shock-absorbing and nonflammable, is never removed from the cartridge. The 3340 disk storage drive contains only the mechanical and electrical components that are required to house, load, air-filter, and drive the 3348 Data Module.

The 3348 Data Module is shown in Figure 20.20.2. The access mechanism in a 3348 Data Module is an L-shaped carriage that moves back and forth on a cylindrical shaft mounted within the data module. When the data module is not loaded, the access mechanism is latched in the home position so that it cannot move. In this position, the access mechanism is located such that the read/write heads rest on nondata areas on the disk surfaces.

Three models of the 3348 Data Module, all of which are the same physical size, are available. The 3348 Model 35 has a maximum capacity (assuming full-track records) of approximately 35 million bytes that are accessed by movable read/write heads. The 3348 Model 70 has a maximum capacity of approximately 70 million bytes that are accessed by movable read/write heads. The 3348 Model 70F also has a maximum capacity of 70 million bytes of which approximately 502,000 bytes maximum (60 logical tracks) are accessed by fixed read/write heads and the balance by movable read/write heads.

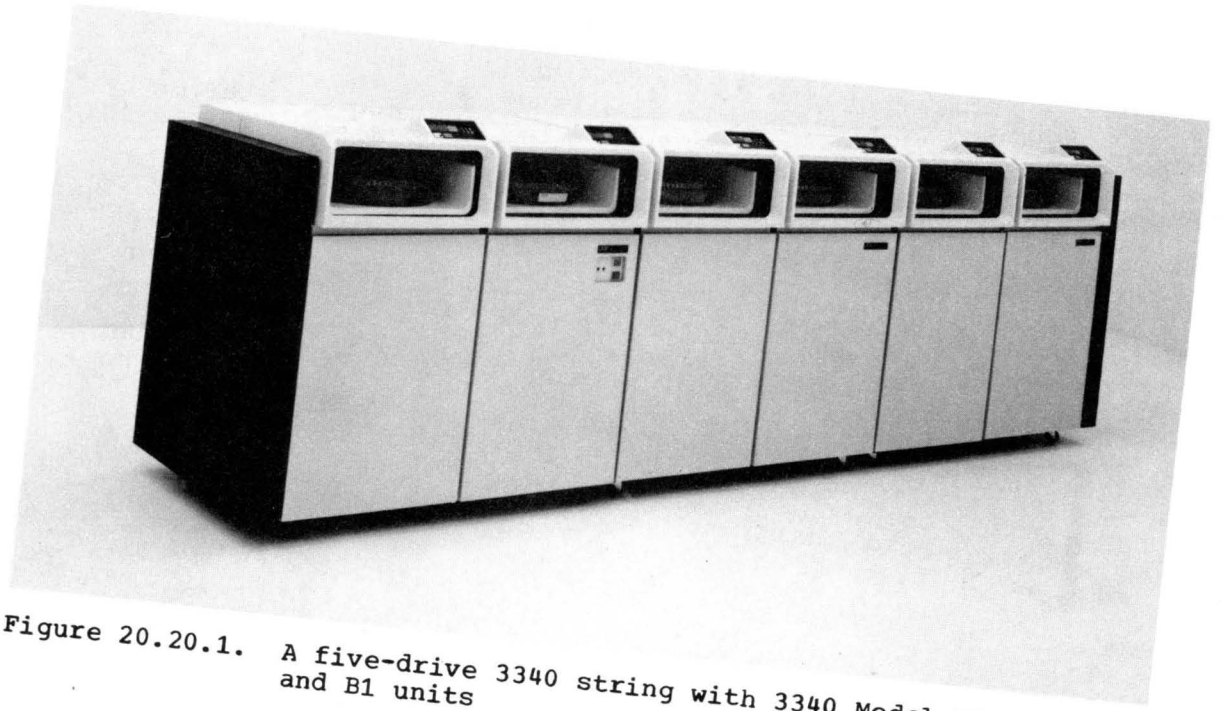


Figure 20.20.1. A five-drive 3340 string with 3340 Model A2, B2, and B1 units

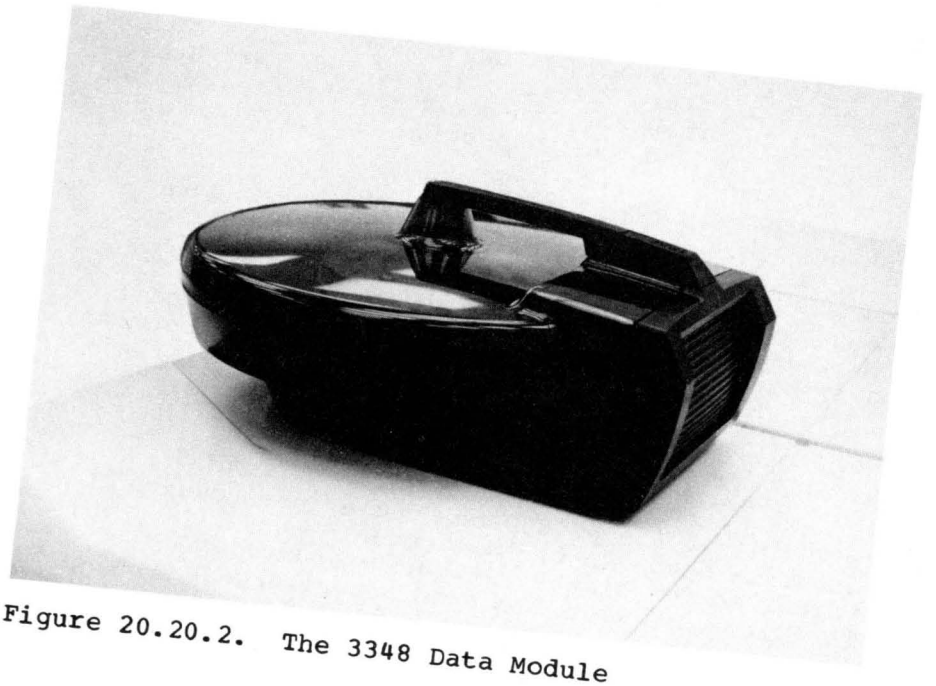


Figure 20.20.2. The 3348 Data Module

A purchased 3348 Model 35 can be upgraded to a Model 70 at the plant of manufacture. The upgrading of a 3348 Model 35 or 70 to a Model 70F and the alteration of a Model 70 to a Model 35 are not available as data module conversions.

The 3348 Model 70F can operate only on a 3340 drive (Model A2, B2, or B1) that has the optional field-installable Fixed Head feature installed. When installed on a 3340 A2 or B2 unit, the Fixed Head feature is installed on both drives. The presence or absence of this feature in a 3340 drive can be determined by programming at any time by issuing a SENSE command and inspecting the Fixed Head feature bit in the sense bytes read. The Fixed Head feature and the the Two-Channel Switch, Additional feature (for four-channel switching) are mutually exclusive for the same 3340 string.

A Model 70F data module can be mounted on a 3340 drive that does not have the Fixed Head feature installed and made ready without any notification of the error by the hardware. However, the first I/O operation issued to the 3340 drive causes an intervention-required unit check condition and the drive is taken out of ready status. When this situation occurs in an OS/VS environment, a message is given to the operator and the affected job must be canceled in order to recover. To avoid such situations, it is recommended that 3340 units with and without the Fixed Head feature not be mixed within a string. If one 3340 unit has the feature, all should have the feature.

Models 35 and 70 of the 3348 Data Module can be used with any 3340 drive (Model A2, B2, or B1) whether or not it has the Fixed Head feature installed. No indication is given if a Model 35 or 70 is placed in a 3340 drive with the Fixed Head feature. In such cases, the fixed head capability of the drive is not utilized.

The 3340 Direct Access Storage Facility is unlike other IBM direct access storage in that the capacity of an individual 3340 drive is determined by the model of 3348 Data Module mounted on the drive rather than by the model of the drive itself. The capacity of the 3348 Data Module that is mounted on a 3340 drive can be determined by programming at any time by issuing a SENSE command and inspecting the data module size bits in the sense bytes read.

The capability of having two capacity options per drive means the capacity of a 3340 string can be increased by using larger capacity data modules on existing drives as well as by adding drives to the string. A 3340 string can vary in capacity from 70 million bytes (two Model 35 data modules) to a maximum capacity of 560 million bytes (eight Model 70 or 70F data modules) in 35- and/or 70-million-byte increments (assuming full-track records).

Reliability and the Sealed Cartridge Design

The sealed cartridge design of the 3348 Data Module, the advanced design used for the read/write heads in the data module, and improvements in the physical design of the 3340 drive make the 3340 Direct Access Storage Facility more reliable than direct access storage devices with removable disk packs, as explained below. No preventive maintenance is scheduled for a 3340 facility because of its reliability features.

Reliability is improved by the removal of head-to-disk alignment problems. Each read/write head within a 3348 Data Module is dedicated to certain tracks on one data surface. Therefore, each head reads only the data it wrote previously, regardless of the 3340 drive that is used. Since common head alignment across all 3340 drives is not required, the critical alignment tolerances that are normally necessary to achieve

data interchangeability among drives are not needed for 3348 Data Modules. It is the less critical alignment tolerances for the read/write heads in a 3348 Data Module that minimize the chance of errors caused by incorrect alignment of a head to its dedicated tracks.

There is also less chance of damaging read/write heads. If a data module is dropped, the only read/write heads that can be affected are those in that data module. If a disk pack is damaged, it can cause damage to the read/write heads in more than one drive if it is moved from drive to drive in an attempt to find a drive that can read the pack. The outside covers of a 3348 Data Module are made of a highly durable material that is designed to enable a data module to withstand more severe blows without damage than can a disk pack.

Reliability is improved because the exposure of the disk surfaces in a 3348 Data Module to outside contamination is greatly reduced when compared to the contamination exposure of a disk pack. A 3348 Data Module is opened only when it is mounted on a 3340 drive and only when the drive cover is closed. Contamination on disk surfaces can be a major cause of head and disk damage.

In addition, the possibility of head crashes is minimized by the improved flying characteristics of the read/write heads in a data module. The low mass of the read/write heads and the low loading force used enable the heads to fly over the rotating disks at a very low height. This near contact (or proximity) recording capability of the read/write heads in the 3348 permits a bit with a weaker than normal signal to be read correctly.

The recording density in bits per inch of a track in a 3348 Data Module is approximately 2.5 times greater than the recording density of a track in a 2316 pack (10 percent greater than 3330-series Model 11 density and more than two times greater than 3330-series Model 1 and 2 density). The advanced head design used for the 3348 Data Module enables greater density to be achieved, together with improved reliability.

Reliability of the 3340 Direct Access Storage Facility is also improved because many critical mechanical parts have been eliminated, such as a complex head load/unload mechanism. In other cases, electronic functions have replaced mechanical functions. While the 3340 drive contains more electronics than the 2314, higher density logic cards are used in the 3340, which results in significantly fewer logic cards. (A 3340 drive also contains approximately one-third the number of logic cards as a 3330-series drive.)

The sealed cartridge design implemented in the 3348 Data Module provides several advantages in addition to improved reliability, such as simplified data module loading and unloading. Operations that are required for disk pack loading and unloading (tightening the pack on the spindle, cover removal, cover replacement, untightening the pack for removal) are not required for a 3348 Data Module. In addition, the possibility of hub wear or hub damage as a result of loading and unloading operations is eliminated for a 3348 Data Module.

After the top cover of the 3340 drive to be used is raised, the operator places the data module in the exposed drive shroud recess. After closing the cover, the operator initiates automatic loading of the module by putting the start/stop switch on the operator panel of the drive in the start position. This causes the cover of the drive to be locked, which is indicated by a light on the operator panel, and the data module to be loaded.

The following occurs during data module loading. The shroud containing the seated data module moves to the back of the 3340 drive

where the voice coil motor is located. While the data module is in motion, the data module door in the rear of the 3348 is rolled down. Electrical, mechanical, and filtered air connections between the 3348 Data Module and the 3340 drive are then made through the open data module door. The access mechanism is then unlatched and the disks are brought up to rotational speed. The access mechanism is moved to physical track 0. This entire loading process requires approximately 20 seconds. When the loading process is completed, the ready light on the operator panel is turned on to indicate the 3348 Data Module is ready for processing.

To unload a data module, the operator places the start/stop switch in the stop position. The unloading procedure consists of a reversal of the operations performed during loading. The access mechanism moves to the home position in the data module, where it is latched, disk rotation is stopped, the data module is disconnected from the drive, the data module door is closed, and the data module moves to the front of the drive. The cover-locked indicator light is turned off as soon as the unloading procedure is completed. Unloading requires approximately 20 seconds. The cover of the 3340 drive can be raised as soon as the cover-locked indicator light is turned off and the 3348 Data Module can then be removed.

The possibility of contaminating the disk surfaces of a data module during loading and unloading operations is minimized because the data surfaces are exposed to the air within the closed 3340 drive through the open data module door for only slightly more than one second. Further, as soon as a seal between the 3340 drive and the 3348 Data Module has been made, the filtered air system displaces the air within the data module several times to remove any contaminants that may have entered via the open data module door.

The sealed cartridge also offers two other unique features. First, a read-only function (not available for the 2314) is provided on a data module basis rather than a drive basis (as implemented for 3330-series disk storage). The read-only function is enabled for a 3348 Data Module by turning an inset in the handle of the 3348 (see Figure 20.20.2) to the read-only position before placing the data module in the 3340 drive. This inset causes the read-only switch that is part of each 3340 drive and the read-only indicator on the operator panel to be turned on when the 3348 is loaded in a 3340 drive.

When the read-only function is enabled for a 3348 Data Module and an attempt is made to write on the data module, an interruption occurs and IBM-supplied programming support terminates the program that issued the write. The advantage of this approach is that once the read-only inset in a 3348 Data Module is set to inhibit writing, the data module can be used with any 3340 drive at any time and the operator need not remember to turn on a read-only switch on the drive.

Second, external label handling is improved. An external label can be placed on a 3348 Data Module after it is removed from the 3340 drive. Placing an external label on the top surface of a disk pack instead of on the cover, to avoid mislabeling a disk pack by placing the wrong cover on it, can be done only when the disk pack is mounted on a drive. In addition, since the outside cover is never removed from a data module, the volume identification label on the cover is legible through the front window of the cover of the 3340 drive even when the data module is loaded and being accessed.

Layout of Tracks, Cylinders, and Read/Write Heads in 3348 Data Modules

The layout of physical and logical tracks on a data surface of any model of the 3348 Data Module and the relative position of the

read/write heads for a data surface are shown in Figure 20.20.3. A data surface contains 700 physical tracks with a small space between the first 350 physical tracks and the second 350 physical tracks. There is also unused space after the second group of 350 physical tracks. Two logical tracks, one even-numbered and one odd-numbered, are written on each physical track. A logical track has a maximum capacity of 8368 data bytes (for full-track records).

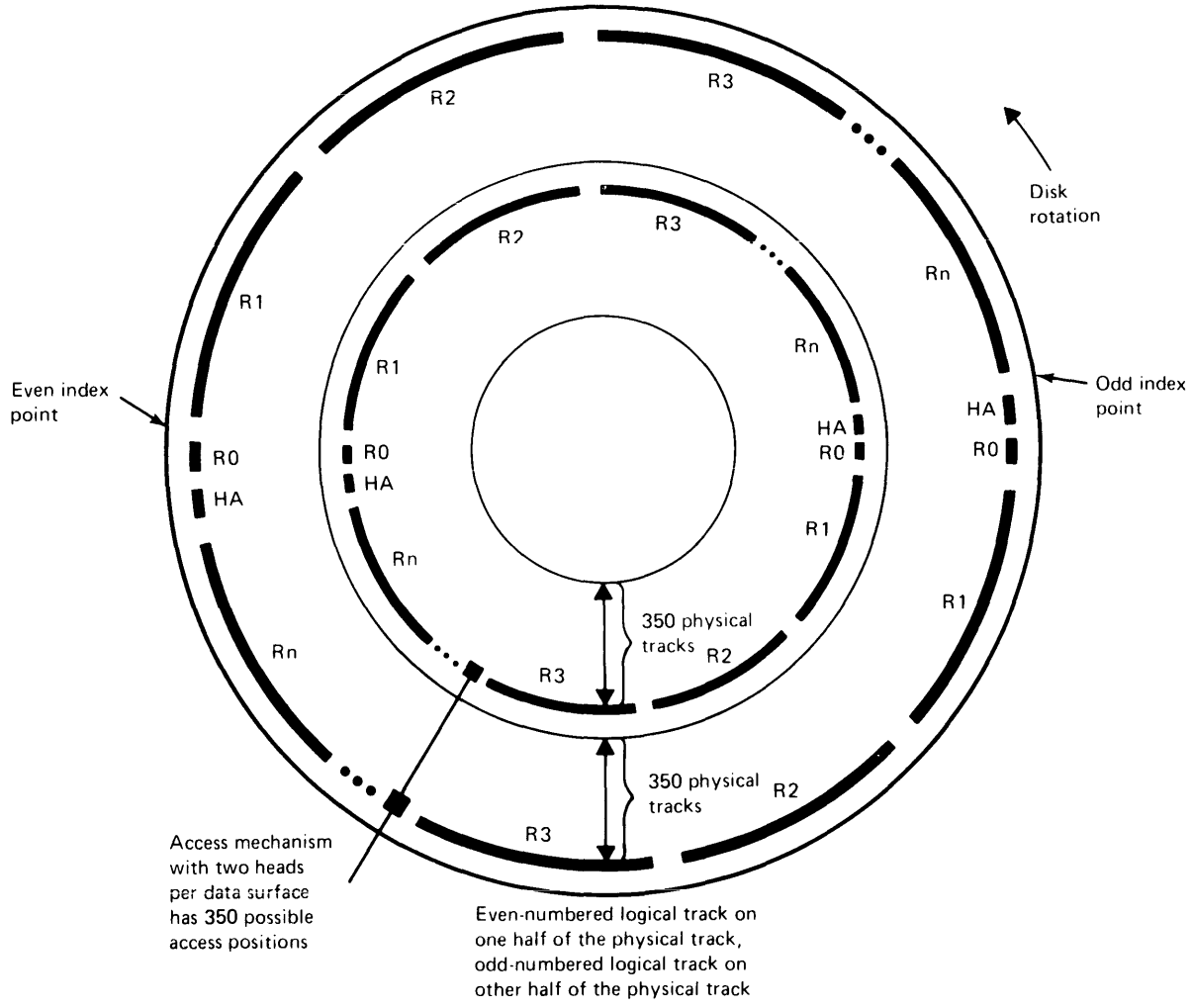


Figure 20.20.3. Location of physical and logical tracks and read/write heads on a data surface in a 3348 Data Module

There are two read/write heads associated with each data surface. They are positioned a little more than 350 physical tracks apart, as shown in Figure 20.20.3. While starting and stopping the data module, the read/write heads are positioned over the unused portions of the data surface.

The access mechanism can be placed at any one of 350 access positions on the data surface. Therefore, an outermost head on the access mechanism can access physical tracks 0 to 349 on its associated data surface while an innermost head can access physical tracks 350 to 699. At any of the 350 possible access mechanism positions, two physical tracks (four logical tracks) can be accessed on a data surface. However, only one read/write head in a data module can be active at a time.

The bottommost surface in all 3348 Data Modules is used as the servo surface. This surface contains information for the servo system that is used to control seek operations, positioning of the heads over tracks, data clocking (the synchronization of data with rotational speed during writing operations), index generation, and signal generation required by the RPS feature. Functionally, the 3340 servo system is like that used in 3330-series drives. However, design improvements, such as elimination of the electromechanical tachometer, have been made.

The required servo information is prerecorded on the servo surface of each 3348 Data Module at the plant of manufacture and is read by a servo read head at the bottom of the access mechanism. The servo information on this surface cannot be read or written using 3340 commands. The servo surface on a 3348 Model 70F Data Module also contains the 60 logical tracks that are read by the fixed heads.

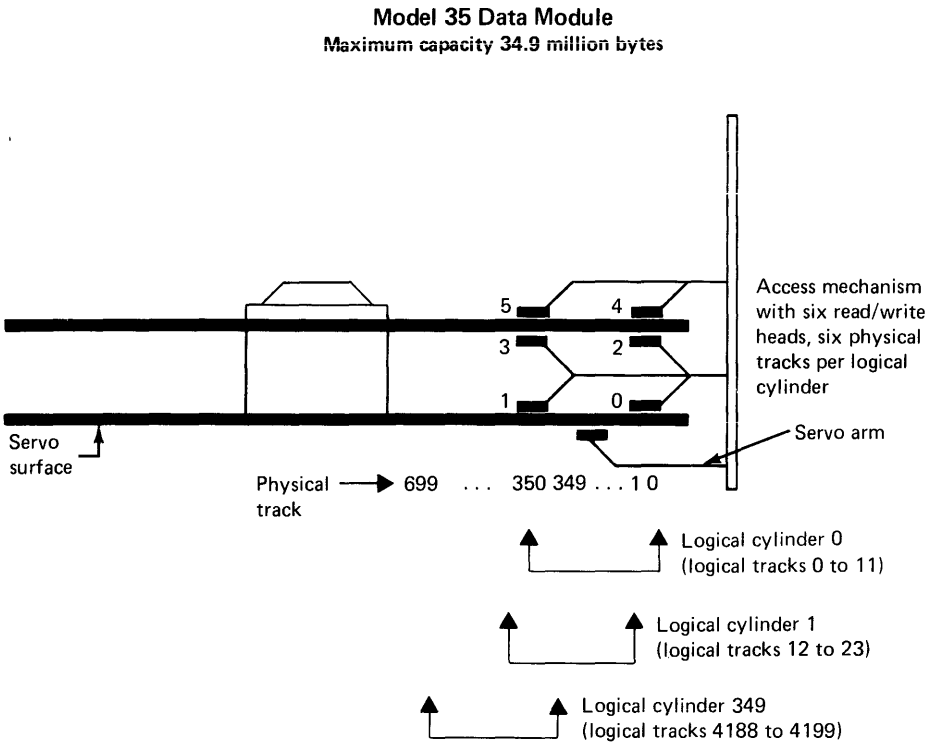
The access mechanism in a 3348 is driven by a voice-coil motor. This motor and the servo system provide fast, precise access mechanism positioning, which minimizes head-settling time.

Figure 20.20.4 shows the layout of cylinders and read/write heads for the 3348 Model 35 Data Module. A Model 35 contains two recording disks. Three of the data surfaces on the two recording disks are used for data recording in a Model 35 Data Module. The three data surfaces are accessed by six read/write heads (0 to 5). The six physical tracks that can be accessed at any given position of the access mechanism constitute a logical cylinder and contain twelve logical tracks. Head 0 accesses logical tracks 0 and 1, head 1 accesses logical tracks 2 and 3, etc.

A four-byte field (CCHH) is used to address the logical tracks in a 3348 Data Module. The two-byte CC (cylinder address) field specifies the logical cylinder address, which can be 0 to 348 for the primary and alternate logical tracks of a Model 35 data module. The two-byte HH field, which normally specifies the actual head address (for 2314 and 3330-series drives, for example), specifies the number of the logical track within the logical cylinder, a value from 0 to 11, instead of a head address of 0 to 5. The drive selects the appropriate head, using the logical track number.

In Figure 20.20.4, the access mechanism is shown positioned at logical cylinder 0, where physical tracks 0 and 350 on each of the three data surfaces can be accessed. There are 350 logical cylinders in the Model 35 data module. The first 348 are used for data, logical cylinder 348 is the alternate cylinder, and logical cylinder 349 is the CE cylinder. The CE cylinder is designed to be used only by the CE for testing the read/write capability of a 3340 drive. It contains a prewritten area for read testing and an area in which write tests can be performed.

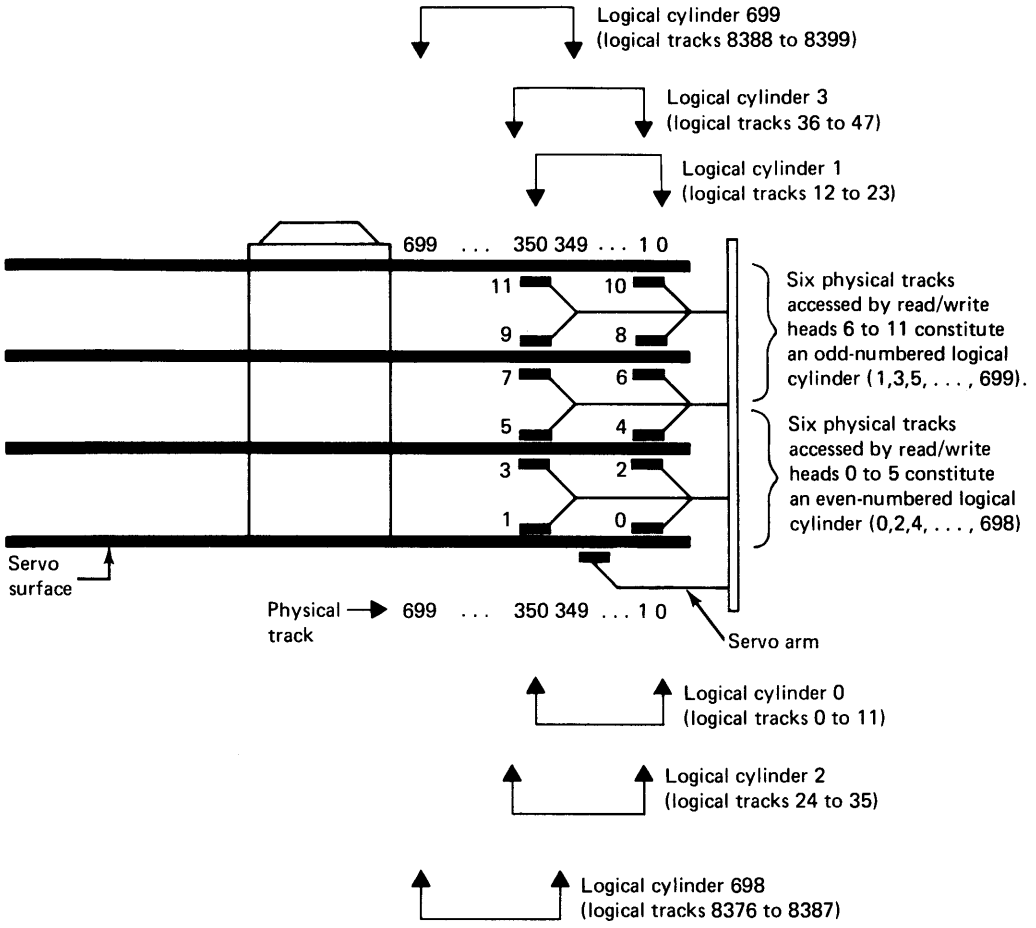
Figure 20.20.5 shows the layout of cylinders and read/write heads for the 3348 Model 70. A Model 70 contains four recording disks. Six data surfaces on the four recording disks, each of which is accessible by two read/write heads, are used for data recording in the Model 70. As for the Model 35, the six physical tracks that can be accessed by the lower six read/write heads (0 to 5) at a given position of the access mechanism constitute a logical cylinder of twelve logical tracks. In a Model 70, however, the logical cylinders addressed by read/write heads 0 to 5 are all even-numbered (0, 2, 4, ..., 698). The six physical tracks that can be accessed by the upper six read/write heads (6 to 11) at a given position of the access mechanism also constitute a logical cylinder of twelve logical tracks. The logical cylinders addressed by read/write heads 6 to 11 are all odd-numbered (1, 3, 5, ..., 699). Thus, on a Model 70 two logical cylinders (24 logical tracks) can be accessed at each of the 350 possible access mechanism positions.



Number of recording disks	2	
Number of data surfaces	3	
Number of read/write heads	6	
Number of physical tracks per physical cylinder	6	
Number of physical tracks per logical cylinder	6	
Number of logical tracks per logical cylinder	12	
Number of logical cylinders per data module	350	
Number of logical tracks per data module	4200	(4176 data) (12 alternate) (12 CE)
Number of access mechanism positions	350	
Number of logical cylinders accessed per access mechanism position	1	

Figure 20.20.4. Cylinder and read/write head layout for a 3348 Model 35 Data Module

Model 70 Data Module
Maximum capacity 69.8 million bytes



Number of recording disks	4	
Number of data surfaces	6	
Number of read/write heads	12	
Number of physical tracks per physical cylinder	12	
Number of physical tracks per logical cylinder	6	
Number of logical tracks per logical cylinder	12	
Number of logical cylinders per data module	700	
Number of logical tracks per data module	8400	(8352 data) (24 alternate) (24 CE)
Number of access mechanism positions	350	
Number of logical cylinders accessed per access mechanism position	2	

Figure 20.20.5. Cylinder and read/write head layout for a 3348 Model 70 Data Module

There are 700 logical cylinders in the Model 70 data module. The first 696 (0-695) are used for data. Logical cylinders 696 and 697 are used as alternate logical cylinders while logical cylinders 698 and 699 are CE cylinders. The method of addressing a logical track in a Model 70 data module is the same as described for a Model 35. The CC value can vary from 0 to 697 for data and alternate logical cylinders while the HH value can vary from 0 to 11.

Figure 20.20.6 shows the layout of cylinders and read/write heads for the 3348 Model 70F. This model is identical to the Model 70 except for the following. Seven surfaces, six data surfaces and the servo surface, on the four recording disks are used for data recording. Logical cylinders 1 to 5 are recorded on the servo surface. They are written on 30 physical tracks that are accessed by 30 fixed read/write elements, which are mounted on a plate under the servo surface, as shown in Figure 20.20.6. The first six physical tracks contain logical cylinder 1, the second six physical tracks contain logical cylinder 2, etc. Logical cylinders 0 and 6 to 699 are recorded on the six data surfaces just as in a Model 70 data module.

Addressing a logical track in a Model 70F data module using a CCHH field is the same as described for the Model 70. When a command is received that addresses a logical track in logical cylinders 1 to 5 of a Model 70F, the 3340 drive automatically selects the fixed read/write element associated with the specified logical track instead of the movable head. Therefore, a Model 70F and a Model 70 data module can be accessed using the same 3340 channel programs. This means no special programming support is required to use a Model 70F instead of a Model 70.

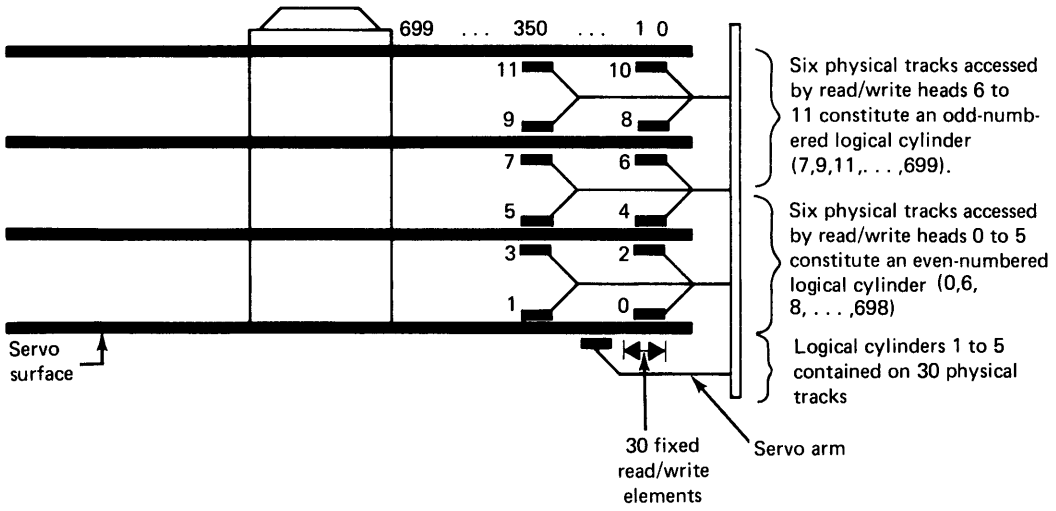
The physical tracks that contain logical cylinders 1 to 5 in a Model 70 are not used in a Model 70F and cannot be accessed by the user or a customer engineer because of the way in which head selection is performed. Hence, the data capacity of Models 70F and 70 is the same. Seek time for logical cylinders 1 to 5 in a Model 70F is zero. Seek times for logical cylinders 0 and 6 to 695 in a Model 70F are the same as Model 70 seek times.

A data set or file can be contained both in logical cylinders 1 to 5 of a Model 70F data module and in logical cylinders that are accessed by movable heads. A 3340 drive, however, can perform only one operation at a time. Therefore, a seek, search, or data transfer operation involving a fixed head in a Model 70F Data Module cannot be performed at the same time a movable head is involved in a seek, search, or data transfer operation.

The best performance gains can be achieved when Model 70F data modules are used by assigning the fixed head logical tracks to small active system data sets (such as the DOS/VS or DOS/VSE page data set, system catalog, TCAM message queue), small active user data sets, large active data sets that can be segmented (OS/VS1 page data set, partitioned data sets, ISAM index levels, for example), and data sets with major activity concentrated at the beginning of the data set (such as the OS/VS job queue).

The assignment of such data sets to the fixed head logical tracks in a Model 70F data module is a user responsibility. DOS/VSE EXTENT and OS/VS DD statements for these data sets must specifically request by actual address locations within the fixed head logical cylinders. Note also that the device type code in the device table that is generated in the control program during a system generation (DOS/VSE PUB table, OS/VS UCB table) does not differentiate between 3340 drives with and without the Fixed Head feature. Therefore, if generic device type assignment by device type (3340) is used in a configuration that contains 3340 drives with and without the Fixed Head feature, either type drive can be selected by the operating system.

Model 70F Data Module
Maximum capacity 69.8 million bytes



Number of recording disks	4	
Number of data surfaces	6	plus servo surface
Number of read/write heads	12	movable
	30	fixed
Number of physical tracks per physical cylinder	12	
Number of physical tracks per logical cylinder	6	
Number of logical tracks per logical cylinder	12	
Number of logical cylinders per data module	700	
Number of logical tracks per data module	8400	(8352 data - 60 fixed head and 8292 movable head) (24 alternate) (24 CE)
Number of movable head access mechanism positions	350	
Number of logical cylinders accessed per access mechanism position	2	except for first 3 positions

Figure 20.20.6. Cylinder and read/write head layout for a 3348 Model 70F Data Module

The assignment of a 3340 drive with the Fixed Head feature can be assured in an OS/VS environment by specifying a user-defined device class name for such 3340 drives at system generation and using this name (instead of UNIT=3340) in the appropriate DD statements. DOS/VSE users utilizing the generic I/O device assignment capability can specify an address list of the 3340 drives with fixed heads in the ASSGN statement for files that are to be located on fixed head tracks in a 3340 drive.

Alternate tracks that are accessed by fixed heads are not provided for logical cylinders 1 to 5 in a Model 70F data module. Logical cylinders 696 and 697, which provide alternate tracks for the logical tracks accessed by the movable heads, also provide alternate tracks for the logical tracks in logical cylinders 1 to 5. This approach is taken because the probability a fixed head track in logical cylinders 1 to 5 will develop a defect is lower than that for movable head tracks and the possibility of a defect occurring in a movable head track is very low (for the reasons discussed later).

The low probability of defects occurring in fixed head logical cylinders 1 to 5 of a Model 70F data module results in part from the fact that these cylinders are recorded on the servo surface, which is a specially manufactured surface because of its primary function. In addition, the fixed head tracks are recorded on the outer edge of the servo surface, which results in a lower bit density for these tracks. The width of a fixed head physical track is six times greater than that of a movable head track on a data surface.

If an uncorrectable error does occur on a fixed head logical track in a Model 70F data module, the logical track should be flagged and an alternate track should be assigned. This can be done using the IEHATLAS, IEHDASDR, or IBCDASDI utility of OS/VS. IEHDASDR or IBCDASDI should then be used to test the flagged fixed head track to determine whether the track is really defective. If the track is found not to be defective, the flag is removed and the assigned alternate track is released. If the track is defective, the data module can be returned to the plant of manufacture for repair if the loss of performance resulting from using an alternate movable head track instead of the fixed head track is not acceptable.

The physical and capacity characteristics of 3348 Data Modules and the 2316 Disk Pack are given in Table 20.20.1. Table 20.20.2 gives the timing characteristics of the 3340 Direct Access Storage Facility and the 2314 facility.

Track Formatting and Data Module Initialization

Self-formatting records consisting of count, key, and data or count and data areas are written on the logical tracks of a 3348 Data Module just as on the tracks of a 2316 pack. However, each home address, count, and key area written on a 3348 track has a six-byte detection code field appended to it for data validity checking by the 3830 Model 2. The detection code used can detect all single-error bursts of eleven bits span or less.

A six-byte correction code field is appended to each data area written on a 3348 track. The correction code used has the same detection capability as the detection code and the capability of correcting single-error bursts of three bits span or less. The actual error correction procedure must be performed by programming (error recovery routines) using corrective bits that are supplied by the control unit as discussed later.

Table 20.20.1. Physical and capacity characteristics of 3348 Data Modules and the 2316 Disk Pack

Characteristic	3348 Model 35	3348 Model 70	3348 Model 70F	2316
Number of data disks per data module/pack	2	4	4	11
Disk diameter in centimeters (inches)	35.6(14)	35.6(14)	35.6(14)	35.6(14)
Number of surfaces used per data module/pack	3 data 1 servo	6 data 1 servo	6 data 1 servo and data	20 data
Number of read/write heads per recording surface	2	2	2 plus 30 read/ write elements for the servo surface	1
Number of cylinders per data module/pack	348 plus 1 alternate and 1 CE	696 plus 2 alternates and 2 CE	696 plus 2 alternates and 2 CE	200 plus 3 alternates
Number of logical tracks per cylinder	12	12	12	20
Number of data tracks recorded per data module/pack	4176	8352	8352	4000
Full-track capacity in bytes	8368	8368	8368	7294
Cylinder capacity in bytes	100,416	100,416	100,416	145,880
Maximum capacity in bytes per data module/pack	34,947,768	69,889,536	69,889,536 (502,080 in logical cylinders 1 to 5, 69,387,456 in logical cylinders 0 and 6 to 695)	29,176,000
Data module/pack weight in kilograms (pounds)	7.7(17)	8.8(19.5)	9(20)	6.8(15)

Table 20.20.2. Timing characteristics of 3340 and 2314 disk storage

Characteristic	Models 35 and 70	Model 70F		2314
		Cylinders 1-5	Cylinders 0, 6-699	
Seek time (ms)				
Maximum	50 (350 cyl-Model 35) (700 cyl-Model 70)	0	50 (700 cylinders)	130
Average	25 (350 cyl-Model 35) (700 cyl-Model 70)	0	25 (700 cylinders)	60
Cylinder to cylinder				
Model 35	10			25
Models 70, 70F	Even to next odd - 0	0	0	
	Even to next even - 10	0	10	
	Odd to next even or odd - 10	0	10	
Rotation time (ms)	20.2	20.2	20.2	25
Rotation speed (rpm)	2964	2964	2964	2400
Data transfer rate (KB/sec)	885	885	885	312
Sectors per track	64	64	64	--
Sector time (microseconds)	316	316	316	--
Load time (secs) (time to ready status after mounting)	20	20	20	60
Unload time (secs)	20	20	20	15

The home address and count areas written on a logical track in a 3348 contain two new fields in addition to the same fields as are written in home address and count areas on 2316 tracks. The home address and each count area on a 3348 logical track contain a two-byte skip-defect field and a two-byte physical address field in front of the flag byte. The automatic surface defect skipping capability of the 3340 allows valid data to be written before and after a surface defect on a logical track. The skip-defect bytes are used to indicate the location of the center of the surface defect relative to the index point of the logical track. Bits in the flag byte field indicate whether the surface defect is located in the next count, key, or data area.

