GC20-1878-0
File No. S/370-01

Systems

A Guide to the IBM 4331 Processor

IBM

GC20-1878-0
File No. S/370-01

Systems

A Guide to the IBM 4331 Processor

This guide presents hardware, I/O device, programming systems, and other pertinent information describing the significant new features and advantages of the IBM 4331 Processor. Knowledge of System/360 hardware and I/O devices is assumed. The content of the guide is intended to acquaint the reader with the 4331 Processor and to be of benefit in planning for its installation.

First Edition (February 1979)

This guide is intended for planning purposes only. It will be updated from time to time; however, the readers should remember that the authoritative sources of system information are the system library publications for the 4331 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

Sections 01 through 50 of this publication assume that 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. These sections highlight and discuss the significance of only those 4331 Processor hardware, I/O, and programming systems features that are different from those of System/360 Models 22 and up.

Sections 01 through 50 are also designed for readers who are knowledgeable about System/370 architecture, channels, I/O devices, and programming systems. Compatibility between the architectures implemented in 4300 Processors and System/370 processors and their programming systems support is also discussed.

Section 60 assumes a reader with System/360 Model 20 (Submodel 5) knowledge. This section highlights the hardware and I/O device differences between 4331 Processor and Model 20 configurations that also exist between the Model 20 and Models 22 and up. Therefore, after reading Section 01, the Model 20 user should read Section 60 before continuing with Sections 05 through 50.

CONTENTS

Section 01: Highlights 1

Section 05: Technology, Architecture, and Physical Components. . . 14.1
05:05 Technology 14.1
Introduction 14.1
System/360 Technology. 14.2
System/370 Technology. 14.3
4331 Processor Technology. 14.4
05:10 Design Objectives and Architectures. 15
Design Objectives. 15
Architectures Implemented. 15
System/360 Program Compatibility with the 4331 Processor 18
System/370 Program Compatibility with the 4331 Processor 19
05:15 Physical and Logical Components. 20

Section 10: The 4331 Processor 21
10:05 The Instruction Processing Function. 21
General Description. 21
Control Registers. 22
Basic Control Mode 22
Extended Control Mode. 23
Expanded Instruction Set 28
Byte-Oriented Operands 31
Monitoring Feature 31
Architecture Implementation Alterations. 32
Interval Timer 32
Time-of-Day Clock. 32
Clock Comparator and CPU Timer 33
10:10 Storage. 34
Processor Storage. 34
Control Storage. 37
10:15 The Support Processor Subsystem. 41
Components and Functions 41
System Initialization. 42
The System Diskette Drive. 43
The Support Bus Adapter and Support Bus. 44
10:20 Channels and I/O Adapters. 44
General Description. 44
Display/Printer Adapter. 50
Byte Multiplexer Channel 51
Block Multiplexer Channel. 52
DASD Adapter 57
8809 Magnetic Tape Unit Adapter. 59
5424 Adapter 60
Diskette Drive 60
Communications Adapter 60
10:25 Block Multiplexing Operations with Fixed Block Architecture Devices 64
Fixed Block Architecture Design. 65
Track Formatting 66
Command Set. 66
Read and Write Command Execution 70
Differences Between FBA and CKD Channel Programs 71
Advantages 72
10:30 Standard and Optional Features 72
Standard Features. 72
Optional Features. 74

Section 12: Operator Console 75
12:05 General Description. 75
Operator/Operating System Communication Modes. 75
Operator's Control Panel 77
Keyboard 78
Display Controls and Indicators. 79
12:10 Operator Displays. 80
Mode Selection Display 80
Program Load Display 82
Display/Alter Display. 83
Address Compare Display. 85
Check Control and Interval Timer Displays. 85
Native Displays and Printers Displays. 86
User Diskette Control and Diskette Device Address Displays 86
Communications Adapter Displays. 86
Machine Status Area. 86
12:15 Maintenance. 87

Section 15: Virtual Storage and Address Translation. 88
15:05 Virtual Storage Concepts, Advantages, and Terminology. . . 88
The Need for Larger Address Space. 88
Virtual Storage and Address Translation Concepts 92
General Advantages Offered by IBM Operating Systems that Support a Virtual Storage Environment 99
Virtual Storage and Address Translation Terminology. . . 106
15:10 Address Translation Facility for the 4331 Processor Operating in System/370 Mode 111
Virtual Storage Organization 111
Operation of Dynamic Address Translation Hardware. . . . 113
Features To Support Demand Paging. 124
Channel Indirect Data Addressing 125
15:15 Address Translation Facility for the 4331 Processor Operating in ECPS:VSE Mode 128
Virtual Storage Organization 129
Operation of Address Translation 134
15:20 System Performance in a Virtual Storage Environment. . . . 138
System Resources Required To Support a Virtual Storage Environment. 139
New Factors that Affect System Performance 141
Relationship Between Virtual Storage Size and System Performance. 144
Increasing System Performance in a Virtual Storage Environment. 147

Section 18: Virtual Machines 152
18:05 Definition and General Operation 152
ECPS:VM/370. 159
18:10 General Advantages of a Virtual Machine Environment . . . 163

Section 20: I/O Devices. 165
20:05 I/O Device Support 165
20:10 3310 Direct Access Storage 167
String Configurations and Capacities 167
Powering 168
Track Formatting and Data Organization 168
Disk Initialization. 169
Command Operation. 169
Error Retry Functions. 172
Usage, Error, and Overrun Statistics 173

Advantages of 3310 Direct Access Storage for 2314/2319 Disk Storage Users 174
Advantages of 3310 Direct Access Storage for 3340 Disk Storage Users. 176
The 2311/2314/2319/3310 Direct Access Storage Compatibility Feature. 177
20:15 3340 Direct Access Storage 185
3340 Disk Storage and the 3348 Data Module 185
Attachment via the DASD Adapter. 200
System/3 Data Import Feature 202
Summary. 202
20:20 The 3803/3420 Magnetic Tape Subsystem. 205
3803 Tape Control Model 1 and Models 3, 5, and 7 of the 3420 Magnetic Tape Unit. 205
3803 Tape Control Model 2 and Model 4 of the 3420 Magnetic Tape Unit. 212
Summary. 216
20:22 The 3410/3411 Magnetic Tape Subsystem. 221.1
20:25 The 8809 Magnetic Tape Unit. 222
Models 222
Modes of Operation 223
Commands 224
Transport Design 224
Serviceability Features. 225
20:30 The 3203 Model 5 Printer 228

Section 30: Programming Systems Support. 231
30:05 DOS/VSE. 231
DOS/VSE Functions. 231
DOS/VSE Support of 4331 Processor Features and I/O Devices. 233
Program Products for DOS/VSE 236
Using DOS/VS Release 34. 242
Using DOS Release 26 243
30:10 OS/VS1 245
30:15 VM/370 247

Section 40: Emulators. 250
40:05 The IBM Systems 1401/1440/1460 Emulator Program. 250
General Operation. 250
Support of 1401/1440/1460 Features 256

Section 50: Reliability, Availability, and Serviceability (RAS). . 260
50:05 Introduction 260
50:10 Recovery Features. 262
ECC Validity Checking and Defect Dictionary for Processor Storage. 262
I/O Operation Retry. 263
Expanded Machine Check Facilities. 264
Machine Checks on System/360 Models 30 and 40. 269
Machine Check Analysis and Logging to the System Diskette. 269
Power System 271
Recovery Management Support for DOS/VSE and OS/VS1 . . . 273
Error Recovery Procedures for DOS/VSE and OS/VS1 274
Recovery Management Support Recorder for DOS/VSE 274
Environment Recording, Edit, and Print Program for DOS/VSE and OS/VS1 275
OLTEP and OLTs--DOS/VSE and OS/VS1 275
50:15 Diagnostic Facilities. 276
IML Testing and Inline Tests 276
Microtests and Manual Operations 277
Remote Support Facility. 277

Section 60: Hardware and I/O Differences Between Model 20 and 4331 Processor Configurations. 280
60:05 Hardware Differences Between the Model 20 and the 4331 Processor 280
60:10 Channel and I/O Device Differences Between the Model 20 and the 4331 Processor 284
Channels 284
I/O Devices. 286

Section 70: Comparison Table of Hardware Features of System/360 Models 20, 30, and 40 and the 4331 Processor 292
70:05 Hardware Features - System/360 Models 20, 30, and 40 and the 4331 Processor 293

Index 299

FIGURES

05.05.1 SLT substrate. 14.2
05.05.2 The 704-circuit logic chip resting on a paper clip . . . 14.5
05.05.3 An MST and 704-circuit logic chip surrounded by salt crystals 14.6
05.05.4 The MLC logic module used for instruction processing function logic in the 4331 Processor without and with its cap 14.7
05.05.5 One MLC 50 logic module containing approximately 4200 circuits and 700 MST logic modules containing the same number of circuits 14.8
05.05.6 One MLC 50 logic module with six chips and 23 MST logic cards with the same number of circuits 14.8
05.05.7 The 64K-bit storage chip resting on a coin 14.13
05.05.8 A processor storage card containing 512K bytes 14.16
05.05.9 Logic and one megabyte of processor storage for the Model 115 and the 4331 Processor 14.16
05.15.1 The 4331 Processor (design model shown right to left with 8809 Magnetic Tape Units, 3310 Direct Access Storage, the 3203 Model 5 Printer, the 3505 Card Reader, the 3525 Card Punch, the 3287 Printer, and the 3278 Model 2A Display Console. 20
10.05.1 BC and EC mode PSW formats 24
10.05.2 BC mode fixed processor storage locations 0 to 511 . . . 26
10.05.3 EC mode fixed processor storage locations 0 to 511 . . . 27
10.10.1 Processor and program sections of processor storage. . . 35
10.10.2 Calculation of processor storage unavailable for program use for a sample 4331 Processor configuration. . 41
10.20.1 The basic data flow between processor storage and the channels and I/O adapters in the 4331 Processor. 47
12.10.1 The mode selection display 80
12.10.2 The program load display 83
12.10.3 The display/alter display for System/370 mode. 84
12.10.4 The display/alter display for ECPS:VSE mode. 84
12.10.5 The address compare display. 85
15.05.1 Names and location of instructions and data in a virtual storage environment. 94
15.05.2 Relationship of virtual storage, direct access storage, and real storage 95
15.05.3 Conceptual illustration of real storage utilization in a mixed batch and online virtual storage environment . . 105
15.05.4 Layout of virtual storage, external page storage, and real storage 108
15.10.1 Virtual storage address fields for a 64K segment 114
15.10.2 Segment table and page tables used for dynamic address translation. 116

15.10.3 Storage levels and real-to-processor storage mapping for System/370 mode in the 4331 Processor. 117
15.10.4 Dynamic address translation procedure for System/370 mode 119
15.10.5 TLB for the 4331 Processor 121
15.10.6 Example of IDALs required for a CCW list when page size is 2K 127
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. 145
15.20.2 General effect on system performance of the paging factor only. 146
15.20.3 General effect of the paging factor on system performance for various active-to-passive page ratios. . 146
15.20.4 General system performance curve for a virtual storage environment. 147
18.05.1 Conceptual illustration of the real and virtual machine environment that is supported by VM/370. 155
18.05.2 Conceptual illustration of the implementation of virtual storage in a virtual machine environment 156
18.05.3 Segment table and page tables built when a virtual storage operating system executes in a virtual machine . 157
| 20.10.1 Two four-drive 3310 strings (design models). 168
20.15.1 A five-drive 3340 string with 3340 Model A2, B2, and B1 units 186
20.15.2 The 3348 Data Module 186
20.15.3 Location of physical and logical tracks and read/write heads on a data surface in a 3348 Data Module. 190
20.15.4 Cylinder and read/write head layout for a 3348 Model 35 Data Module. 192
20.15.5 Cylinder and read/write head layout for a 3348 Model 70 Data Module. 193
20.15.6 Cylinder and read/write head layout for a 3348 Model 70F Data Module. 195
20.20.1 Tape-switching configurations for the 3803/3420 Magnetic Tape Subsystem. 210
20.20.2 Sample tape-switching configuration with mixed 3803 and 3420 models. 215
| 20.22.1 The 3410 Magnetic Tape Unit. 221.2
| 20.22.2 3410/3411 tape subsystem physical layouts. 221.3
20.25.1 A string of four 8809 Magnetic Tape Units. 222
20.25.2 Top view of an 8809 tape unit showing the reel-to-reel transport design. 225
20.30.1 The 3203 Model 5 Printer (design model). 228
50.10.1 Data representation used in processor storage in the 4331 Processor 262
50.10.2 Data representation used in Model 30 and 40 processor storage and in the 4331 Processor in other than processor storage 263
50.10.3 4331 Processor machine check code. 266