Surface defect skipping is implemented by including in each logical track of a 3348 Data Module a reserved area called a surface defect gap in which no data is written. If a logical track has no surface defects, the surface defect gap is located at the end of the logical track. If there is a surface defect, the surface defect gap is placed over the

defective portion of the logical track at the time of manufacture. One or more surface defects that together occupy an area of up to 16 bytes in length per logical track can be handled by the defect skipping technique while the stated full logical track capacity of 8368 bytes is maintained.

The error detection and correction code capabilities of the 3340 facility permit successful recovery from an error within the data portion of a physical record even when it contains a surface defect gap.

Partial initialization of all 3348 Data Modules is performed at the plant of manufacture. A home address record and track descriptor (R0) record are written on each logical track in the data module. If a single skippable defect is found during the analysis of the surface of a logical track, the appropriate SD bytes and flag byte are written in the home address to indicate this fact. If no surface defect is found, the SD bytes are written as zeros.

The SD bytes and flag byte are supplied in the count area field in virtual storage only for a WRITE HOME ADDRESS command. When R0 is written during data module initialization and thereafter whenever a formatting write is performed, the SD and flag bytes for the count area to be written on disk are supplied by the control unit, which reads them from the record immediately preceding the record to be written.

When a record is written with a formatting write command on the portion of a logical track that contains an identified surface defect, the defect gap area is maintained in the defective portion of the logical track and data is written before and after the defect gap as appropriate. Whenever a nonformatting write or a read is issued for this record, the surface defect gap is automatically skipped over by the hardware without programming assistance or any error notification, just as if no surface defect existed.

The OS/VS IBCDASDI, IEHDASDR, or IEHATLAS utilities can be used to assign an alternate track if a physical track becomes defective during its use in an installation. If data cannot be read from a 3348 Data Module and recovery of this data is critical, the data module can be returned to the plant of manufacture where recovery will be attempted.

The two physical address bytes in home address and count areas on a 3348 logical track contain the physical cylinder and track address of the logical track on which they are written. When a seek command is issued, the control unit converts the logical cylinder and track address specified by the seek command to a physical cylinder and track address that is actually used by the drive in the seek operation. This physical address is saved in the control unit for later use in seek verification.

The physical address bytes are automatically written and read by the control unit and are not processed by programming. That is, when a home address or count area is written, the physical address bytes are automatically supplied by the control unit and are not contained in the home address or count area field in virtual storage that is indicated by the write command. Similarly, when a home address or count area is read, the control unit reads the physical address bytes but they are not placed in the home address or count field area in virtual storage.

The physical address bytes are used by the control unit for seek verification during normal operations and by the 3340 microdiagnostic routines. When a home address or count area is processed during a read, search, or clock operation, the physical address bytes read are compared with the most recent seek address (physical cylinder and track address) that was saved in the control unit when the last seek command was issued. If the two physical addresses are not equal, the command is

terminated and a unit check condition results. Seek check is indicated in the sense bytes.

ATTACHMENT VIA 3830 STORAGE CONTROL MODEL 2

The 3830 Storage Control Model 2 unit contains the control functions required to operate one or two 3340 strings of from two to eight drives each. If the 32 Drive Expansion and Control Store Extension optional features are installed on a 3830 Model 2, up to four 3340 strings of from two to eight drives each can be attached to it. These two features are field-installable. A maximum of two of the 3340 strings attached to a 3830 Model 2 can contain 3344 units.

Cabling between the 3830 Model 2 and the 3340 Model A2 can be a maximum of 45.7 meters (150 feet) in length. The 3830 Model 2 attaches to a block multiplexer channel in the 4341 Processor via cabling up to 45.7 meters (150 feet) in length. Intermixing 3340 and 3330-series strings on an attachment is discussed later in this subsection.

Standard features of the 3830 Model 2 when used with 3340 disk storage are record overflow, multiple requesting, and rotational position sensing. The command retry facility of the 3830 Model 2 that is implemented for 3330-series drives is not implemented for 3340 drives. When multiple requesting is used, the 3830 Model 2 can control concurrent operation of up to 32 channel programs (when 32 Drive Expansion is installed), one on each of its drives. Only one of the 2 to 32 drives attached to a 3830 Model 2 can be transferring data at a time.

Rotational position sensing is an optional field-installable feature for 3340 units. It must be installed on each unit (both drives in an A2 or B2 3340 unit) that is to use the standard rotational position sensing capability of the 3830 Model 2. For performance reasons (see Section 60) it is recommended that the RPS feature be installed on all of the 3340 units in a given string or on none of the units in the string. The presence or absence of the RPS feature in a 3340 drive can be determined by programming at any time by issuing a SENSE command and inspecting the RPS feature bit in the sense bytes read.

If a SET SECTOR command is issued to a 3340 drive that does not have the RPS feature installed, no operation is performed, track orientation is lost, and channel end and device end status are presented. If a READ SECTOR command is issued to a 3340 drive without RPS installed, a sector value of zero is returned together with channel end and device end status. Thus, channel programs containing sector commands can operate on 3340 drives that do not have RPS installed.

The 3830 Model 2 supports all the 2314 commands (except the file scan commands) in addition to new commands not available for the 2314, such as RPS and diagnostic commands. The command set for the 3340 is the same as that for 3330-series disk storage.

The Two-Channel Switch feature, identical in function to the same feature for the 2314 facility, can be installed on a 3830 Model 2 to allow it to be attached to two channels. The Two-Channel Switch, Additional feature can be added to this configuration to permit the 3830 Model 2 to be attached to four channels. A maximum of two of the four channels can be present in the same system. The channels to which a 3830 Model 2 with one or both of these features is connected must each have one control unit position and, if block multiplexing is to be used, eight nonshared subchannels available. An enable/disable switch on the 3830 Model 2 can be set to dedicate the 3830 to any subset of the two to four channels.

The optional String Switch feature can be installed on 3340 Model A2 drives. This field-installable feature enables the 3340 Model A2 and

its attached Model B2 and B1 units to be connected to two control-unit-type attachments instead of only one. The attachments can be any two of the following:

- 3830 Storage Control Model 2 attached to a block multiplexer channel in a 4341 Processor or System/370 processor (Models 115 and up)
- 3880 Storage Control attached to a block multiplexer channel in a 4341 Processor or Sytem/370 processor (Models 145 and up)
- Integrated Storage Control for the System/370 Model 145 or 148
- 3345 Storage and Control Frame Models 3, 4, and 5 for the System/370 Model 145 Model 0
- Integrated Storage Controls for System/370 Models 158 and 168 (or the two logical controls in one ISC)
- 3330/3340-series IFA for the System/370 Model 135 or 138
- Direct Disk Attachment of a System/370 Model 115 Model 2 or Model 125 Model 2

Except for the Direct Disk Attachment, the two attachments to which a 3340 Model A2 with the String Switch feature is connected can be attached to the same or different channels in the same processor, or to channels in two different processors. In addition, channel-switching features can be installed on one or both of the attachments (except for the Direct Disk Attachment).

For Model 2 of Models 115 and 125, the String Switch enables two strings of 3340 drives to be attached to the 4341 Processor or any System/370 processor (except a Model 115 Model 0 or Model 125 Model 0) and a Model 115 Model 2 or Model 125 Model 2.

The String Switch feature for 3340 disk storage is functionally similar in its operation to the Two-Channel Switch. A switch on the 3340 Model A2 can be set to allow the 3340 string to be accessed via both attachments, one at a time. In effect, this setting provides two control unit paths to the string. Switching is accomplished dynamically under program control. Alternatively, the switch can be set to dedicate the string to one attachment or the other so that the string can be accessed only via that attachment.

Figure 20.20.7 illustrates string switching for two 3340 strings attached to a 3830 Model 2 unit. In the configuration shown, both strings can be accessed via two channels and two control units. Channel switching, string switching, and 32 Drive Expansion features can be used to enhance the availability of 3340 Direct Access Storage Facilities and to extend backup capabilities when two System/370 systems (the same or different models) are present in an installation.

The 3830 Model 2 control unit is microcode controlled. Read/write monolithic storage contained in the control unit is used for microcode residence. The 3830 Model 2 also contains a diskette device that reads interchangeable disk cartridges. This device is used for microcode backup storage and for storage of nonresident diagnostics for the 3340 string. During a 3830 Model 2 power-on sequence, the functional microcode is loaded from the device into control storage within the 3830 Model 2 control unit. Therefore, microcode engineering changes can be installed merely by replacing the current disk cartridge with another that contains the new microcode.

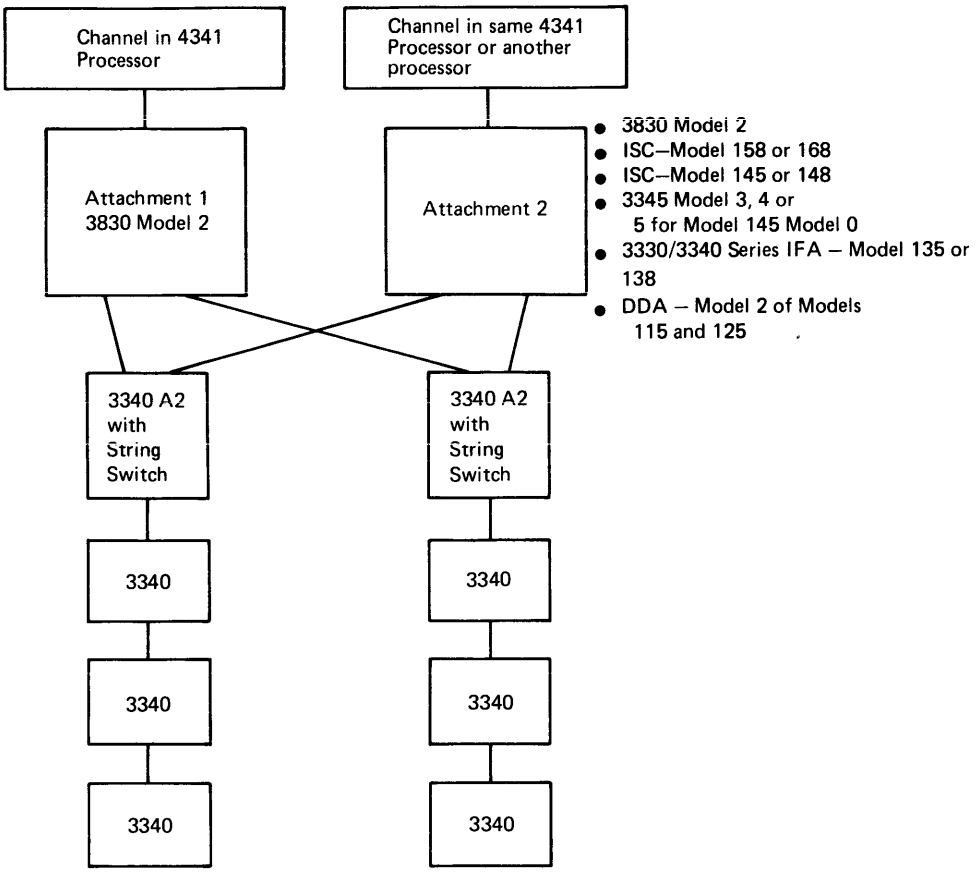


Figure 20.20.7. String switching for 3340 facilities attached to a 3830 Model 2

The 3830 Model 2 incorporates error detection, correction, and logging features that are designed to improve its availability and serviceability. For the 3340, the 3830 Model 2 provides the following facilities that are not implemented in System/360 direct access devices:

- I/O error routine correction of recoverable data errors on read operations with data supplied by the control unit in sense bytes. When the 3830 Model 2 detects a correctable data error during the reading of the data portion of a physical record, it generates the information necessary to correct the erroneous bytes. The sense bytes presented by the 3830 Model 2 contain a pattern of corrective bits and a displacement value to indicate which of the bytes transferred to processor storage contain the errors. The disk error recovery program need only EXCLUSIVE OR (logical operation) the corrective bit pattern with the error bytes in the input area in processor storage to correct the errors.

- Statistical usage recording by the 3830 Model 2. Statistical usage counters for each drive in a 3340 string are continuously maintained by the 3830 Model 2. These counters indicate the number of bytes read/searched, number of seeks issued, and number of command and data overruns for each device. When a counter reaches its threshold or a data module is removed from a drive, the 3830 Model 2 indicates the condition via a unit check when the next I/O operation is initiated to the drive or a data module is made ready on the drive.

Counter data can be obtained and counters can be reset by issuing a READ AND RESET BUFFERED LOG command.

- Inline diagnostic testing of a malfunctioning drive. (Inline diagnostics are provided only for 2314 facilities.) A 3830 Model 2 control unit can execute diagnostic tests on a malfunctioning drive while normal operations take place on the remaining drives in the string. Diagnostic tests can be loaded into a transient area of the control storage of the 3830 Model 2 and executed on the malfunctioning drive. This can be done in an online environment using OLTEP or the CE panel on the 3830 Model 2. OLTSEP can be used in a standalone environment. This inline testing allows CE diagnosis and repair of most 3340 drive failures without the necessity of taking the entire 3340 string out of the system configuration.

A 3340 drive can be placed in CE mode (offline to the system) by means of a switch that is located inside the rear door of the drive so that maintenance functions can be performed. To take the 3340 drive out of CE mode and return it to online status, the attention pushbutton must be pressed. This also causes the access mechanism to move to physical track 0.

The 3830 Model 2 can have a mixture of 3340, 3330-series, and 3350 strings attached. Each string must contain all 3340 (but no 3344) drives, all 3330-series, or all 3350 drives as usual. A 3340 string that contains 3344 units cannot be intermixed with 3330-series or 3350 strings on an attachment.

SUMMARY

The hardware features of the 3340 and 2314 Direct Access Storage Facilities are summarized in Table 20.20.3. Table 20.20.4 compares the capabilities of the 3830 Model 1 and 3830 Model 2 for 3340, 3344, 3330-series, and 3350 disk storage.

When compared with the 2314 facility, the 3340 facility offers the following major advantages:

- Faster access to data
  - Data transfer rate almost three times that of the 2314
  - Seek times approximately 40% those of the 2314 for movable head accesses
  - Zero seek time provided by the fixed heads in a 3348 Model 70F Data Module
  - Rotational delay interval approximately 20% shorter than for the 2314

- Larger capacity per drive
  - 17% for the Model 35 data module
  - 175% for Model 70 and 70F data modules

- Two capacity options per drive for expanded growth flexibility

- Multiple requesting and rotational position sensing capabilities for use with block multiplexer channels

- Operational improvements
  - Cover tightening/untightening and removal/replacement operations are eliminated, speeding up data module loading and unloading.
  - Load time to ready status for a mounted data module is three times faster.
  - Write protection is provided on a data module basis.

External labeling procedures are more flexible and leave less chance of erroneous data module labeling.

- Significantly increased reliability
  - Sealed cartridge design eliminates head-to-disk alignment problems, minimizes the possibility of disk surface contamination, and eliminates hub wear and damage.
  - Advanced head design makes head crashes a remote possibility and permits increased recording density without any loss of reliability.

- Improved error handling capabilities
  - Error correction data is provided by the hardware for use by programmed error recovery procedures.
  - Surface defect skipping reduces the need to use the error correction capability.

- Improved availability and serviceability
  - No preventive maintenance is scheduled, because of the reliability features of the 3340 and 3348.
  - Faster error isolation and correction is possible because the 3340 contains fewer circuit cards.
  - Expanded microdiagnostics can test more than 95% of the circuits in a 3340.

Table 20.20.3. Summary of the hardware features of 3340 and 2314 disk storage facilities

Feature	3340 attached to 3830 Model 2	2314 (A-Series)
Number of drives per string or facility	Two to eight in one-drive increments	One to eight in one-drive increments. (A ninth can be included as a spare only.)
Number of strings or facilities per control unit	One to four	one maximum
Data medium used	Removable interchangeable data module (sealed cartridge)	Removable interchangeable disk pack
Read-only feature on drive or data medium	Yes, on data module	No
Removable address plugs on drive	No	Yes
Attachment of a string or facility to two control units in the same or a different processor	Yes, via optional string switch feature. Only one data transfer operation permitted per string.	Yes, via 2844 Auxiliary Storage Control. Two concurrent data transfer operations per facility permitted.
Two-Channel Switch	Optional	Optional

Table 20.20.3 (continued)

Feature	3340 attached to 3830 Model 2	2314 (A-Series)
Attachment of the control unit to four channels	Yes, using the optional Two-Channel Switch and Two-Channel Switch, Additional features	Yes, using the optional Two-Channel Switch and 2844 Auxiliary Storage Control
Record Overflow	Standard	Standard
File Scan	Not available	Standard
Multiple track operations	Standard	Standard
Multiple requesting	Standard	Not available
Rotational Position Sensing	Optional (on 3340 drives)	Not available
Error correction data presented by control unit	Yes	No
Surface defect skipping	Yes	No
Writable storage in control unit loaded from a disk cartridge	Yes	No
Statistics logging by the control unit in its storage	Yes	No
Inline diagnostics executed under OLTEP or via the CE panel	Yes	Yes

Table 20.20.4. Summary of the features of 3830 Storage Control Models 1 and 2

Characteristic	3830 Model 1	3830 Model 2
Power source	Contains own for itself and all the drives that can be attached to it	Contains own for itself only
Attaches to	Block multiplexer channel	Block multiplexer channel
Devices attaching to it	3330 Models 1 and 2	3333 Models 1 and 11 (optionally with 3330 Model 1, 2, and 11 units attached) 3340 Model A2 (optionally with 3340 Model B1 and B2 and/or 3344 B2 and B2F units attached) 3350 Model A2 or A2F (optionally with 3350 Model B2, B2F, C2, and C2F units attached)
Number of drives in a string	1 to 8	2 to 8
Standard number of strings attachable	One maximum	Two maximum
32 Drive Expansion feature for attachment of two additional strings	Not available	Optional for a maximum of four strings
Intermixing string types on an attachment	No	Yes. 3330-series 3340, and 3350 strings can be attached to one unit. A 3340/3344 string cannot be intermixed with other string types.
Two-Channel Switch	Optional	Optional
Two-Channel Switch, Additional (for four-channel switching)	Optional	Optional

Table 20.20.4 (continued)

Characteristic	3830 Model 1	3830 Model 2
String switching capability	Not available	Yes, for 3330-series strings via optional 3333 String Switch feature. Yes, for 3340, 3340/3344, and 3350 via optional String Switch Feature.
Multiple requesting	Standard	Standard
Rotational Position Sensing	Standard	Standard on control unit (standard on 3330-series, 3344, and 3350 drives, optional on 3340 drives)
Multiple track operations	Standard	Standard
Record Overflow	Standard	Standard
Command retry	Standard	Standard for 3330-series and 3350 strings. Not available for 3340 and 3340/3344 strings.
Surface defect skipping	Not implemented	Implemented for 3340, 3340/3344, and 3350 strings. Not implemented for 3330-series strings.
Inline diagnostic tests	Standard	Standard
Error logging by control unit	Standard	Standard

20:25 3344 DIRECT ACCESS STORAGE

MODELS, FEATURES, AND STRING CONFIGURATIONS

Models

The 3344 Direct Access Storage is large capacity, modular, fixed-media disk storage that is available in Models B2 and B2F. The Model B2F differs from the Model B2 in that it contains some fixed head storage while the Model B2 contains only movable head storage. A 3344 unit consists of two drives and attaches only to 3340 Model A2 Disk Storage and Control to form a 3340/3344 string. The 3340 Model A2 provides control functions not present in the 3344 unit.

The 3344 does not have a storage medium that is removable by the operator. The read/write head mechanism and recording disks of the 3344 are assembled as units and fixed in the 3344 drive. The head/disk assembly can be removed only by the customer engineer if it is necessary to send the assembly to the plant of manufacture for repair or data retrieval.

A 3340/3344 string consists of up to four 3340 and 3344 units. Movable head and fixed head models can be mixed in a string. A 3340 Model A2, which must be the first unit in a 3340/3344 string, can have attached a mixture of 3344 B2/B2F and 3340 B2/B1 units up to a maximum of three units in the string. If a Model B1 unit is included in a 3340/3344 string, it must be the last unit in the string.

A 3340/3344 string can be attached to a block multiplexer channel in the 4341 Processor via the 3830 Model 2 or 3880 Storage Control. When a 3340/3344 string is attached to the 3830 Model 2 or 3880, 3330-series and 3350 strings cannot also be attached to the same 3830 Model 2 unit or storage director in a 3880, respectively. Up to two 3340/3344 strings and two 3340 strings can be attached to a 3830 Model 2 or 3880 storage director (see maximum string configuration discussion below).

A 3344 drive contains four logical volumes, each of which has the same maximum capacity (for full-track records) as a 3348 Model 70 Data Module (approximately 70 million bytes). Thus, the maximum capacity of a 3344 drive is 279.5 million bytes. The maximum capacity of a 3340/3344 string is more than 1.8 billion bytes.

Each logical volume in a 3344 drive is addressed separately. Therefore, eight logical device addresses are used for a 3344 unit (four per drive). An operation can be performed only on one logical volume in a 3344 drive at a time. When a channel program is active on one logical volume, a busy indication results if an attempt is made to start an I/O operation on any of the other three logical volumes in the 3344 drive.

The timing characteristics of a 3344 drive are the same as those for a 3340 drive: 885 KB/sec data transfer rate, rotation time of 20.2 milliseconds, maximum seek time of 50 milliseconds, average seek time of 25 milliseconds, and physical cylinder to physical cylinder seek time of 10 milliseconds. Note that the maximum and average seek times apply to the entire four logical volumes in a 3344. Therefore, when activity is confined to one of the logical volumes, average and maximum seek times are less than 25 and 50 milliseconds.

A 3344 Model B2F drive has approximately 1 million bytes that are accessed by fixed read/write heads for which seek time is zero. The fixed head storage in a Model B2F is associated with the first of the four logical volumes in the 3344 drive. Seek times for the movable head storage in the Model B2F are the same as for the Model B2.

A 3344 Model B2 can be field converted to a 3344 Model B2F. No other model changes involving 3344 units (3340 to 3344, for example) are possible.

An operator panel is located on the top of each 3344 drive. This panel contains a start/stop switch to start and stop rotation of the disks and a switch to provide read-only protection for the disks. When the latter switch is in the read position, no write or erase commands can be executed on the drive. When the switch is in the read/write position, all commands can be executed.

Features

Rotational Position Sensing and Record Overflow are standard on all 3344 drives, which have no optional features. It is recommended that RPS be installed on the 3340 drives in a 3340/3344 string for performance reasons. A 3830 Model 2 to which a 3340/3344 string is attached must have the Control Store Extension and Register Expansion features installed.

String switching can be installed on a 3340 Model A2 to which 3344 units are attached to enable the 3340/3344 string to be accessed by two control-unit-type attachments, as for 3340 strings. In addition, a 3830 Model 2 or 3880 to which 3340/3344 strings are attached can have channel switching features installed (Two-Channel Switch and Two-Channel Switch, Additional for the 3830 Model 2 and Two-Channel Switch Pair and Two-Channel Switch Pair, Additional for the 3880).

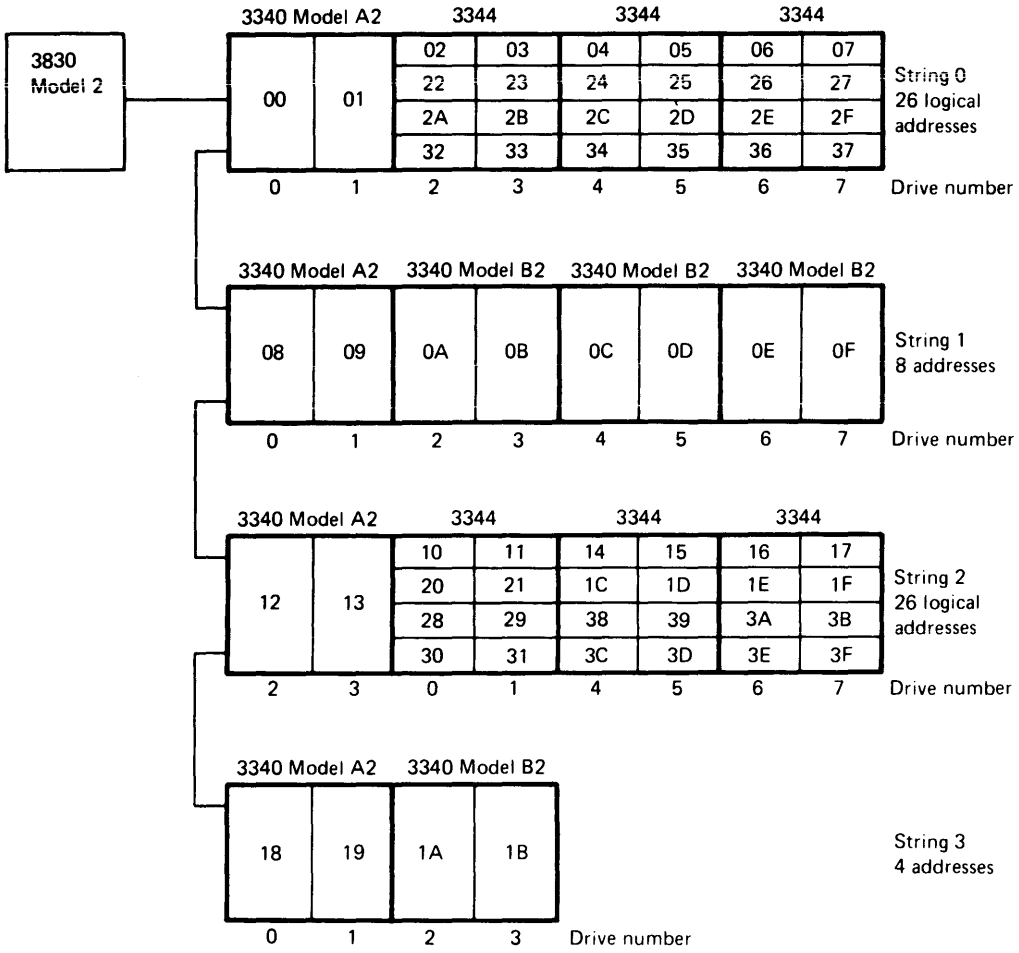
Maximum String Configuration

The maximum string configuration that can be attached to a 3830 Model 2 when 3340/3344 strings are included in the configuration is shown in Figure 20.25.1. Two full 3340/3344 strings (each with 26 logical device addresses), one full 3340 string, and one short (four-drive) 3340 string can be attached to the 3830 Model 2 or a 3880 storage director for a maximum of 28 drives with 64 logical device addresses.

The limit of two 3340/3344 strings per control function results from the addressing scheme used for a 3344 drive to support up to 64 logical device addresses. Bits 2 through 7 of byte 3, which is the I/O device address byte in a START I/O instruction, specify the logical device address (0 to 63). Bits 0 and 1 of byte 3 specify the storage control address. This permits 3340/3344 strings to be attached only to strings 0 and 2, as shown in Figure 20.25.1. The addresses shown in the figure must be used because of the special characteristics of the address transformation algorithm.

TRACK, CYLINDER, AND READ/WRITE HEAD LAYOUT

The layout of physical and logical tracks on a data surface in a 3344 and the relative position of the read/write heads for a surface are shown in Figure 20.25.2. A data surface contains 1122 physical tracks. There are two read/write heads associated with each data surface. They are positioned 561 physical tracks apart, as shown in Figure 20.25.2. Two logical tracks, one even-numbered and one odd-numbered, are written on each physical track. A logical track in a 3344 drive has a maximum capacity of 8368 data bytes (for full-track records), as in a 3348 Data Module.



To obtain the permissible hexadecimal device address(es) for a 3340/3344 drive, add the one address (3340) or four addresses (3344) shown for the drive to the control unit address X00, X40, X80, or XC0, where X is the channel number. For example, if the control unit address 140 is chosen, the four permissible addresses for 3344 drive 5 on string 0 are 145, 165, 16D, and 175.

Figure 20.25.1. Maximum 3830 Model 2 string configuration for 3340/3344 strings

The access mechanism can be placed at any one of 561 access positions on the data surface. Therefore, an outermost head on the access mechanism can access physical tracks 0 to 560 on its associated data surface while an innermost head can access physical tracks 561 to 1121. At any of the 561 possible access mechanism positions, two physical tracks (four logical tracks) can be accessed on a data surface. However, only one read/write head in a 3344 drive can be active at a time.

Figure 20.25.3 shows the layout of cylinders, read/write heads, and logical volumes in 3344 Direct Access Storage. A 3344 contains eight recording disks. Fifteen of the surfaces are used for data recording. The 15 recording surfaces are accessed by 30 read/write heads addressed 0 to 29.

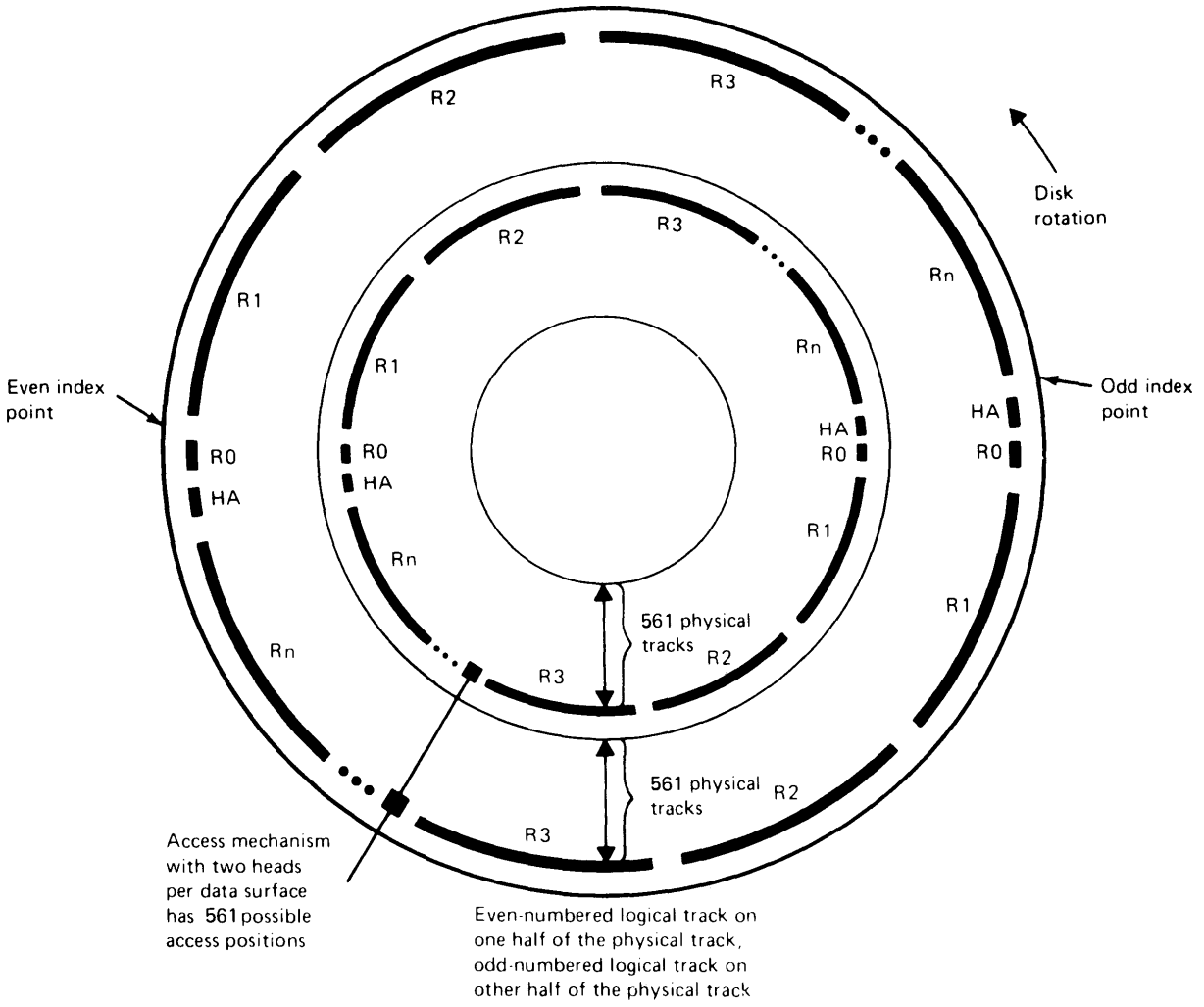
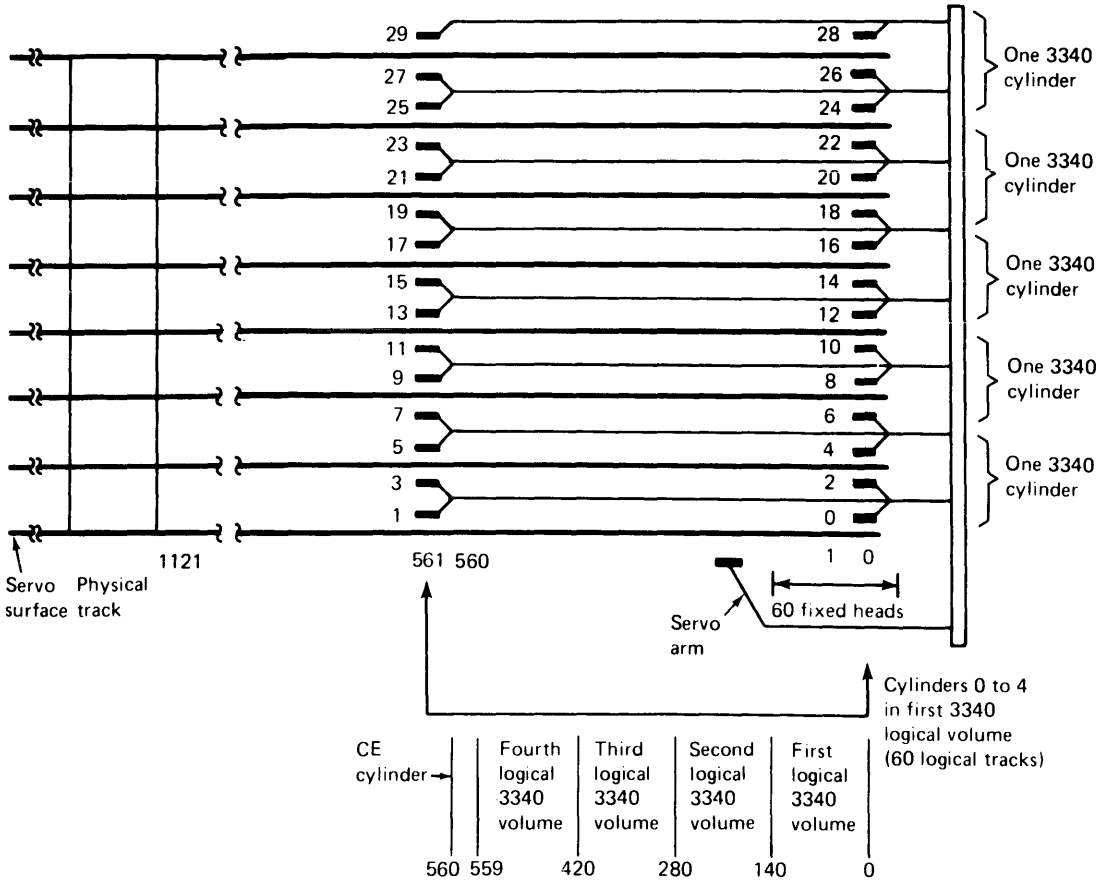


Figure 20.25.2. Location of physical and logical tracks and read/write heads on a data surface in 3344 Direct Access Storage

The bottommost surface in a 3344 is used as the servo surface, as it is in a 3348 Data Module. This surface contains information for the servo system that is used to control seek operations, positioning of the heads over tracks, data clocking (the synchronization of data with rotational speed during writing operations), index generation, and signal generation required by the RPS feature.

The required servo information is prerecorded on the servo surface at the plant of manufacture and is read by a servo read head at the bottom of the access mechanism. The servo information on this surface cannot be read or written using 3344 commands. The servo surface on a 3344 Model B2F also contains the logical tracks that are read by the fixed heads.

The 30 physical tracks that can be accessed at any given position of the access mechanism constitute five logical 3340 cylinders containing 60 logical tracks. Each three surfaces represent one logical 3348 Model 70 cylinder that contains twelve 3340 logical tracks. That is, heads 0 through 5 access six physical tracks that constitute the first 3340 logical cylinder of twelve logical tracks, heads 6 through 11 access the second logical 3340 Model 70 cylinder, etc.



Number of recording disks	8
Number of recording surfaces	15
Number of 3340 logical tracks per physical track	2
Number of movable read/write heads	30
Number of fixed read/write heads (Model B2F only)	60
Number of surfaces per 3340 cylinder (twelve logical tracks)	3
Number of 3340 cylinders per physical cylinder	5
Number of physical cylinders per 3340 logical volume	140
Location of first 3340 logical volume	0 to 139
Location of second 3340 logical volume	140 to 279
Location of third 3340 logical volume	280 to 419
Location of fourth 3340 logical volume	420 to 559

Figure 20.25.3. Cylinder, read/write head, and logical volume layout for 3344 Direct Access Storage

Therefore, the first logical 3348 Model 70 volume in a 3344 is contained in physical cylinders 0 through 139 (5 logical 3340 cylinders per access position times 140 access positions gives 700 logical 3340 cylinders). The second logical volume is contained in 3344 physical cylinders 140 through 279, the third logical volume in 3344 physical cylinders 280 through 419, and the fourth logical volume in 3344 physical cylinders 420 to 559. The physical cylinder at access position 560 is a CE cylinder.

The CE cylinder is designed to be used only by the customer engineer for testing the read/write capability of a 3344 drive. It contains a prewritten area for read testing and an area in which write tests can be performed.