TABLES

10.10.1 Functional microcode group requirements for standard and optional 4331 Processor features 38
10.10.2 Reloadable control and processor storage requirements for 4331 Processor features by functional microcode group. 39
10.20.1 Channel attachment and subchannel mode for frequently used I/O devices for the 4331 Processor. 56
15.10.1 Number and size of segments and pages for a 16-million-byte virtual storage 112
15.10.2 Virtual and real storage addresses used by, and supplied to, programs in the 4331 Processor operating in System/370 mode. 123
20.05.1 I/O devices attachable to the 4331 Processor 165
20.15.1 Physical and capacity characteristics of 3348 Data Modules and the 2316 Disk Pack 197
20.15.2 Timing characteristics of 3340 and 2314 disk storage . . 198
20.15.3 Summary of hardware features of 3340 and 2314 disk storage. 203
20.20.1 3803 Model 1 control unit configurations and capabilities with Single Density, Dual Density and Seven Track features 209
20.20.2 3803 Model 2 control unit configurations 214
20.20.3 3420, 2420, and 2401 Magnetic Tape Unit Characteristics. 217
20.22.1 3410, 2401, and 2415 Magnetic Tape Unit Characteristics. 221.5
30.05.1 4331 Processor features and major I/O devices not supported by DOS Release 26. 244
40.05.1 1401/1440/1460 I/O devices and features supported by the IBM Systems 1401/1440/1460 Emulator Program and corresponding 4331 Processor devices 258
40.05.2 1401/1440/1460 CPU features supported by the IBM Systems 1401/1440/1460 Emulator Program. 259
40.05.3 1401/1440/1460 I/O devices not supported by the IBM Systems 1401/1440/1460 Emulator Program. 259
50.10.1 4331 Processor machine check interruptions 266
60.05.1 4331 Processor instructions not available on the Model 20 282
60.05.2 Model 20 and 4331 Processor I/O device correspondence. . 287
60.05.3 I/O devices for the 4331 Processor that are supported by DOS/VSE but not attachable to the Model 20. 291

SECTION 01: HIGHLIGHTS

The 4331 Processor is an intermediate-scale, general purpose processor. It is one of the IBM 4300 Processors and is compatible with the 4341 Processor. The 4331 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 4331 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 4331 Processor provides a growth path for users of System/360 Models 30 and 40. It offers such installations advanced functions, significantly increased internal performance, greatly improved price performance, and hardware and programming systems compatibility. These characteristics also make the 4331 Processor a suitable growth processor for installations with a System/360 Model 20.

The advanced functions provided by the 4331 Processor, its price performance, emphasis on ease of installation, and reduced power and cooling costs make it an attractive processor for businesses without an intermediate-scale System/360 or System/370 processor installed. These characteristics, together with the transition aids available, also make the 4331 Processor a suitable growth system for installations with large IBM System/3 configurations.

The 4331 Processor can be installed as a standalone system to handle all traditional application areas. Its compatibility with System/360 and System/370 and its programming systems support also enable the 4331 Processor to be incorporated in the communications networks of large installations and multiple-system installations without disruption of normal production processing.

In addition, the telecommunications capabilities of the 4331 Processor (which include synchronous data link control support), its low-cost direct access storage, and its compact physical design (which results from large-scale integrated technology and integrated I/O attachments) enable the 4331 Processor to function well in a distributed data processing environment. The 4331 Processor can operate as a systems network architecture (SNA) node or as a host to other SNA terminals (see discussion at the end of this section).

Transition from System/360 Model 30 and 40 configurations to a 4331 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 4331 Processor. More effort is required for a Model 20 user to convert to a 4331 Processor because of basic differences between the Model 20 and other System/360 processors (see Section 60).

Compatible growth from System/360 operating systems to a 4331 Processor virtual storage environment can be achieved using Disk Operating System/Virtual Storage Extended (DOS/VSE) and Operating System/Virtual Storage 1 (OS/VS1), which is extended to support the 4331 Processor as of Release 7. DOS/VSE is a compatible extension of DOS/VS Release 34, which utilized DOS Version 4 as a base, while OS/VS1 is based on OS MFT. DOS/VS Release 34 will also operate correctly on the 4331 Processor but will not be modified to support its features.

In addition to supporting virtual storage, the operating systems that support the 4331 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), a System/360 operating system, namely DOS Version 3 (Release 26), can be used on a 4331 Processor.

Transition with little or no reprogramming is also provided for Model 30 and 40 users who are emulating 1401/1440/1460 systems. Improved emulators for these 1400 systems that operate under DOS/VSE, DOS/VS, and DOS control on the 4331 Processor are available.

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 4331 Processor. VM/370 Release 5 with the appropriate PLC level will also operate correctly on the 4331 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.

Highlights of the 4331 Processor, when compared with System/360 Models 30 and 40, are as follows:

- Upward compatibility with most System/360 and System/370 architecture and programming systems has been maintained in the 4331 Processor through implementation of the System/370 mode of processor operation. System/370 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 4331 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.

- When DOS Release 26 and a commercial workload (RPG II payroll application, etc.) were executed, the instruction execution speed of the 4331 Processor measured from 3.3 to 5.6 times faster than that of the System/360 Model 30 processor executing the same workload under DOS Release 26.

  When DOS/VSE and ECPS:VSE mode were used in a 4331 Processor with 3310 Direct Access Storage attached, the 4331 Processor executing a representative batch jobstream was measured to have an instruction execution rate of 4.4 to 5.4 times that of a System/370 Model 115 Model 0, 2.5 to 4.7 times that of a System/370 Model 115 Model 2, and 2.0 to 3.3 times that of a System/370 Model 125 Model 2 (all with 3340 disks attached), using DOS/VS Release 34 and the same representative batch jobstream.

  With DOS/VSE executing in the 4331 Processor and ECPS:VSE mode in effect, a reduction of up to 20 percent of the total processor time used was measured as compared with DOS/VSE executing the same workload on an identical 4331 Processor configuration operating in System/370 mode.

- The following are instruction processing function features of the 4331 Processor.

  Implementation of a System/370 mode and an ECPS:VSE mode is standard. Both modes provide the ability to support virtual storage. 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 4331 Processors and is architecturally equivalent to BC mode for System/370.

  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 is architecturally equivalent (with a few exceptions) to the EC mode of System/370.

  When System/370 and EC modes are in effect, the standard dynamic address translation (DAT) facility can be enabled to provide translation of instruction addresses (but not addresses in channel command words) during program execution. The standard channel indirect data addressing function is provided for this 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 4331 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 with 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 and 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.

  The cycle time of the instruction processing function varies from 200 nanoseconds to 1600 nanoseconds in 100-nanosecond increments, depending on the function performed.

  The standard instruction set for the 4331 Processor includes many new general purpose and control-program-oriented instructions in addition to the powerful System/360 instruction set. The standard 4331 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 (see Section 10:30 for the instructions not available for System/360).

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 10 milliseconds resolution, at location 80 in processor storage, that can improve job accounting accuracy is standard. (A 16.6-ms resolution timer is available for Models 30 and 40.)

A time-of-day clock is included as a standard feature to provide more accurate time-of-day values than does the interval timer. The clock has a 16-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 16 microseconds, 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.

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 operate only when the 4331 Processor is in EC mode.

The Extended Control Program Support:Virtual Machine/370 (ECPS:VM/370) feature is optional. This feature is designed to increase the performance of VM/370 when it operates in the 4331 Processor, since it causes certain control program functions of VM/370 to be executed in hardware instead of via programming. This feature cannot be installed with the 1401/1440/1460 Compatibility feature and can be enabled only when System/370 mode is in effect.

Functionally improved relocatable emulators are available that operate under operating system control. Concurrent execution of 4331 Processor programs with any combination of 1401, 1440, and 1460 programs in a DOS/VSE, DOS/VS Release 34, or DOS Release 26 multiprogramming environment is supported for the 4331 Processor. The IBM Systems 1401/1440/1460 Emulator Program (a program product) supports 1400 emulation using the no-charge optional 1401/1440/1460 Compatibility feature for the 4331 Processor or a 1400 simulator program. Use of the compatibility feature provides better performance. The 1401/1440/1460 Compatibility feature cannot be installed with the 8809 Magnetic Tape Unit Adapter or ECPS:VM/370 features.

- The following are significant storage features of the 4331 Processor.

  All storage in the 4331 Processor--local, control, and processor (main)--is implemented using monolithic technology instead of discrete ferrite cores. The technology used for processor and control storage in the 4331 Processor provides much denser storage chips than are used in System/370 (64K-bits per chip and 18K-bits per chip for processor and control storage, respectively, versus 2K-bits per chip in most System/370 processor storage).

  512K and 1024K of processor storage are available--16 times the maximum amount of main storage available on the Model 30 and 4 times the maximum amount available on the Model 40. Processor storage has a read cycle time of 900 nanoseconds and a write cycle time of 1300 nanoseconds for a full word.

  A portion of the processor storage installed is used to contain certain instruction processing function microcode, processor tables and work areas, and (optionally) 231X and/or 3340 emulation buffers. The amount of processor storage reserved for processor use varies depending on the optional features installed and mode (System/370 or ECPS:VSE) in effect.

  The minimum amount of reserved processor storage is 16K bytes (optional Control Storage Expansion feature installed, System/370 mode of operation or ECPS:VSE mode with one megabyte of virtual storage defined, channel-attached 231X direct access storage, and no optional I/O adapters or other microcoded features). The minimum reserved processor storage requirement without Control Storage Expansion installed is 52K.

  64K of reloadable control storage (RCS) is standard. The optional Control Storage Expansion feature provides an additional 64K of RCS and is required when more than 64K of RCS is needed to contain the microcode for the selected optional features.

  Control storage contains the most frequently used processor microcode, including that needed for both standard and optional features. 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 processors), 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.

  Spare bits are also included in the processor storage in each 4331 Processor. Via the processor storage defect dictionary, the customer engineer can assign spare bits as substitutes for malfunctioning primary processor storage bits.

- The following channel and I/O adapter features are provided for the 4331 Processor.

  Several integrated I/O adapters are available that offer lower-cost attachment of I/O devices. The adapters enable I/O devices to be attached to the 4331 Processor directly without a channel and control unit. In addition, channels can be installed. All optional I/O adapters and channels available can be installed in the same 4331 Processor when certain configuring rules are observed.

  Optionally, one byte multiplexer channel with a maximum data rate of 18 thousand bytes per second (KB/sec) in byte mode and 500 KB/sec

maximum data rate in burst mode can be installed. Functionally, the byte multiplexer channel for the 4331 Processor is equivalent to that for System/360 and System/370 processors.

Optionally, one block multiplexer channel with a maximum data rate of 500 KB/sec can be installed. Functionally, the block multiplexer channel 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 4331 Processor can also operate in selector channel mode.

The Display/Printer Adapter is standard. It provides the ability to connect the required operator console device and up to seven additional display and/or printer devices to the 4331 Processor without the need for a channel and control unit. The optional Display/Printer Adapter Expansion feature can be installed to permit an additional eight units (required console and up to 15 displays and/or printers) to be attached to the Display/Printer Adapter. The 3278 Display Station Model 2, 3287 Printer Models 1 and 2, 3262 Line Printer Model 1, and 3289 Line Printer Model 4 can be attached directly to the 4331 Processor via this adapter.