Of the 700 logical cylinders in a logical volume in the 3344, the first 696 (0-695) are used for data. Logical cylinders 696 and 697 are used as alternate logical cylinders while logical cylinders 698 and 699, which are CE cylinders in a 3348 Model 70 Data Module, are unused in a logical volume in the 3344, since physical cylinder 560 is used as the CE cylinder.

The method of addressing a logical track in a logical volume in a 3344 is the same as for a 3348 Model 70 Data Module. That is, a four-byte CCHH field is used in seek commands, where the CC value can vary from 0 to 697 and the HH value from 0 to 11. Once the physical cylinder address is determined from the logical cylinder specified in the seek command, an offset of a multiple of 140 is added to the physical address depending on the logical volume being addressed (0 for logical volume 1, 140 for logical volume 2, etc.) in the START I/O instruction.

Note that the cylinder-to-cylinder seek time of 10 milliseconds applies to the physical cylinders within a 3344. Thus, cylinder-to-cylinder seek time for the four logical 3340 cylinders after the first in a physical cylinder is zero.

In a 3344 Model B2F, 60 fixed read/write heads are contained on a plate under the servo surface, as shown in Figure 20.25.3. Logical cylinders 1 through 10 of the first logical volume only are recorded on the servo surface and accessed by the 60 fixed heads. The first six physical tracks contain logical cylinder 1, the second six physical tracks contain logical cylinder 2, etc. Logical cylinders 0, and 11 through 697, of the first logical volume are recorded on physical cylinders 0 and 2 through 139, just as in a 3344 Model B2. The fixed head cylinders can contain a maximum of 1,004,160 bytes.

Addressing a fixed head logical track in the first logical volume in a 3344 Model B2F using a CCHH field is the same as described for the 3344 Model B2. When a command is received that addresses a logical track in logical cylinders 1 to 10 of the first logical volume in a Model B2F, the 3344 drive automatically selects the fixed read/write element associated with the specified logical track instead of the movable head. Therefore, all logical volumes in a 3344 Model B2 can be accessed using the same 3344 channel programs. This means no special programming support is required to access a 3344 Model B2F instead of a 3344 Model B2.

The physical tracks in access positions 0 and 1 that normally contain logical cylinders 1 through 10 in the first logical volume are not used in a 3344 Model B2F and cannot be accessed by a user or customer engineer because of the way head selection is performed. Thus, the data capacity of the first logical volume in a 3344 Model B2F is the same as that of one of the other logical volumes. Seek time for the logical cylinders accessed by the fixed heads is zero. Seek times for the other cylinders in the first logical volume (0 and 11 to 697) are the same in 3344 Models B2 and B2F.

A data set or file can be contained both in logical cylinders 1 to 10 of the first logical volume of a 3344 Model B2F and logical cylinders that are accessed by movable heads. A 3344 drive, however, can perform only one operation at a time. Therefore, a seek, search, or data transfer operation involving a fixed head in a Model B2F cannot be performed at the same time a movable head is involved in a seek, search, or data transfer operation on any logical volume in the 3344 drive.

The best performance gains can be achieved for 3344 Model B2F drives by assigning the fixed head logical tracks to small active system data sets/files (such as the DOS/VS or DOS/VSE page data set, system catalog, TCAM message queue, VTOC, OS/VS pageable link pack area, JES2 spool file), small active user data sets/files (containing tables, indexes, work areas, data, etc.), large active data sets/files that can be segmented (OS/VS1 page data set, partitioned data sets, ISAM index levels, for example), and data sets/files with major activity concentrated at the beginning of the data set/file (such as the OS/VS job queue).

The assignment of such data sets/files to the fixed head logical tracks in a 3344 Model B2F is a user responsibility. DOS/VS or DOS/VSE EXTENT and OS/VS DD statements for these files and data sets must specifically request, by actual address, locations within the fixed head logical cylinders. Note also that the device type code in the device table that is generated in the control program during a system generation (DOS/VS or DOS/VSE PUB table, OS/VS UCB table) is that for a 3340 and does not differentiate between drives with and without fixed heads. Therefore, if generic device type assignment by device type (3340) is used in a configuration that contains 3344 drives with and without fixed heads, either type drive can be selected by the operating system.

The assignment of a logical 3340 volume with fixed heads within a 3344 drive can be assured in an OS/VS environment by specifying a user-defined device class name for such logical 3340 drives at system generation and using this name (instead of UNIT=3340) in the appropriate DD statements. DOS/VS or DOS/VSE users utilizing the generic I/O device assignment capability can specify an address list of the logical 3340 drives with fixed heads in the ASSGN statements for files that are to be located on fixed head tracks in a 3344 Model B2F drive.

Logical cylinders 696 and 697, which provide alternate tracks for the logical tracks accessed by the movable heads, can also provide alternate tracks for the logical tracks in logical cylinders 1 through 10. This approach is taken because the probability a fixed head track in logical cylinders 1 to 10 will develop a defect is lower than that for movable head tracks and the possibility of a defect occurring in a movable head track is very low, as for a 3348 Data Module.

If an uncorrectable error does occur on a fixed head logical track in a 3344 Model B2F, normally the customer engineer will replace the head/disk assembly instead of assigning an alternate movable head track to avoid loss of performance. If an uncorrectable error occurs on a movable head logical track, the logical track should be flagged and an alternate track should be assigned. This can be done using the IEHATLAS, IEHDASDR, or IBCDASDI utility of OS/VS. IEHDASDR or IBCDASDI should then be used to test the flagged fixed head track to determine whether the track is really defective. If the track is found not to be defective, the flag is removed and the assigned alternate track is released.

Note that the defective track testing capability of OS/VS IEHDASDR and IBCDASDI is not provided by any DOS/VS or DOS/VSE utility. DOS/VS users can obtain IBCDASDI, the standalone utility, by ordering the OS/VS1 system.

The Analysis Program-1 (AP-1) utility is provided as a problem determination aid for 3344 drives. When errors occur on a 3344, AP-1 can be used to determine whether the drive is failing or a problem exists on a recording disk. The AP-1 program operates as a problem program under DOS/VS, DOS/VSE, or OS/VS.

The 3344 is supported as a 3340 by DOS/VS, DOS/VSE, and OS/VS because of the compatibility between the two drives. Thus, any processing program that accesses a 3340 (with or without the fixed head feature) using a standard disk access method (SAM, ISAM, DAM, VSAM, or OS/VS PAM) can access a 3344 (Model B2 or B2F) without modification unless they contain timing dependencies.

When converting from 3340 drives to 3344 drives, consideration should be given to the use patterns of the 3340 volumes that are placed in the four logical 3340 volumes in the same 3344 drive to maximize performance. For example, one high-activity and three low-activity 3340 volumes that are accessed at the same time or four high activity 3340 volumes that are never accessed concurrently should be placed on the same 3344 drive. Multiple volumes of the same data base that will have only one request outstanding at a time can also be placed on the same 3344 drive.

The physical, capacity, and timing characteristics of 3340, 3344, 3330, 3350, and 2314 drives are compared in Table 20.25.1.

TRACK FORMATTING AND INITIALIZATION

Self-formatting records consisting of count, key, and data or count and data areas are written on the logical tracks in a 3344 drive just as on the logical tracks of a 3348 Data Module. The same six-byte error detection and six-byte error correction codes are used for 3344 and 3340 records. Defect skipping and seek verification like that implemented for 3348 Data Modules are implemented for the 3344.

The commands for the 3344 are the same as those for the 3340, except that one additional command, READ MULTIPLE COUNT-KEY-DATA, is implemented for the 3344. This command causes reading to begin with the count field of the next record that passes under the read/write heads (unless it is R0). Reading continues until all remaining records on the logical track are read. This command enables all the records on a logical track to be read using one command. This command can also be used on the 3340 drives that are attached to a 3830 Model 2 that has a 3340/3344 string attached.

Initialization of the disks in a 3344 drive is performed at the plant of manufacture. Four logical volumes are formatted. A home address and track descriptor (R0) record are written on each logical track in each logical volume. If a skippable defect is found during analysis of the surface of a logical track, the appropriate SD bytes and flag byte are written in the home address to indicate this fact.

If data cannot be read from a 3344 track during its use in an installation and recovery of this data is critical, the head/disk assembly unit can be sent to the plant of manufacture, where recovery will be attempted.

Table 20.25.1. Physical, capacity, and timing characteristics of 2314, 3340, 3344, 3330, and 3350 drives

Characteristic	2314	3340 Model 35/70 Data Module	3344	3330 Model 1/11	3350 (native mode)
Number of data disks per drive	11	2/4	8	10	8
Disk diameter in inches	14	14	14	14	14
Number of surfaces used per drive	20 data	3/6 data 1 servo	15 data 1 servo	19 data 1 servo	15 data 1 servo
Number of read/ write heads per recording surface	1	2 (30 fixed heads for servo surface on Model 70F)	2 (60 fixed heads for servo surface of Model B2F)	1	2 (60 fixed heads for servo surface Models A2F, B2F, C2F)
Number of logical cylinders per drive	200 plus 3 alternates	348 plus 1 alternate and 1 CE 696 plus 2 alternates and 2 CE	4 x 696 plus 4 x 2 alternates and 1 CE	404 plus 7 alternates 808 plus 7 alternates	555 plus 5 alternates and 1 CE
Number of physical/logical tracks per cylinder	20	12	12	19	30
Number of data tracks recorded per drive	4000	4176/8352	4 x 8352	7676/15,352	16,650
Full-track capacity in bytes	7294	8368	8368	13,030	19,069
Cylinder capacity in bytes	145,880	100,416	100,416	247,570	572,070
Maximum capacity per drive in bytes	29,176,000	34,947,768 69,889,536 (502,080 fixed head bytes in 30 logical tracks)	4 x 69,889,536 (1,004,160 fixed head bytes in 120 logical tracks)	100,018,280 200,036,560	317,498,850 (1,144,140 fixed head bytes in 60 logical tracks)
Seek time (ms)					
Maximum	130	50	50 (across 4 logical volumes)	55	50
Average	60	25	25 (across 4 logical volumes)	30	25
Cyl-to-cyl	25	10 (or 0)	10 (or 0)	10	10
Rotation time (ms)	25	20.2	20.2	16.7	16.7
Data transfer rate (KB/sec)	312	885	885	806	1198

ADVANTAGES

The 3344 is a growth direct access device for medium and large 3340 installations. The 3344 offers significantly larger capacity at a lower cost per bit than the 3340, timing characteristics equivalent to the 3340, minimum conversion effort through compatibility with the 3348 Model 70 and 70F Data Modules, and additional reliability and serviceability features. The 3344 can be used to increase online direct access capacity where data portability is not required.

A 3344 unit is equivalent in size to a 3340 Model B2 unit but contains four times the capacity through the implementation of a greater recording density and the use of twice the number of recording disks as in a Model 70 Data Module. However, power and air conditioning requirements for a 3344 and 3340 are the same. A 3340 cannot be converted to a 3344.

The technology used in the 3344 is based on that used in the 3340. Low-mass read/write heads, proximity recording, and servo track following are utilized in the 3344, as in the 3340, to improve reliability. Increased reliability that is achieved for the 3340 via the implementation of the sealed head/disk assembly design is also achieved for the 3344 via its fixed-media design. That is, 3344 reliability is aided by (1) the removal of head-to-disk alignment problems (a track is always read by the same head that wrote it), (2) elimination of a head loading and unloading mechanism, and (3) near elimination of disk contamination because the disks are sealed within the 3344 drive.

Reliability of the 3344 is improved over the 3340 through the elimination of any operator handling of the disks and by implementation of an improved power design. This power design results in fewer components, fewer power regulators, improved power isolation, and paralleling of critical circuits. As a result of its reliability features, no preventive maintenance is scheduled for 3344 drives.

Serviceability improvements for the 3344 are designed to help the customer engineer find and correct failures more quickly. The following are the major serviceability improvements:

- An enhanced fault symptom index has been added to provide faster problem isolation. It contains a list of "possible causes", an action/checklist, and cross references.
- Maintenance information manuals for the customer engineer have been improved by the addition of summaries of adjustments and items to check. Additional theory and recovery actions have been developed, especially in the "no trouble found" sections.
- Microdiagnostics have been enhanced and are more powerful in fault isolation capability. In addition, they are much faster and execute with less CE intervention.
- System-level online test programs provide better read/write tests for the CE than have been available for 2314 and 3330-series disk storage.

20:30 THE 2305 FIXED HEAD STORAGE MODULE AND 2835 STORAGE CONTROL MODEL 2

The 2835 Model 2 attaches to a block multiplexer channel in the 4341 Processor. It can be attached to channels 1 through 4 but not channel 5. One or two 2305 Fixed Head Storage Modules can be attached to 2835 Storage Control. Each module contains six nonremovable rotating disks on which data is recorded. Read/write heads, called recording elements, are fixed in position to access each track on the twelve recording surfaces so that no arm motion is required. (See Tables 20.30.1 through 20.30.3 at the end of this subsection for a comparison of 2305 Model 2 and 2303 Drum Storage characteristics and capacities.)

Spare, or alternate, tracks are provided in 2305 modules and must be wired in by a customer engineer to replace defective recording tracks. However, one spare track is available for assignment by the alternate track assignment utility program when a permanent track error occurs on a recording track during processing. Once a spare has been assigned as an alternate track, the hardware automatically accesses the alternate track when the defective recording track is addressed. This is called alternate track sparing. Switching to an assigned alternate track during processing is a programmed action for currently announced direct access devices.

The 2835 control unit provides new error correction facilities similar to those of the 3830 control unit. Recorded data areas within self-formatting records have ten correction code bytes appended to them instead of a two-byte cyclic check code. When certain types of data errors occur during the reading of the data portion of a record, the control unit can determine the bits in error and generate correction data. This recovery information is presented to the error routine via the sense bytes and can be used to correct the invalid record in processor storage (as described for 3330-series disk storage).

A command retry feature is implemented in the 2835. This feature permits certain types of failing commands to be reissued automatically by the channel, when requested by the control unit, without an intervening I/O interruption. For example, when a count or key area is read erroneously, the control unit retries the command once. If the error is not corrected by the retry, the control unit corrects the data in its own buffer, reexecutes the failing read, and presents the corrected data from the buffer instead of reading it from the track. Like the 3830 control unit, the 2835 contains a device that reads disk cartridges containing the control unit microprogram and diagnostic routines.

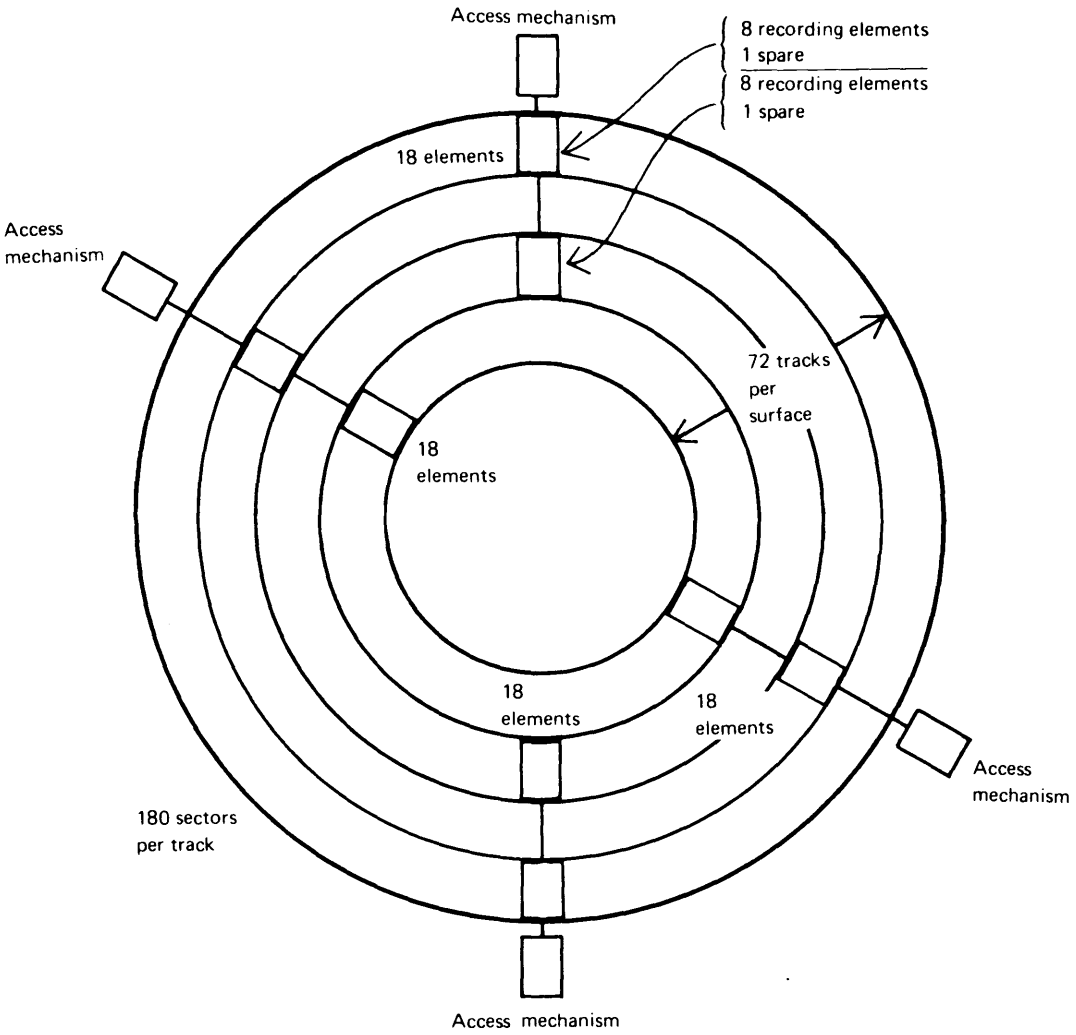
DATA RECORDING

Data tracks on the 2305 Model 2 facility are formatted in the same manner as on System/360 direct access devices, except for the absence of a home address on each track. There are 768 recording tracks and 96 spare tracks in one module. One recording element is positioned over each track. Each of the twelve surfaces contains 72 tracks, 64 recording and 8 spare. The spare tracks are interspersed among the 72 tracks so that every ninth track is a spare. Data is recorded serially by bit on each track.

Four nonmovable access mechanisms are positioned around the rotating disks as shown in Figure 20.30.1. Each access mechanism contains two groups of nine recording elements per surface (for a total of 16 recording and two spare elements) and accesses one-quarter of the tracks on each surface. A group of eight recording elements accesses every other track. The outermost element group of the access mechanism at the top of Figure 20.15.1 accesses data tracks 1, 3, 5, ..., 15, while data

tracks 2, 4, 6, ..., 16 are accessed by the outermost element group of the access mechanism at the bottom of the diagram.

There are 180 sectors per track on the 2305 Model 2. When RPS is used, the search time, from sector found to beginning of desired record, ranges from a minimum of 112 microseconds to a maximum of 167 microseconds.



64 recording tracks/surface x 12 = 768 recording tracks/module
8 spare tracks/surface x 12 = 96 spare tracks/module
216 recording elements (read/write heads)/access mechanism x 4 = 864 recording elements

Figure 20.30.1. Top view of a 2305 Model 2 disk surface

ROTATIONAL POSITION SENSING AND MULTIPLE REQUESTING

RPS is a standard feature of the 2835 control unit as is the other new capability called multiple requesting, which allows up to eight channel programs to be active concurrently on each of the two 2305 modules that can be attached to the control unit. In other words, a 2305 module can be viewed as eight logical devices, although physically it is only one device.

As described previously, rotational position sensing and block multiplexing permit a direct access device to disconnect during set

sector operations. These facilities, used in conjunction with the multiple requesting feature, permit concurrent operations to take place on each 2305 module in a facility. Thus, the effective data rate of a module can be increased substantially.

The multiple requesting capability is implemented by associating eight logical device addresses (0-7) with a 2305 module. Each logical device address is also assigned to a specific subchannel of a block multiplexer channel and a specific register (0-7) in the 2835 control unit. When a channel program is initiated, it is associated with an available logical device address by data management (I/O supervisor). (Logical device addresses are not permanently assigned to specific tracks or data sets in a module.) When the SET SECTOR command is issued, its specified sector number is stored in the register in the control unit that is assigned to the logical device address being used for the channel program. Then the control unit disconnects from the channel.

At this point, another channel program with a SET SECTOR command can be accepted by the channel and control unit (assuming neither is busy). This channel program will be initiated using another available logical device address and its assigned control unit register. This process can be repeated for up to eight SET SECTOR commands, so that eight channel programs can be executing concurrently per 2305 module.

Whenever the control unit is not executing a command or is not otherwise busy, it monitors the rotational position counter in the 2305 module that is being incremented each sector time period. When the sector number in the counter of a module compares equal with the sector number stored in one of the registers in the control unit and the channel is free, the control unit reconnects and resumes execution of the suspended channel program associated with the logical device address assigned to the control unit register that compared equal (see Figure 20.30.2).

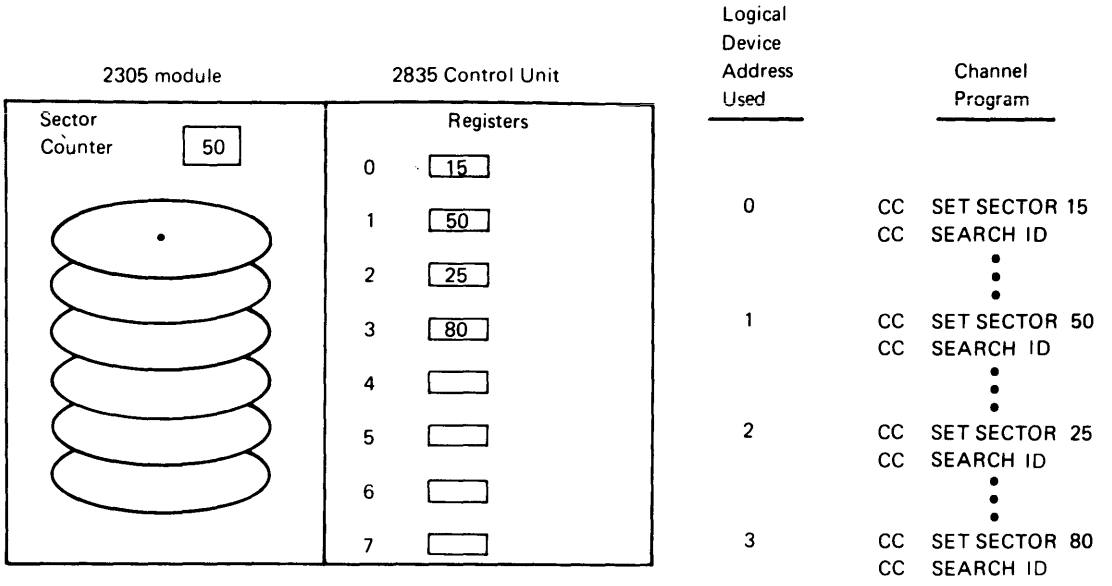


Figure 20.30.2. Multiple requesting on the 2305 facility

It should be noted that one 2305 module requires eight logical device addresses, each of which requires one subchannel on a block multiplexer

channel. Since a 2835 control unit can have two modules, one 2305 facility can use 16 device addresses and 16 nonshared block multiplexer subchannels.

Table 20.30.1. 2305 Model 2 facility and 2303 Drum Storage characteristics

Characteristic	2305 Model 2 Module	2303 Drum Storage
Device type	Six rotating disks with twelve recording surfaces	Rotating drum
Module capacity in bytes (full-track records, no key)	11,258,880	3,913,000
Number of recording tracks	768	800
Number of bytes per track (R0, R1 written without key)	14,660 Home address is never written on a track	4892 Home address is always written on each track
Number of read/write heads (recording elements) per module	864 One positioned to access each of 768 recording and 96 spare tracks	880 One positioned to access each of 800 recording and 80 alternate tracks
Rotation time (ms)	10	17.5
Seek time (ms) Maximum Average	 10.25 5.0	 17.5 8.6
Time channel busy searching when SET SECTOR is used (ms) Minimum Maximum	 .112 .167	 - -
Rotation speed (rpm)	6000	3400
Data transfer rate (MB/sec)	1.5	.303
Data validity checking	Ten correction code bytes (CCB) appended to each area written	Two-byte cyclic check (CC) code appended to each area written
Error recovery performed by the control unit	1. Command retry to retry a failing command without an I/O interruption 2. Correction of data errors that occur in the data area of a record by programming using sense byte information	Not provided
Rotational position sensing	Standard feature (180 sectors per track)	Not available

Table 20.30.1. (continued)

Characteristic	2305 Model 2 Module	2303 Drum Storage
Multiple requesting (allows concurrent I/O operations on one module)	Up to 8 concurrent operations per 2305 module (up to 7 per module supported by data management)	One I/O operation at a time is supported.
Record Overflow	Standard feature	Optional feature
Two-Channel Switch	Optional feature	Optional feature
Device contained in the control unit	Yes, to read the control unit microprogram and diagnostic programs	No

Table 20.30.2. Effective capacity of the 2305 Model 2 and the 2303 for various block sizes (DL column) with a 25-byte key

KL = 25 DL in Bytes	Effective Capacity in Bytes	
	2305 Model 2	2303
100	2,688,000	1,200,000
200	4,300,800	1,760,000
300	5,529,600	1,920,000
400	6,144,000	2,240,000
500	6,912,000	2,400,000
600	7,372,800	2,400,000
700	7,526,400	2,240,000
800	7,987,200	2,560,000
900	8,294,400	2,160,000
1000	8,448,000	2,400,000
2000	9,216,000	1,600,000
3000	9,216,000	2,400,000
4000	9,216,000	3,200,000

Table 20.30.3. Effective capacity of the 2305 Model 2 and the 2303 for various block sizes (DL column) when records are written without key

KL = 0 DL in Bytes	Effective Capacity in Bytes	
	2305 Model 2	2303
100	3,763,200	1,600,000
200	5,683,200	2,080,000
300	6,681,600	2,400,000
400	7,372,800	2,560,000
500	8,064,000	2,400,000
600	8,294,400	2,400,000
700	8,601,600	2,800,000
800	8,601,600	2,560,000
900	8,895,600	2,880,000
1000	9,216,000	2,400,000
2000	9,216,000	1,600,000
3000	9,216,000	2,400,000
4000	9,216,000	3,200,000

20:35 THE 3203 MODEL 5 PRINTER

The 3203 Model 5 Printer is a line printer with back printing and new features that, like those of a 3211 Printer, are designed to reduce operator intervention. The 3203 Model 5 Printer contains its own control function and attaches to a 4341 Processor via the byte or a block multilplexer channel. Figure 20.35.1 shows the 3203 Model 5 Printer.

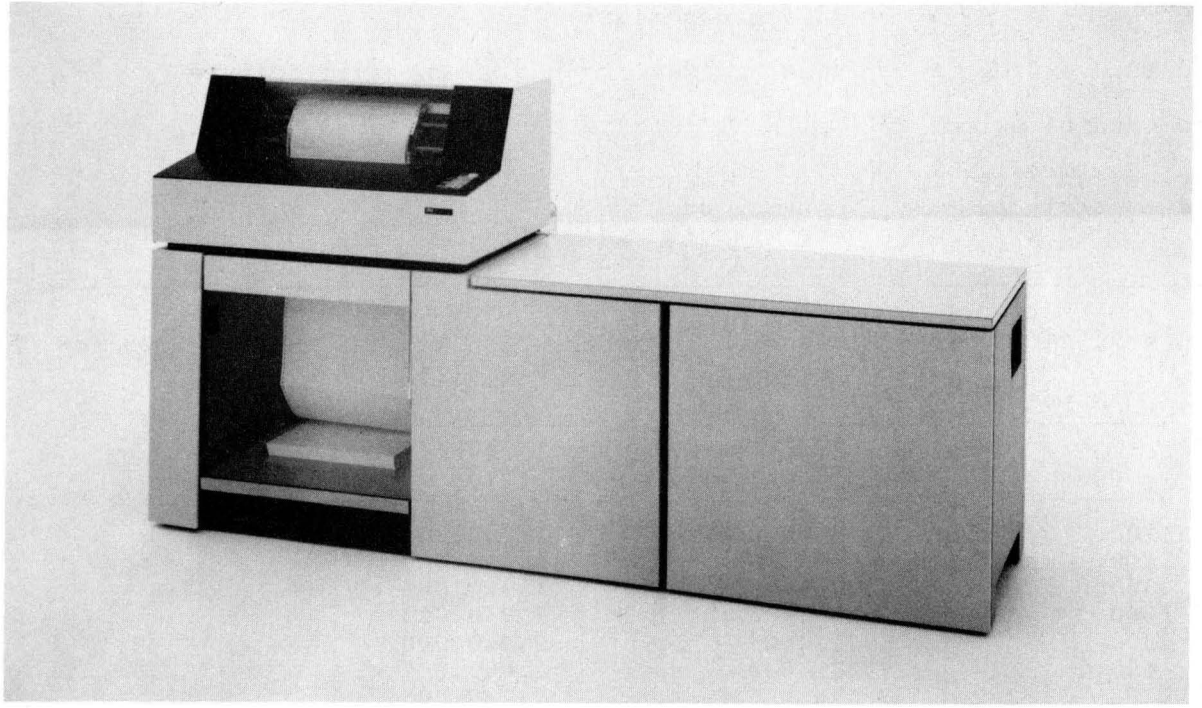


Figure 20.35.1. The 3203 Model 5 Printer (design model)

The 3203 Model 5 can print 1200 alphameric lines per minute (with a 48-character set) and is designed for installations that do not require the higher speed of the 3211 Printer. The 3203 Model 5 Printer is a standalone version of the 3203 Model 2 Printer that is available for attachment to System/370 Models 115 and 125. It offers higher reliability than the 1403 Printer, quieter operation, and a more compact design.

The 3203 Model 5 has a 132-print-position line and can print eight or six lines per inch. The Universal Character Set (UCS) feature and the 1416 Interchangeable Train Cartridge with 240 characters (which is also used with the 1403 Model N1 Printer) are standard. The cartridge arrangement is not restricted and can contain alphabetic, numeric, and special characters in any combination. The maximum speed of the 3203 Model 5 (1565 lines per minute) can be attained using a 32-character set.

The 3203 Model 5 can also be used to print documents that can be read by optical character readers, such as the 1287 Optical Reader, 1288 Optical Page Reader, 3881 Optical Mark Reader, and 3886 Optical Character Reader. Optical character printing is a standard capability of the 3203 Model 5 and several optical character train arrangements, each with 48 characters, are available. Use of optical character printing on multiple-part forms is not recommended.

The commands used for a 3203 Model 5 are a subset of those used for a 3211 Printer. Commands for the two are identical for the same functions. The indexing capability available for the 3211 is not available for the 3203 Model 5.

New features of the 3203 Model 5 include a powered forms stacker, tapeless carriage, and built-in vacuum-cleaning system. The powered stacker mechanism is self-adjusting and automatically rises in increments as the stack of paper mounts. This ensures that the stacker mechanism is always the same distance above the top of the stack of forms. The rate of rise during each increment is determined by the setting of the stacker rate knob, which can be adjusted by the operator to produce the best condition for the thickness of the forms being stacked. The stacker also can be raised or lowered manually.

The forms stacker is surrounded by a transparent acoustic enclosure that is hinged on the left side of the stacker. The function of the acoustic enclosure is to reduce noise during the operation of the printer. In order to access the forms stacker, the operator must open the enclosure.

Forms control paper carriage tape loading and unloading by the operator is eliminated by implementation of a tapeless carriage feature for the 3203 Model 5. Forms spacing and skipping are controlled by a program-loaded forms control buffer (FCB) contained in the printer attachment.

The FCB contains 192 storage positions, each of which corresponds to a print line, that is, to a single space of the carriage. Up to twelve channel codes (1-12), corresponding to the twelve channel positions of the paper carriage tape used on a 1403 Printer, can be stored in the appropriate buffer line positions to control carriage skipping. The FCB can be considered to contain a storage image of a carriage control tape.

A carriage control address register is used to address the FCB and maintain correct line position with respect to the form. This register is incremented as space and skip commands, which cause the form to advance, are issued. When a SKIP TO CHANNEL command is executed, the carriage control address register is incremented and the form moves until the channel specified is sensed in a line position in buffer storage. If the requested channel number is not found in the buffer, forms movement stops after address position 1 (line 1) has been sensed twice. This prevents runaway forms skipping.

A flag in a buffer storage line position is used to indicate the last line of the form for forms shorter than 192 lines. A flag bit is also used in the first buffer storage position to indicate six or eight lines per 2.54 centimeters (inch) spacing. The FCB is loaded with the desired forms spacing characters via a LOAD FCB command issued by a program. An error indication is given if an end-of-page flag is not present or if an invalid carriage code is loaded.

Although the 3203 Model 5 does not support the print indexing facility available for the 3211, if the first byte of an FCB load is coded as the print indexing byte, byte 2 instead of 1 of the load is interpreted as the first printer line position, just as for a 3211. This enables FCB loads for a 3211 with and without print indexing specified and FCB loads for a 3203 Model 5 to be interchangeable.

A vacuum-cleaning system, consisting of a motor-driven suction unit, vacuum tube and nozzle, and waste container, is mounted under the side cover of the 3203 Printer. The vacuum-cleaning system is always in operation during printing operations to continuously draw fibers of paper dust and ribbon lint from the type slugs. A separate hose is provided that is stored under the top front cover of the printer. The operator can attach this hose to the vacuum tube to manually clean the area around the print train cartridge before beginning printer operation during a shift.

Twenty-four bytes of sense information are provided to identify 3203 Printer and printer attachment malfunctions instead of only one byte, as is provided for the 1403. The first four bytes are identical to the first four sense bytes provided for the 3211 Printer. Certain errors (such as a parity check in the print line buffer) that might be corrected by programmed retry of the print operation are identified in the sense bytes, and carriage motion is suppressed. This permits error recovery without operator intervention if the retry is successful.

20:40 THE 3803/3420 MAGNETIC TAPE SUBSYSTEM

The 3803/3420 Magnetic Tape Subsystem consists of 3803 Tape Control Models 1 and 2 and a family of six 3420 Magnetic Tape Units. Models 3, 5, and 7 of the 3420 read, read backward, and write nine-track 1600-BPI, phase-encoded recorded, half-inch magnetic tape, and have a data rate of 120 KB/sec, 200 KB/sec, and 320 KB/sec, respectively. Models 4, 6, and 8 of the 3420 read, read backward, and write nine-track, 6250-BPI, group-coded recorded, half-inch magnetic tape, and have a data rate of 470 KB/sec, 780 KB/sec, and 1250 KB/sec, respectively.

This tape subsystem, which embodies a completely different control unit technology from 2400-series tape control units, offers price performance improvements, compatibility with existing seven- and nine-track tape volumes and programs, enhanced reliability, availability, and serviceability features, lower cost tape-switching capabilities, and standard automated tape-handling features previously available only on 2420 Magnetic Tape Units. (Table 20.40.4 at the end of this subsection compares 3420 and 2401 tape unit characteristics.)

The tape commands, status responses, and basic sense data of the 3803/3420 tape subsystem are compatibile with those of 2400-series tape units. Thus, any correctly written, non-time-dependent System/360 program for 2400-series tape units will operate without change on the 4341 Processor (subject to the constraints stated in Section 05:10) to handle operations on 3803/3420 subsystems with equivalent features installed. That is, existing nine-track 1600-BPI phase-encoded (PE), nine-track 800-BPI non-return-to-zero (NRZI), and seven-track 556/800-BPI NRZI-encoded tapes can be processed on appropriate 3420 tape unit models using existing programs without change to the tape volumes or programs.

3803 TAPE CONTROL MODEL 1 AND MODELS 3, 5, AND 7 OF THE 3420 MAGNETIC TAPE UNIT

Up to eight 3420 tape units, in any combination of Models 3, 5, and 7, can be attached to the 3803 Model 1. Models 4, 6, and 8 of the 3420 cannot be controlled by a 3803 Model 1. However, when required, a 3803 Model 1 can supply power for a 3420 Model 4, 6, or 8, as discussed later.

Models 3, 5, and 7 of the 3803/3420 tape subsystem offer users with intermediate systems and 2401 Magnetic Tape Units the advantages of advances in tape speed and design while maintaining media compatibility with existing tape volumes and providing enhanced RAS features. Specifically, the following are provided:

- Data rates of 120 KB/sec, 200 KB/sec, and 320 KB/sec at 1600-BPI density
- Phase-encoded data recording that automatically detects and corrects single-bit read errors in flight
- A tape transport design that minimizes tape wear and increases reliability, a single-capstan drive to control tape movement that provides faster data access times and rewinds, and more precise control of motor speed to help minimize damage to tape media
- Cartridge loading of tape, automatic tape threading, and a new automatic tape reel hub latch, all to reduce tape setup time
- Dual Density and Seven Track (mutually exclusive) features to enable a 3420 tape unit to handle either nine-track 800-BPI NRZI and 1600-BPI PE tape or seven-track 200/556/800-BPI NRZI (BCD or binary) tape

- Flexible, lower cost tape switching implemented in a new compact physical design. A two-channel switch is available also.

- Features such as new technology to improve subsystem reliability and new diagnostic facilities to aid serviceability and thereby increase subsystem availability

Phase-encoded recording. The phase-encoded (PE) recording technique that is used for 3420 Models 3, 5, and 7 offers superior error detection and reliability as compared to the conventional non-return-to-zero (NRZI) technique. In both cases, magnetic recording of one and zero bits is accomplished by means of flux reversals or changes in polarity. In NRZI recording, only one bits are recorded as magnetized spots, and a flux reversal occurs only for one bits. In PE recording both zero and one bits are recorded (the zero bit and one bit being opposite in polarity), and a flux reversal is required in every bit position. Thus, the PE dual flux recording technique differentiates between no recording and the presence of a zero bit. The absence of any signal is detected as an error.

The positive recording of all zero and one bits in PE eliminates the need for horizontal parity bits (longitudinal redundancy check used in NRZI recording), and vertical parity bits are used to correct single-bit read errors in flight. During reading, if a single track fails to respond with a suitable pulse in any bit position, reading of the rest of that track is immediately disabled for the remainder of the data block, and the remaining bits for that track are automatically generated by use of the vertical parity bits. In-flight single-track error correction eliminates the time normally lost in backspacing and rereading NRZI tape for correction of single-track dropouts or defects.

Phase encoding offers other advantages. If a string of zeros is recorded on tape, successful reading in NRZI requires close synchronization to "count" the correct number of zeros. With PE, this synchronization is provided by the flux reversal in every bit position; hence, PE recording (and reading) is self-clocking. In addition, each block written on a PE tape is preceded and followed by a coded burst of bits in all tracks to set up the individual track-clocking rates. The read circuitry is designed to recognize these bursts and thereby minimize the effect of noise in the gap.