Optionally, one DASD Adapter can be installed for the direct attachment of up to four strings of direct access devices. The DASD Adapter functions as a block multiplexer channel with an attached disk storage control unit that has multiple requesting and rotational position sensing capabilities. Rotational position sensing is implemented for direct access devices to enable them to utilize the block multiplexing capability.

The DASD Adapter provides for the direct attachment of 3310 and 3370 Direct Access Storage (up to four strings in any combination of 3310 strings and 3370 strings). When the optional 3340 Direct Attach feature is installed with the DASD Adapter, 3340 direct access devices can be attached to the 4331 Processor via the DASD Adapter (maximum of two 3340 strings). The DASD Adapter can have 3310, 3370, and 3340 strings attached when 3340 Direct Attach is installed.

Optimum processor and direct access device performance is achieved when 3310 and/or 3370 disk drives are installed. Attachment of 3340 disk is provided to facilitate the coexistence of the 4331 Processor with System/370 and 4341 processors. The 3340 Direct Attach feature cannot be installed when 231X disk devices are attached to the block multiplexer channel of the 4331 Processor.

Optionally, the System/3 Data Import feature can be installed together with the DASD Adapter. This feature enables 3340 direct access devices (A and B models) to be attached to the DASD Adapter and permits the reading (but not writing) of 3348 Data Modules that were created by IBM System/3 Model 12 or 15 processors. The files read can then be placed on a direct access device type that the 4331 Processor can read and write. This feature is designed to be used as a conversion aid for System/3 users. The System/3 Data Import feature cannot be installed if 231X devices are attached to the block multiplexer channel.

The optional 2311/2314/2319/3310 Direct Access Storage Compatibility feature is a disk conversion aid. It enables 3310 drives attached to the DASD Adapter to be used to simulate 2311 or 2314/2319 disk

drives. This feature enables DOS, DOS/VS, and DOS/VSE programs that read from, and write to, 2311 or 2314/2319 drives to execute in the 4331 Processor without modification and access the 231X files on 3310 drives once the 231X files have been transferred to 3310 drives.

Optionally, one 8809 Magnetic Tape Unit Adapter can be installed to permit up to six 8809 Magnetic Tape Units to be attached directly to the 4331 Processor. This adapter cannot be installed if 231X disk devices are attached to the block multiplexer channel or if the optional 1401/1440/1460 Compatibility feature is installed.

The optional Communications Adapter for the 4331 Processor offers lower-cost communications line attachment than is available for Models 30 and 40. This adapter permits direct attachment to the 4331 Processor of up to eight low-speed (up to 9600 bits per second) lines. The maximum aggregate data rate of the adapter is 64,000 bits per second (bps). One high-speed line with a data rate over 9600 bps but not higher than 56,000 bps can be installed and can operate concurrently with other lines attached to the Communications Adapter as long as the maximum aggregate data rate of 64,000 bits per second is not exceeded and no other line operates at a data rate higher than 9600 bits per second.

Start/stop, binary synchronous, and synchronous data link control communications are supported by the Communications Adapter but only two of the three line control types, in any combination, can be installed in a 4331 Processor. Two autocall units can be installed as well. Communications lines can also be attached to the 4331 Processor via 2701, 2702, 2703, 3704, and 3705 transmission control units.

Optionally, one 5424 Adapter can be installed to permit one 5424 Multi-Function Card Unit to be attached directly to the 4331 Processor. This feature is provided as another conversion aid for System/3 users and to assist transition from System/370 processors that use the 5425 Multifunction Card Unit.

Optionally, one diskette drive (which includes its own adapter) can be installed. This drive reads and writes diskettes, in EBCDIC code, that are interchangeable with diskettes for the diskette drives in other IBM devices and processors.

- 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 is standard and printer-keyboard mode can be installed as an option. 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 the 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 diagnosing processor malfunctions.

  Optionally, a 3287 Model 1 or 2 Printer can be installed via the standard Display/Printer Adapter to provide hard-copy backup for the operator console when display mode is used. A 3287 Printer is required for printer-keyboard mode, which is made available 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 4331 Processor.

- I/O devices for the 4331 Processor include the following.

  Most I/O devices for System/360 Models 30 and 40 can be attached. In addition, several I/O devices that cannot be attached to Models 30 and 40 because of their channel data rate or to any System/360 processors are attachable to the 4331 Processor.

  The 3505 Card Reader and the 3525 Card Punch with optional card read capability (not attachable to Models 30 and 40) can be channel-attached to the 4331 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 2540 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 2540 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 3289 Line Printer Model 4, with a print speed of 400 alphameric lines per minute for a 48-character set, can be attached via the Display/Printer Adapter. This printer (not attachable to Models 30 and 40) provides economical printing capability for installations that do not have a high volume of print activity.

  The 3262 Model 1 Line Printer, with a print speed of 650 lines per minute for a 48-character set, can be attached to the Display/Printer Adapter. This printer (not attachable to Models 30 and 40) is similar in construction to the 3289 Model 4 but provides faster print speeds. A maximum of two line printers, any combination of 3262 Model 1 and 3289 Model 4 printers, can be attached to the Display/Printer Adapter.

  The 3203 Model 5 Printer (not attachable to Models 30 and 40), with a tapeless carriage and print speed of 1200 alphameric lines per second (with a 48-character set), can be channel attached. The 3203 Model 5 contains its own control function and 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 other 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 channel 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 30 and 40) can be channel 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 8809 Magnetic Tape Unit (not attachable to System/360 or System/370) can be attached via the 8809 Magnetic Tape Unit Adapter. An 8809 Magnetic Tape Unit can operate in start/stop mode like 2400- and 3400-series tape units, starting and stopping between interblock gaps during tape block processing, or in a streaming mode not implemented for these tape units.

When operating in streaming mode, an 8809 tape unit does not stop the tape between interblock gaps and maintains tape velocity while passing over an interblock gap, the assumption being another command will be received during the gap-passing time interval. An 8809 tape unit can operate at a 160-KB/sec instantaneous maximum data rate during streaming operations and at a 20-KB/sec instantaneous maximum data rate for start/stop mode. For the 4331 Processor, the 8809 tape unit can be used as a high-speed save/restore device for non-removable direct access devices, such as 3310 and 3370 disk storage, and for traditional tape functions.

The 3410/3411 Magnetic Tape Subsystem, Models 1, 2, and 3, can be channel 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 channel 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. Model 4 of the 3420 Magnetic Tape Unit (not attachable to Models 30 and 40) can also be attached and has a data rate of 470 KB/sec at 6250-BPI recording density. The advanced recording technique used for the Model 4 provides automatic correction of all single- and double-bit read errors in flight.

The 3803/3420 tape subsystem offers significantly faster data rates than 2400-series tape units at 6250-BPI density; 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 improved reliability, availability, and serviceability features.

3310 Direct Access Storage (not attachable to System/360 or System/370) is intermediate-capacity, fixed media disk storage with movable heads. Each drive has a capacity of 64.5 million bytes. Up to four strings of from one to four drives each (in one-drive increments) can be attached to the 4331 Processor only via the DASD Adapter to provide a maximum of sixteen 3310 drives with a capacity of 1,032 million bytes.

A 3310 drive has a 1.031-MB/sec data transfer rate, average seek time of 27 ms, and full rotation time of 19.2 ms. A fixed-block

recording technique (fixed block architecture) is used for data on 3310 tracks instead of the self-formatting (count, key, data) recording technique implemented for System/360 and System/370 direct access devices. Fixed block architecture is designed to utilize fully the block multiplexing capability of the DASD Adapter (rotational position sensing is a standard capability of the 3310). It provides reduced storage control function cost and data mobility advantages over the self-formatting recording technique.

3370 Direct Access Storage (not attachable to System/360 or System/370) is very large capacity, high-speed, fixed-media direct access storage with movable heads. Data is stored on the nonremovable disks using the fixed block architecture that is utilized for 3310 Direct Access Storage. The 3370 can be attached to the 4331 Processor only via the DASD Adapter.

A 3370 drive has a data transfer rate of 1.86 MB/sec, average seek time of 20 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 capacity of 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. Thus, a maximum of approximately 9.2 billion bytes of 3370 Direct Access Storage can be attached to the 4331 Processor via the DASD Adapter.

Each 3370 contains two logical drives, one per actuator. While only one actuator can transfer data at a time, the other actuator can be performing seeking and rotational positioning operations.

The 3310 and 3370 disk storage units offer many advantages over both 2314 and 3340 disk storage, such as lower cost; reduced space, power, and cooling requirements; and increased reliability.

3340 Direct Access Storage can be attached to the 4331 Processor via the DASD Adapter when the 3340 Direct Attach feature is installed. The 3340 is intermediate-capacity direct access storage that, because of its unique design and advanced technology, offers advantages over 2314 disk storage. It utilizes the count, key, data recording technique.

The storage medium for 3340 disk storage 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.

The 3340 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 or 70 million bytes (assuming full-track records), 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. One or two 3340 strings can be attached to the 4331 Processor only via the DASD Adapter. Any model of the 3348 can be mounted on a 3340 drive. Therefore, 3340 string capacity can vary from 70 million bytes to 560 million bytes in increments of 35 and/or 70 million bytes (assuming full-track records).

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 procedures.

- The technology implemented in the 4331 Processor for logic, processor storage, and control 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, significantly reduced cost, and maintenance improvements. Power consumption by the 4331 Processor is also reduced by certain other design features (see Section 50:10).

- Extensive hardware and programming systems error recovery and repair features not implemented for System/360 are provided to enhance 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 replaceable unit or the procedure to follow to locate the malfunction.

  In addition, remote diagnosis of hardware failures by IBM support center personnel is supported via the optional Remote Support Facility. Remote diagnostic facilities are not provided for System/360 processors or intermediate-scale System/370 processors.

As the highlights indicate, the 4331 Processor offers Model 30 and 40 users a wide variety of improved features and additional functions that provide improved throughput and expanded capabilities. Specifically, the 4331 Processor offers the following advantages when compared with Models 30 and 40. (Advantages for the Model 20 user are discussed in Section 60.)

Larger, Faster Processor Storage

Processor storage sizes of 512K and 1024K are provided. The Model 30 can have a maximum of 64K and the Model 40 a maximum of 256K. The cycle time of processor storage in the 4331 Processor is faster than that of Models 30 and 40 (900 or 1300 nanoseconds versus 1500 nanoseconds for the Model 30 and 2500 nanoseconds for the Model 40), during which four bytes instead of one (Model 30) or two bytes (Model 40) can be accessed. This improved cycle time increases internal performance and permits faster I/O devices to be attached to the 4331 Processor.

The availability of two processor storage sizes for the 4331 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 4331 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 4331 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 4331 Processor and the new facilities supported by operating systems that support the 4331 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 the operational flexibility of the installation (see discussion in Section 15:05).

Greatly Expanded I/O Capabilities

The fast internal performance of the 4331 Processor, together with the expanded use of multiprogramming, requires that more data be available faster than on Models 30 and 40.

The 4331 Processor supports more and faster concurrent high-speed I/O operations than Models 30 and 40. It also provides the block multiplexing capability, which is not available for System/360 processors. Integrated I/O adapters, not available for the Model 30 or 40, are also offered.

The I/O features of the 4331 Processor provide:

- Attachment of the 3505 reader and 3525 punch, not attachable to Models 30 and 40
- Direct attachment of one diskette drive for data interchange
- Lower-cost direct attachment of two 3289 Model 4 and/or 3262 Model 1 Line Printers via the standard Display/Printer Adapter. In addition, a larger variety of printers can be channel attached to the 4331 Processor than to Models 30 and 40 (3203 Model 5 Printer and 3800 Printing Subsystem in addition to 1403 and 3211 Printers).
- Lower-cost direct attachment of 3278 Model 2 displays and 3287 Printers via the Display/Printer Adapter
- Lower-cost direct attachment of up to eight communications lines via the Communications Adapter and support of synchronous data link control communications

- Lower-cost direct attachment of lower-cost disk storage (3310 and 3370 drives)

- Potential increases in channel and DASD Adapter 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 30 and 40 to balance the higher performance capabilities of the 4331 Processor. The 4331 Processor is capable of achieving an aggregate data rate in excess of 2 MB/sec.