The critical nature of vertical skew (alignment of bits within a byte) that is imposed by NRZI recording is minimized by this individual track-clocking scheme (one clock per track versus one clock for the entire tape subsystem), and by the use of one-byte (nine-bit) capacity skew buffers that can be in the process of collecting up to four data bytes at the same time, as the tape passes the read head. Because of the positive recording of all bits, once a skew buffer contains nine bits, one from each horizontal data track, it is an indication that a byte has been read. Thus, 3420 Models 3, 5, and 7 can handle the situation in which the tape is not exactly aligned, and bits from up to four adjacent bytes can be read concurrently.

Like 2400-series tape units, all models of the 3420 (3 through 8) utilize a two-gap read/write head that performs readback checking during write operations. All models of the 3420 also have a separate erase head that erases the entire width of the tape during any write operation before writing occurs. Full-width erasure reduces the likelihood of leaving extraneous bits in interblock gaps or skip areas and minimizes the interchangeability problems that can occur when tape is written on one tape unit and read on another.

Advanced engineering design. The tape path in 3420 Model 3, 5, and 7 tape units is designed for "soft handling" of tape volumes to minimize tape wear and thus improve tape reliability. Other features, such as

the single-capstan drive and optical tachometers, result in faster data access and rewind times than those of the 2401.

On a 3420 tape unit, the tape reel is mounted on the right side of the tape transport, instead of on the left as on a 2401 tape unit, so that an inverted tape path exists. As a result, when the tape is loaded in the columns, the recording side touches only the tape cleaner and read/write head. Friction and tape wear are also reduced by the presence of air bearings in the tape transport that provide a thin film of air between the nonrecording surface and each metal bearing.

Use of a single-capstan drive transport for tape movement and optical tachometers for control of motor speed result in several advantages. First, faster access times than those of 2401 tape unit models are achieved. Access time is defined as the time interval from initiation of a write or forward read command (given when the tape is not at load point) until the first data byte is read or written, assuming the tape is brought up to speed from stopped status. Nominal access times for 3420 Models 3, 5, and 7 are 4.0 ms, 2.9 ms, and 2.0 ms, respectively.

Second, the single-capstan drive can be made to operate faster than normal read/write speed, and in-column rewind is thus implemented. Full reel rewind speeds average 1041.4, 1219.2, and 1625.6 centimeters per second (410, 480, and 640 inches per second) for 3420 Models 3/4, 5/6, and 7/8, respectively. In addition, less time is required to rewind less than a full reel on a 3420 as compared to a 2401 because of faster rewind times achieved by in-column rewinding.

Last, three optical tachometers that monitor motor speed are used to achieve precise control of the speed of both the capstan motor and the tape reel motors. The capstan tachometer measures the size of the interblock gaps (IBG's) created during tape writing. The result is a more consistent IBG size than is created by 2400-series tape units, which enables more accurate calculation of tape-passing time. IBG passing times are 8.0 ms, 4.8 ms, and 3.0 ms for 3420 Models 3, 5, and 7, respectively. A 1.524-centimeter (.6-inch) IBG is written on 1600-BPI-density tapes. These times would be used in calculations for command-chained tape operations (reading or writing more than one tape block with a single START I/O instruction). More precise capstan motor speed also results in smoother starts and stops, thereby minimizing tape stretching and breaking.

The two tape reel tachometers measure tape speed as the tape enters and leaves the vacuum columns, and tape speed is adjusted when necessary. The 3420 tape unit is, therefore, less sensitive to voltage changes. More precise control of tape reel motor speed improves rewind speed and minimizes erratic tape stacking during rewinds so that there is less chance of damaging tape edges.

Automatic threading and cartridge loading. These advanced features are standard on all 3420 models (3 through 8) and significantly reduce tape mounting and demounting time. Tape threading is automatic for tape reels not enclosed in a wraparound cartridge once the reel--26.7-centimeter (10.5-inch), 21.6-centimeter (8.5-inch), or minireel--is mounted on the tape unit with the tape end placed in the threading chute and the load-rewind button is depressed. The power window is closed, the tape is threaded on the takeup reel, and the tape is loaded in the columns and positioned at load point within ten seconds after the button is depressed for 3420 Models 3, 4, 5, and 6. On 3420 Models 7 and 8, only seven seconds are required. In addition, unload and rewind/unload operations cause the tape to be completely rewound on the tape reel and the power window to be lowered so that the reel is ready for immediate demounting.

If the tape is enclosed in a wraparound cartridge--26.7-centimeter (10.5-inch) reels only--an operator need only mount the cartridge and does not have to place the tape end on the threader chute. Once the load-rewind button is depressed, ten seconds are required to open the cartridge and perform automatic threading. If automatic threading fails on the first try, the 3420 unit automatically rewinds the tape and retries threading. Unload operations rewind and close the cartridge automatically. In addition to fast tape reel mounting, the use of a wraparound cartridge offers other advantages. Handling of the tape reel itself is not required when the tape is used, because the wraparound cartridge is also the shelf storage container. The only time the cartridge need be opened is when it is opened by the 3420 during use. This enhances the reliability of the tape media.

The 3420 tape unit also has a new automatic reel latch instead of the snap-type hub latch implemented on newer 2400-series tape units. The operator places the tape reel on the hub and the automatic latch pneumatically locks it in position.

The advantage of these features can be shown by comparing setup times for tape units with and without the autothread feature. A tape study using experienced operators indicated the total time required to remove a tape reel, mount a new reel, thread the tape, and come to ready status was the following:

2401 tape unit - 40 seconds
Autothread tape unit without cartridge - 29 seconds
Autothread tape unit with cartridge - 13 seconds

Single Density, Dual Density, and Seven Track features. These three features are provided for both the 3803 Model 1 control unit and 3420 Model 3, 5, and 7 tape units. They are mutally exclusive features. The Dual Density or the Seven Track NRZI feature can be field installed on a 3420 Model 3, 5, or 7 tape unit only if it is replacing another NRZI feature. (For example, Dual Density can be field installed to replace the Seven Track but not the Single Density feature.) The Dual Density and Seven Track features facilitate efficient conversion of existing NRZI-recorded tapes to 1600-BPI phase-encoded format and permit tape volume interchange with other systems that use seven-track 200/556/800-BPI or nine-track 800-BPI tape.

A 3803 Model 1 control unit with the Single Density feature (without a switching feature) can handle up to eight 3420 tape units (Models 3, 5, and 7) with the companion Single Density feature installed. Only 1600-BPI PE tape can be read and written. When the Dual Density feature is present on the 3803 Model 1 control unit, both nine-track 1600-BPI PE and nine-track 800-BPI NRZI tape operations can be performed on 3420 units (Models 3, 5, and 7) with the companion Dual Density feature installed. (Tape units with the Single Density feature still handle only nine-track 1600-BPI PE tape.)

When the Seven Track feature is present on the 3803 Model 1 control unit, seven-track 556/800-BPI NRZI operations (both BCD and binary format) can be performed on 3420 tape units (Models 3, 5, and 7) with the companion Seven Track feature installed. The data convert and translate facilities are a standard part of the Seven Track feature. An RPQ is available to enable a 3803 Model 1 to handle 200-BPI density for tape units with the Seven Track feature. The capability of handling 200-BPI density is a standard part of the Seven-Track feature for 3420 Model 3, 5, and 7 tape units. Table 20.40.1 summarizes 3803 Model 1 control unit capabilities without and with these features.

Tape mode setting for 3420 Models 3, 5, and 7 is handled as follows. For write operations on nine-track tape units with the Dual Density feature, a MODE SET command must be issued to establish 1600-BPI PE or

800-BPI NRZI recording mode prior to the first write. Tapes written in PE mode have a format identification burst recorded at load point that differentiates them from NRZI-mode tapes. During reading, sensing of this burst automatically puts the tape unit in 1600-BPI PE mode. Failure to sense the burst establishes NRZI mode if both the tape unit and control unit have the Dual Density feature. If an attempt is made to read NRZI-mode tape on a unit without the Dual Density feature, an error indication results. Once PE or NRZI mode is established for read operations, it is retained until the tape returns to load point.

For seven-track read and write operations, NRZI mode, density, parity, and use of the data converter or translator are established by issuing a single MODE SET command.

Table 20.40.1. 3803 Model 1 control unit configurations and capabilities with Single Density, Dual Density, and Seven Track features

3803 Model 1 with Single Density feature	3803 Model 1 with Dual Density feature	3803 Model 1 with Seven Track feature (includes data convert and translate)
1. Nine-track, 1600-BPI PE tape on 3420 Models 3, 5, and 7 with Single Density feature	1. Nine-track, 1600-BPI PE tape on 3420 Models 3, 5, and 7 with Single Density feature 2. Nine-track, 800-BPI NRZI tapes and nine-track, 1600-BPI PE tapes on 3420 Models 3, 5, and 7 with the Dual Density feature	1. Nine-track, 1600-BPI PE tape on 3420 Models 3, 5, and 7 with Single Density feature 2. Seven-track, 556/800-BPI, NRZI BCD and binary tapes on 3420 Models 3, 5, and 7 with the Seven Track feature. If the appropriate RPQ is installed on the 3803 Model 1, 200-BPI density tapes can be handled on Models 3, 5, and 7.
Note: The Single Density, Dual Density, and Seven Track features are mutually exclusive on the same control unit or the same tape unit.		

Tape-switching features. Tape subsystem configuration flexibility is provided by field-installable tape-switching options that permit up to four control units to be switched among up to 16 tape units. While this capability is provided for 2400-series tape units via the 2816 Switching Unit, tape switching for the 3803/3420 subsystem offers the advantages of compact design, reduced cost, and enhanced subsystem availability.

The switching features are built into the 3803 Model 1 or 2 control unit itself so that space for standalone switching units is not required. The fact that tape-switching features are contained in the 3803 control units being switched (rather than in one unit) also enhances tape subsystem availability. When a switch failure occurs in one control unit, that unit can be switched offline, eliminating the necessity of removing the entire tape-switching subsystem from the operative system configuration.

Using combinations of the Communicator and the Two-Control Switch, Three-Control Switch, or Four-Control Switch optional features, two, three, or four control units can be configured to be switched among up to 8 or up to 16 tape units. The Communicator must be present in all control units that are to be switched. It allows the control unit in

which it is installed to address tape units that are attached to an interconnected control unit. Figure 20.40.1 shows the switching feature requirements for permissible switching combinations. The switch combinations shown for switching control units among up to 16 tape units are the same that are required for switching control units among up to 8 tape units.

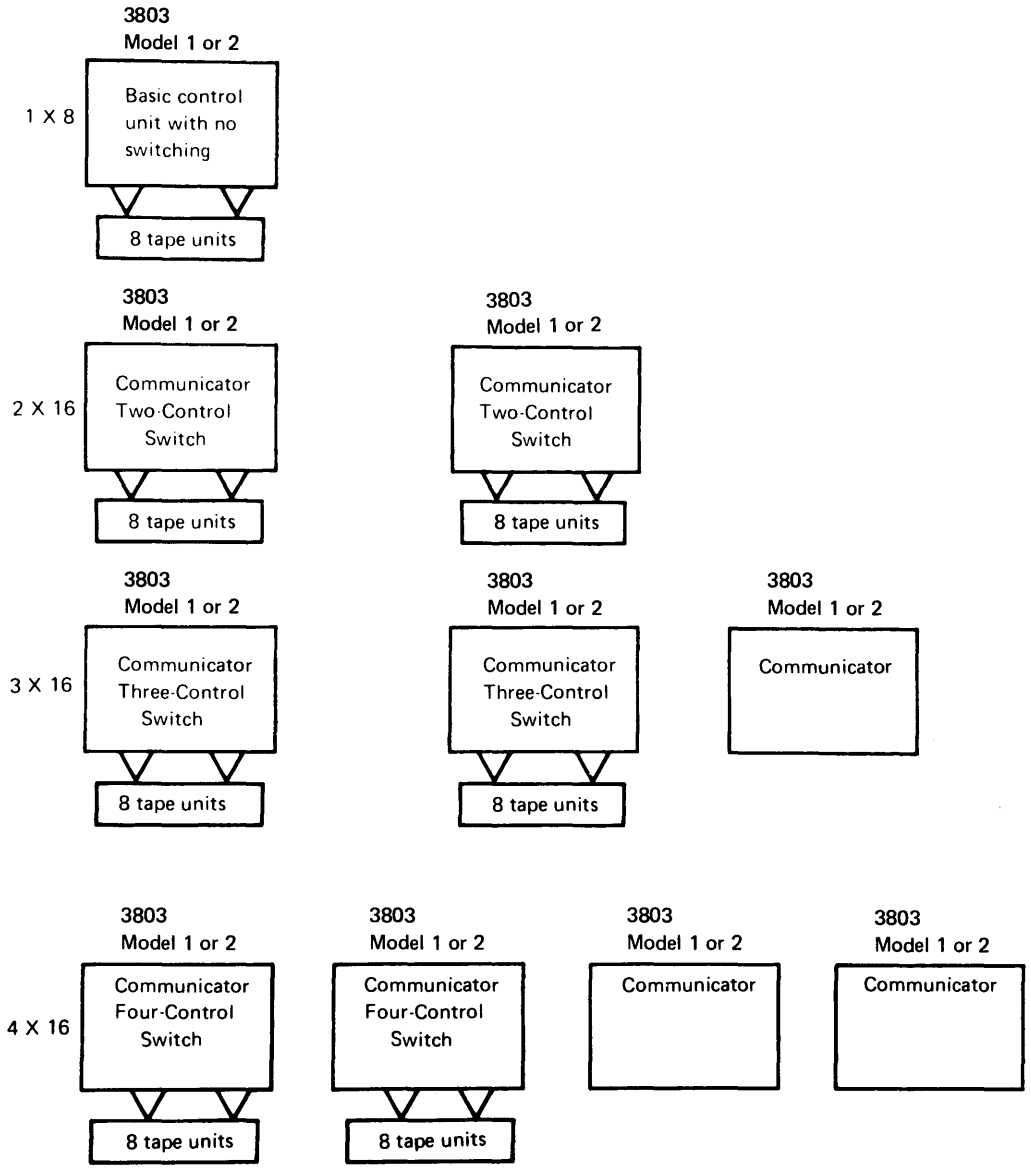


Figure 20.40.1. Tape-switching configurations for the 3803/3420 Magnetic Tape Subsystem

A two-control-unit switching configuration is required to replace the 2804 and 2404 read-while-write control units. The advantage of the tape-switching approach is that for a small price increment better performance is possible, because any two tape operations can be active concurrently in a switched configuration (including two reads or two writes), while the degree of simultaneity achieved using a read-while-write control unit is application dependent. That is, the application must lend itself to reading, then writing (or vice versa).

Two-Channel Switch. A 3803 control unit (Model 1 or 2) with the Two-Channel Switch installed can be attached to two channels in the same system or in two different systems. This feature can be present on a 3803 that also has tape-switching features installed.

When the Two-Channel Switch is installed on a 3803, the switch can be set to permit only one channel or the other to access the control unit and its tapes, or the switch can be set to allow access to the control unit and its tapes by both channels, one channel at a time. In the latter case, if channel A requests an operation when the control unit is busy performing an operation on channel B, channel A must wait until the control unit becomes available again.

The Two-Channel Switch can be used to connect a 3803 to two channels in the same system to provide the capability of switching the tapes from one channel to another or to provide two channel paths to the tapes connected to the 3803.

Similarly, the Two-Channel Switch can be used to connect a 3803 to two channels in two different systems to provide the capability of switching all the tapes from one system to the other, for backup purposes. Alternatively, the 3803 can be set to allow access to it by both systems. In this case, each tape connected to the 3803 can be accessed by only one processor. Those that are to be accessed by processor A must be varied offline to processor B, and vice versa, by the operator. Partitioning of the tapes in a two-system environment is strictly the responsibility of the operator. There are no physical controls on a 3803 with a Two-Channel Switch that provide for partitioning the tapes between the two processors, nor is there any programming systems support that checks whether or not a tape is enabled to both systems.

The Two-Channel Switch for the 3803/3420 subsystem offers configuration flexibility not generally available to 2400-series tape unit users. For System/360, a Two-Channel Switch was provided only for a 2803 Model 1 control unit and could be used only in Model 67 and in Model 65 multiprocessing configurations.

Data security erase command. This command is implemented for all 3420 models (3 through 8). When a data security erase command is issued, the tape unit selected erases tape from the point at which the operation is initiated until the tape indicate (end-of-file) marker is sensed. This command is not provided for 2400-series tape drives.

Reliability, availability, and serviceability features. The 3803/3420 hardware subsystem has several RAS features, in addition to the reliability and availability features already discussed for the tape media itself.

The 3803 control unit (Models 1 and 2) embodies a totally new design. Monolithic logic technology is used in the 3803 control unit, and it therefore offers greater reliability and more compact physical design in comparison with the 2803 control unit. (The 3803, Model 1 or 2, is approximately half the size of a 2803 control unit.) In addition, both logic circuitry and mechanical components in the control and tape units are functionally packaged to enable more rapid fault location and faster replacement.

As a diagnostic aid, additional sense bytes are generated by the microprogram-controlled 3803 control unit. The 3803 uses ROS for microprogram residence. Twenty-four sense bytes are provided by 3803 Models 1 and 2 (instead of the six generated by the 2803), certain of which can be used in tracing control unit microprogram malfunctions. Some of the other additional sense bytes identify the control unit and

tape unit by serial number, optional features, and engineering change (EC) level.

Two other very significant new serviceability features are microdiagnostics resident in the 3803 control unit and radial attachment of 3420 tape units to the 3803.

Resident microdiagnostics in the 3803 (Models 1 and 2) enhance test operations for the 3803/3420 subsystem by relieving the CPU of the execution of most time-dependent tests. Diagnostics in the 3803 are executed via use of a diagnostic command issued by a program.

The 3803 also contains diagnostics that are operative during normal tape processing operations. These diagnostics perform operations such as the monitoring of measurement functions of the tape units. If an irregularity is noted, the control unit generates sense bits to inform the executing program of the malfunction.

Tape subsystem availability is improved by radial attachment of 3420 tape units to the 3803 control unit (Models 1 and 2). That is, each 3420 is cabled directly to the control unit so that any malfunctioning tape unit can be disconnected from the tape subsystem for servicing without disturbing the other tape units. When tape units are attached to the control unit in series (each tape unit cabled to the next tape unit), as are 2400-series units, the entire tape subsystem must be taken offline to uncable a tape unit.

These new features, combined with the use of fewer adjustable parts, are designed to provide optimum tape subsystem availability through better reliability and reduced maintenance time.

3803 TAPE CONTROL MODEL 2 AND MODELS 4, 6, AND 8 OF THE 3420 MAGNETIC TAPE UNIT

Models 3 through 8 of the 3420 can be attached to a 3803 Model 2. Up to eight 3420 units, in any combination of Models 3 through 7, can be controlled by a 3803 Model 2, which provides the signal requirements (user data and signals needed by the tape subsystem for control and error checking and correction) and power requirements for these models. Model 2 of the 3803 also provides the signal requirements for up to eight 3420 Model 8 tape units but supplies enough power for a maximum of only six of these models. When fewer than six 3420 Model 8 tape units are attached to a 3803 Model 2, it can also supply the power required for various combinations of 3420 Models 3 through 7.

If a 3803 Model 2 cannot supply the power required by the combination and number of 3420 models that are attached to it, the additional required power can be supplied from other 3803 Model 1 or 2 control units in the tape configuration. In order to achieve this, the power and signal connections to the tape units for which additional power is required must be made via two separate cables instead of one. For example, if eight 3420 Model 8 tape units are attached to a 3803 Model 2, power must be supplied for two of the Model 8 tape units from a second 3803, either Model 1 or 2. Therefore, six of the 3420 Model 8 tape units would be power- and signal-connected to the 3803 Model 2 via one cable for each unit, as usual. The other two 3420 Model 8 tape units would be signal-connected to the 3803 Model 2 via one cable each and power-connected to a second 3803 via another two cables.

Table 20.40.2 lists the maximum number of 3420 tape units and the model combinations for which one 3803 Model 1 or 2 can supply power. If a permissible maximum configuration is exceeded for a 3803 control unit, than separate cabling, as previously described, can be used to distribute power to the 3420 tape units as required. If separate

cabling still does not provide the necessary power or if only one 3803 Model 2 is present in the configuration and the tape units for which it must supply power exceed the permissible maximum, a no-charge power supply feature can be installed on the 3803 Model 2. This feature enables a 3803 Model 2 to supply power for up to eight 3420 tape units in any combination of Models 3 through 8. This power feature requires a 100-amp power source.

Table 20.40.2. Model combinations and maximum number of 3420 tape units that can be powered from one 3803 Model 1 or 2. (Each line in the table indicates a permissible combination.)

Number of 3420 Model 8 tape units	Number of 3420 Model 7 tape units	Number of 3420 Models 3 through 6 tape units
6	0	0
5	1	0
5	0	2
4	2	1
4	1	2
4	0	3
3	4	0
3	3	1
3	2	2
3	1	4
3	0	5
2	5	0
2	4	2
2	3	3
2	2	4
2	1	5
2	0	6
1	*	*
0	**	**

*Seven additional 3420 tape units in any combination of Models 3 through 7

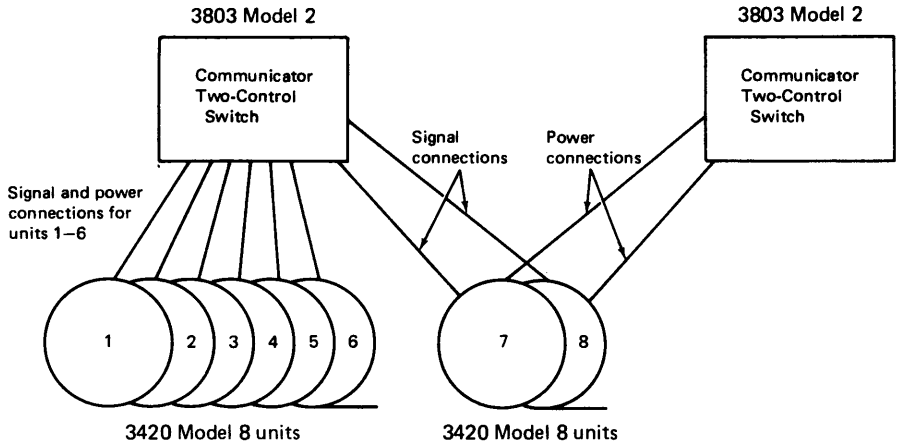
**Eight 3420 tape units in any combination of Models 3 through 7

Figure 20.40.2 illustrates the use of separate power and signal cabling to supply the required power for two sample tape unit configurations. Note that the control units from which the additional power needed by another control unit is required are also subject to the limitations listed in Table 20.40.2.

The signal and power support provided by 3803 Model 1 and 2 units can be summarized as follows. The 3803 Model 1 provides the signal requirements for up to eight 3420 tape units, in any combination of 3420 Models 3, 5, and 7, and power requirements for the combinations of 3420 Models 3 through 8 listed in Table 20.40.2. Models 4, 6, and 8 of the 3420 cannot be signal-connected to a 3803 Model 1. The 3803 Model 2 provides the signal requirements for up to eight 3420 tape units, in any combination of Models 3 through 8, and power requirements for the combinations of 3420 Models 3 through 8 listed in Table 20.40.2.

The 3803 Model 2 attaches only to a block multiplexer channel in the 4341 Processor. It cannot be attached to the byte multiplexer channel. Model 1 of a 3803 can be field converted to a Model 2. Field upgrades of 3420 tape units is also possible. A Model 3, 5, or 7 can be converted to a Model 4, 6, or 8. A Model 4 can be converted to a Model 6 or 8 and a Model 6 can be converted to a Model 8.

Example 1. 2 x 8 switching of eight 3420 Model 8 units



Example 2. No switching

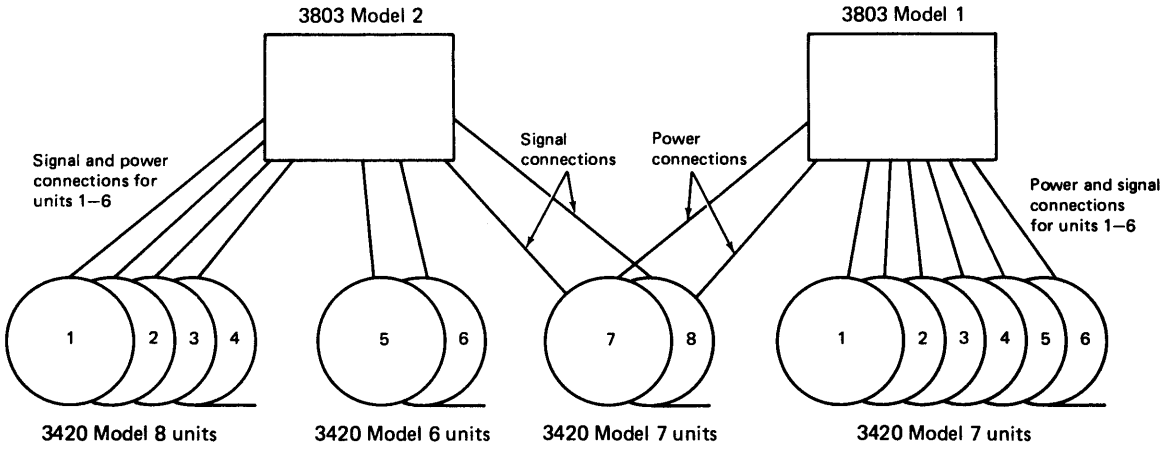


Figure 20.40.2. Sample 3420 tape unit configurations for which separate power and signal cables are required

The 3803 Model 2 and 3420 Models 4, 6, and 8 offer 2401-series tape unit users the advantages of the 3803 Model 1 and 3420 Models 3, 5, and 7 as well as the following additional features:

- Density of 6250 BPI, almost four times that of 3420 Models 3, 5, and 7
- Data rates of 470 KB/sec, 780 KB/sec, and 1250 KB/sec at 6250 BPI
- Up to three times the data capacity per tape reel, depending on the blocking factor, as a result of the higher recording density and an IBG of .762 centimeters (.3 inches) instead of 1.524 centimeters (.6 inches)

- Faster read/write access to a data block as a result of the shortened IBG and other design improvements

- Automatic in-flight detection and correction of all double-bit as well as single-bit errors using group-coded recording and other reliability features, such as a new tape-cleaning mechanism and automatic read amplification

- Compatibility with existing tapes via support of 1600-BPI density for 3420 Models 4, 6, and 8, and of 200-, 556-, 800-, and 1600-BPI density for Models 3, 5, and 7

Group-coded recording A group-coded recording technique is used by 3420 Models 4, 6, and 8 when reading and writing 6250-BPI-density nine-track tape. The group-coded technique enables both single- and double-bit read errors to be corrected in flight since error checking and correction bits, as well as data bits, are recorded within a tape record.

Engineering design. The significant engineering design features that are described for 3420 Models 3, 5, and 7 also apply to Models 4, 6, and 8 (inverted tape path, single-capstan drive, in-column rewind, and optical tachometers). However, the capstan tachometer in Models 4, 6, and 8 is a dual-phase tachometer instead of a single-phase tachometer, as is used in 3420 Models 3, 5, and 7.

Since the IBG size on 6250-BPI-density tapes is only .762 centimeters (.3 inches), instead of 1.524 centimeters (.6 inches) as on 1600-BPI-density tapes, more precise control of tape starting and stopping is required. Tape acceleration during starting must be faster so that the correct speed is achieved by the time the data block is reached and deceleration during stopping must be faster to ensure that tape motion consistently stops at the proper location within the shorter IBG. The dual-phase capstan tachometer in 3420 Models 4, 6, and 8 provides the more precise motion and control that is needed.

Nominal read and write access times for Models 4, 6, and 8, which are faster than access times for Models 3, 5, and 7, respectively, even at 1600-BPI density, are shown below.

	Model 4	Model 6	Model 8
Read access (ms) 1600 BPI	4.0	2.6	1.7
Write access (ms) 1600 BPI	3.0	2.0	1.3
Read access (ms) 6250 BPI	2.3	1.6	1.1
Write access (ms) 6250 BPI	2.1	1.5	.95

Tape rewind speeds for Models 4, 6, and 8 are the same as those for Models 3, 5, and 7, respectively. However, when 6250-BPI density is used instead of 1600-BPI, less tape is required to contain a given data set/file because of the .762-centimeter (.3-inch) IBG as well as because of the higher density. Therefore, less time is required to rewind the given data set/file when it is recorded at 6250-BPI instead of 1600-BPI.

Density features. The capability of handling 6250-BPI and 1600-BPI densities is standard on a 3803 Model 2 control unit. Optionally, the 9-Track NRZI field-installable feature can be added to a 3803 Model 2 to enable it to handle nine-track, 800-BPI-density, NRZI-recorded tape. In

addition, the optional 7-Track NRZI feature, which requires the 9-Track NRZI feature as a prerequisite, can be installed on a 3803 Model 2 to enable it to handle seven-track, 200-, 800-, and 556-BPI density, NRZI-recorded tape. When these two optional features are installed on a 3803 Model 2, both seven-track and nine-track 3420 tape units can be attached to the 3803 and multiple densities and recording techniques can be handled. If an attempt is made to read a seven-track tape of any density or a nine-track 800-BPI-density tape via a 3803 Model 2 without the requred seven- or nine-track feature, an error results.

Either the 6250 Density or the 6250/1600 Density feature must be installed on a 3420 Model 4, 6, or 8 to enable it to handle 6250-BPI-density nine-track tapes only or both 6250- and 1600-BPI-density nine-track tapes, respectively. These two features are field installable and mutually exclusive. Models 4, 6, and 8 of the 3420 cannot handle 800-BPI, NRZI, nine-track tapes or any seven-track tapes (the Dual Density and Seven Track features that are available for 3420 Models 3, 5, and 7 cannot be installed on 3420 Models 4, 6, and 8). Table 20.40.3 summarizes 3803 Model 2 control unit configurations.

Tape mode setting for 3420 Models 4, 6, and 8 is handled as follows. For write operations on tape units with the 6250/1600 Density feature installed, the default density is 6250 BPI. Therefore, a MODE SET command must be issued prior to the first write to establish 1600-BPI density. Tapes written in these two densities have a unique 6250-BPI or 1600-BPI format identification burst recorded at load point. During reading, sensing of this burst automatically places the tape unit in the proper mode (6250-BPI group-coded or 1600-BPI PE). If an attempt is made to read a 1600-BPI-density tape on a 3420 Model 4, 6, or 8 without the 6250/1600 Density feature installed, an error results. Similarly, an error results if an attempt is made to read a seven-track tape of any density or a nine-track, 800-BPI-density tape on a 3420 Model 4, 6, or 8.

Table 20.40.3. 3803 Model 2 control unit configurations

1 3803 Model 2 with no optional features	2 3803 Model 2 with 9-Track NRZI feature	3 3803 Model 2 with 9-Track NRZI and 7-Track NRZI features
1. Nine-track 6250-BPI tapes on 3420 Models 4, 6, and 8 with the 6250 Density or the 6250/1600 Density feature 2. Nine-track 1600-BPI tapes on 3420 Models 4, 6, and 8 with the 6250/1600 Density feature 3. Nine-track 1600-BPI tape on 3420 Models 3, 5, and 7 with the Single Density or Dual Density feature	1. Same as configuration 1 2. Nine-track 800-BPI NRZI tapes and 1600-BPI PE tapes on 3420 Models 3, 5, and 7 with the Dual Density feature	1. Same as configuration 2 2. Seven-track 200/556/800-BPI NRZI BCD and binary tapes on 3420 Models 3, 5, and 7 with the 7-Track feature

Tape-switching and Two-Channel Switch features. All of the control unit and channel-switching facilities described for the 3803 Model 1 are also available for the 3802 Model 2. When separate cabling is used in a switching configuration to distribute the required power, the tape units

with two cables must be signal-connected to a control unit in the configuration that has a control switch feature installed.

Tape-switching configurations can contain a mixture of 3803 and/or 3420 models. When separate cabling is used, a tape unit must be signal-connected to a 3803 control unit that has the optional density features that are required by the tape unit. In addition, any 3420 tape unit in a switched configuration can be accessed only via 3803 control units in the configuration that have the required optional features. Figure 20.40.3 illustrates a switching configuration with mixed 3803 and 3420 models and indicates the control units by which each tape unit can be accessed.

Reliability, availability, and serviceability features. In addition to group-coded recording and the features discussed for the 3803 Model 1 and 3420 Models 3, 5, and 7, two other reliability features are implemented in 3420 Models 4, 6, and 8, namely, automatic read amplification and a cleaning mechanism. When reading a 6250-BPI-density tape on a 3420 Model 4, 6, or 8, the read amplifier gain of the tape unit is automatically adjusted to the amplification of that individual tape. This facility enables successful reading of 6250-BPI tapes with a wide range of amplitudes.

The cleaning blade, which is part of the read/write assembly and which is implemented in 2400-series tape units and in 3420 Models 3, 5, and 7, is also present in 3420 Models 4, 6, and 8. However, 3420 Models 4, 6, and 8 also have another cleaning mechanism that is designed to protect the read/write head when possible and to remove contaminants from the tape media.

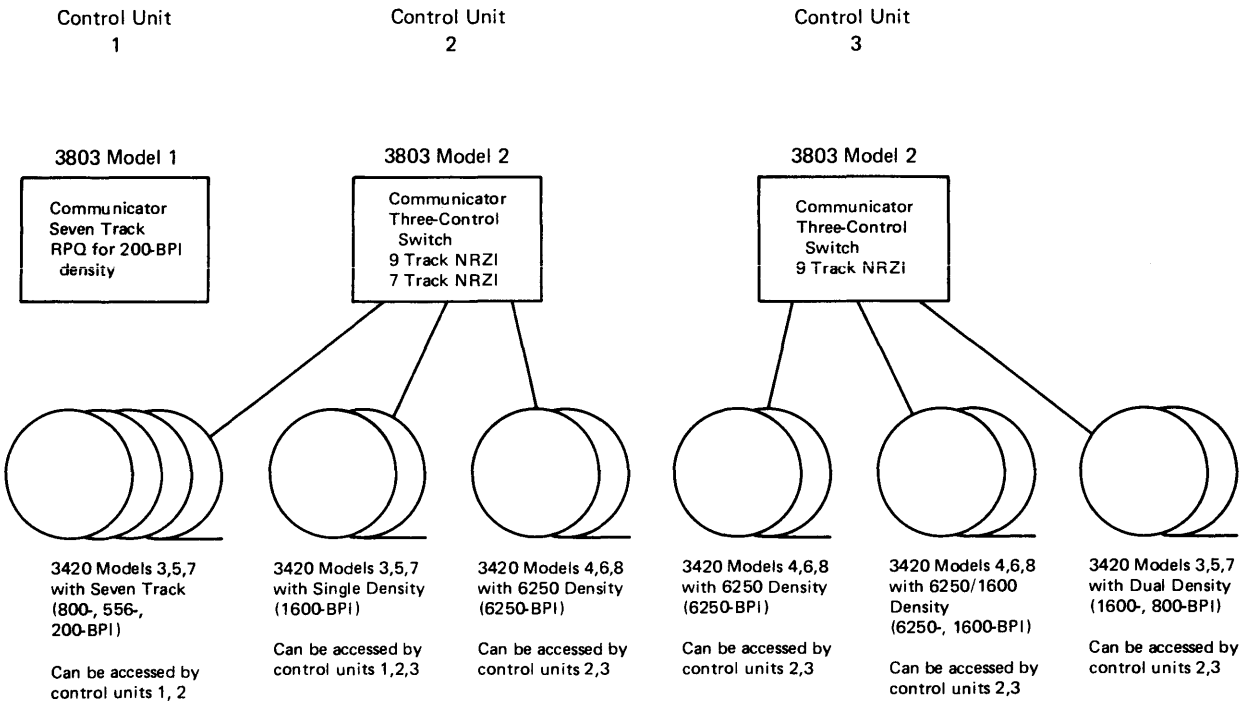


Figure 20.40.3. Sample tape-switching configuration with mixed 3803 and 3420 models

The cleaning mechanism consists of a cleaning web, which is a .945-centimeter (3/8-inch) wide continuous band of material, that remains positioned between the read/write head and the tape during tape loading,

high-speed rewind, and tape unloading operations. Positioning of the cleaning web under the head is performed automatically before any of these operations begin. During the time the tape is being read or written, the cleaning web is in a retracted position so that the read/write head can come in direct contact with the tape. Retraction is also performed automatically at the time it is required.

This cleaning mechanism is designed to minimize possible contamination of the read/write head by the tape media at the times it is most likely to occur, which is during high-speed rewind, unloading, and, in particular, loading operations. Without this cleaning mechanism, during tape loading operations, the read/write head comes in contact with the tape leader, which is normally more exposed to contamination than any other portion of the tape. While this cleaning facility offers additional protection against contamination of the read/write head, it is not meant to be a replacement for tape unit cleaning by the operator.

SUMMARY

In conclusion, the 3803/3420 Magnetic Tape Subsystem offers users of 2400-series tape units the following advantages:

- Increased throughput for tape operations because of faster data rates, faster access times, and less rewind time for tapes recorded at 6250-BPI density. In-flight correction of single-bit read errors for Models 3, 5, and 7 and of single- and double-bit read errors for Models 4, 6, and 8 eliminates a backspace and reread procedure and reduces the number of permanent read errors.