Faster I/O Devices with Increased Data Capacity

The 4331 Processor supports faster magnetic tape units than do Models 30 and 40. Models 7 and 4 of the 3420 with data rates of 320 and 470 KB/sec, respectively, at 6250 BPI cannot be attached to Models 30 and 40. A 3420 Magnetic Tape Unit at 6250-BPI density supports a tape reel capacity approximately three times greater than a 2400-series tape volume.

An I/O configuration for the 4331 Processor can also include significantly more lower-cost, higher-capacity, faster direct access storage. Models 30 and 40 are limited to having 2314 facilities on only one channel and cannot have 3310, 3370, or 3340 disk storage attached.

The 3310, 3370, and 3340 direct access devices provide considerably more capability and faster data access than 2314 drives because of higher data transfer rates, faster rotation, and new features. They also offer higher availability through use of new hardware-only and program-assisted error correction features as well as improved reliability. The 3310 and 3370 offer additional advantages inherent in the design of fixed block architecture.

The 3310 and 3370 provide large storage capacity and fast access for a lower cost per bit than 23XX and 3340 disk storage. They are growth devices that offer improved price performance for the 2314 facility, the 2321 Data Cell Drive, and 3340 disk storage. Like 2314 disk storage, the 3310 and 3370 are designed to be used in every area in which direct access storage is needed, for example:

- As a system residence device and for program library storage

- As external page storage in a virtual storage environment

- As high-speed work storage for sorting, assembling, and link editing

- For residence of data indexes, such as for VSAM data sets

- In data base/data communications applications, such as management information systems

- In time-sharing (or interactive) environments as swap devices and for online work storage (for program and data residence)

Summary

Since hardware features and programming systems support for the 4331 Processor are upward compatible with that of System/360, the 4331 Processor offers Model 30 and 40 users significantly expanded computing capabilities without the necessity of a large conversion effort. Little or no time need be spent modifying operational System/360 problem programs or the 1400 programs currently being emulated.

Existing processor-bound System/360 programs can execute faster in a 4331 Processor because of the significantly increased internal performance of the 4331 Processor, while I/O-bound programs can benefit from the use of more processor storage, more channel capability, block multiplexing, and faster I/O devices. The 4331 Processor also offers economical and flexible entry into communications-based applications.

The increased power and new functions of the 4331 Processor provide the base for expanded applications installation and penetration of previously marginal application areas. New applications installation and transition to online operations can be easier when a virtual storage environment is implemented. The greatly improved price performance of the 4331 Processor offers the System/360 user the opportunity to widen his data processing base for a significantly lower cost than was previously possible.

Users of large System/3 configurations who need to expand their data processing functions will also find the low price and high performance of the 4331 Processor attractive. The hardware and programming conversion tools provided for the transition to a 4331 Processor will ease the conversion process.

For large installations that want undisrupted growth and decentralization of their data processing facilities, the 4331 Processor provides economical and easy entry into (or expansion of) distributed data processing operations. The following outlines the distributed application environments in which the 4331 Processor can be used:

- Distributed applications (periodic data transfer, remote job entry, and pass through). The 4331 Processor can provide powerful processing capability in a department or branch location of a larger enterprise, with teleprocessing 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 4331 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 4331 Processor via RJE facilities. RJE support is provided by DOS/VSE SNA and BSC program products, VM/370 RSCS Networking, and the OS/VS1 Host Remote Node Entry System (HRNES) IUP (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 4331 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 processors. Data most frequently used locally may be stored on direct access storage attached to the 4331 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 4331 Processors to the host, or to 8100 Information Systems. With a complete network, both data and processor loads can be spread to larger processors, to distributed 8100 Information Systems, or between 4331 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 4331 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 processor and control storage in the 4331 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 SLT logic (Solid Logic Technology) and core storage implemented in System/360, the large-scale integrated logic and SAMOS storage technologies implemented in IBM 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 4331 Processor, which contain several times more circuits than the MST logic chips and modules utilized in System/370. The very high density of the logic chip used in the 4331 Processor, 704 circuits maximum per chip, and its packaging for the 4331 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

The SAMOS technology used for processor storage in the 4331 Processor also provides greatly improved chip density compared to the processor storage chip used in most System/370 processors. In the 4331 Processor, a processor storage chip contains 64K bits, while a 2K-bit chip is used for processor storage in most System/370 processors. An 18K-bit SAMOS technology storage chip is used for control storage in the 4331 Processor.

The higher densities of the SAMOS storage chips used in the 4331 Processor are achieved by (1) utilizing fewer elementary components (such as transistors) to form an individual storage cell (that can contain a 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 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 10 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 chips used in the 4331 Processor, the SAMOS technology and design of the chip, and the new manufacturing facility result in the following advantages for processor storage in the 4331 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 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.

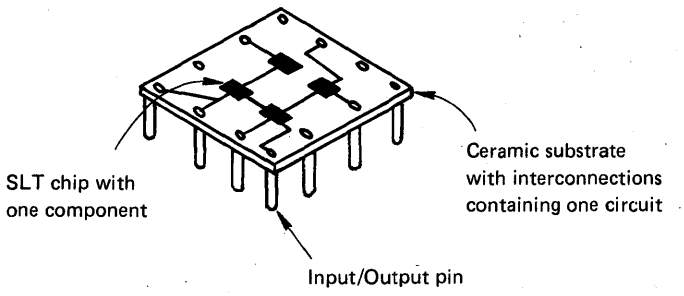


Figure 05.05.1. SLT substrate (shown four times its actual size)

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-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 three thirty-seconds 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 8 to 12 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 2048 (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) in size and contains a large number of interconnected circuits that form 2K storage cells and 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. 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 general advantages of a static monolithic storage over core storage are:

- Faster storage speeds can be obtained, first, because of the shorter paths between storage circuitry and, second, because of 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 only identify 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.