- Reduced tape setup time because of automatic tape threading and cartridge loading

- Significantly reduced tape library size and tape reel handling using 6250-BPI density

- Less tape wear as a result of the transport design and automatic threading and less tape damage caused by handling if wraparound cartridges are used for tape volume mounting and storage

- Reduced maintenance time because of the transport design (fewer adjustable parts), functional packaging of components, expanded sense bytes, and microdiagnostics resident in the control unit

- Increased tape subsystem availability because of reduced maintenance requirements

- Compatibility with existing 2400-series tape volumes and programs

Table 20.40.4. 3420, 2420, and 2401 Magnetic Tape Unit characteristics

	3420 Tape Units						2420 Tape Units	2401 Tape Units	
Characteristic	Model 3	Model 5	Model 7	Model 4	Model 6	Model 8	Model 5 Model 7	Model 2 Model 5	Model 3 Model 6
Density (bytes/inch)	1600	1600	1600	6250	6250	6250	1600 1600	800 1600	800 1600
Data rate (KB/sec)	120	200	320	470	780	1250	160 320	60 120	90 180
Tape speed (inches/sec)	75	125	200	75	125	200	100 200	75 75	112.5 112.5
Recording technique	PE	PE	PE	Group coded	Group coded	Group coded	PE PE	NRZI PE	NRZI PE
Nominal interblock gap size in inches (nine-track)	.6	.6	.6	.3	.3	.3	.6 .6	.6 .6	.6 .6
Nominal read/write access to data (ms)	4.0	2.9	2.0	2.3/2.1	1.6/1.5	1.1/.95	3.9 2.5	8 8	5.3 5.3
In-column rewind	Yes	Yes	Yes	Yes	Yes	Yes	Yes Yes	No No	No No
Nominal rewind and unload time (secs)	76	66	51	76	66	51	78 66	90 90	66 66
Nominal rewind to ready status--full 2400-foot reel (secs)	70	60	45	70	60	45	72 60	84 84	60 60
Automatic threading	Standard	Standard	Standard	Standard	Standard	Standard	Standard. Standard	Not available. Not available	Not available. Not available
Time to ready status after load button pressed (secs)	10	10	7	10	10	7	10 7	- -	- -

Table 20.40.4. 3420, 2420, and 2401 Magnetic Tape Unit characteristics (continued)

Characteristic	3420 Tape Units						2420 Tape Units	2401 Tape Units	
	Model 3	Model 5	Model 7	Model 4	Model 6	Model 8	Model 5 Model 7	Model 2 Model 5	Model 3 Model 6
Cartridge loading (10.5-inch reels only)	Standard	Standard	Standard	Standard	Standard	Standard	Standard Standard	Not available. Not available	Not available. Not available
Automatic reel latch	Yes	Yes	Yes	Yes	Yes	Yes	No No	No No	No No
Recording medium (1/2-inch magnetic tape)	IBM Series/ 500 Dynexcel, Heavy Duty, or equivalent. 10.5", 8.5", 6.5" reels. (Use of Mylar* is not recommended.)	Same as Model 3	Same as Model 3	Same as Model 3	Same as Model 3	Same as Model 3	Same as 3420. Same as 3420	Same as 3420 plus Mylar. Same as 3420	Same as 3420 plus Mylar. Same as 3420
Inverted tape path, single-capstan drive optical tachometers	Yes	Yes	Yes	Yes	Yes	Yes	Yes except for optical tachometers. Same as Model 5	No No	No No
Error checking									
Error correction during reading	Automatic single-track	Automatic single-track	Automatic single-track	Automatic single- and double-track	Same as Model 4	Same as Model 4	Automatic single-track. Automatic single-track	Programmed. Automatic single-track	Programmed. Automatic single-track
Longitudinal redundancy check	No	No	No	No	No	No	No No	Yes No	Yes No
Automatic read amplification	No	No	No	Yes	Yes	Yes	No No	No No	No No

*Trademark of E. I. Dupont deNemours & Co. (Inc.)

Table 20.40.4. 3420, 2420, and 2401 Magnetic Tape Unit characteristics (continued)

	3420 Tape Units						2420 Tape Units	2401 Tape Units	
Characteristic	Model 3	Model 5	Model 7	Model 4	Model 6	Model 8	Model 5 Model 7	Model 2 Model 5	Model 3 Model 6
Additional cleaning web mechanism	No	No	No	Yes	Yes	Yes	No No	No No	No No
Number of sense bytes	24	24	24	24	24	24	6 6	6 6	6 6
Microdiagnostics in control unit	Yes	Yes	Yes	Yes	Yes	Yes	No No	No No	No No
Separate erase head	Yes	Yes	Yes	Yes	Yes	Yes	Yes Yes	Yes Yes	Yes Yes
Data security erase command	Yes	Yes	Yes	Yes	Yes	Yes	No No	No No	No No
Seven Track feature	Optional	Optional	Optional	Not available	Not available	Not available	Not available. Not available	Optional. Not available	Optional. Not available
Densities (BPI)	800 556 200	800 556 200	800 556 200	-	-	-	- -	800 556 200 -	800 556 200 -
Data rate (KB/sec)									
800 BPI	60	100	160	-	-	-	-	60	90
556 BPI	41.7	69.5	111.2					41.7	62.5
200 BPI	15	25	40					15	22.5
								-	-
Recording technique	NRZI	NRZI	NRZI	-	-	-	- -	NRZI -	NRZI -
IBG size (inches)	.75	.75	.75	-	-	-	- -	.75 -	.75 -
Translator	Standard	Standard	Standard	-	-	-	- -	Optional -	Optional -
Data Converter	Standard	Standard	Standard	-	-	-	- -	Optional -	Optional -

Table 20.40.4. 3420, 2420, and 2401 Magnetic Tape Unit characteristics (continued)

	3420 Tape Units						2420 Tape Units	2401 Tape Units	
Characteristic	Model 3	Model 5	Model 7	Model 4	Model 6	Model 8	Model 5 Model 7	Model 2 Model 5	Model 3 Model 6
Dual Density feature (800/1600 BPI)	Optional	Optional	Optional	Not available	Not available	Not available	Not available. Not available	Not available. Not available	Not available. Not available
Data rate (KB/sec) at 800 BPI	60	100	160	-	-	-	-	-	-
Recording technique at 800 BPI	NRZI	NRZI	NRZI	-	-	-	- -	- NRZI	- NRZI
IBG size at 800 BPI (inches)	.6	.6	.6	-	-	-	- -	- .6	- .6
6250/1600 Density	Not available	Not available	Not available	Optional	Optional	Optional	Not available. Not available	Not available. Not available	Not available. Not available
Data rate (KB/sec) at 1600 BPI	-	-	-	120	200	320	- -	- -	- -
Recording technique at 1600 BPI	-	-	-	PE	PE	PE	- -	- -	- -
IBG size at 1600 BPI	-	-	-	.6	.6	.6	- -	- -	- -
Nominal read/write access to data (ms)	-	-	-	4.0/3.0	2.6/2.0	1.7/1.3	- -	- -	- -
Control unit	3803 Model 1 or 2 Read while write (RWW) capability is not provided.	Same as Model 3	Same as Model 3	3803 Model 2 Read while write capability is not provided.	Same as Model 4	Same as Model 4	2803 Model 2. 2803 Model 2	2803, 2804 (RWW) Model 1 or 2. 2803, 2804 (RWW) Model 2	Same as Model 2. Same as Model 5
Attachment of tape units to control unit	Radial	Radial	Radial	Radial	Radial	Radial	Serial Serial	Serial Serial	Serial Serial

Table 20.40.4. 3420, 2420, and 2401 Magnetic Tape Unit characteristics (continued)

	3420 Tape Units						2420 Tape Units	2401 Tape Units	
Characteristic	Model 3	Model 5	Model 7	Model 4	Model 6	Model 8	Model 5 Model 7	Model 2 Model 5	Model 3 Model 6
Tape switching	2 x 16 3 x 16 4 x 16 (Switching features in 3803 control units)	Same as Model 3	Same as Model 3	Same as Model 3	Same as Model 3	Same as Model 3	2 x 16 3 x 16 4 x 16 (Requires one or two 2816 units.) Same as Model 5	Same as 2420. Same as 2420	Same as 2420. Same as 2420
Two-Channel Switch	Optional	Optional	Optional	Optional	Optional	Optional	Not available. Not available	Optional on 2803 Model 1 for Model 67 and MP65 systems only. Not available	Same as Model 2. Not available

30:05 DOS/VSE

DOS/VSE FUNCTIONS

The 4341 Processor is supported by DOS/VSE, which is system control programming (SCP). DOS/VSE is based on DOS/VS Release 34 and offers many significant new functions. DOS/VSE provides basic processor support that is required to operate IBM 4300 Processors (4341 and 4331 Processors) and most System/370 processors (Models 115 to 158 and the 3031 Processor Complex).

The DOS/VSE SCP alone is designed to support system generation and hardware servicing functions only. DOS/VSE is designed to operate with the VSE/Advanced Functions program product, which should be installed to ensure the successful execution of IBM and user programs.

DOS/VSE contains the following:

- The IPL program, supervisor, job control program, librarian, linkage editor, all system utilities, SDAID, RMS routines, and console support for the supported processors
- SAM, DAM, and ISAM access methods, diskette I/O modules, and language translator I/O modules
- The Assembler language translator
- Environment recording, editing, and print (EREP) program (includes the same facilities as are available in the EREP for OS/VS and most of the output from the DOS/VSE EREP is compatible with that of the OS/VS EREP)
- Analysis Program-1 (AP-1) for 3344, 3350, and FBA direct access devices
- Maintain System History Program (includes the functions of PTFHIST for DOS/VS Release 34 and provides new support)
- OLTEP (which can execute in any partition, not the background only)

The DOS/VSE SCP supports all the processors, I/O devices, and functions supported by the DOS/VS Release 34 SCP except the following:

- 2321 Data Cell Drive
- 2495 Tape Cartridge Reader
- QTAM
- VSAM, BTAM, VTAM, and POWER/VS. (Each is now available as a program product instead of an SCP component.)
- Seek separation
- Label cylinder. (A label area in virtual storage is used instead.)

- PDAIDS. (The new SDAID program in DOS/VSE combines the functions of the SDAID and PDAID components of DOS/VS Release 34 and contains new functions.)
- Interval timer at location 80 in processor storage
- Model 20 emulator
- EXTM support

The 1400 emulator program that operates with DOS/VS Release 34 is not distributed with DOS/VSE and must be ordered separately if required. This 1400 emulator program can be used to generate a 1400 emulator that will operate under DOS/VSE on System/370 processors but not on 4300 Processors. Simulation of 1400 programs on the 4341 Processor (and emulation of 1400 programs on the 4331 Processor) is provided via the IBM Systems 1401/1440/1460 Emulator Program (program product), which is discussed in Section 40. A simulator, but not an emulator, program generated using this program product will operate on System/370 processors.

The DOS/VSE Standalone Restore program must be used for the initial installation of DOS/VSE. Certain differences between the system generation macros for DOS/VSE and DOS/VS exist because many features that are optional for DOS/VS are standard for DOS/VSE, new functions are supported by DOS/VSE, and some functions of DOS/VS are not supported by DOS/VSE.

DOS/VSE contains the following facilities that are not available in DOS/VS Release 34:

- Support of 4341 and 4331 Processors in System/370 and ECPS:VSE modes and the 3278 Model 2A Display Console for these processors
- Full support of the 3031 Processor Complex
- Support of 3310 (not attachable to the 4341 Processor), 3370, 8809 (not attachable to the 4341 Processor), 5424, 3284, 3286, 3287, and 3289 I/O devices and 3880 Storage Control.
- Functional improvements, such as channel switching for direct access devices; use of the time-of-day clock, clock comparator, and CPU timer for all timing facilities; and Fast Copy Disk Volume Utility support of 2311 and 2314 disk storage. (The CPU timer and clock comparator are required by DOS/VSE.)
- Miscellaneous usability and performance improvements

User-written problem programs (program phases) that operate under a DOS/VS Release 34 supervisor and access the same I/O devices can operate under the control of a DOS/VSE supervisor that supports System/370 or ECPS:VSE mode subject to the hardware compatibility constraints listed in Section 05:10 and the following operating system constraints:

- Programs that use internal DOS/VS supervisor interfaces described in the program logic or serviceability and debugging aids publications for DOS/VS may not execute correctly because of changes in supervisor code and/or control blocks. Those programs that use the standard interfaces (that is, IBM-provided Assembler Language macros or high-level language statements) will operate without change.
- Programs that depend upon facilities of DOS/VS Release 34 that are not implemented in DOS/VSE (as previously listed) will not operate.

- Programs that issue the EXCP macro with the REAL parameter specified must be modified if these programs are to operate with an ECPS:VSE mode supervisor.

- Programs that have been link edited to an absolute address must be re-link edited to produce relocatable program phases.

- Programs that use DTFSL must be reassembled and link edited.

- Programs that contain BTAM modules must be re-link edited with new BTAM modules generated using the BTAM-Extended Support program product.

- Programs that are generated using DOS Type 1 language translators (COBOL LCP, COBOL D, ANS COBOL, or PL/I D) may not execute correctly under DOS/VSE. The preceding language translators themselves and the following DOS Type 1 programs may not execute correctly under DOS/VSE: Util-Group-1, Util-Group-2, Util-Group-3, Tape/Disk Sort/Merge, and MPS Utility Macros.

DOS/VSE SUPPORT OF 4341 PROCESSOR FEATURES AND I/O DEVICES

The following discusses DOS/VSE support of specific 4341 Processor hardware features and I/O devices.

Modes of processor operation. Both System/370 and ECPS:VSE modes are supported to provide a virtual storage environment. A DOS/VSE supervisor can be generated to support one mode or the other (but not both) and a given 4341 Processor installation can generate one or more DOS/VSE supervisors that support System/370 mode and/or one or more supervisors that support ECPS:VSE mode.

An ECPS:VSE mode supervisor provides increased performance over a System/370 mode supervisor because channel program translation is eliminated, less page management processing is required, and the internal mapping mechanism used for address translation in ECPS:VSE mode is faster than the DAT facility of System/370 mode.

A DOS/VSE supervisor generated to support the 4341 Processor operating in System/370 mode (utilizing 33XX direct access storage) is a minimum of 92K bytes and operates with System/370, EC, and DAT modes in effect in the 4341 Processor. A DOS/VSE supervisor generated to support the 4341 Processor operating in ECPS:VSE mode (utilizing FBA direct access storage) is a minimum of 86K bytes and operates with ECPS:VSE and EC modes in effect. The resident supervisor size can be reduced by approximately 12K bytes by making part of the supervisor pageable during IPL.

Instructions. The Assembler language translator supports all the instructions, except the MOVE INVERSE instruction, for the 4341 Processor, including those instructions that operate only with ECPS:VSE mode.

Interval timer. This timer is not supported.

Time-of-day clock. This clock is used to provide time-of-day support and to support interval timing.

Clock comparator and CPU timer. The clock comparator is used with the time-of-day clock to support interval timing. The CPU timer is used by the job accounting facility.

Store and fetch protection. Store protection is supported. Fetch protection is not supported.

Reference and change recording. The reference and change bits are used by the page replacement routine of the page supervisor.

Program event recording. This feature is used by the SDAID program.

Monitoring feature. This feature is used by the SDAID program.

Console support. The 3278 Model 2A Display Console operating in display (3277) mode is supported as the operator console device by Display Operator Console (DOC) support. Optionally, a natively attached 3287 Model 1 or 2 Printer can be used for hard-copy backup of the 3278 Model 2A when it is operating in display mode. Printer-keyboard mode of operation is also supported for the 3278 Model 2A and requires a 3287 Printer.

Channels. The byte multiplexer channel and five block multiplexer channels are supported. Channel program translation (utilizing channel indirect data addressing when necessary) and page fixing/unfixing in System/370 mode supervisors is unchanged from DOS/VS Release 34. In ECPS:VSE mode supervisors, the construction of a new channel program with translated addresses and, when necessary, indirect data address lists is eliminated, since address translation is done by hardware during channel program execution.

For ECPS:VSE mode only, supervisor scanning of channel programs to determine the pages that must be fixed can be eliminated by using the new IORB macro instead of the CCB macro. The IORB macro permits specification of the areas that must be fixed before the I/O operation is started. Alternatively, the IORB can specify that all required pages have already been fixed. For the latter specification, the channel program is started with no processing by the supervisor.

231X disk storage. The 2311 and 2314/2319 are supported by all the disk access methods in DOS/VSE, and for all system files, except that a 2311 cannot be the system residence device.

3330-series, 3340/3344, and 3350 Direct Access Storage. All these disk devices are supported by SAM and DAM (and VSE/VSAM). ISAM does not support the 3350 or the 3330-series Model 11. All are supported for system files. Rotational position sensing is supported.

2305 Model 2 disk storage. This device is not supported.

3370 Direct Access Storage. The 3370 is supported by SAM (and VSE/VSAM) and for all system files. The same functions are supported as for other direct access devices. ISAM programs can access files on 3370 drives using the ISAM Interface Program of VSE/VSAM.

The 3370 Surface Analysis Utility (a standalone system utility) performs surface analysis on 3370 tracks and permits reclamation of alternate blocks, except those flagged at the plant of manufacture. This is a separately orderable utility.

The 3370 is supported by a System/370 mode supervisor only when the supervisor executes in a virtual machine under VM/370 control, and the supervisor must include VM/370 linkage support (VM=YES option of the VSE/Advanced Functions Release 1 program product). The 3370 is always supported by ECPS:VSE mode supervisors. Analysis Program-1 (AP-1) has been updated to support FBA devices.

3880 Storage Control. This control unit is supported.

3203 Model 5 Printer. This printer is supported as an I/O device and as a system (SYSLST) printer, as are the 3211 and 3800 printers. The

train cleaning utility supports the 3203 Model 5 (and the 1403 if the UCS feature is installed).

PROGRAM PRODUCTS FOR DOS/VSE

Several system-oriented program products are provided for operation under DOS/VSE:

- VSE/Advanced Functions (the Advanced Functions-DOS/VS program product for DOS/VS Release 34 will not operate with DOS/VSE). This program product should be installed in order to utilize any other system-oriented or industry-oriented program product with DOS/VSE.
- VSE/Virtual Storage Access Method (VSE/VSAM). The VSAM component for DOS/VS releases will not operate with DOS/VSE.
- VSE/POWER. (The POWER/VS component for DOS/VS releases will not operate with DOS/VSE.)
- VSE/Interactive Computing and Control Facility (VSE/ICCF). The field developed program IBM Entry Time Sharing System (ETSS) will not operate with DOS/VSE.
- BTAM-Extended Support (BTAM-ES), ACF/VTAM Release 2, and ACF/VTAME
- VSE/Access Control-Logging and Reporting
- VSE/Interactive Problem Control System (VSE/IPCS)
- High-level language translators, sorts, and utility programs
- IBM Systems 1401/1440/1460 Emulator Program
- VSE/IBM System/3-3340 Data Import
- VSE/Data Interfile Transfer, Testing, and Operations Utility (VSE/DITTO)
- System/3 DOS/VS RPG II Conversion Preprocessor
- Data communications/data base (DB/DC) program products (such as CICS/VS, DL/I DOS/VS Version 2, and DL/I Entry DOS/VS)
- Other system-oriented program products (VS APL, for example)

The following discussions highlight the new facilities offered by certain of the system-oriented program products provided for use with DOS/VSE. Details of the 1400 emulator program product are covered in Section 40.

VSE/Advanced Functions Release 1

The VSE/Advanced Functions program product and the DOS/VSE SCP together provide the required operating environment for other DOS/VSE program products and user-written programs. VSE/Advanced Functions Release 1 contains all the functions included in the Advanced Functions-DOS/VS program product for DOS/VS Release 34 and the following new functions, usability features, and performance features not provided in Advanced Functions-DOS/VS:

- Support of seven partitions for all supported system residence disk device types

- FBA device support (3370 for the 4341 Processor) for System/370 mode supervisors that execute in a virtual machine under VM/370 control. The DOS/VSE supervisor must include the VM=YES option of VSE/Advanced Functions. Note that a DOS/VSE supervisor that contains the VM=YES option can execute in a 4341 Processor only in a virtual machine in a VM/370 environment.

- Implicit link support that reduces the number of job control statements required for a compile, link-edit, and execute operation

- Job-to-job communication via a 256-byte area in the system GETVIS area using the new JOBCOM macro

- Multiple label areas (one per processor) on the SYSRES volume to facilitate sharing of a SYSRES volume by two or more processors

- Console usability improvements (such as retention of messages with an outstanding reply on the display screen)

- Elimination of the requirement for a LBLTYP linkage editor control statement to reserve space in a partition for label processing

- Automated system initialization using IPL information in cataloged procedures

- Dump improvements (the writing of system, partition, operator-invoked, and standalone dumps to one or a pair of dump files, allowing selective printing of dump types at a later time)

- Fetching of highly used B-transient and C-transient routines from the SVA instead of from the core image library

- Fast open of the hard-copy file during IPL

- A high-level System Directory List (SDL) search that minimizes page faults during directory searching

- Support of page data sets with multiple extents. Up to 15 maximum on the same or different type volumes (maximum three extents per volume) with the same track formatting are supported. That is, the extents must reside on only CKD or only FBA devices.

- Option to disable fast CCW translation for the duration of one job step

- Improved operation of DOS/VSE with VM/370

VSE/Advanced Functions Release 2

The VSE/Advanced Functions Release 2 program product provides all the functions of VSE/Advanced Functions Release 1 and the following additional facilities:

- Direct access device sharing across processors (using the LOCK and UNLOCK gating facility). VSE/VSAM and VSE/POWER use this facility for their shared DASD support.

- Up to 208 user tasks active concurrently (maximum 32 in one partition) and up to 12 partitions

- Extended label area support that includes (1) user-defined label areas supported on any disk device except 2311 (not just SYSRES), (2) loading permanent labels for foreground partitions from the

background partition, and (3) adding and deleting labels via the label ADD and label DELETE functions

- Operating system tailoring at IPL (determination of the I/O devices present in the configuration)
- Improved librarian facilities, such as multiple procedure libraries, concurrent access to the same library of any type by multiple partitions, and concatenated libraries (up to 30 maximum) within a job step
- Generation of a DOS/VSE and VSE/Advanced Functions Release 2 system in a partition of a DOS/VSE multiprogramming environment
- Device independence for direct access devices utilizing SAM or DAM (dynamic modification of the DTF at OPEN time to reflect the actual disk device type assigned)
- A list log utility to print all job-related SYSLOG messages on SYSLIST at end-of-job time whenever a job is canceled. The utility can be invoked explicity as well.
- Support required by the VSE/VSAM Space Management for SAM program product (additional DLBL parameters, SYSLNK in VSE/VSAM managed disk space, and extended VOLUME command)
- B-transient area contention removal by having the most frequently used or longest running B-transient routines (such as OPEN, EOJ, and attention routines) execute in the SVA

The VSE/Advanced Functions Release 2 program product must be installed instead of VSE/Advanced Functions Release 1 when VSE/POWER Release 2, VSE/VSAM Release 2, or the Fast Copy Data Set Program is to be installed.

VSE/POWER Release 1

VSE/POWER (Priority Output Writers, Execution Processors, and Input Readers) operates in a partition under control of the DOS/VSE supervisor to provide services for other partitions that are defined as POWER-controlled partitions when started. VSE/POWER is designed to improve system throughput in a multiprogramming environment by providing job scheduling by priority within class and automatic data transcription to and from unit record devices (card readers, printers, and punches) overlapped with job step execution.

The VSE/POWER program product provides all the functions available in the POWER/VS component of DOS/VS Release 34 plus the following major additional facilities:

- Support of FBA devices for VSE/POWER files
- SYSIN on magnetic tape. Jobs can be submitted to VSE/POWER in blocked/unblocked, standard labeled/unlabeled files on 2400-series and 3400-series magnetic tape units for the 4341 Processor (and the 8809 for the 4331 Processor).
- Save and restore capability that enables VSE/POWER queues to be dumped from disk to tape and then restored to disk. VSE/POWER queues that are not being processed can also be transferred to another VSE/POWER installation.
- Improved nesting of SLI (Source Library Inclusion) statements

- Integration of support for the 3800 Printing Subsystem
- Expanded PALTER and PHOLD commands
- Support of SNA exchange media inbound, allowing the interchange of diskettes between different SNA workstations
- Full 3741 support

VSE/POWER is fully compatible with POWER/VS so that programs that execute in a DOS/VS Release 34 environment under POWER/VS control will execute in a DOS/VSE environment under VSE/POWER control without modification.

The VSE/POWER Release 1 program product supports only local spooling and job scheduling in a multiprogramming environment. The separately orderable VSE/POWER Remote Job Entry Feature program product can be installed with VSE/POWER to provide transmission of jobs between a VSE/POWER environment and remote BSC terminals, SNA terminals, and SNA workstations. In addition, the Remote Job Entry Feature provides a BSC multileaving facility that supports the transmission of VSE/POWER jobs between different VSE/POWER installations. To support SNA terminals or workstations, ACF/VTAM Release 2 or ACF/VTAME must be included in the DOS/VSE system being used.

VSE/POWER Release 2

The VSE/POWER Release 2 program product provides the same functions as VSE/POWER Release 1 and the following additional facilities:

- Spooling and job scheduling support for up to eleven partitions
- Improved cross-partition communication interface
- Shared spooling support via the separately orderable VSE/POWER Shared Spooling Feature program product

The Shared Spooling Feature permits common VSE/POWER queue and data files (and optionally a common account file) to be accessed concurrently by different VSE/POWER Release 2 systems, each of which is operating in a different processor. The VSE/POWER Remote Job Entry Feature program product that is available for VSE/POWER Release 1 can also be installed with VSE/POWER Release 2.

Installation of VSE/POWER Release 2 requires installation of VSE/Advanced Functions Release 2. Jobs that execute correctly under VSE/POWER Release 1 will execute correctly under VSE/POWER Release 2 without modification.

VSE/VSAM Release 1

The VSE/VSAM Release 1 program product provides all the facilities available in the VSAM component of DOS/VS Release 34 and the following additional functions:

- Support of FBA disk devices
- Selection, via a new DEFINE parameter, of either fast catalog access or optimum disk space utilization
- Access to additional information about the data or index records of a catalog via an additional parameter on the SHOWCAT macro

- Ability to assign special classes to any area of direct access space for the purpose of controlling space allocation (in order to control the assignment of the fixed head area of a disk, for example)
- Automatic correction of a duplicate record situation if a control interval split for a key-sequenced file is interrupted by a processor failure
- A reduction in the time required to create and reload backup and portable copies of VSAM files and volumes
- Support of a larger set of physical record sizes for CKD devices to better utilize track capacity (512 to 8192 bytes in multiples of 512 bytes)

VSAM files, catalogs, and programs created utilizing DOS/VS releases can be used with VSE/VSAM under DOS/VSE. Similarly, VSAM files, catalogs, and programs created utilizing VSE/VSAM can be processed by the VSAM component of DOS/VS releases if they do not utilize any of the new functions listed above.

VSE/VSAM Release 2

The VSE/VSAM Release 2 program product provides the facilities of VSE/VSAM Release 1 and the following additional support:

- Sharing of VSAM catalogs and files between DOS/VSE operating systems that have Release 2 of VSE/Advanced Functions installed (processor sharing of VSAM data)
- Improved performance of catalog requests through improved scanning algorithms and a reduction in the number of scans
- Improved performance when SHAREOPTIONS 4 is specified
- Automatic correction for keyed sequential access processing of a duplicate record condition that occurs as a result of a processor failure during a control area split of a key-sequenced file
- A CANCEL command to permit the operator to cancel the current job step or job while Access Method Services is being used
- Improvements in VSAM space management functions

Optionally, the separately orderable VSE/VSAM Space Management Feature for SAM Files program product can be installed for use with VSE/VSAM Release 2. This program provides VSAM space management functions for DOS/VSE sequentially organized (SAM) files, as follows:

- Definition and deletion of a SAM file in VSAM direct access space
- Access to a SAM file in VSAM direct access space
- Dynamic secondary allocation of a SAM file in VSAM direct access space

VSE/Interactive Computing and Control Facility

The VSE/ICCF program product provides the means for any DOS/VSE user to move from batch-oriented processing to interactive computing. It is

an extension of the ETSS I (Entry Time Sharing System I), ETSS II, and TCS/VS Field Developed Programs and supports the following:

- Interactive operation of DOS/VSE
- Interactive program development and testing (using any of the DOS/VSE language translators--Assembler, BASIC, COBOL, FORTRAN IV, PL/I, and RPG II)
- Personal computing and problem solving (using VS BASIC, FORTRAN, and PL/I)
- Improved DOS/VSE usability by interactive execution of applications
- Improved protection for all of the online data in the installation

The above facilities improve the ease of use of DOS/VSE and can help increase the productivity of personnel. Installations that require larger interactive environments than VSE/ICCF supports can utilize VM/370 and CMS.

VSE/ICCF requires the use of (1) CICS/DOS/VS and the BTAM-ES, ACF/VTAM Release 2, or ACF/VTAME program product for integrated terminal control or (2) the Terminal and Transaction Facility (TTF) of VSE/ICCF. Terminals supported are the 3270 Information Display System, 2740/2741, 3278, and 3767 in 2740/41 mode.

VSE/Fast Copy Data Set Program

The VSE/Fast Copy Data Set Program program product provides the same functions as the Fast Copy system utility in the DOS/VSE SCP and the following additional features:

- Full-track read for CKD disk devices using the read multiple command for devices that can execute this command
- Copy/restore of SAM files with relocation to a different disk area
- Partial volume dump/copy
- NOREWIND option for dump to tape to permit the dumping of several files to a single tape volume
- Multivolume file copy/dump/restore
- Selective restore of files from a dump volume

This utility supports all the tape and disk devices supported by DOS/VSE. It requires installation of the VSE/Advanced Functions Release 2 program product.

VSE/Access Control-Logging and Reporting Release 1

The VSE/Access Control-Logging and Reporting (VSE/LOGREP) program product provides improved data protection by auditing the access to specific data in a DOS/VSE installation. This program product works in conjunction with VSE/ICCF, which must be installed in order to use VSE/LOGREP.

The data protection functions provided when the VSE/LOGREP and VSE/ICCF program products are installed with DOS/VSE are:

- Determination of the data protection environment (user identification and authorization) and checking whether access to data, libraries, and programs is allowed or considered to be a security violation. Data is generated for each security violation and, optionally, any access to protected data. These functions are provided by DOS/VSE and VSE/ICCF.
- Formatted lists of security violations, of all access to protected data, or of both security violations and all access to protected data

VSE/Interactive Problem Control System

The VSE/IPCS program product is designed to assist in problem determination by providing a uniform mechanism for reporting and diagnosing programming failures. DOS/VSE contains facilities that collect problem data resulting from errors detected within the supervisor, subsystems, or problem programs. This data is recorded on a direct access device.

VSE/IPCS provides problem dump data management, creation of a problem description report with a symptom string, problem dump data display, special formatting of DOS/VSE and VSE/POWER control blocks, and assistance with APAR reporting. VSE/IPCS can execute either in a DOS/VSE partition or a VSE/ICCF partition.

USING DOS/VS RELEASE 34

A DOS/VS Release 34 operating system generated for a System/370 processor can be utilized in a 4341 Processor operating in System/370, EC, and DAT modes. Problem programs that execute under the Release 34 supervisor can operate in the 4341 Processor without change subject to the hardware constraints listed in Section 05:10.

The 4341 Processor can be set to normal mode for machine checks so that MCAR recording of machine check records and processor-independent recovery actions take place as for a System/370 processor. The logging of processor-dependent machine check data and the reference code to the system diskette occurs as usual before a machine check interruption is taken and MCAR receives control.

The operator console device utilized by the DOS/VS Release 34 supervisor can be a display device operating in 3270-compatible mode (but not 115/125 DOC mode), such as the 3278 Model 2A, or a 3210/3215 Console Printer-Keyboard. The 3287 Printer is not supported as a hard-copy device for display mode operations by DOS/VS Release 34. If the generated system utilizes a 3210/3215 printer-keyboard as the operator console device, the 3278 Model 2A must be used in printer-keyboard mode.

A 1400 emulator program that operates under DOS/VS Release 34 control cannot be used in a 4341 Processor. However, if the IBM Systems 1401/1440/1460 Emulator Program is obtained, a 1400 simulator program that can execute under DOS/VS Release 34 control in the 4341 Processor can be generated (see discussion in Section 40).

DOS/VS Release 34 does not support FBA devices. However, it does support the 3203 Model 5.

USING DOS RELEASE 26

A DOS Release 26 supervisor generated to support a System/360 processor can execute in the 4341 Processor operating in System/370 and BC modes without change subject to the hardware compatibility constraints listed in Section 05:10. Since RMS routines for the 4341 Processor are not present in a System/360 supervisor and the fixed logout area in the 4341 Processor is larger than in System/360 processors, the 4341 Processor should also be set to stop after log mode, using the check control display of the operator console.

In order to use DOS Release 26, which supports the 1052 Printer-Keyboard as the operator console device, the 3278 Model 2A Display Console must be used as a 1052 Printer-Keyboard. In addition, only I/O devices and hardware features of the 4341 Processor that are supported by DOS Release 26 can be utilized.

If 1400 simulation is to be performed, the IBM Systems 1401/1440/1460 Emulator Program must be obtained. A 1400 simulator program can be generated to operate with DOS Release 26 so that simulation of a 1400 system can be performed on a 4341 Processor (see Section 40).

DOS Release 26 is System/360-oriented and thus does not support the new hardware capabilities or I/O devices of the 4341 Processor. In addition, DOS/VSE and its program products provide a significant number of facilities that are not supported by DOS. Therefore, DOS should be used primarily as an aid for transition from a System/360 processor to the 4341 Processor. Table 30.05.1 lists the 4341 Processor hardware features and major I/O devices that DOS Release 26 does not support.

Table 30.05.1. 4341 Processor features and major I/O devices not supported by DOS Release 26

- Dynamic address translation, channel indirect data addressing, and ECPS:VSE mode. (That is, a virtual storage environment is not supported.)
- Time-of-day clock, clock comparator, and CPU timer. (The interval timer is used for the timing facilities available in DOS.)
- Monitoring and program event recording
- Display mode for the 3278 Model 2A Display Console and additional printers/displays that natively attach to the 4341 Processor
- Block multiplexing. (The block multiplexer channels will operate in selector mode.)
- 3370, 3330-series, 3340, 3344, and 3350 direct access storage. (The 2311 and 2314/2319 devices are supported.)
- 3400-series magnetic tape units. The 2400-series support can be utilized for these devices but the extended sense data is not recorded.
- 3505 Card Reader and 3525 Card Punch. (The 1442 and 1402 Card Read Punches are supported.)
- 3211, 3203, and 3800 printers. (The 1443 and 1403 Printers are supported.)

30:10 OS/VS1

As of Release 7, OS/VS1 supports 4341 Processors and provides several additional enhancements that are not specifically related to these processors. The 4341 Processor operating in System/370 mode will be supported by OS/VS1 Release 7. OS/VS1 does not support the 4341 Processor operating in ECPS:VSE mode. Modifications to OS/VS1 to support the 4341 Processor include recognition of its processor identification (as supplied by the STORE CPU ID instruction), support of the operator console (3278 Model 2A), and MCH/CCH support of the machine check and channel logout data.

When OS/VS1 determines it is executing in a 4341 Processor during initialization, it will perform processor-dependent initialization where required. For example, processor storage will be tested for errors during IPL processing, no extended logout area is defined, RMS routines are initialized to handle the machine check types and channel logouts that occur in the 4341 Processor, and I/O load balancing routines are set to use the 4341 Processor values.

OS/VS1 RMS routines need only be initialized to support the subset of logout provided by the 4341 Processor. The same level of support is provided for the machine check types and channel logouts that are implemented in the 4341 Processor as for the identical machine check types and channel logouts in System/370. The processor-independent EREP routine for 4300 Processors will be included in the OS/VS1 operating system generated for a 4341 Processor.

The following discusses OS/VS1 support of specific 4341 Processor features and I/O devices.

Instructions. The Assembler language translator supports all the instructions for the 4341 Processor except MOVE INVERSE and those instructions that operate only when ECPS:VSE mode is in effect.

Interval timer. Supported for timing facilities, except time of day, unless the extended timer option is included in the VS1 supervisor.

Time-of-day clock. Supported for time of day.

Clock comparator and CPU timer. Supported for job step timing and interval timing when the extended timer option is included in the VS1 supervisor.

Store and fetch protection. Both are supported.

Reference and change recording. The reference and change bits are used by the page replacement algorithm of the page supervisor.

Program event recording. This feature is not supported.

Monitoring feature. Supported by the Generalized Trace Facility.

Console support. The 3278 Model 2A Display Console is supported by the 3277 display routines that support the display console for Models 138 and 148. Display and printer-keyboard modes are supported and the same functions will be provided. The only difference is 3278 Model 2A support will handle a display of 20 lines instead of the 24 lines supported for the Model 138/148 display.

Alternate consoles and additional consoles are supported. Multiple Console Support (MCS) is required to support multiple consoles and Device Independent Display Operator Console Support (DIDOCS) is required to support display devices (such as the 3278 Model 2A) as primary and secondary consoles. MCS is also required to support the 3287 Printer.

Channels. The byte multiplexer channel and five block multiplexer channels are supported.

231X Direct Access Storage. The 2314/2319 is supported. The 2311 is not supported.

3330-series, 3340, 3344, 3350, and 2305 Model 2 Direct Access Storage. These devices are supported by all the disk access methods and for system data sets. Rotational position sensing is supported.

3370 Direct Access Storage. The 3370 is not supported.

| 3880 Storage Control. The 3880 with 3340/3344, 3330-series, and 3350
| disk storage attached is supported.

3203 Model 5 Printer. This printer is supported as are the 1403, 3211, and 3800 printers.

ECPS:VS1

ECPS:VS1, a portion of the standard Extended Control Program Support feature for the 4341 Processor, is designed to improve the performance of VS1 by providing for the execution of certain frequently used VS1 supervisor routines in hardware to reduce the amount of processor time used by the VS1 supervisor. The reduction that results from the use of ECPS:VS1 in the 4341 Processor is up to 7 percent.

Unlike ECPS:VM/370, which is enabled and disabled using bits in control register 6, ECPS:VS1 is always available for use when it is activated during an IML. The use of ECPS:VS1 and ECPS:VM/370 in a 4341 Processor at the same time is not possible and ECPS:VS1 cannot be used by an OS/VS1 operating system that is executing in a virtual machine.

Privileged instructions are defined for the VS1 supervisor routines that are supported by ECPS:VS1. These instructions are six bytes in length and have a storage-to-storage format and the same operation code to identify them as VS1 assist instructions. An extended operation code byte in each instruction uniquely identifies each individual instruction.

These privileged instructions are valid only when supervisor state and storage protect key 0 are in effect. An operation exception program interruption occurs if an attempt is made to execute a VS1 assist instruction in a System/370 processor that does not have a VS1 assist or a 4341 Processor for which ECPS:VS1 was not loaded during IML.

The ECPS:VS1 hardware instructions are designed to be used only in the VS1 operating system and are not provided for general use in installations. Assembler Language mnemonics are not provided for these instructions.

Several ECPS:VS1 instructions are provided that perform the same functions as certain VS1 supervisor routines in the following functional areas:

- CCW translation
- Fixed page frame list scan
- Short-term fix and unfix
- GETMAIN and FREEMAIN routines in the I/O supervisor
- SVC first-level interruption handler
- Dispatcher
- Trace recording
- Identify feature level

ECPS:VS1 for the 4341 Processor provides a subset of the support provided by the VS1 hardware assist function for System/370 Models 138, 148, 135 Model 3, and 145 Model 3. ECPS:VS1 is functionally equivalent (with a few minor exceptions) to the OS/VS1 Extended Control Program Support feature for the System/370 Model 158 and 3031 Processor Complex. Page measurement, page supervisor queuing routines, invalidate page table entry, and GETMAIN/FREEMAN virtual storage management functions provided in the Model 138/148 VS1 hardware assist function are not provided by ECPS:VS1 for the 4341 Processor.

Support of ECPS:VS1 must be specifically requested when a VS1 supervisor is generated. The default is not to include such support in the VS1 resident nucleus. Three system generation options are provided for ECPS:VS1 support:

- Exclude support of the assist (the default). This option enables a VS1 supervisor to execute in any processor it supports without use of a VS1 assist feature if it is present in the processor.

- Include support of the assist using the VS1 assist privileged instructions only. The VS1 supervisor is generated with the VS1 assist privileged instructions and without the VS1 supervisor routines they replace. This option provides the best performance improvement but makes the VS1 supervisor portable only among processors that have a similar VS1 assist feature.

  Note that a VS1 supervisor generated for a 4341 Processor that contains only the ECPS:VS1 privileged instructions can also execute in a 3031 Processor and Models 135 Model 3, 138, 145 Model 3, 148, and 158 and can utilize the VS1 hardware assist function in these processors. Similarly, a VS1 supervisor generated for a 3031 Processor or Model 158 can execute in a 4341 Processor and utilize ECPS:VS1. However, a VS1 supervisor with only the VS1 assist privileged instructions that is generated for a Model 135 Model 3, 138, 145 Model 3, or 148 cannot execute in a 4341 Processor, Model 158, or 3031 Processor, since the VS1 assist in the latter three processors is only a subset of that implemented in the other four processors.

- Include support of the assist using both the VS1 assist privileged instructions and the supervisor routines they replace. This option makes the VS1 supervisor portable among processors with and without a VS1 assist feature and enables the supervisor to be used if the VS1 assist feature is malfunctioning. Such a VS1 supervisor is slightly larger than one that includes only the VS1 assist instructions and provides a smaller performance increase than the above option. The performance reduction results from the larger size of the resident VS1 supervisor, which reduces the amount of pageable real storage available, and the execution of additional instructions during system operation to determine whether to use the VS1 assist instructions or the supervisor routines.

When a VS1 supervisor with VS1 assist support is IPLed, it determines whether the processor has a VS1 assist feature by issuing one of the privileged instructions. If the assist function is present and the supervisor contains only the VS1 assist instructions, the supervisor is initialized to use the VS1 assist feature. If a VS1 assist feature is not present and the supervisor contains only the VS1 assist instructions, an error message is issued and the processor enters the wait state.

When a supervisor containing both the VS1 assist instructions and their corresponding supervisor routines is IPLed in a processor with the VS1 assist feature, the VS1 supervisor issues a message asking the

operator whether or not the VS1 assist instructions should be used. The reply determines whether VS1 assist instructions or the supervisor routines are activated (a bit in the communications vector table that is tested during system operation is set).

When the processor does not have a VS1 assist and a VS1 supervisor with both the VS1 assist instructions and their corresponding supervisor routines is IPLed, the supervisor routines are automatically activated.

30:15 VM/370

Support of the 4341 Processor and its I/O devices is provided by the CP component of VM/370 and the VM/Basic System Extensions Release 2 program product. As of Release 6, the CP component of VM/370 supports the 4341 Processor as a real machine in which it executes. The 3278 Model 2A Display Console is supported as the console for the real machine, as an alternate console, and as a virtual operator's console. Both display and printer-keyboard modes are supported for the 3278 Model 2A.

CP supports the 3203 Model 5 Printer, as of Release 6, for the same functions as other supported printers. ECPS:VM/370 is also supported. Various other functional improvements have been made to CP as well.

The operating systems that can execute in a virtual machine under Release 6 of VM/370 are DOS/VSE, OS/VS1 Release 7, VM/370 Release 5, and the operating systems supported by VM/370 Release 5. A DOS/VSE supervisor that executes in a virtual machine must support System/370 mode.

Support of certain I/O devices that attach to the 4341 Processor is provided by Release 2 of VM/Basic System Extensions (VM/BSE), which also provides other support not included in the VM/BSE Release 1. VM/BSE Release 2 requires VM/370 Release 6 as a base. VM/BSE Release 2 contains all the functions provided by VM/BSE Release 1 and the following new functions:

- Support of FBA disk storage (the 3370 for the 4341 Processor) for the same functions as other supported direct access device types
- CMS/DOS uplevel to DOS/VSE. The CMSDOS discontiguous shared segments are updated to support the DOS/VSE SVCs and VSE/VSAM and DOS/VSE files on FBA devices.
- Interactive help facility in a CMS virtual machine that guides the operator in using CP and CMS commands. The virtual operator's console must be a display device.
- Several improvements to CMS file management, including support of FBA devices, removal of existing limitations on CMS disk size and number of files per CMS disk, and support of more physical block sizes for CMS files (512, 800, 1024, 2048, and 4096 bytes)
- Larger block size (4K) for CMS tape command dump/load operations
- CMS use of CP page management interfaces to use CP paging facilities more efficiently
- CP performance improvements
- Option to generate a smaller resident CP (by making more of the CP pageable) to increase the real storage available for paging
- Improvements in 3270 Information Display System support

Many new program products are available for operation with VM/370 Release 6. Certain of these program products provide more support than their counterpart components in the VM/370 Release 6 SCP.

The Remote Spooling Communications Subsystem (RSCS) Networking program product provides more function than the RSCS component of VM/370 and the VM/370 Networking PRPQ. The RSCS Networking program product can be used together with the RSCS component in a given VM/370 system but not with the VM/370 Networking PRPQ.

The VM/Directory Maintenance program product improves the VM/370 directory processing facilities provided by CP. The VM/Interactive Problem Control System Extension (VM/IPCS Extension) program product provides capabilities not included in the IPCS component of VM/370. The Display Management System/CMS (DMS/CMS) and VM/Interactive File Sharing (VM/IFS) program products are provided to extend the capabilities of CMS. Both require installation of VM/BSE Release 2.

VM/370 support of ECPS:VM/370 for the 4341 Processor and the VM/370 assist features available for other processors is designed to enable VM/370 systems that are generated with and without support of these features to operate in processors with and without the features installed.

When a VM/370 system that contains VM/370 assist support is IPLed, CP issues the STECPSVM privileged instruction of the Control Program Assist component to determine whether ECPS:VM/370 for a 4300 Processor or the VM/370 hardware assist function for Models 135 Model 3, 138, 145 Model 3, and 148 is present in the processor. If either assist is present (instruction executes without an operation exception program interruption), CP compares the identification level code stored to its own identification level code to determine whether they are the same. If so, CP enables the assist (all components). If the level codes are not the same, CP issues an error message and does not enable the assist.

When the STECPSVM instruction causes an operation exception program interruption, ECPS:VM/370 or the VM/370 hardware assist function is not present in the processor. CP then changes all the Control Program Assist privileged instructions it contains to NOP instructions so that it can operate without causing further operation exception interruptions via these instructions. CP then checks for the presence of the Virtual Machine Assist feature, which is a component of ECPS:VM/370 and available for most System/370 processors. If it is present, CP enables the Virtual Machine Assist feature.

These checks enable a VM/370 system with ECPS:VM/370 or VM/370 hardware assist function support to operate in 4341 Processors, 4331 Processors with and without ECPS:VM/370 installed, System/370 processors with the VM/370 hardware assist function, System/370 processors with the Virtual Machine Assist feature, and System/370 processors without Virtual Machine Assist installed.

Similarly, a VM/370 system that supports only the Virtual Machine Assist feature can operate in a 4300 Processor with ECPS:VM/370 or System/370 processor with the VM/370 hardware assist function with only the Virtual Machine Assist component enabled (bits 0, 6, and 7 in control register 6 set to 100). A VM/370 system that does not support a VM/370 assist feature operates in a 4341 Processor without any ECPS:VM/370 component enabled.

40:05 THE IBM SYSTEMS 1401/1440/1460 EMULATOR PROGRAM

GENERAL OPERATION

The 4341 Processor continues the advantages of integrated emulation available to intermediate-scale System/370 processor users. The IBM Systems 1401/1440/1460 Emulator Program (program product) is provided to support 1400 emulation for 4300 Processors. The emulator program product can be used to generate a 1400 simulator program that executes as a problem program under DOS Release 26, DOS/VS Release 34, or DOS/VSE in a 4341 Processor.

A generated 1400 simulator can be used in a batch-only system environment or can operate in the background and batched foreground partitions of a multiprogramming system. Therefore, multiple 1401/1440/1460 emulator programs can execute concurrently with each other and with 4341 Processor programs. Additionally, 1400 simulator jobs and DOS, DOS/VS, or DOS/VSE jobs can be intermixed in a single jobstream.

The 1401/1440/1460 simulator for 4341 Processor consists of the following basic parts:

- Instruction simulator module that executes all 1401/1440/1460 instructions except EDIT, HALT, input/output, and invalid instructions
- Input/output simulation module that recognizes and handles all I/O instructions issued by 1401/1440/1460 programs. Control is transferred to the DOS interface module, which actually executes the I/O operations requested.
- DOS interface module that executes all requested 1401/1440/1460 I/O operations by creating the necessary interface to DOS, DOS/VS, or DOS/VSE data management routines. For certain functions (tape simulation, column binary operations, CS/30/40-format disk simulation, and fixed block architecture format disk simulation), the EXCP level is used. This module is dependent on the specific operating system, DOS Release 26, DOS/VS Release 34, or DOS/VSE, that is to be used with the 1400 simulator program.
- Operator's interface routines that handle operator commands, error messages, initialization processing, and messages to the operator (including handling of the 1401/1440/1460 HALT instruction)

The IBM Systems 1401/1440/1460 Emulator Program offers 4341 Processor users the following advantages:

- Simulators can run concurrently in all partitions of a multiprogramming system. They are relocatable and can be link-edited to run in any partition.
- 1401/1440/1460 simulator programs and DOS, DOS/VS, or DOS/VSE programs can be executed concurrently and intermixed in a single jobstream.
- DOS, DOS/VS, or DOS/VSE supervisor and data management services are available to the user. This provides job control facilities,

standard disk and tape label processing, and common data formats for simulator files and DOS, DOS/VS, or DOS/VSE files.

- 1400 unit record input/output operations can be made device-independent and can be emulated on 4341 Processor unit record devices, magnetic tape units, or direct access storage devices.

Simulator Program Generation and Execution

The specific 1401/1440/1460 simulator program required by a given 4341 Processor installation must be generated. A 1401/1440/1460 simulator program can be generated to execute in the 4341 Processor in one of the following environments with the specified DOS, DOS/VS, or DOS/VSE supervisor:

- System/370 and BC modes with DOS Release 26
- System/370, EC, and DAT modes with DOS/VS Release 34 and DOS/VSE
- ECPS:VSE and EC modes with DOS/VSE

A different simulator program must be generated to support each of the three environments listed above. The simulator program product package consists of a set of simulator program modules, a sample 1400 program with control cards to be used to verify correct operation of the generated simulator program, the pre-postprocessor utility for converting tape files, and an initialize disk utility that must be used to initialize and preformat any fixed block architecture devices that are to be used for emulating 1400 disk files. The package is distributed in standard SYSIN format with the control statements necessary to catalog the simulator modules in the appropriate DOS, DOS/VS, or DOS/VSE libraries.

A simulator is assembled by the use of macro instructions. The macro instructions describe the 1400 CPU, input/output devices, special features, data files, simulator buffers, and the desired user options. When assembled, the macros provide an object module and linkage to preassembled modules stored in the system relocatable library. The preassembled modules are combined with the simulator object module by the linkage editor for cataloging in a core image library. Any number of 1400 simulators can be assembled and cataloged in a core image library to run in any partition.

The simulator generation macros used for a 1400 simulator for the 4341 Processor are the same as those used to generate a 1400 simulator for a System/370 processor except for the addition of a MODE macro (to indicate in which of the three environments the simulator will execute, as described above) and the removal of a few parameters for System/370 options that are standard or not supported by the 1400 simulator for the 4341 Processor.

The simulator program generated will emulate, without change, 1400 programs written in accordance with IBM 1400 Principles of Operation manuals, subject to the following conditions:

- 1400 programs that purposely depend on the absence of a 1400 feature or on error conditions may not execute properly.
- Programs with undetected programming errors and those that depend on timing of 1400 I/O operations yield unpredictable results.

A simulator program is handled by DOS, DOS/VS, or DOS/VSE in the same manner as any problem program. When using the 1401/1440/1460 simulator,

1400 programs may be cataloged to, and fetched from, a core image library for execution or loaded from a card, tape, or direct access storage device. Standard DOS, DOS/VS, or DOS/VSE job control statements are used to prepare the system for a simulator job. The EXEC job control statement causes the specified simulator program to be loaded and control is passed to the simulator program.

Emulator control statements are read by the simulator from the card reader or can be entered via the operator console. CS-format control cards are also accepted by the simulator.

Emulation with the 1401/1440/1460 simulator consists of three main steps:

1. Initialization. Emulator control statements supplied by the user are read and interpreted. This information overrides, for the execution of the simulator, information specified at simulator generation.

2. Loading or fetching. The 1400 program is loaded from a card reader, magnetic tape unit, or direct access storage device. A 1400 program can also be fetched from a core image library if it has been cataloged.

3. Execution or precataloging. When loaded, the 1400 program is executed. The 1400 instructions are fetched, interpreted, and executed by the simulator until an end-of-job condition is recognized. The 1400 program can either be executed or converted to a DOS, DOS/VS, or DOS/VSE object module (precataloged). This module can be subsequently link-edited and cataloged in a core image library.

Input/output errors are processed by DOS, DOS/VS, or DOS/VSE device error recovery procedures. Input/output errors that cannot be corrected, such as permanent input/output errors and wrong-length records, are passed to the 1400 program.

Console simulation and operator communication with the simulator program are provided by the exchange of simulator commands and messages between the operator and the simulator program. The simulator provides messages to inform the operator of errors or other conditions that require his attention or a response. Simulator commands can be entered from the operator console keyboard and are handled in the same way as operator communications are handled by the DOS, DOS/VS, or DOS/VSE release being used.

User-written routines can be included in the generated 1400 simulator program to support the following:

- 1400 operation codes not supported by the simulator (unsupported valid I/O operation codes and invalid 1400 operation codes)

- Any I/O operation

- EDIT and HALT instructions

- The console user exit. When a console user exit routine is provided, all messages issued by the simulator or the 1400 program being simulated are passed to the exit routine instead of being displayed on the operator console. The routine must determine the content of the message and return a response to the simulator. Such a routine can be used to avoid having the operator respond to simulator program messages.

A set of diagnostic commands (like those provided for System/370 1400 simulators) is provided that can be used during simulator program execution to obtain information about the job step in progress or to modify the 1400 program being simulated. In addition, the DOS/VS or DOS/VSE serviceability aids can be used without restriction with simulator programs.

The following diagnostic commands are provided:

- CONVERT to convert a 1400 core decimal address to its corresponding hexadecimal address in the mapped storage area
- CLEAR to reset the 1400 core to a given character with or without a wordmark
- SET to set any 1400 register to a given value
- TN/TP to either turn on or turn off any sense switch or the inquiry indicator
- ALTER to modify any part of the 1400 emulated core storage area
- DISPLAY to display the value, contents, setting, or assignment of the 1400 sense switches, registers, or status
- DUMP to dump part or all of (1) processor storage in the 4341 Processor in hexadecimal format using the PDUMP macro or (2) the 1400 storage in decimal format
- DEBUG to emulate the 1400 address stop and step-to-step functions. In addition, this command prints on SYSLST each 1400 instruction executed with the contents of the 1400 registers and the executed instruction.

Tape and Disk Emulation

The option of processing tape files in 1400 format or in spanned variable-length record format is provided. A tape formatting program, the pre-postprocessor utility, is provided to convert tape files from 1400 format to spanned record format, and vice versa. Mixed-density tapes are not supported by the simulator or the tape formatting utility.

The simulator accepts as input and produces as output two tape file formats:

1. 1400 format, which is produced by a 1400 system, a standalone simulator, CS/30, CS/40, the tape pre-postprocessor utility, a System/370 1400 emulator/simulator, a 1400 emulator for a 4331 Processor, or a 1400 simulator for a 4341 Processor
2. Spanned variable-length record format, which is produced by the tape pre-postprocessor utility, a System/370 1400 emulator/simulator, a 1400 emulator for a 4331 Processor, or a 1400 simulator for a 4341 Processor. A physical tape record can be a maximum of 32,767 bytes in length.

Processing tape files in spanned variable-length record format provides several advantages:

- Blocking short records reduces the time for emulating I/O operations.
- The pre-postprocessor utility can be run concurrently with simulator programs in a multiprogramming environment.

- Files in spanned variable-length record format can be used by 4341 Processor programs if the programs provide for handling the 1400 label records and 1400 tapemark records.

- The pre-postprocessor utility can be used to convert a file in spanned variable-length record format back to 1400 format for use in a 1400 system.

Tape files in spanned record format have standard DOS, DOS/VS, or DOS/VSE labels and 1400 labels are treated as data records, since they are processed by the 1400 program. The 1400 tapemarks appear as special data records and are recognized by the 1400 simulator. Tape I/O operations are executed using the EXCP macro.

The character codes supported by the 1400 simulator for magnetic tape data are:

- BCD representation in even and odd parity for seven-track tape (data translator on) in 1400 format

- BCDIC-8 representation for nine-track tapes in either 1400 or spanned record format, and for seven-track tapes (data converter on) in spanned record format. This character code, which is the eight-bit representation of BCD, is used to simulate parity. In normal mode, bit 1 is set to one for even parity, to zero for odd parity. In alternate mode, bit 1 is always set to one and no distinction is made between even and odd parity.

The pre-postprocessor utility is provided for converting tape files. The preprocessor portion converts seven-track or nine-track tapes in 1400 format to seven-track (data converter on) or nine-track tapes in spanned variable-length record format with standard DOS, DOS/VS, or DOS/VSE labels. The postprocessor portion converts seven-track or nine-track tapes in spanned record format to seven-track or nine-track tapes in 1400 format. This utility operates under operating system control in a partition.

The simulator accepts three disk formats: CS simulator; count, key, data; and fixed block architecture. The DOS, DOS/VS, or DOS/VSE Clear Disk utility program is required to preformat 4341 Processor count, key, data type disk volumes that are used to emulate 1400 disk files. Each 4341 Processor disk record represents one 1400 disk track. Each 4341 Processor disk record is fixed-length, its length being a function of the emulated 1400 device and mode rather than the amount of 1400 data on each track.

The disk utility provided with the simulator package must be used to format fixed block architecture devices used for 1400 file emulation. Each 1400 disk track is mapped to a consecutive group of fixed blocks. The number of blocks required is calculated at simulator generation time based on the 1400 device type.

A 1400 disk file can occupy one or more extents on 4341 Processor disk volumes but only one extent per volume. Extents must be allocated complete cylinders. When a file requires more than one 4341 Processor disk volume, the 4341 Processor volumes must be the same type. Two different 1400 files can be placed on the same disk volume but this arrangement may increase seek time if both files are processed at the same time.

Character codes supported by the 1400 simulator for disk files are:

- EBCDIC representation for disk operations in move mode
- BCDIC-8 representation for disk operations in load mode. (Data written in load mode must be converted to EBCDIC if it is to be used by programs other than the simulators.)

Disk files in 1400 format, which are created on a 1400 system or under standalone emulation, must be converted to a standard fixed-length record format on a disk volume of a type that attaches to the 4341 Processor before emulation. Disk files created under CS/30 or CS/40 can be processed by the 1401/1440/1460 simulator if the CS option is specified at simulator generation and 2311- or 2314-type disks are used for emulation.

To convert disk files in 1400 format, or CS/30 or CS/40 disks if desired, to a standard format on a 4341 Processor disk volume, the user must dump and restore the data as follows:

1. Dump the disk device, using a 1400 disk-to-tape or disk-to-card utility program. When converting files on 1301, 1311, or 1405 disk devices that were created on a 1400 system, the utility is executed on the system used to create the file. When converting files on 2311 or 2314 disks that were created under standalone emulation, CS/30, or CS/40, the utility is executed on a System/360 under control of the simulator used to create the disk file.

2. Use the appropriate DOS, DOS/VS, or DOS/VSE disk utility program to format the previously initialized disk volumes to be used for the 1400 data.

3. Restore the 1400 data to a formatted 4341 Processor disk volume, using a 1400 tape-to-disk or card-to-disk utility program under control of a 1400 simulator for a 4341 Processor.

The direct access method (DAM) or EXCP macro is used to execute 1400 disk operations, depending on the format of the disk data. DAM is used to process 1400 disk files emulated on count, key, data devices. EXCP is used for CS-format disks emulated on count, key, data devices and for 1400 files emulated on fixed block architecture devices.

Emulator performance will vary depending on user options, such as number and size of buffers, the instruction mix of the 1401/1440/1460 programs, the format of tape files, and the priority of the partition in which the simulator is running.

Emulator performance is improved by:

1. Using double buffers and spanned record format for tape files in lieu of single or shared buffers and 1400 record format. A shared buffer can be used by more than one I/O device. The shared buffer option for tape is provided primarily for users of DOS Releases 26 and 27 (which do not support virtual storage) to reduce tape buffer storage requirements. The shared buffer option is not provided for use with emulated disk files.

2. Specifying device independence for emulating unit record operations on a magnetic tape or direct access storage device. (The sequential access method is used to emulate 1400 unit record operations on 4341 Processor unit record devices.)

3. Generating the simulator without support for the 51-Column Interchangeable Read Feed and Column Binary features and the select stacker instruction

SUPPORT OF 1401/1440/1460 FEATURES

The size of the partition required for emulation depends on the 1400 system being emulated, including standard and special features, input/output devices, buffers, etc. The processor storage required for the 1401/1440/1460 simulator is equal to the combined sizes of:

- Simulated 1401/1440/1460 storage. Each position of 1400 storage is simulated in one byte of 4341 Processor storage (for example, 8000 positions = 8000 contiguous bytes). This storage is marked permanently fixed during simulator program execution. Note that the communication region for the simulator is also permanently fixed.
- Emulator routines required to emulate the 1401/1440/1460 system instructions, features, and I/O operations
- Tape, disk, and unit record buffers. The number and size of tape and disk buffers are specified by the user.

Approximate 1401/1440/1460 simulator program processor storage requirements for simulation of a 1400 system with unit record operations only, unit record/tape operations, unit record/disk, and unit record/tape/disk operations are shown below.

Emulated 1400 Configuration	Storage Requirement in Bytes
1401 unit record system with 8000 positions of storage	25,000
1401 unit record/six tape system with 16,000 positions of storage	41,000
1440 unit record/three disk system with 12,000 positions of storage	36,000
1401 unit record/two tape/two disk system with 12,000 positions of storage	43,500

The storage requirement given for the four configurations includes storage required for emulated 1400 core storage, data management routines, support of the advanced programming feature, support of sense switches, standard (not CS) format for 1311 disks, 1400-format tape files, two 1000-byte tape buffers, one 2164-byte disk buffer, and one 440-byte unit record buffer.

The 1400 CPU features and 1400 I/O devices and special features supported and the 4341 Processor devices used for 1401/1440/1460 emulation are given in Tables 40.05.1 and 40.05.2. Table 40.05.3 lists the 1400 I/O devices that are not supported.

Table 40.05.1. 1401/1440/1460 I/O devices and features supported by the IBM Systems 1401/1440/1460 Emulator Program and corresponding 4341 Processor devices

1401/1440/1460 Device and Features	Corresponding 4341 Processor Device
1402, 1442, 1444 Card Read Punch with stacker selection The following are supported: • Column Binary or Card Image • 51-Column Interchangeable Read Feed • Punch Feed Read • Punch Column Skip • Binary Transfer • Processing Overlap • Read Punch Release Not supported: • Multiple card reader/punch operations in one program	1442, 2520, 2540 Card Read Punch, 2501 Card Reader, 3505 Card Reader, and 3525 Card Punch Note: Card reader and card punch operations may be emulated using a magnetic tape or direct access storage (SYSRDR, SYSPCH, or SYSIPT) device.
1403, 1404, and 1443 Printers The following are supported: • Numerical Print • Processing Overlap • Space Suppression Not supported: • Selective Tape Listing • Multiple printer operations • Cut-card operations and read compare	1403, 1443, 3211, or 3203 Model 5 Printers Note: Printer operations may be emulated using a magnetic tape or direct access storage (SYSLST) device.
1407, 1447 consoles	Operator console (3278 Model 2A Display Console)
729, 7330, and 7335 Magntic Tape Units The following are supported: • Binary tape instructions • Processing Overlap Compressed-tape and mixed-density tape are not supported.	2400- and 3400-series magnetic tape units (any density) • Seven-Track feature is required if processing seven-track tapes.
1301, 1311, and 1405 Disk Storage The following are supported: • Direct Seek • Scan Disk • Track Record • Additional access arm (1405) A 1405 cannot be emulated in combination with a 1301 or 1311.	2311, 2314/2319, 3330-series, 3340/3344, 3350, and 3370 direct access devices

Table 40.05.2. 1401/1440/1460 CPU features supported by the IBM Systems 1401/1440/1460 Emulator Program

Storage from 1400 to 16,000 positions	Multiply-Divide
Expanded Print Edit	Sense Switches
Inverted Print Edit	Advanced Programming
High-Low-Equal Compare	Indexing and Store Address Register
Move Binary Code and Decode	Bit Test

Note: Translate feature is not supported.

Table 40.05.3. 1401/1440/1460 devices not supported by the IBM Systems 1401/1440/1460 Emulator Program

1445 Printer	1404 Printer in cut-card mode
Paper tape readers	7340 Hypertape Drive
Paper tape punches	Teleprocessing devices
Magnetic character readers	Audio response units
Optical character readers	

SECTION 50: RELIABILITY, AVAILABILITY, AND SERVICEABILITY (RAS)

50:05 INTRODUCTION

With the growth of more and more online data processing activities, as distinguished from traditional batch accounting functions, the availability of a data processing system becomes an essential factor in company operations, and complete system failure is extremely disruptive. Because of the growing frequency of online processing and the fact that the 4341 Processor is designed to operate in such an environment, extensive availability and serviceability features are implemented for the 4341 Processor and its reliability is improved.

The objectives of the RAS features of the 4341 Processor are to (1) reduce the frequency and impact of system interruptions that are caused by hardware failure and necessitate a re-IPL and (2) reduce the time required to locate and repair malfunctions. RAS features are as follows:

- Hardware reliability is enhanced through use of inherently more reliable technology.
- Recovery facilities, both hardware and program supported, not available for System/360 Models 50 and 65, are provided to reduce the number of failures that cause a complete system termination. This permits deferred maintenance.
- Diagnostic facilities not available for System/360 or intermediate-scale System/370 processors are provided in addition to commonly used diagnostic procedures. These facilities are designed to reduce problem location and repair time.

Each availability and serviceability feature is discussed in the remainder of this section. The following recovery/repair features are implemented in hardware:

- Automatic retry of instructions when an instruction processing function error occurs during instruction execution
- ECC validity checking on processor storage to correct all single-bit and detect all double-bit and many multiple-bit errors
- I/O operation retry facilities, including channel retry data provided in the limited channel logout area and channel/control unit command retry procedures to correct failing I/O operations
- Expanded machine check interruption facilities to support better error recording and recovery procedures
- Machine check error diagnosis (reference code generation) and logging done by the support processor to aid the customer engineer in faster problem determination and to provide the ability to record errors even when the instruction processing function malfunctions
- Microcode-controlled power sequencing and power and temperature monitoring performed by the support processor

The following recovery features are provided by programming systems:

- Recovery management support (RMS) to handle the expanded machine check interruption and channel retry data. Machine check analysis and recording (MCAR) and channel check handler (CCH) routines for the 4341 Processor are provided in DOS/VSE. Machine check handler (MCH) and CCH routines for the 4341 Processor are provided in OS/VS1.

- Error recovery procedures (ERP) to retry failing I/O device and channel operations utilizing channel logout data (DOS/VSE and OS/VS1)

- Outboard recorder (OBR) and statistical data recorder (SDR) routines (OS/VS1) and DOS/VSE recovery management support recorder (RMSR) to record statistics for I/O errors

- Environment recording, edit, and print program (EREP) for DOS/VSE and OS/VS1 to format and print error log records

- I/O RMS routines (OS/VS1)--alternate path retry (APR) and dynamic device reconfiguration (DDR)--to provide additional recovery procedures after channel or I/O device failures

- Checkpoint/restart (DOS/VSE and OS/VS1) and warm start facilities (OS/VS1) to simplify and speed up system restart procedures after a failure necessitates a re-IPL

The following diagnostic facilities are provided:

- Online Test Executive Program (OLTEP) and Online Tests (OLTs) that execute under operating system control (DOS/VSE and OS/VS1) and provide online diagnosis of channel-attached I/O devices for most devices that attach to the 4341 Processor

- System test for checking the basic operating capability of the 4341 Processor

- Microdiagnostics for the components of the 4341 Processor (instruction processing function, processor storage, support processor, etc.)

- Manual operations that the customer engineer can perform using the operator console and appropriate support documentation

- A Remote Support Facility that enables the on-site customer engineer to access a remote data bank and allows malfunction diagnosis to be performed by a remote customer engineer at a support center

The hardware and programmed recovery aids are designed to improve system availability. In many cases, the system can run in a degraded mode so that maintenance can be deferred. When solid failures do occur, their impact can be reduced by utilizing the expanded diagnostic facilities, which can provide faster isolation and repair of the malfunction than is possible for System/360.

The diagnostic facilities for the 4341 Processor also represent improvements in the facilities available for System/370 Models 125 to 145. Specifically, the 4341 Processor offers the following:

- More extensive diagnostics with better problem isolation capabilities than those for Models 125 to 145

- More checking circuits included in the basic design of the processor than in Models 125 to 145

- Microcode-controlled power sequencing and power and temperature monitoring (not implemented in Models 125 to 145)
- Reference code generation to aid in faster malfunction diagnosis and repair (not implemented in Models 135 to 145)
- Remote Support Facility to aid in diagnosing problems (not implemented in Models 125 to 145)

50:10 RECOVERY FEATURES

Additional hardware, which attempts correction of many hardware errors without programming assistance, is included as a basic part of the 4341 Processor. The control program can be notified, via an interruption, of both intermittent and solid hardware errors so that error recording and recovery procedures can take place.

AUTOMATIC INSTRUCTION RETRY

If a hardware error is detected during the execution of certain instructions, the failing instruction is automatically retried once by the processor without programming assistance. The data required for a retry is saved during the execution of instructions.

The following instructions are retryable at any point during their execution (see System/370 Principles of Operation for the instructions in each instruction group):

- All general instructions except SET PROGRAM MASK (includes binary arithmetic instructions)
- All decimal and floating-point instructions
- Control instructions whose names begin with STORE except for STORE THEN AND SYSTEM MASK and STORE THEN OR SYSTEM MASK

The instructions not contained in the groups listed above are initially retryable but become unretryable at some point during their execution.

If the instruction retry is successful, a machine check interruption is taken, if the processor is enabled for recovery interruptions, so that error recording can be done. If the instruction cannot be retried at all or if the retry is unsuccessful in correcting the error, a machine check interruption occurs, if the processor is enabled for such interruptions, and programmed error recovery procedures should be executed.

The instruction retry feature provides the 4341 Processor with the ability to recover from many intermittent processor failures that would otherwise cause a system halt and necessitate a re-IPL or that would cause an executing program to be terminated. Corrected errors are logged by recovery routines for later diagnosis, thereby increasing system availability. Retry of failing processor operations in Models 50 and 65 is not provided by hardware.

ECC VALIDITY CHECKING ON PROCESSOR STORAGE

The ECC method of validity checking on processor storage provides automatic single-bit error detection and correction. It also detects all double-bit and many multiple-bit processor storage errors but does not correct them. Checking is handled on an eight-byte basis, using an eight-bit code, rather than on a single-byte basis, using a single parity bit. However, parity checking is still used to verify other data in a 4341 Processor that is not contained in processor storage. Models 50 and 65 use parity checking for main storage data verification.

As data enters and leaves storage, ECC logic performs validity checking on each doubleword. When a doubleword (72 bits, as shown in Figure 50.10.1) is fetched from processor storage, the eight-bit ECC code is checked to validate the 64 data bits. If the data is correct, the appropriate parity bit for each of the eight data bytes is generated and the doubleword is reformatted to look as shown in Figure 50.10.2. If a single-bit error is detected, the identified data bit in error is corrected automatically by ECC logic.

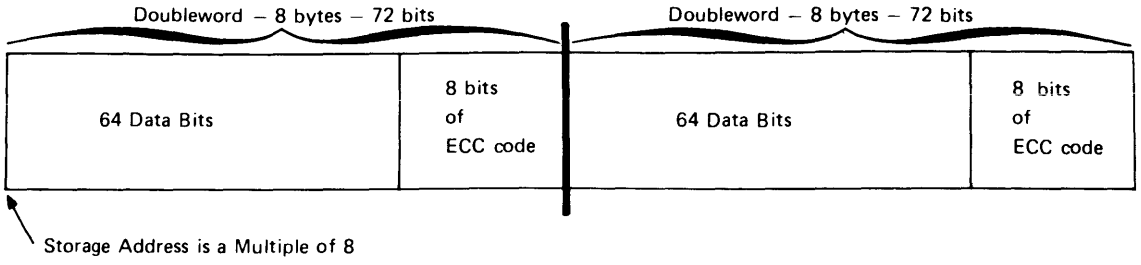
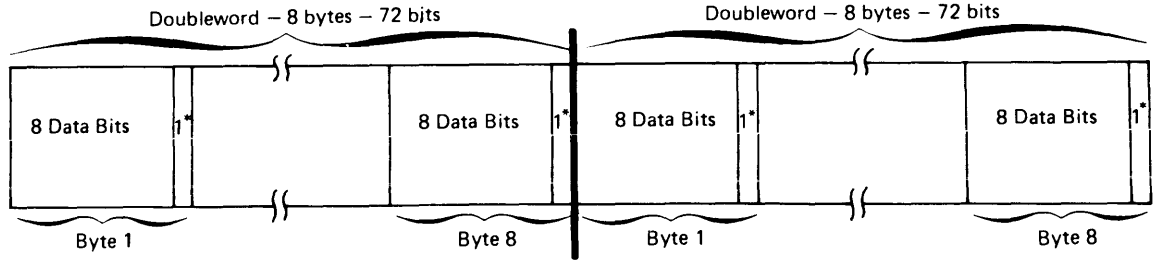


Figure 50.10.1. Data representation used in processor storage in the 4341 Processor



*Parity Bit

Figure 50.10.2. Data representation used in Model 50 and 65 processor storage and in the 4341 Processor in other than processor storage

When a doubleword is to be placed in processor storage by a program, the eight parity bits are removed and the eight-bit ECC code is generated and appended to the 64 data bits. The 72 bits are then stored as shown in Figure 50.10.1.

If a double- or multiple-bit error occurs during instruction execution and the instruction is retryable, it is retried one time. If the storage error is of an intermittent type and a single-bit error results after the retry, the error is corrected as usual and processing continues.

A machine check interruption occurs, if the processor is enabled for these interruptions, after the occurrence of a double- or multiple-bit storage error. However, a single-bit correction in processor storage does not cause a machine check condition.

If a machine check interruption is taken after a double- or multiple-bit storage error, identification of the failing processor storage address is provided in a fixed storage area (discussed in the machine check interruption explanation).

When a double- or multiple-bit processor storage error occurs during an I/O operation, it is reported during the ensuing I/O interruption so that error recording and I/O retry procedures can be executed.

The ECC feature increases the availability of the 4341 Processor by permitting system operation to continue normally after single-bit processor storage errors occur and are corrected. Any processor storage errors on Models 50 and 65 necessitate at least termination of the processing program involved, since neither hardware nor programmed retry of processor storage errors is provided for these systems.

I/O OPERATION RETRY

Channel retry and command retry features (like those for System/370 processors) are provided to reduce the number of abnormal program terminations and unscheduled system halts that occur because of I/O errors.

Channel retry is implemented to ensure that most failing channel operations can be retried by error-handling routines. A limited channel logout area is implemented. When a channel error or a processor error associated with a channel operation occurs, the channel status word (CSW) and a limited channel logout word are stored in the fixed lower processor storage area (locations 176 to 179) during the I/O interruption. The limited channel logout data provides additional, more exacting status information about the channel failure. The CCH routine passes this data to a device-dependent error recovery routine to be used in the retry of the failing I/O operation.

Channel error retry routines (channel check handlers) for System/360 processors are provided only for Models 65 and higher. However, after a channel error occurs, these processors do not always present enough information to the error recovery routines to enable them to retry the failing operation. In other cases, the channel may be left in a condition in which retry is impossible after a channel malfunction. Hardware improvements in the 4341 Processor eliminate these two situations in most instances.

Command retry is a channel/control unit procedure that can cause an improperly executed command in a channel program to be retried automatically by hardware so that an I/O interruption and programmed error recovery are not required. An indication is presented when the control unit recognizes this situation.

In the 4341 Processor, the command retry capability is implemented in the block multiplexer channels and can be utilized with 3370, 3330-series, 3350, and 2305 disk storage (as discussed in Section 20).

EXPANDED MACHINE CHECK FACILITIES

Implementation of the machine check class of interruption in the 4341 Processor is expanded in order to improve error recording and error recovery procedures. Programming support of the extended machine check

interruption is provided by the MCAR and MCH routines of DOS/VSE and OS/VS1, respectively.

The machine check interruption facilities of the 4341 Processor differ from those of Models 50 and 65 as follows:

- Four types of machine check are defined.
- Machine check interruption masking is expanded to handle selective disabling and enabling of the processor for the interruption types defined.
- The size of the logout area in lower processor storage is increased to accommodate the storing of additional machine status and diagnostic information when a machine check interruption occurs.
- Error conditions are defined that cause the 4341 Processor to stop functioning immediately (enter the check stop state) because the nature of the malfunction prevents valid processing from continuing.

The 4341 Processor presents one of four types of machine check interruption conditions, depending on the specific machine malfunction, and each type of interruption is maskable. Machine check interruption conditions are either repressible or exigent (formerly called soft or hard).

A repressible machine check condition exists in the 4341 Processor after an error has occurred that does not prevent continued successful execution of instructions (successful instruction retry, for example). An interruption can occur after a repressible machine check condition so that the failure can be recorded. System operation continues after the error is logged.

An exigent machine check condition exists when an uncorrectable error (such as an unretryable or uncorrectable instruction failure) occurs. Exigent conditions are those that prevent the successful execution of the current instruction.

For 4300 Processors, the logout area for every processor type is 512 bytes and no processor-dependent data is stored within these 512 bytes, although every processor may not utilize every defined field or bit (the machine check interruption types implemented in the 4331 Processor and the 4341 Processor are not exactly the same, for example).

This approach is different from that implemented in System/370 in which processor-dependent data is stored in certain fields in locations 0 to 511 and a processor-dependent extended logout is also stored when a machine check interruption occurs (usually beginning at location 512). The length of the processor-dependent extended logout varies by System/370 processor.

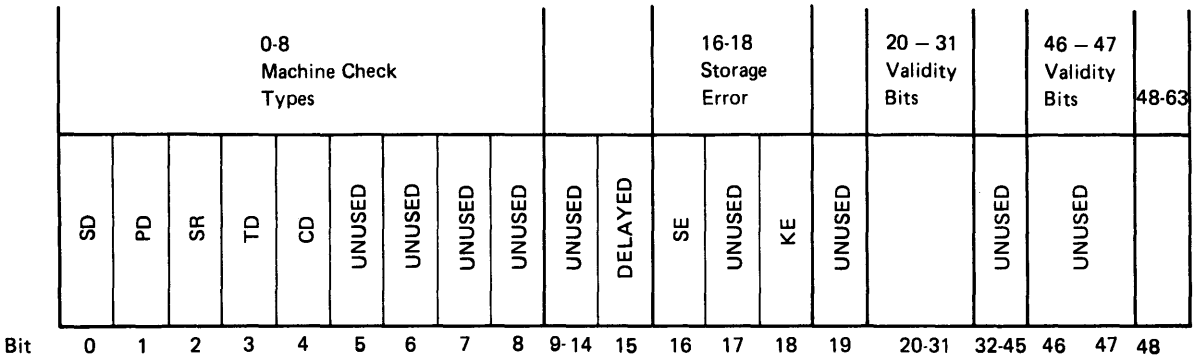
The approach taken in 4300 Processors permits a processor-independent (1) fixed logout area size, (2) machine check handler routine, and (3) logout data interpreting and printing routine (EREP) to be used. This approach removes operating system portability restraints for 4300 Processors that exist for System/370 processors because of processor-dependent logouts.

A logout to processor storage locations 0 to 511 occurs in the 4341 Processor when any type of machine check interruption is taken. The fixed logout area data indicates the reason for the interruption in the machine check code (locations 232 to 239). The save areas in locations 216 to 511 in the logout area preserve the status of the processor at

the time of the machine check interruption and contain the contents of the general, floating-point, and control registers as well as CPU timer and clock comparator values.

Figure 50.10.3 shows the layout and contents of the eight-byte machine check code for the 4341 Processor that is stored in processor storage locations 232 to 239. The machine check code indicates which type of interruption occurred and the validity of certain fields stored in the fixed logout area.

Fixed Logout Area Locations 232-239



Bit	Interruption Type
0	SD – System Damage
1	PD – Instruction Processing Damage
2	SR – System Recovery
3	TD – Timer Damage
4	CD – Timing Facilities Damage

Bit	Storage Error
15	Delayed
16	Storage Error Uncorrected
18	Key in Storage Error Uncorrected

Bit	Valid Fixed Area Data
20-23	Machine Check Old PSW (48-55)
	20 AMWP
	21 Masks and Protect Key
	22 Program Mask and Condition Code
	23 Instruction Address
24	Failing Storage Address (248, 249)
25	Unused
26	Unused
27	Floating Point Registers (352-383)
28	General Registers (384-447)
29	Control Registers (448-511)
30	Unused
31	Storage (Validity of processor storage being processed by instructions when interruption occurred)
46	CPU Timer Value
47	Clock Comparator Value

Figure 50.10.3. 4341 Processor machine check code

Table 50.10.1 lists the machine check types defined for the 4341 Processor. They are described in the discussion that follows. The mask bits used to enable or disable the processor for interruptions for each type are indicated and the setting of the machine check code is discussed.

PSW bit 13 and two other mask bits are used to enable and disable the processor for machine check interruptions. The recovery report (R) and external damage mask (E) bits are contained in control register 14 and operate subject to PSW bit 13. If PSW bit 13 is off, the processor is disabled for all machine check interruptions. If PSW bit 13 is on, the settings of the two additional mask bits determine whether or not

interruptions, other than instruction processing damage and system damage, will be taken.

Note that these mask bits control logouts to processor storage locations 0 to 511 only. They do not control logging to the system diskette, which is controlled by the setting established via the operator console (check control display).

Table 50.10.1. 4341 Processor machine check interruptions

Mask Bit(s)	Interruption Type and Cause	Machine Check Condition
PSW 13 and E	Interval Timer Damage and Timing Facilities Damage •Interval timer error •Time-of-day clock error •Clock comparator error •CPU timer error	Repressible
PSW 13 and R	Recovery Report •Error during instruction execution corrected by instruction retry	Repressible
PSW 13	System Damage •The damage cannot be isolated to one of the other machine check types	Exigent
PSW 13	Instruction Processing Damage One of the following occurs during instruction execution: •An unretryable malfunction •A retryable malfunction, such as a double- or multiple-bit processor storage error, access control key failure, etc., is not corrected by instruction retry	Exigent

Repressible Machine Check Interruptions

Repressible machine check interruptions are the following:

- Interval Timer Damage and Timing Facilities Damage. This interruption occurs if PSW bit 13 and the external damage mask bit are on. It indicates damage to the interval timer, time-of-day clock, CPU timer, or clock comparator. The TD and CD bits are stored in the machine check code. No differentiation between these two interruption types is made because of the way in which timing hardware is implemented in the 4341 Processor. That is, if one timing facility is failing, none are usable.

  An interval timer/timing facilities damage machine check interruption is generated when the time-of-day clock enters the error state as a result of a detected malfunction that could have affected the validity of the clock value or when damage to the interval timer, CPU timer, or clock comparator occurs. This

interruption (with instruction processing damage as well as timer damage indicated) is also taken when (1) a SET CPU TIMER or STORE CPU TIMER instruction is issued to a damaged CPU timer or (2) a SET CLOCK COMPARATOR or STORE CLOCK COMPARATOR is issued to a damaged clock compartor.

- Recovery Report. This interruption occurs if PSW bit 13 and the recovery report mask bit are on. It indicates the instruction retry facility was successful in correcting a retryable malfunction that occurred during the execution of an instruction.

Exigent Machine Check Interruptions

Exigent machine check interruptions are the following:

- Instruction Processing Damage. This interruption occurs if PSW bit 13 is on. The PD bit in the stored machine check code (bit 1) is used to indicate that an error occurred during the execution of the instruction indicated by the machine check old PSW. This error is posted when a nonretryable malfunction occurs or a retryable malfunction is not corrected after one retry is performed. If failure of an access control key or a double- or multiple-bit processor storage error caused the error, the failing address field will contain the address of the 2K storage block associated with the malfunctioning access control key or the failing address itself, respectively.

- System Damage. This interruption occurs if PSW bit 13 is on. The SD bit in the stored machine check code (bit 0) is used to indicate that a processor failure occurred that was not a result of the execution of the instruction indicated in the machine check old PSW. An unsuccessful interruption attempt, control register damage, etc., are examples of system damage errors.

Modes of System Operation for Machine Check Interruptions

Using the check control display, the operator can set the 4341 Processor to operate in normal, hard stop, no retry, disable, or stop after log mode after a machine check error condition occurs. When set to operate in normal mode, a logout to the system diskette and machine check interruption and logout to processor storage take place after a machine check condition occurs (when the processor is enabled for the specific machine check interruption).

Two modes of system operation for machine check interruptions can be established by the control program when the processor is set to operate in normal mode after a machine check: full recording mode and quiet, or nonrecording, mode. In full recording mode the processor is enabled for all machine check interruption types, and all types cause an interruption to be taken and logouts to occur. This is the normal mode of 4341 Processor operation.

In quiet mode, the processor is disabled for all repressible machine check interruptions. Quiet mode can be used to permit system operation without error recording for repressible conditions when the 4341 Processor is operating under the control of an operating system without 4341 Processor repressible machine check handling routines included.

A check-stop state is defined for the 4341 Processor (formerly called hard-stop state). If a check-stop condition occurs with the 4341 Processor set to normal mode for machine checks, a logout to the system diskette occurs after which the 4341 Processor stops without the occurrence of a logout to the fixed area in processor storage (locations 0 to 511). Check stop is initiated by hardware rather than by

programming. Implementation of a check-stop state prevents system operations from continuing when the nature of the machine malfunction prevents the processor from presenting meaningful status data.

The check-stop function is always active. That is, the occurrence of a check stop is not controlled by a check-stop control bit as in System/370. However, for compatibility reasons, bit 0 of control register 14, which is the check stop control bit in System/370, is set to one during a reset. When the 4341 Processor enters the check-stop state, the start key and restart function are made inoperative. Processor operation can be resumed only after a system reset or IPL is performed.

The following conditions cause a check-stop for the 4341 Processor when it is in normal mode for machine check conditions:

- Any hardware error is detected when the processor is in the stopped state and manual operations are in process.
- A system damage or instruction processing damage condition occurs and PSW bit 13 is zero.
- A system damage or instruction processing damage condition occurs during the processing of a previous system damage or instruction processing damage condition.

When hard-stop mode is in effect, after any type of machine check, the 4341 Processor enters the check-stop state immediately without any retry or reference code generation. There is no logout to the system diskette and no machine check interruption and logout to processor storage locations 0 to 511 before the stop. The suppressed log is kept until the 4341 Processor is started again, at which time the logouts to the system diskette and processor storage occur.

When the 4341 Processor is set to operate in no-retry mode, the instruction retry function is inhibited. When disable mode is in effect, the processor is prevented from entering the check-stop state and taking any machine check interruptions. When a machine check condition occurs, the processor attempts to continue operation without a logout to the system diskette or a machine check interruption.

The stop after log mode is provided for use with System/360 operating systems. When this mode is in effect, retry occurs after an error but system operation stops after a logout to the system diskette occurs to prevent erroneous continued operation because of instructions or data overlayed by the logout to processor storage.

The state of the 4341 Processor for machine check interruptions after IPL or a system reset is:

1. External damage interruptions are enabled. Damage to any one of the timing facilities causes a machine check interruption.
2. Recovery report interruptions are disabled.
3. PSW bit 13 normally is set to one by the IPL PSW (it is set to zero by the IPL system reset procedure) to enable the processor for system damage and instruction processing damage interruptions.

MACHINE CHECKS ON SYSTEM/360 MODELS 50 AND 65

A machine check situation in Models 50 and 65 results from hardware detection of a machine malfunction or parity error. Bad parity can

occur in main storage, local storage, a register, an adder, etc. Error correction is not attempted by Model 50 and 65 hardware when a machine check occurs. If the machine check mask bit in the current PSW is on, a machine check causes an interruption and a diagnostic scan-out occurs, starting at location 128. The number of bytes logged is model-dependent.

If the DOS machine check recording and recovery (MCRR) routine for the Model 50 is present, it gains processor control after a machine check interruption, and the error is logged. A retry of the failing operation is not provided by this routine and the affected program is terminated abnormally. If a recovery routine is not present, the system is placed in a wait state when a machine check interruption occurs.

MACHINE CHECK ANALYSIS AND LOGGING TO THE SYSTEM DISKETTE

While the 4300 Processors have expanded machine check capabilities similar to those implemented in System/370, 4300 Processors perform a machine check analysis function that is implemented only in System/370 for Models 115 and 125. In addition, the analysis and writing of logout data is handled by the support processor so that a logout is obtained even when the instruction processing function cannot perform a machine check and logout to processor storage.

When a hardware error is detected in the 4341 Processor, the support processor receives control and a logout of scan ring data occurs in the support processor. The support processor inspects the logout data and determines whether a check-stop condition exists. If not, the support processor determines whether the error is retryable and if so, initiates an instruction retry by the instruction processing function.

If a retry is not possible, a machine check interruption request is presented to the instruction processing function. If the processor is enabled for the machine check type, the interruption is taken and a logout of processor-independent information is placed in program processor storage locations 0 to 511. The machine check handler routine then writes the 512-byte logout to the operating system logout data set/file (SYS1.LOGREC in OS/VS1 or SYSREC in DOS/VSE) and takes appropriate recovery action.

While the instruction processing function is performing a retry or machine check interruption, the support processor writes the processor-dependent logout data to the system diskette. If the error was not retryable, an error analysis routine is loaded into the support processor from the system diskette. The error analysis program inspects the data logged to the support processor and, if possible, generates an eight-digit reference code to identify the malfunction.

The error analysis program then writes the reference code to the system diskette and displays it on line 23 of the display console. The audible alarm is sounded. The reference code remains displayed until replaced by another reference code or a system reset is performed. The operator should document any reference code that is displayed on the console. Whenever the customer engineer is called, the reference code(s) should be communicated to him. This may enable him to bring additional spare parts that might be required.

In order to diagnose errors that occur in most System/370 processors, the customer engineer utilizes the processor-dependent data written to the operating system logout data set/file. When diagnosing errors that occur in a 4300 Processor, the customer engineer will utilize the reference code supplied by the operator (or obtained from the system diskette) and, if necessary, processor-dependent logout data contained on the system diskette, rather than the operating system logout data

set/file, to diagnose errors. The reference code is not written to the operating system logout data set/file.

The processor-independent logouts contained in the operating system logout data set/file can be used in 4300 Processor installations for the following:

- As a history of the machine check conditions that occurred. The system diskette keeps logout data for only the latest machine check conditions (see discussion under "System Diskette Logouts"). The EREP program can be used to print the logout data set/file periodically.

- To inform the customer engineer of exactly which program was executing at the time the failure occurred (which is not recorded on the system diskette), if this information is requried

- To enable the customer engineer to determine the type of error that has occurred before actually obtaining dedicated use of the 4341 Processor. The EREP program can be executed concurrently with normal customer processing.

In order to diagnose a malfunction for which a reference code was generated, the customer engineer will first invoke the code analysis processor (CAP), which is resident on the system diskette and all diagnostic diskettes, and specify the generated reference code. The CAP program will display on the console suspected field replaceable units for the reference code, if any. This program will also indicate the page in the maintenance analysis procedure (MAP) documentation that should be consulted when no field replaceable units are listed or replacement of the listed units does not correct the malfunction.

The MAP page identified by the reference code contains information about the malfunction and the actions that should be taken to repair it. Output from the diagnostics is either another reference code that will indicate the card to be replaced or the additional diagnostic actions that are to be taken.

SYSTEM DISKETTE LOGOUTS

Four basic types of processor-dependent logouts are written to the 4341 Processor system diskette: support processor, instruction processing function, reference code history, and power. Power logouts are discussed under "Power Monitoring". The support processor logouts consist of one support processor summary record and up to eight support processor detail records. The summary record contains summary data about the existing detail records, which contain status data about the support processor and its adapters for the last eight support processor errors.

The instruction processing function logouts consist of one processing unit summary record, one processing unit directory record, and two status (check in system and scan rings) log records for each logout (up to nine) listed in the directory record.

The processing unit directory record identifies the processor-dependent logout data currently recorded on the system diskette. Logout data for the last three correctable errors and the last uncorrectable error is maintained in the status log records. The processing unit directory record contains the following for each logout: the logout identification number of the log, the reference code, the date and time of the error, and the error type (recoverable or nonrecoverable).

The checks in system and scan rings log records for a given error contain detailed processor-dependent information about the status of the processor at the time of the failure and the logout identification number of the error.

The reference code history record contains the last 31 reference codes generated by the support processor and each is time stamped.

The customer engineer can display the contents of each type of support processor and instruction processing function log record and the reference code history record on the display console using the general selection display when CE mode is in effect. The customer engineer can also print the contents of log records on a natively attached 3287 Printer (via the copy key) and purge log records using the console. If a 3287 Printer is not installed, log records can be printed on a channel-attached printer using the standalone error log print and edit program. This program will be provided on punched cards.

The MAP guides for the 4341 Processor will contain all the currently known information for locating and repairing malfunctions based on their reference code. If a reference code cannot be generated for a specific malfunction, the maintenance guide indicates the steps to take to attempt to locate the failure.

However, if the customer engineer cannot repair a malfunction using the maintenance guide, the customer engineer can utilize the Remote Service Facility to access a remote data bank system. Using the Data Bank mode of RSF, the customer engineer can search the data bank files (reference code and Symptom Fix) for additional information about the malfunction whether or not a reference code was generated (see discussion in Section 50:15 under "Remote Support Facility").

When a customer engineer locates a malfunction without the aid of the maintenance guide or as a result of procedures not specified for the diagnostic program used, the customer engineer will report the diagnosed failure, the failing reference code, symptoms and procedures used for the analysis, and the field replaceable unit that actually failed to a field problem feedback file. This information will be used to generate a new service procedure for the specified reference code that is then stored in the Symptom Fix File. All additional service procedures for the same reference code are stored together in the Symptom Fix File and have a format almost the same as that for procedures listed in the maintenance guide.

POWER SYSTEM

The power system in the 4341 Processor consists of the following functional units:

- A hardwired sequence that provides conventional sequencing for powering the support processor, its adapters, and the system diskette drive
- Power controller hardware that provides digital and analog sense points to permit power on/off control and power monitoring via microcode
- The power controller adapter diagnostic program, which is executed during the power-on procedure before the instruction processing function is powered up, as part of the hardware check-out procedure
- The power feature table that contains configuration-dependent power information for use by the microcode power sequencing and power monitoring programs

- Power sequencing, monitoring, logging, and shutdown programs
- A partial power-up/power-down program the customer engineer can use to individually power up/down the instruction processing function or the Channel-to-Channel Adapter. The support processor subsystem must be completely operational to execute this program.
- A CE service panel that enables the customer engineer to power the 4341 Processor on and off and make power adjustments

The power controller adapter is attached to the support bus of the support processor to provide two-way communication between the support processor and the power and environmental subsystem. Via the power controller adapter, the support processor controls power on/off sequencing, monitors voltage and current conditions, and monitors thermal, airflow, and EMC (power line transients and electrostatic discharge) sensors in the 4341 Processor. The adapter contains circuitry that performs digital and analog sensing as well as digital to analog conversions. This circuitry provides accurate voltage and temperature monitoring and measurement.

Once the support processor is powered on via conventional hardwired sequencing, the power-on sequence for the rest of the 4341 Processor system is microcode controlled. During a power on, the power diagnostic program is executed to test the operation of the power controller adapter. If an error is found during these tests, the power-on procedure is terminated and a reference code that identifies the power failure is displayed on the operator console.

During the IML procedure, the power feature table is generated based on the 4341 Processor configuration specified on the system diskette. The table contains configuration-dependent power sensor and control data information in the form of strings and masks. This table is used by the power monitoring program.

The power monitor program is resident in control storage of the support processor during system operation. Once a power on is successfully completed, the power monitoring program is executed every 500 milliseconds.

The power monitor program reads all the analog and digital sense points to determine whether any power or thermal fault conditions exist. The sense data read is compared against the mask in the power feature table. If the comparison indicates no fault exists, execution of the power monitor program is terminated. If the comparison detects a fault, the power monitor program rereads all the sense points and repeats the comparison to suppress spurious faults.

If the readings still show a fault, the power monitor program determines whether the readings indicate power or thermal conditions are critical enough to warrant a power down of the processor. If so, a reference code is generated, a power logout record is written to the system diskette that includes the reference code, the reference code is displayed on the operator console, and a power down of the processor is initiated. Neither the operating system nor the operator is notified that a power down is to occur.

When the readings indicate power or thermal conditions do not warrant a power down, a power logout that contains a reference code is written to the system diskette and processor operation continues. The reference code is not displayed.

The power interrupt handler program performs the same functions as the power monitor program except that it is invoked to read sense data

as a result of a power adapter interruption. Such an interruption occurs when a digital sense point becomes active.

The system diskette contains the four most recent power error logouts. These logouts contain power sensor and control latch status at the time of the logout. The last eight power reference codes generated are also recorded. During processor operation, EMC levels are sensed and stored in latches and the highest and lowest temperatures sensed at a particular gate are kept. During a power down sequence, one EMC and one temperature trace log record are written that contain the data accumulated during processor operation. The power logout records can be displayed using the operator console.

Additional programs are provided that enable the customer engineer to display voltage and thermal data on the operator console. A voltage tracking program can be used to display selectable simulated voltage wave shapes that depict the timing relationships among up to 3 of 20 selectable voltages. A digital display program displays all digital sense points, while a temperature display program displays the temperatures read at the thermal sense points.

The power system implemented in the 4341 Processor offers serviceability and availability advantages. Microcode-controlled power sequencing, versus hardwired control, is a more flexible method of control. It enables engineering changes to be installed more rapidly and sense point data to be obtained more quickly. The customer engineer can display the status of sense points on the console and need not manually obtain these readings by scoping. The power monitoring facility can provide early warning of potential power failures and may prevent catastrophic power failures.

RECOVERY MANAGEMENT SUPPORT FOR DOS/VSE AND OS/VS1

DOS/VSE machine check analysis and recording and channel check handler routines for the 4341 Processor represent an extension of the recovery support provided by DOS for System/360 Models 30, 40, and 50. The MCRR routine, which provides machine check and channel error handling, is offered as an option for these models.

MCAR, CCH, and RMSR are included automatically in any DOS/VSE supervisor generated for a 4341 Processor. RMSR performs all error recording functions. It replaces the following provided in DOS Version 3: outboard recorder, statistical data recorder, and the recording functions of MCAR, CCH, tape error by volume (TEBV), and error volume analysis (EVA).

OS/VS1 RMS for the 4341 Processor consists of extensions to the facilities offered by RMS routines provided in OS for Models 65 and up. The two RMS routines, machine check handler and channel check handler, are included automatically in OS/VS1 control programs generated for the 4341 Processor.

The two primary objectives of RMS are (1) to reduce the number of system terminations that result from machine malfunctions and (2) to minimize the impact of such incidents. These objectives are accomplished by programmed recovery to allow system operations to continue whenever possible and by the recording of system status for uncorrected hardware errors.

Machine Check Analysis and Recording and Machine Check Handler Routines

The DOS/VSE MCAR or OS/VS1 MCH routine receives control after a machine check interruption occurs. These routines record the processor-

independent logout data in the DOS/VSE SYSREC file or OS/VS1 SYS1.LOGREC data set, respectively, and then attempt to continue processor operation when possible. For example, continued operation is not possible after a system damage interruption. However, after an instruction processing error occurs, the MCAR/MCH routine determines whether the control program or a partition is affected. In the latter situation, the affected partition is canceled and processor operation continues.

As a result of the way in which machine check error recording is handled by 4300 Processors (all processor-dependent logout data is written to the system diskette), the MCAR and MCH routines are processor-independent for 4300 Processors.

Channel Check Handler

CCH receives control after a channel error occurs and constructs an error record. In a DOS/VSE environment, RMSR records the error record in SYSREC. In an OS/VS1 environment, CCH records the error record in SYS1.LOGREC. CCH passes the limited channel logout data and other pertinent status information to the appropriate I/O error recovery procedure, unless analysis of the error indicates that system operation cannot continue (the error involved SYSRES, for example).

If the ERP can correct the error using retry data from the limited channel logout, operations continue. If a permanent channel error exists, CCH records a permanent error and cancels the partition affected. The operator is notified.

The recovery support provided by the MCAR and CCH routines represents an extension of the facilities provided by the optional MCRR routine of DOS, which does not contain any repair or channel retry procedures.

ERROR RECOVERY PROCEDURES FOR DOS/VSE AND OS/VS1

The error recovery procedures are device-dependent error routines that are a standard part of the control program generated for any DOS/VSE or OS/VS1 environment. The limited channel logout provided by the DOS/VSE CCH routine is handled by a set of CCH ERP routines. The DOS/VSE CCH ERPs are an addition to the set of DOS/VSE ERPs. The latter are used without modification. For OS/VS1, the ERP routines are modified to accept and use limited channel logout data formatted by the CCH routine.

When a channel or I/O device error occurs in a 4341 Processor, the appropriate ERP is scheduled to perform recovery procedures. If the error is corrected, operations continue normally. If the error cannot be corrected and a user-written permanent error handling routine is not present, the affected task is abnormally terminated and the operator is notified.

DOS/VSE records only permanent I/O erros. In an OS/VS1 environment, temporary and permanent I/O errors are recorded by the statistical data recorder (SDR) and outboard recorder (OBR) routines, respectively.

RECOVERY MANAGEMENT SUPPORT RECORDER FOR DOS/VSE

The recovery management support recorder is a generalized error recording routine that replaces the error recording functions provided in DOS Version 3. RMSR provides more comprehensive error recording than DOS Version 3 and writes records in SYSREC that are compatible in format with those written in the OS/VS1 error recording data set SYS1.LOGREC. This enables a SYSREC file created by DOS/VSE and a SYS1.LOGREC data set

created by OS/VS1 to be processed by the same routines. This capability is not available for System/360, since a SYSREC file created by DOS Version 3 recording routines is not compatible in format with SYS1.LOGREC.

RMSR is given control to record the following types of error records:

- Records created by MCAR and CCH for processor and channel errors
- Unit check records that RMSR creates to provide statistics about permanent I/O device errors. This type of data is recorded by the OBR routine in DOS Version 3.
- Counter overflow records that contain statistics about the number of retries performed in correcting temporary I/O device errors. SDR handles this recording in DOS Version 3.
- Tape error statistics records that the TEBV routine records in DOS Version 3. The functions provided by the TEBV and EVA routines are handled by the tape error statistics portion of RMSR.
- IPL records that indicate the reason for this IPL and EOD records that are written when the operator issues the ROD command
- Miscellaneous data recordings for errors specific to a device, such as 3211 Printer buffer errors

ENVIRONMENT RECORDING, EDIT, AND PRINT PROGRAM FOR DOS/VSE AND OS/VS1

EREP is a standard system utility that can be initiated as a job step via standard job control statements at any time. It is designed to edit and print the error records written by RMS routines.

EREP performs the following:

1. Edits and prints all error records contained in SYS1.LOGREC or SYSREC. These records have been constructed and/or written by machine and I/O error handling routines such as OS/VS1 MCH, CCH, OBR, and SDR or DOS/VSE MCAR, CCH, and RMSR.
2. Accumulates a history of specified record types from SYS1.LOGREC or SYSREC by creating or updating an accumulation data set
3. Edits and prints a summary of selected records from SYS1.LOGREC or SYSREC or an accumulation data set
4. Clears SYS1.LOGREC or SYSREC

I/O RMS FOR OS/VS1

I/O RMS routines are optional, processor-independent routines supported in OS/VS1 environments. These reconfiguration procedures attempt to minimize the number of abnormal job terminations and unscheduled system halts that occur because of errors on channels or I/O devices.

The alternate path retry routine provides for the retry of a failing I/O operation on another channel path to the device involved, if one is available, when an uncorrectable channel error occurs. Thus APR, if present, is entered from a device-dependent ERP when a permanent error is deemed to exist after retry procedures have been attempted. If the I/O error is corrected using the alternate channel path, operations continue. If a permanent error still exists, the task is abnormally

terminated unless the dynamic device reconfiguration routine is present. A malfunctioning channel path can be varied offline by the operator, if necessary.

The dynamic device reconfiguration routine permits the operator to move a demountable volume from one device to another of the same type when a permanent hardware error occurs and provides repositioning of the volume so that the failing I/O operation can be retried. A volume can also be demounted so that device cleaning procedures can be performed and it can then be remounted on the same device. The DDR option also supports demountable system residence devices and unit record equipment.

DDR is entered from a device-dependent ERP after a permanent channel or device error occurs on a demountable device. Task termination occurs if the error cannot be corrected and a user-written permanent error handling routine is not present.

OLTEP AND OLTS FOR DOS/VSE AND OS/VS1

OLTEP is designed to operate as a processing program under operating system control. It handles the required interface between the operating system and the device-dependent OLTs. One OLTEP is provided for operation under DOS/VSE and another for execution under OS/VS1. These two OLTEPs support functions are not provided by the DOS OLTEP and OS OLTEP programs for System/360.

The inclusion of OLTEP in an operating system is automatic for a DOS/VSE system generated for a 4341 Processor, unless OLTEP is specifically excluded by a system generation parameter. OLTEP is standard in all VS1 operating systems. A standalone version of OLTEP, called OLTSEP, is available as well.

OLTEP directs the selection, loading, and execution of device-dependent OLTs for the purpose of I/O device testing and error diagnosis. OLTEP is also designed to verify I/O device repairs and engineering changes. OLTEP and OLTs are used by the customer engineer.

As with any other job step, OLTEP is invoked with job control and executes with a user-assigned priority. The input stream or system console device can be used to supply the parameters required for test operations (devices to be tested, options desired, etc.).

Both DOS/VSE and OS/VS1 OLTEP ensure the protection and security of user data files and storage in use while OLTs are operating. OS/VS1 OLTEP also ensures that the devices to be tested are online or offline (as far as the operating system is concerned) as required by the particular device type.

The OLTEPs for OS/VS1 and DOS/VSE also have the new capability of being able to access history records describing previous I/O errors on the device being tested. In addition, multiple devices can be tested during one OLTEP execution. If a console is used to define the test run, the new prompting facility can be requested as an aid to the user supplying the definition.

OLTEP and the OLTs reside in a DOS/VSE core image library. In OS/VS1 environments portions of OLTEP reside in both SYS1.LINKLIB and SYS1.SVCLIB, while the OLTs can be placed in a user-designated disk library (partitioned data set).

OLTEP and OLTs can operate concurrently with other executing jobs in a multiprogramming environment and provide online I/O device testing, eliminating the necessity for complete system unavailability while many types of errors are being diagnosed.

50:15 DIAGNOSTIC FACILITIES

Several diagnostic programs are provided that can be used to isolate an error to a field replaceable unit when the customer engineer cannot locate the malfunction utilizing procedures associated with the reference code for the error, if any. A basic system checkout test is also provided, as for System/370 processors.

For the 4341 Processor, the following diagnostic programs are provided:

- System Test. The System Test is available on tape or disk.
- Support processor subsystem diagnostics (basic, extended, and optional tests). The basic tests are in read-only control storage of the support processor and on the system diskette. The extended and optional tests are contained on a diagnostic diskette.
- Power controller adapter diagnostics. These are on the system diskette.
- Instruction processing function diagnostics (basic diagnostics and machine speed microdiagnostics). These are on the four diagnostic diskettes.

SYSTEM TEST

The System Test, available on tape or disk, performs a functional test of the system hardware components, including the support processor, storage, channels, and most locally channel-attached I/O devices. A new version of the System Test program for System/370 is utilized for the 4341 Processor that includes testing of the ECPS:VSE mode of operation.

The System Test can be used for the following purposes:

1. To locate a malfunctioning unit in the system. (The specific error component within the unit is not identified.)
2. To determine a working hardware configuration before an IPL
3. To verify correct system operation after a malfunctioning unit has been repaired

SUPPORT PROCESSOR SUBSYSTEM DIAGNOSTICS

Basic diagnostics for support processor subsystem components are contained on the system diskette and in read-only control stroage in the support processor. When the support processor is powered on or re-IMLed (power on/IML pushbutton is pressed), these diagnostics are executed to test for the correct operation of the support processor, diskette drive and its adapter, console attachment adapters, and the operator console. These tests execute regardless of the diskette mounted or the setting of the CE mode switch. If an error is found, a reference code is generated and displayed on the operator console or an error indication is displayed on the CE panel.

The extended and optional diagnostics for the support processor subsystem provide more extensive testing of the components of the support processor subsystem than the basic tests. Reference codes are displayed on the display console to indicate errors.

POWER CONTROLLER ADAPTER DIAGNOSTICS

The system diskette contains a complete set of power controller adapter tests. Whenever, the support processor is powered on, these tests are automatically executed after successful execution of the support processor subsystem diagnostics that are resident in the support processor. When CE mode is in effect, execution of these tests can be invoked using the console and error information is displayed on the console.

INSTRUCTION PROCESSING FUNCTION DIAGNOSTICS

The basic diagnostics and machine speed microdiagnostics for the instruction processing function operate in conjunction with the test case monitor and machine speed microdiagnostics monitor programs. These monitor programs control the loading and execution of the diagnostic programs and, utilizing the operator console, provide communication between the customer engineer and the diagnostics. The test case monitor program is loaded when the IBM TESTS function on the CE mode general selection display is selected or when the DIAG key is activated.

Basic Diagnostics

The basic diagnostics execute in the support processor under control of the test case monitor program, which is contained on the system diskette and all four diagnostic diskettes. The basic diagnostics (which are contained on three diagnostic diskettes) test the error checking hardware and other hardware in the processor that is required to execute the machine speed microdiagnostics. Error checking hardware, maintenance chips, scan rings, control storage, and clock distribution are tested via the support bus adapter interface to the instruction processing function. Errors are indicated via the displaying of reference codes on the display console.

Machine Speed Microdiagnostics

The machine speed microdiagnostics execute in instruction processing function control storage under the control of the machine speed microdiagnostics monitor program. The microdiagnostics and monitor are contained on the same diagnostic diskette as the extended and optional support processor subsystem diagnostics and are loaded under control of the test case monitor.

The machine speed microdiagnostics exercise the instruction processing function at processor speed to test the interaction between control storage, the instruction processing function, the storage controller, and channel hardware. Errors are indicated via the displaying of reference codes on the display console.

ERROR LOGOUT ANALYSIS PROGRAM

The error logout analysis program can be used to analyze processor logout data contained on the system diskette and provide fault isolation.

REMOTE SUPPORT FACILITY

The Remote Support Facility (RSF) is an optional (specify), no-charge feature of the 4341 Processor that is designed to improve its serviceability by reducing the time required to locate and repair a malfunction when the on-site (local) customer engineer cannot diagnose a problem using local maintenance procedures. For System/370 processors, such a facility is provided only for large-scale processors, such as the Model 158 and 168, and 303X Processors.

The Remote Support Facility provides two basic service facilities. First, via the Data Bank mode of operation, RSF enables the local customer engineer to access a data bank to obtain the latest service aids for the 4341 Processor and history information regarding the specific 4341 Processor installation with the malfunction. Second, via the Remote Console mode, RSF enables a remote customer engineer specialist to control operation of a malfunctioning 4341 Processor for diagnostic purposes utilizing a data bank system terminal or 3275 terminal.

Remote console operation from any data bank system terminal is through the Data Link facility. While in this mode, the remote specialist can perform online diagnosis as though he were at the customer installation. Logout data stored on the system diskette can be placed in the data bank system during data link operations for later offline analysis. In addition, microcode patches can be applied to a system diskette remotely.

Operation of the remote support facility is controlled by the support processor. The RSF feature consists of a communication common adapter attached to the I/O bus of the support processor and a 1200-baud modem in the 4341 Processor that handle communications functions for the Remote Support Facility. Autodial and autoanswer are not supported.

Support processor microcode for the Remote Support Facility is standard. However, in order to utilize RSF, a 4341 Processor installation must install a point-to-point, switched communications line to connect the modem to the data bank system, which is connected to remote 3270 display devices that are operated by customer engineer specialists. This communication facility can be utilized only for Remote Support Facility functions (not for any customer communications applications).

In order to activate RSF, the local customer engineer must select the configuration display from the general selection display. CE mode must be in effect in order to select the configuration display. Using the configuration display, the customer engineer selects the Data Bank Initialization display to activate Data Bank mode or the Remote Console Initialization display to activate Remote Console mode.

Data Bank Mode

When the local customer engineer cannot locate the cause of a problem utilizing the generated reference code and MAPS documentation or when a reference code is not generated, the customer engineer can invoke RSF to access the data bank. When Data Bank mode of RSF is activated, the 3278 Model 2A console of the 4341 Processor appears to be a terminal with a processor-to-processor interface to a special data bank system dial port.

The 4341 Processor must be in hard-stop mode before Data Bank mode can be invoked and only one 3278 Model 2A console and one 3287 hard-copy printer can be active during Data Bank mode operations. When Data Bank mode is terminated, the state of the 4341 Processor is unpredictable.

Thus, a re-IML or re-IPL will be necessary, depending on the state in effect.

When the Data Bank option is selected from the configuration display, the data bank initialization program is invoked and the local customer engineer must enter information to sign on the data bank system as a data bank system terminal.

The local customer engineer can perform one of the following operations when Data Bank mode is active:

- When a reference code was generated, search the reference code data bank utilizing the generated reference code as the search argument. The reference code data bank contains additional information for reference codes that is not yet included in the MAPS documentation. This information is generated by 4341 Processor engineering. If the generated reference code is present in the data bank, the data supplied to the local customer engineer may enable him to locate the cause of the problem.

  When the generated reference code is not found in the data bank, the local customer engineer can create an installation incident entry in the system that provides information about the problem. This information will be analyzed by engineering in an effort to locate the cause of the problem.

- When a reference code was not generated, search the Symptom Fix file using symptoms of the problem as the search argument. This file may indicate steps that should be taken to locate the cause of the problem.

- When installing an engineering change (EC) or optional feature, search the Tip/Diag file for any additions or corrections to the documentation that describes installation of the EC or feature.

If the customer engineer still cannot locate the error using the facilities of Data Bank mode, the Field Support Center should be contacted.

Remote Console Mode

When the Field Support Center (FSC) is contacted by a local customer engineer, the FSC may instruct the local customer engineer to invoke the Remote Console mode of RSF to enable a remote customer engineer specialist to control operation of the malfunctioning 4341 Processor for diagnostic purposes.

To activate Remote Console mode, the local customer engineer must (1) select the Remote Console initialization display from the configuration display, (2) select the remote console entry, (3) enter identifying information about the 4341 Processor (for example, branch office and customer number), and (4) dial the data bank system to establish a communication link. This causes the data bank system to perform a security check on the specified 4341 Processor and the link is disconnected if the 4341 Processor is not registered.

If the security check is successful, the data bank system determines whether there are any messages stored for this 4341 Processor and, if so, indicates the fact on the initialization display. The console of the 4341 Processor is under control of the remote customer engineer via the data bank system as soon as the message INITILIAZATION COMPLETE is shown on the local and remote consoles.

Using the Remote Console mode of RSF, the remote customer engineer can perform the following functions that are provided for local customer engineers:

- Display and analyze logouts contained on the system diskette
- Execute microdiagnostics and display their results
- Execute any customer engineer manual (CE mode) operations
- Execute support processor interpreter manual operations
- Display voltage measurements
- Execute operator manual operations (non-CE mode functions)
- Apply temporary patches to instruction processing function microcode or support processor microcode
- Display system status data
- Control execution of an operating system

The results of any diagnostic facility utilized by the remote specialist are displayed on both the local and the remote consoles so that the local customer engineer can monitor all operations invoked by the remote specialist and enter any necessary commands.

The local and remote customer engineers can communicate with each other via the console displays when Remote Console mode is in effect. If voice communication is desired, a separate communications line must be used.

Communication can be requested by the local or remote customer engineer by pressing the COMM REQ key on the display console. A message indicating communication is requested is displayed on both consoles and the audible alarm is sounded on both consoles. Communication via the displays is actually initiated when the remote specialist selects the communication display from the configuration display. The communication display appears on both consoles.

If the data bank system is not available when the remote specialist wishes to perform remote diagnosis on a malfunctioning 4341 Processor, the remote specialist can utilize a 3275 terminal with the dial feature for point-to-point communication with the malfunctioning 4341 Processor (via a dial-up, 1200-baud BSC line). To establish this type of remote console mode connection, remote console via 3275 must be selected from the Remote Console Initialization display.

When a 3275 terminal is used, the remote specialist can perform the same functions as when a data bank terminal is used. The local and remote customer engineers can communicate via their displays and voice utilizing the same communications line. The remote specialist can perform all the same console functions as the local customer engineer except power on/off, IML, support processor microcode trace, initial I/O control, voltage tracking, and utilization of the power displays with the continuous sample option. In addition, certain other restrictions on console key utilization by the remote specialist exist.

Advantages

The basic design of the Remote Support Facility includes customer security features. First, operation of RSF can be requested only from the customer installation and a security check is performed before the

facility is initiated. Second, via the local operator console, the customer can monitor all operations performed while RSF is active and the facility can be deactivated immediately at any time by depression of the LINE DISC key on the local operator console. (RSF can also be deactivated by the remote specialist via the data bank or 3275 terminal.)

Installation and use of the Remote Support Facility are optional. However, use of this facility offers the following advantages:

- A local customer engineer can obtain the services of a specialist quickly when he cannot locate a malfunction.
- For difficult problems remote analysis can make critical information available and enable parts to be sent to an installation before a specialist arrives at the installation to perform further diagnosis.
- Corrections to service aids and patches for known problems can be provided quickly.

SECTION 60: PLANNING OPTIMAL SYSTEM PERFORMANCE, USING BLOCK MULTIPLEXER CHANNELS AND ROTATIONAL POSITION SENSING DEVICES

Block multiplexing, rotational position sensing, and multiple requesting provide the user with another tool that can improve total system throughput in the area of multiprogramming. However, the effectiveness of this tool for a given installation depends largely on proper planning for its use. This section indicates how to use block multiplexer channels and RPS devices more effectively.

The guidelines outlined indicate how best to configure a system with rotational position sensing devices, how to plan job scheduling, and what to consider when determining disk data set characteristics. Explanations follow the statement of each guideline. These guidelines apply to DOS/VS, DOS/VSE, and OS/VS1 environments. The guidelines given apply to CKD disk devices that have the RPS feature (3330-series, 3340, 3344, and 3350) and FBA devices (3370) for the 4341 Processor.

All guidelines presented are not necessarily practical for all users. Each item should be evaluated in terms of the processing requirements and hardware configuration of an installation.

60:05 SYSTEM CONFIGURATION AND GENERATION

Guidelines for system configuration and generation are as follows:

1. Multiple 33XX strings should be placed on a single block multiplexer channel.

   Performance improvement occurs (1) as a result of overlapping the rotational positioning time of disk devices and (2) because more I/O requests can be initiated in a given period of time, since the channel is free more often. When many disk devices are active concurrently on a block multiplexer channel, there is more potential for such overlap.

2. Direct access devices with RPS should be placed on separate channels from I/O devices without RPS. Alternatives are as follows:

   a. If it is necessary to place non-RPS devices on the same block multiplexer channel with RPS devices, give first choice to non-RPS devices with a buffered control unit, such as the 2540 Card Read Punch, 3505 Card Reader, 3525 Card Punch, and 3203 Model 5 and 3211 Printers. These devices disconnect from a block multiplexer channel during the relatively long mechanical portion of their cycle, thereby freeing the channel for other operations.

   b. Tape units should not be placed on a block multiplexer channel with RPS devices unless absolutely necessary, because channel disconnection does not occur during any of their channel operations. If this is not possible, try to plan job scheduling to avoid having jobs using tape units and jobs using RPS support active on a block multiplexer channel at

the same time. If this is not feasible, try to assign very low-activity data sets to these tape units.

A device without channel disconnect capability can monopolize the block multiplexer channel for relatively long periods of time, thereby preventing (1) the initiation of other I/O operations on the channel and (2) the reconnection and completion of disk RPS channel programs already in operation on the channel. For example, a direct access device without RPS retains use of the channel during its search operations as well as during its reads and writes. If the device is a 2314 and block size is half a track, the channel is busy for 25 ms on the average (12.5 ms average rotational delay plus 12.5 ms read/write) for each I/O operation started for the non-RPS 2314 facility. Even if the block size used is relatively small, the channel can still be monopolized by the non-RPS device if there is high activity on the device.

3. The 2305 facility normally should not be placed on a block multiplexer channel with any other device.

   Exclusive use of a channel ensures optimum performance of the 2305 facility as a system residence device.

4. Consideration should be given to attaching buffered high-speed printers like the 3211 and 3203 Model 5 to a block multiplexer channel to improve system efficiency, especially in configurations with a heavily loaded byte multiplexer channel. The improved efficiency results from burst mode operation in which significantly more data is transferred per storage reference and fewer microcode cycles are taken per reference than in byte mode operation. The net effect will lower channel interference with the instruction processing function. Since these devices disconnect from the channel during the relatively long mechanical portion of their cycle, many printers can be attached to the same block multiplexer channel without degradation due to printer interference.

The following should be noted as regards specification of priority and ordered-seek I/O request queuing options for RPS devices at OS/VS1 system generation. The priority queuing option ensures priority I/O request initiation for the device, but because of first-come, first-served handling of I/O operations on the block multiplexer channel, this option does not ensure that priority device channel programs will complete sooner than other RPS channel programs that were started later on the channel. However, the objective of specifying the ordered-seek queuing option (minimization of arm movement on a disk drive) can still be achieved when using RPS.

60:10 JOB SCHEDULING

Guidelines for job scheduling are:

1. If total system throughput improvement, rather than maximum individual job performance increase, is the objective, schedule jobs that use RPS together such that the maximum number of RPS devices are active concurrently on each block multiplexer channel.

   Greater overlap potential exists when more RPS devices are active concurrently on a block multiplexer channel. (See item 1 in Section 60:05.)

2. When jobs that use disk data management functions with RPS support are executed concurrently with jobs that do not use RPS

support, the devices assigned to the former should be on different channels from devices assigned to the latter.

Alternatively, if jobs using RPS and jobs not using RPS must access devices on the same block multiplexer channel concurrently, the jobs without RPS support should have high seek activity such that search and read/write time is small compared to seek time.

3. Allocate a data set that will be accessed using QSAM or BSAM chained scheduling to a device on a channel without active RPS jobs.

   The chained scheduling technique is designed to keep a device active as long as record processing keeps up with record reading or writing. Thus, the channel can be kept busy for long durations, preventing the execution of any other I/O operation on the channel. Note that while QSAM and BSAM support concurrent use of RPS and chained scheduling for access to a disk data set, the performance attained by using chained scheduling alone will not be improved significantly by using RPS as well.

4. When data sets are being processed by an RPS access method in a multiprogramming or multitasking environment and disk device assignment is handled by the user rather than by the control program, allocate as many separate RPS devices as is practical.

   This approach allows the possibility of having more concurrent operations on these data sets and therefore more seek and rotational positioning overlap.

5. If a response-oriented RPS job operates on a block multiplexer channel concurrently with other RPS jobs, job scheduling should ensure that the number of jobs executing simultaneously is such that the performance desired for the response-oriented job can be attained.

   The performance of a block multiplexer channel is affected by the percentage of time the channel is busy searching and reading. The read or write of a particular record may be delayed one rotation because the channel is busy servicing another channel program. The probability of a particular record being delayed is a function of the percentage of channel busy time.

   As block multiplexer channel utilization increases, the probability that individual channel programs will be delayed increases. It is theoretically possible for the read or write of a particular record to be delayed indefinitely because the block multiplexer channel is busy searching for and reading other records. That is, utilization of more and more block multiplexer channel time will normally result in better overall performance, but will increase the likelihood of delayed response from any one data set.

60:15 DATA MANAGEMENT PARAMETERS

Guidelines for using data management parameters are:

1. When organizing direct data sets to be processed using DAM or BDAM on a CKD device, use fixed-length standard records and a

record reference that includes ID (relative block, relative track and ID, or actual address).

For CKD devices, RPS is supported only for fixed-length standard and VBS formats without key reference because record position must be known in order to calculate the sector number required for positioning. However, if a key reference or a variable record format is used, RPS support is provided for write verification and update (after retrieval) operations.

2. Use a large block size for sequentially processed data sets whenever possible, subject to the availability of processor storage for buffers.

   The use of RPS can provide performance gains for both short and long disk record block sizes. However, use of a large rather than short block size reduces the total time required to read or write a given data set because less disk space is required and fewer I/O operations are necessary. Note also that total throughput for a given block multiplexer channel is improved by using block sizes of equal (or nearly equal) size for all data sets being processed on the channel.

3. Use fixed standard records for QSAM and BSAM data sets where possible.

   The channel programs used for fixed standard records free the channel more often than when other record formats are used. A search for the previously read record is not used in order to locate the next sequential record when fixed standard records are read sequentially. The sector number of the next sequential record is obtained by including a READ SECTOR command at the end of the channel program used to read each record. Therefore, the SEARCH command specifies the ID of the desired record and the channel is free during the time it would otherwise have been busy searching for the previously read record.

   Channel time is reduced when fixed standard records are written because the operation required to calculate the remaining number of bytes on a track after each write is eliminated. (Note that the disk control unit is still busy erasing to end of track after formatting write operations even though the channel is freed after the data record has been written.)

4. Use multiple buffers with SAM, QSAM, and BSAM.

   The availability of multiple buffers per data set lowers the probability that a task will have to wait for a particular record. QSAM is designed to initiate an I/O request whenever a buffer becomes available, thus keeping the channel queue as full as possible. When BSAM is used, the programmer must handle the initiation of I/O requests.

60:20 SUMMARY

The following summarizes the advantages of rotational position sensing, multiple requesting, and block multiplexing.

- System throughput increases can be achieved when multiple sequential data sets are processed concurrently on a single block multiplexer channel (using SAM, QSAM, BSAM, ISAM, QISAM, or VSAM) because a higher effective channel data rate results.

- The number of block multiplexer channels required in a given system configuration can be fewer than the number of selector channels that would be required to handle the same amount of data, because more effective channel utilization is achieved by block multiplexing disk operations.

- The performance cost to an installation of verifying disk write operations is sharply reduced.

- The greatest throughput improvement results from use of rotational position sensing with high-activity, transaction-based processing, that is, with applications that include one or more large jobs that:

  1. Use direct processing (DAM or BDAM) with fixed-length standard or VBS records and a record reference that includes ID for CKD devices

  2. Require a multiple-volume data base of small records

  3. Process many additions and updates and use write verify

SECTION 70: COMPARISON TABLES OF HARDWARE FEATURES AND PROGRAMMING SUPPORT - SYSTEM/360 MODELS 50 AND 65, SYSTEM/370 MODELS 125 to 145, AND THE 4341 PROCESSOR

These tables have been included for quick reference. The first compares the hardware features of System/360 Models 50 and 65 and the 4341 Processor. The second table compares hardware features of System/370 Models 125, 135, 138, and 145 and the 4341 Processor.

70:05 COMPARISON TABLE OF HARDWARE FEATURES FOR SYSTEM/360 MODELS 50 AND 65 AND THE 4341 PROCESSOR

Hardware Feature	System/360 Model 50	System/360 Model 65	4341 Processor
I. PROCESSOR			
A. System/370 mode of operation	-	-	
1. BC mode	Comparable to BC mode	Same as Model 50	Standard
2. EC mode	Not implemented	Not implemented	Standard
B. ECPS:VSE mode of operation	Not implemented	Not implemented	Standard
C. Instruction set			
1. Standard set (Binary arithmetic)	Standard	Standard	Standard
2. Decimal arithmetic	Standard	Standard	Standard
3. Floating-point arithmetic	Standard	Standard	Standard
4. Extended precision floating-point	Not available	Not available	Standard
5. New instructions listed in Section 10:35	Not available	Not available	Standard
D. Buffered instruction fetch	No	Instruction unit normally prepares one instruction at a time. Imprecise interruptions occur only for storage protection violations.	Yes. Instruction fetching is overlapped with instruction execution for sequential instruction execution.
E. Processor cycle time	500 nanoseconds, 4-byte parallel flow	200 nanoseconds, 8-byte parallel flow	Variable--150 to 300 nanoseconds 8-byte parallel data flow
F. Address translation hardware	Not available	Not available	Standard
G. Channel indirect data addressing	Not available	Not available	Standard (for use in System/370 mode only)
H. Interval timer	Standard (16.6-ms resolution)	Standard (16.6-ms resolution)	Standard (3.33-ms resolution)
I. Time-of-day clock	Not available	Not available	Standard
J. CPU timer and clock comparator	Not available	Not available	Standard
K. Monitoring feature	Not available	Not available	Standard

Hardware Feature	System/360 Model 50	System/360 Model 65	4341 Processor
L. Program event recording	Not available	Not available	Standard
M. Direct control	Optional	Optional	Not available. External signals feature is standard
N. Interruption for SSM instruction	Not implemented	Not implemented	Standard
O. Compatibility features (all are optional)	1. 1410/7010 2. 7070/7074 (mutually exclusive features)	1. 7070/7074 2. 7080 (for both 705 and 7080) 3. 709/7040/7044/ 7090/7094/7094II	None
P. Control logic	Microcode in ROS	Microcode in ROS	Microcode in reloadable control storage (RCS)
Q. Instruction retry by hardware	No	No	Yes
R. Machine check	Occurs on processor, main storage, and certain channel errors. One mask bit controls this interruption.	Same as Model 50	Occurs after corrected and uncorrected errors. There are four types of machine check and each is individually maskable.
S. Fixed storage area size in lower storage (including logout area for machine and channel errors)	292 bytes (includes processor-dependent data)	328 bytes including processor and channel logouts (includes processor-dependent data)	512 bytes (no processor-dependent data included)
T. Multiprocessor systems	1. A multisystem feature is not available. 2. The support or main processor in an ASP configuration can be a Model 50.	1. Multisystem opticnal feature permits interconnection of two Model 65s. Main storage is shared (512K or more). Direct control is required. 2. The support or main processor in an ASP configuration can be a Model 65. Two or three systems are connected via a Channel-to-Channel Adapter.	Not supported
U. Extended Control Program Support (ECPS:VS1 and ECPS:VM/370)	Not available	Not available	Standard
V. Support processor unit	Not available	Not available	Standard
W. Remote Support Facility	Not available	Not available	Optional, no-charge

Hardware Feature	System/360 Model 50	System/360 Model 65	4341 Processor
II. STORAGE			
A. Processor (main) storage sizes	64K 128K 256K 384K 512K	256K 512K 768K 1024K	2048K 4096K
B. Type of processor storage	Ferrite cores	Ferrite cores	Monolithic technology (dynamic storage)
C. Processor storage cycle	2 microseconds for 4 bytes	750 nanoseconds (for 8 bytes). Two-way interleaving of sequential accesses other than by the channels is provided.	See high-speed buffer
D. High-speed buffer	No	No	8K buffer is standard. Fetch/store for a dcubleword is 225 nanoseconds.
E. Processor storage validity checking	Parity checking by byte. No hardware error correction is provided.	Same as Model 50	ECC checking on a doubleword. Single-bit errors are corrected by hardware.
F. Byte-oriented operands	No	No	Standard
G. Store and fetch protection	Store protect is standard and fetch protect is not available.	Standard	Standard
H. Shared processor storage	Optional. (Model 50 system shares 2361 Core Storage with a Model 50, 65, or 75.)	Optional. (Model 65 system shares 2361 Core Storage with a Model 50, 65, or 75.)	Not available
I. 2361 Core Storage	Optional Up to 8 million bytes can be attached.	Same as Model 50	Cannot be attached
III. CHANNELS			
A. Byte multiplexer	Standard	Optional (2780)	Channel 0 is standard
1. Subchannels/UCWs	16-128. (256 is an option.)	192	128 to 1024 in increments of 32 available to be used with all installed channels
B. Second byte multiplexer	Not available	Yes	Channel 4 is optional

Hardware Feature	System/360 Model 50	System/360 Model 65	4341 Processor
C. Block multiplexer	Not available	Not available	Two are standard and two (if channel 4 is a byte multiplexer channel) or three additional are optional.
1. Maximum individual channel data rate	-	-	Channels 1 through 4--2 MB/sec Channel 5--1 MB/sec
D. Selector	Optional--up to 3 (800 KB/sec data rate)	Optional--up to 6 2860s (1.3 MB/sec data rate)	Selector mode supported by each block multiplexer channel
E. Maximum aggregate data rate for channels	Approximately 1 MB/sec	In excess of 4 MB/sec for one 2870 and six 2860s	4 MB/sec for two block multiplexer channels and 9 MB/sec for five
F. Channel retry data provided after channel error	No	Yes in I/O logout	Yes in a limited channel logout area.
G. Channel-to-Channel Adapter	Optional	Optional on 2860	Optional (one maximum)
IV. OPERATOR CONSOLE DEVICES	1. 1052 Model 7 Printer-Keyboard 15 cps (No alter/display mode) 2. Additional consoles, such as display units, are optional. 3. Remote 2150 Console with operator control panel and/or 1052 Model 7 is optional. 4. Remote 2250 Display Unit containing operator control panel is optional.	1. 1052 Printer-Keyboard (optional) 2. Second 1052 Printer-Keyboard is optional. 3. A 2250 Display Unit and a remote 2150 Console are optional. 4. Other devices can be used as primary and secondary consoles.	1. One 3278 Model 2A Display Console with an operator control panel is required 2. Up to three additional 3278 Model 2A consoles can be directly attached for use as additional and/or alternate consoles. These consoles cannot have the operator control panel.
V. I/O DEVICES			
A. 2301, 2303, 2314, and 2321 direct access devices	All except 2301 drum.	All attach	Same as Model 50
B. 2305 facility			
1. Model 2	No	No	Yes
2. Model 1	No	No	No
C. 3203 Model 5 Printer	No	No	Yes
D. 3211 Printer	Yes	Yes	Yes
E. 3250 Graphics Display System	No	No	Yes
F. 3270 Information Display System	No	No	Yes

TNL GN20-4044 (06 June 80) to GC20-1877-0

Hardware Feature	System/360 Model 50	System/360 Model 65	4341 Processor
G. 3310 Direct Access Storage	No	No	No
H. 3330-series Direct Access Storage	No	No	Yes (all models) via 3830 Models 1 and 2
I. 3340/3344 Direct Access Storage	No	No	Yes (attachment via 3830 Model 2 and 3880)
J. 3350 Direct Access Storage	No	No	Yes (attachment via 3830 Model 2)
K. 3370 Direct Access Storage	No	No	Yes (attachment via 3880)
L. 3410/3411 Magnetic Tape Subsystem	Yes	No	Yes
M. 3505 Card Reader and 3525 Card Punch	No	No	Yes
N. 3540 Diskette Input/ Output Unit	No	No	Yes
O. 3600 Finance Communication System	No	No	Yes
P. 3630 Plant Communication System	No	No	Yes
Q. 3650 Retail Store System	No	No	Yes
R. 3660 Supermarket System	No	No	Yes
S. 3680 Programmable Store System	No	No	Yes
T. 3704, 3705-I, and 3705-II Communications Controllers	Yes (emulation mode only)	Yes (emulation mode only)	Yes (emulation and network control program modes)
U. 3730 Distributed Office Communication System	No	No	Yes
V. 3740 Data Entry System	Yes	Yes	Yes
W. 3767 Communication Terminal	Yes	Yes	Yes
X. 3770 Data Communication System	Yes	Yes	Yes
Y. 3780 Data Communication System	Yes	Yes	Yes

Hardware Feature		System/360 Model 50	System/360 Model 65	4341 Processor
Z.	3790 Communication System	No	No	Yes
AA.	3800 Printing Subsystem	No	No	Yes
BB.	3803/3420 Mangetic Tape Subsystem (Models 3, 5, 7 and 4, 6, 8)	Yes except Models 6 and 8. Model 4 can be attached to one channel only.	Yes except Model 8	Yes
CC.	3838 Array Processor	No	No	Yes
DD.	3850 Mass Storage System	No	No	Yes via 3830 Model 3
EE.	3881 Optical Mark Reader	No	No	Yes
FF.	3886 Optical Character Reader	No	No	Yes
GG.	3890 Document Processor	No	No	Yes
HH.	3895 Deposit Processing System	No	No	Yes

70:10 HARDWARE FEATURES OF SYSTEM/370 MODELS 125, 135, 138, 145, AND THE 4341 PROCESSOR

Hardware Feature	System/370 Model 125 (Models 0 and 2)	System/370 Model 135 (Models 0 and 3)	System/370 Model 138	System/370 Model 145 (Models 2 and 3)	4341 Processor
I. PROCESSOR					
A. Modes of operation	BC and EC modes standard. ECPS:VSE mode not supported.	Same as Model 125	BC and EC modes and Model 135 and 138 modes standard. ECPS:VSE mode not supported.	Same as Model 125	BC and EC modes provided for standard System/370 and ECPS:VSE modes
B. Instruction set					
1. Standard set (Binary arithmetic)	Standard	Standard	Standard	Standard	Standard
2. Decimal arithmetic	Standard	Standard	Standard	Standard	Standard
3. Floating-point arithmetic	Optional	Optional for the Model 0 (no-charge) Standard on the Model 3	Standard	Optional on the Model 2 (no-charge) Standard on the Model 3	Standard
4. Extended precision floating-point	Optional (included in floating-point option)	Optional for the Model 0 Required optional feature for the Model 3	Standard	Optional on the Model 2 (included in floating-point option) Standard on the Model 3	Standard
5. New System/370 mode instructions (listed in Section 10:35)	Standard except no MOVE INVERSE	Clock comparator, CPU timer, and conditional swapping are optional for the Model 0 and standard for the Model 3. INSERT PSW KEY and SET PSW KEY FROM ADDRESS are optional on the Model 0 and standard on the Model 3. CLEAR I/O and MOVE INVERSE are not implemented in the Model 0 or 3	Standard except no MOVE INVERSE	Standard on Model 2 except for clock comparator, CPU timer, conditional swapping, and advanced control program support instructions, which are optional. All are standard on the Model 3. MOVE INVERSE not supported.	Standard

Hardware Feature	System/370 Model 125 (Models 0 and 2)	System/370 Model 135 (Models 0 and 3)	System/370 Model 138	System/370 Model 145 (Models 2 and 3)	4341 Processor
C. Dynamic address translation	Standard	Standard	Standard	Standard	Standard
D. Channel indirect data addressing	Standard	Standard	Standard	Standard	Standard
E. Interval timer	Standard (3.3-ms resolution)	Standard (3.33-ms resolution)	Standard (3.33-ms resolution)	Standard (3.33-ms resolution)	Standard (3.33-ms resolution)
F. Time-of-day clock	Standard (16-microsecond resolution)	Standard (16-microsecond resolution)	Standard (16-microsecond resolution)	Standard (one-microsecond resolution)	Standard (one-microsecond resolution)
G. Clock comparator and CPU timer	Standard	Optional on the Model 0 Standard on the Model 3	Standard	Optional on the Model 2 Requred optional feature for the Model 3	Standard
H. Microinstruction retry by hardware	Standard	Standard	Standard	Standard	Standard
I. Machine check interruption	Occurs after corrected and uncorrected errors. There are six types of machine check and two mask bits.	Occurs after corrected and uncorrected errors. There are five types of machine check and three mask bits.	Same as Model 135	Occurs after corrected and uncorrected errors. There are five types of machine check and each is individually maskable.	Occurs after corrected and uncorrected errors. There are four types of machine check and three mask bits.
J. Fixed lower storage area size (including logout area for machine and channel errors)	512 bytes	512 bytes	512 bytes	705 bytes reducible to 512 if extended logout area is moved	512 bytes
K. Compatibility features (optional unless otherwise indicated)	1. 1401/1440/1460 Compatibility 2. System/360 Model 20 Compatibility 3. 2311 Model 1/ 3330-Series Compatibility 4. 2311 Model 1/ 3340-Series Compatibility 5. 2314/3340-Series Compatibility (all are no-charge)	1. 1401/1440/1460 (no-charge) 2. OS/DOS Compatibility (standard) 3. Model 20 Compatibility (no-charge) 4. 2314/3340 Compatibility	Same as Model 135	1. 1401/1440/1460 (no charge) 2. 1401/40/60, 1410/7010 (no-charge) 3. OS/DOS Compatibility (standard)	None available

Hardware Feature	System/370 Model 125 (Models 0 and 2)	System/370 Model 135 (Models 0 and 3)	System/370 Model 138	System/370 Model 145 (Models 2 and 3)	4341 Processor
L. Processor cycle time	480 nanoseconds, 4-byte parallel data flow	Variable from 275 to 1430 (BC mode) 1485 (EC mode) nanoseconds, 2-byte parallel data flow	Same as Model 135	Variable from 202.5 to 315 nanoseconds, 4-byte parallel data flow (Model 2) Variable from 180 to 270 nanoseconds, 4-byte parallel data flow (Model 3)	Variable from 150 to 300 nanoseconds, 8-byte parallel data flow
M. Direct Control feature	Not available (External Signals is optional)	Optional and includes External Interrupt feature	Same as Model 135	Optional	Not available (External Signals is standard)
N. Monitoring feature	Standard	Standard	Standard	Standard	Standard
O. Program event recording	Standard	Standard	Standard	Standard	Standard
P. Interruption for SSM instruction	Standard	Standard	Standard	Standard	Standard
Q. Virtual Machine Assist	Not available	Optional on Model 0 (no-charge) Not available on Model 3	Not available	Optional on Model 2 (no-charge) Not available on Model 3	Not available (function included in ECPS:VM/370)
R. Extended Control Program Support (VS1 and VM/370 assists)	Not available	Not available on Model 0 Standard on Model 3	Standard	Not available on Model 2 Standard on Model 3	Standard (ECPS:VS1 and ECPS:VM/370)
S. Remote Support Facility	Not available	Not available	Not available	Not available	Optional (no-charge)

Hardware Feature	System/370 Model 125 (Models 0 and 2)	System/370 Model 135 (Models 0 and 3)	System/370 Model 138	System/370 Model 145 (Models 2 and 3)	4341 Processor
II. STORAGE					
A. Processor (main) storage sizes	96K 128K 160K 192K 256K 384K (Model 2 only) 512K (Model 2 only)	For the Model 0 96K 144K 192K 240K 256K 320K 384K 512K For the Model 3 256K 320K 384K 512K	512K 1024K	For the Model 2 256K 384K 512K 768K 1024K 1536K 2048K For the Model 3 192K 320K 448K 704K 960K 1472K 1984K	2048K 4096K
B. Processor storage cycle	480 nanoseconds for two bytes Model 0 320 nanoseconds for two bytes Model 2	For BC mode operations, 770 nanoseconds read and 935 nanoseconds write for 2 data bytes (includes fetch of next micro-instruction). For EC and DAT mode operations, 825 nanoseconds read and 935 nanoseconds write.	Same as Model 135 except for read operations which are 715 nanoseconds for BC mode and 770 nanoseconds for EC mode.	For the Model 2, 540 nanoseconds fetch for 4 data bytes, 607.5 nano-seconds store for 4 data bytes, 540 nanoseconds fetch for 8 instruction bytes. For the Model 3, 405 nanoseconds fetch and 540 nanoseconds store for four bytes.	See item D
C. Processor storage validity checking	ECC checking on a doubleword. Single-bit errors are corrected by hardware.	Same as Model 125	Same as Model 125	Same as Model 125	Same as Model 125
D. High-speed buffer	Not implemented	Not implemented	Not implemented	Not implemented	Standard 8K bytes (225 nanoseconds for fetch/store of 8 bytes)

TNL GN20-4044 (06 June 80) to GC20-1877-0

Hardware Feature	System/370 Model 125 (Models 0 and 2)	System/370 Model 135 (Models 0 and 3)	System/370 Model 138	System/370 Model 145 (Models 2 and 3)	4341 Processor
E. Control storage	Reloadable monolithic 12K standard 4K or 8K additional optional on Model 0. 16K standard and 8K additional optional for Model 2	Reloadable monolithic storage with ECC. 24K, 36K, or 48K for the Model 0, 128K for the Model 3	Reloadable monolithic (128K) with ECC	Reloadable monolithic storage with ECC. 32K, 64K for the Model 2, 128K for the Model 3	Reloadable monolithic with parity checking
F. Byte-oriented operands	Standard	Standard	Standard	Standard	Standard
G. Store and fetch protection	Standard	Standard	Standard	Standard	Standard
III. CHANNELS					
A. Byte multiplexer channel - up to 8 control units	One optional	One standard	One standard	One standard	One standard and one optional
1. Subchannels	32 standard	Number of subchannels per system is not related to storage size. A total of 16, 64, 128, or 256 subchannels are available for the Model 0. A total of 64, 128, or 256 are available for the Model 3.	Same as Model 135 Model 3	Same as the Model 135 Model 0 for the Model 2. 64 standard and 128, 256, or 512 optional with any storage size for the Model 3.	128 to 1024 in increments of 32 standard
2. Data rate (maximum)	25 KB/sec in byte mode 29 KB/sec in in burst mode	41 KB/sec in byte mode 149 KB/sec in burst mode	Same as Model 135	50 KB/sec in byte mode 180 KB/sec in burst mode	64 KB/sec for byte mode and 1 MB/sec for burst mode for channel 0. 88 KB/sec for byte mode and 2 MB/sec in burst mode for channel 4.

Hardware Feature	System/370 Model 125 (Models 0 and 2)	System/370 Model 135 (Models 0 and 3)	System/370 Model 138	System/370 Model 145 (Models 2 and 3)	4341 Processor
B. Integrated attachments	1. Direct Disk Attachment for 3330-series or 3340 disk storage (standard) 2. Optional integrated attachments are available for the following: 1403 Printer 3203 Printer 2560 MFCU 5425 MFCU 3411/3410 Magnetic Tape 3420 Magnetic Tape 3504 Card Reader 3525 Card Punch 5213 Printer 3. Integrated Communications Adapter (optional) for up to six binary synchronous lines or up to 16 start/stop lines	1. One integrated file adapter can be attached. The 2319 IFA can handle from 3 to 8 2314A-type drives. The 3330/3340 series IFA can handle from 2 to 16 3330-series and/or 3340 drives. IFA Conversion feature permits both IFAs to be installed. 2. One integrated communications adapter can be installed to handle from 1 to 8 communications lines. 3. One integrated printer adapter can be installed to handle one 1403 printer Model 2, 7, or N1.	1. Only the 3330/3340-series IFA is available 2. Same as Model 135 3. Same as Model 135 4. Two Integrated 3203 Model 4 Printer Attachments optional 5. One Integrated Console Printer Adapter to attach one 3286 Model 2 Printer is standard	None available	None available (3278 Model 2A and 3287 Printers natively attach)
C. Block multiplexer channels	Not available	Mode optional for all installed selector channels on Model 0. Two optional on the Model 3. Maximum aggregate data rate of two channels is 2.6 MB/sec, 1.3 MB/sec per channel.	Two standard	Model optional for one or more of the installed selector channels on a Model 2. One standard and three more optional on a Model 3. Maximum agrregate data rate for four channels is 1.5 MB/sec without the channel word buffer and 5 MB/sec with the channel word buffer.	Two are standard and three additional are optional. Maximum aggregate data rate is 4 MB/sec or 9 MB/sec for two or five channels, respectively.

Hardware Feature	System/370 Model 125 (Models 0 and 2)	System/370 Model 135 (Models 0 and 3)	System/370 Model 138	System/370 Model 145 (Models 2 and 3)	4341 Processor
D. Selector channels	Not available	Optional. Two maximum on the Model 0 (1.3 MB/sec). Mode standard on all installed block multiplexer channels for the Model 3.	Mode standard for all block multiplexer channels	For a Model 2, channel 1 standard, 2-4 optional. 820 KB/sec without word buffer, 1.85 MB/sec with word buffer. Selector mode standard for all installed block multiplexer channels on the Model 3.	Mode standard for all block multiplexer channels
E. Channel retry data in a limited channel logout area after channel error, and I/O extended logout data	Yes	Yes	Yes	Yes	Only a limited channel logout is provided
F. Channel-to-Channel Adapter	Not available	Not available. (A Model 135 channel can be connected to an adapter on a channel in another System/360, System/370, or 4341 Processor.)	Same as Model 135	Optional (one maximum)	Optional (one maximum)
IV. OPERATOR CONSOLE DEVICES	1. Display console with standard display mode and optional printer-keyboard mode	1. 3210 Model 1 console with alter/display mode (15 cps) 2. 3215 Model 1 console with alter/display mode (85 cps) 3. Additional consoles, such as display units, are optional.	1. Display console with displays, printer-keybaord and 115/125 emulation modes is standard. 3286 or 3287 Printer is optional for all modes.	1. Same as Model 135 2. Same as Model 135 3. Same as Model 135 4. Optional 3210 Model 2 console remote with either (1) or (2) - no alter/display	1. 3278 Model 2A Display Console with standard display and printer-keyboard modes is required. 2. Additional (up to three) 3278 Model 2A display consoles and/or 3287 Printers can be natively attached.

Hardware Feature	System/370 Model 125 (Models 0 and 2)	System/370 Model 135 (Models 0 and 3)	System/370 Model 138	System/370 Model 145 (Models 2 and 3)	4341 Processor
V. I/O DEVICES					
A. 2301, 2303, 2314, and 2321 direct access devices	No	All except 2301 drum. 2314-type disk storage can be attached to the IFA and one channel or both channels when an IFA is not present.	All except 2301 drum. The 2314 can be channel-attached only.	All except 2301 drum. 2314 facilities can be attached to channels 1, 2, and 3.	All except 2301 drum.
B. 2305 facility 1. Model 2	No	No	No	Yes on block multiplexer channels. The word buffer feature is required.	Yes
2. Model 1	No	No	No	No	
C. 2401 and 2420 tape units	No	All can be attached but not to the byte multiplexer channel	Same as Model 135	All can be attached	Yes
D. 2560 and 5424 Multi-Function Card Units	Yes	No	No	No	No
E. 3203 Printer	Models 1 and 2	No	Model 4	No	Model 5
F. 3211 Printer	No	Yes	Yes	Yes	Yes
G. 3250 Graphics Display System	Yes (via byte multiplexer operating in burst mode only)	Yes	Yes	Yes	Yes
H. 3270 Information Display System	Yes	Yes	Yes	Yes	Yes
I. 3310 Direct Access Storage	No	No	No	No	

TNL GN20-4044 (06 June 80) to GC20-1877-0

Hardware Feature	System/370 Model 125 (Models 0 and 2)	System/370 Model 135 (Models 0 and 3)	System/370 Model 138	System/370 Model 145 (Models 2 and 3)	4341 Processor
J. 3330-series Models 1, 2, and 11	Yes (Models 1 and 2 only)	Yes via the 3330/3340 Series IFA or a channel and the 3830 Model 1 or 2	Same as Model 135	Yes on block multiplexer channels (on selector channels also if multiple requesting and RPS are not used) via the 3830 Model 1 or 2 and/or Integrated Storage Control	Yes via the 3830 Model 1 or 2
K. 3340/3344 Direct Access Storage	Yes	Yes via the 3330/3340 Series IFA or a channel and the 3830 Model 2	Same as Model 135	Yes on selector or block multiplexer channels via the 3830 Model 2 and/or Integrated Storage Control	Yes via 3830 Model 2 and/or the 3880
L. 3350 Direct Access Storage	No	Yes via a channel and the 3830 Model 2	Same as Model 135	Same as 3340	Yes via 3830 Model 2
M. 3370 Direct Access Storage	No	No	No	No	Yes via the 3880
N. 3410/3411 Magnetic Tape Subsystem	Yes	Yes (cannot be attached to the byte multiplexer channel)	Same as Model 135	Yes	Yes
O. 3505 Card Reader and 3525 Card Punch	Yes plus native attach of 3504 Card Reader	Yes	Yes	Yes	Yes
P. 3540 Diskette Input/Output Unit	Yes	Yes	Yes	Yes	Yes
Q. 3600 Finance Communication System	Yes	Yes	Yes	Yes	Yes
R. 3630 Plant Communication System	Yes	Yes	Yes	Yes	Yes
S. 3650 Programmable Store System	Yes	Yes	Yes	Yes	Yes
T. 3660 Supermarket System	Yes	Yes	Yes	Yes	Yes
U. 3680 Programmable Store System	Yes	Yes	Yes	Yes	Yes

Hardware Feature		System/370 Model 125 (Models 0 and 2)	System/370 Model 135 (Models 0 and 3)	System/370 Model 138	System/370 Model 145 (Models 2 and 3)	4341 Processor
V.	3704, 3705-I, and 3705-II Communications Controllers	Yes	Yes	Yes	Yes	Yes
W.	3730 Distributed Office Communication System	Yes	Yes	Yes	Yes	Yes
X.	3740 Data Entry System	Yes	Yes	Yes	Yes	Yes
Y.	3767 Communication Terminal	Yes	Yes	Yes	Yes	Yes
Z.	3770 Data Communication System	Yes	Yes	Yes	Yes	Yes
AA.	3780 Data Communications Terminal	Yes	Yes	Yes	Yes	Yes
BB.	3790 Communication System	Yes	Yes	Yes	Yes	Yes
CC.	3800 Printing Subsystem	No	No	No	Yes	Yes
DD.	3803/3420 Magnetic Tape Subsystem					
	Models 3, 5, and 7	Yes (Models 3 and 5 only)	Yes (cannot be attached to the byte multiplexer channel)	Same as Model 135	Yes	Yes
	Models 4, 6, and 8	No	Yes (cannot be attached to the byte multiplexer channel)	Same as Model 135	Yes	Yes
EE.	3838 Array Processor	No	No	No	Yes	Yes
FF.	3850 Mass Storage System	No	No	No	Yes via the 3830 Model 3	Yes via the 3830 Model 3
GG.	3881 Optical Mark Reader	Yes	Yes	Yes	Yes	Yes
HH.	3886 Optical Character Reader	Yes	Yes	Yes	Yes	Yes
II.	3890 Document Processor	No	Yes	Yes	Yes	Yes
JJ.	3895 Deposit Processing System	Model 2 only	Yes	Yes	Yes	Yes

A Guide to the IBM 4341 Processor
Systems
GC20-1877-0

READER'S COMMENT FORM

This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

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.*

Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name and mailing address:

_____

_____

_____

Note: Staples can cause prot with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

What is your occupation? _____

Number of latest Newsletter associated with this publication: _____

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.)

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 824
1133 Westchester Avenue
White Plains, New York 10604

Fold and tape | Please Do Not Staple | Fold and tape

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601

Cut or Fold Along Line

A Guide to the IBM 4341 Processor Systems Printed in U.S.A. GC20-1877-0