4331 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 4331 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 then 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 4331 Processor are significantly reduced from that required by the technology implemented in System/370. Separate logic boards and separate processor storage boards and the cabling between them is eliminated because only three cards are required to contain the logic for the instruction processing

function of the 4331 Processor and only one or two cards are required for processor storage.

Logic Physical Design and Advantages

The logic chip used in the 4331 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 a 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 the 704-circuit logic chip on the right surrounded by salt crystals. The relative sizes of these chips can be seen in the latter figure.

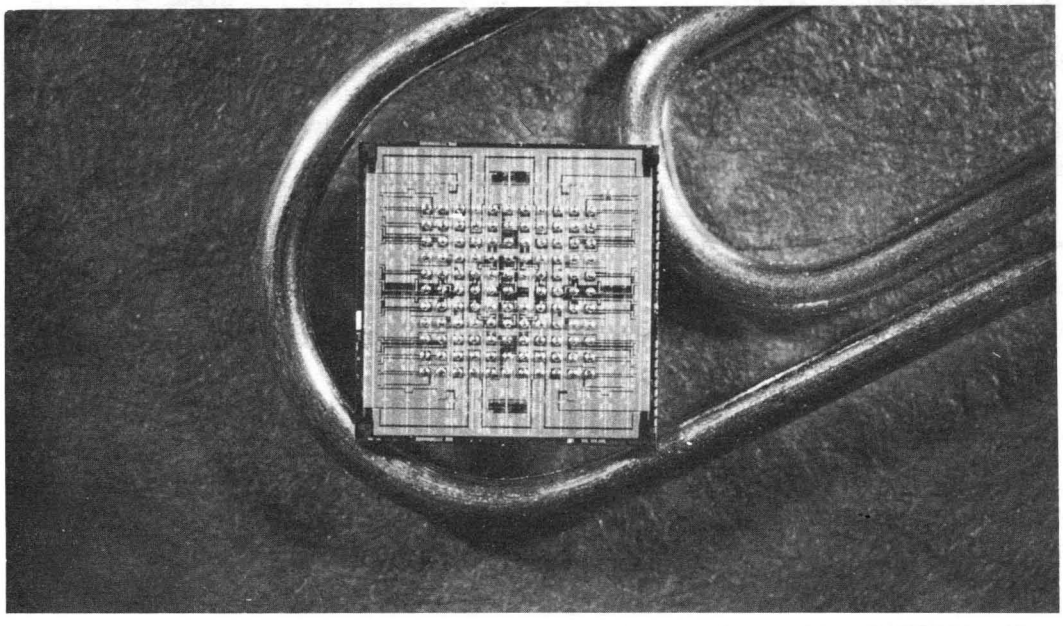


Figure 05.05.2. The 704-circuit logic chip resting on a paper clip

Of the 704 circuits available on a single chip, an average of 650 are actually utilized in the logic implemented in the 4331 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 4331 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 the logic chip in the 4331 Processor, a higher percentage of the circuits available on the chip are used than are used on the MST logic chip. For the 4331 Processor, average circuit utilization on a logic chip is over 90 percent, compared to 75 percent average utilization for the MST logic chip in System/370 processors.

The ceramic substrate used for the logic modules in the 4331 Processor is 50 by 50 millimeters (approximately 2 inches square) and is called a multilayer ceramic (MLC) because it has up to 23 layers of ceramic available to accommodate circuit interconnections within the ceramic itself. An MLC module for the 4331 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.

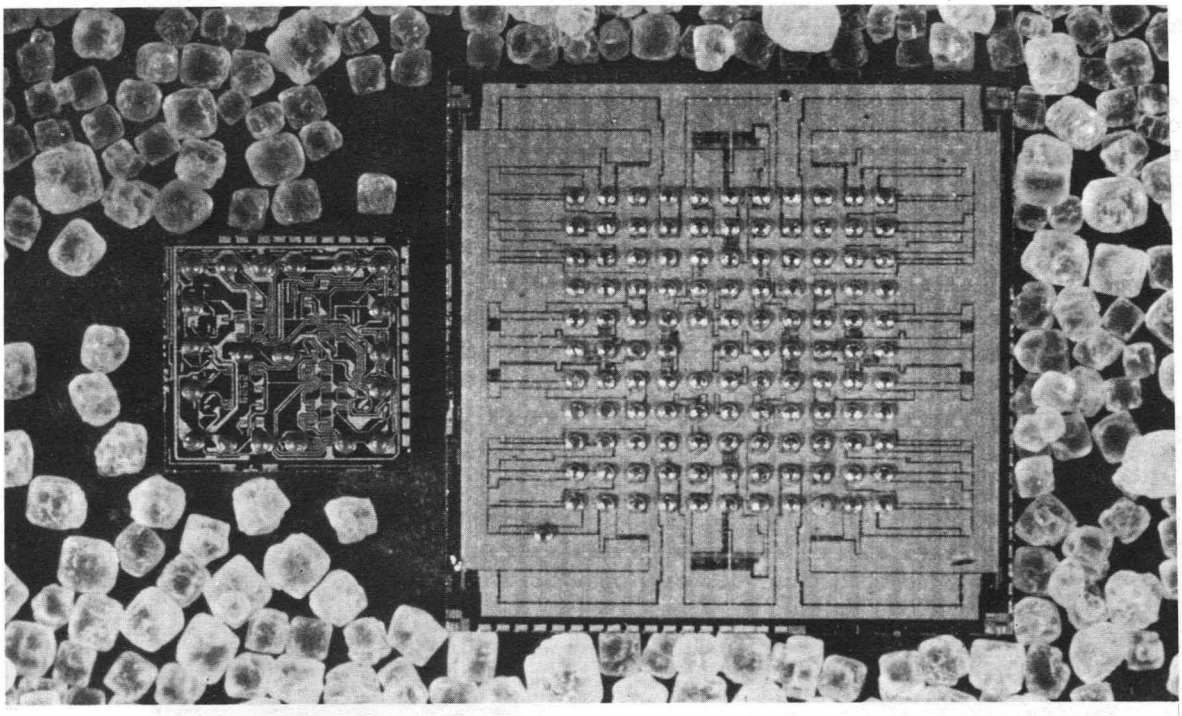


Figure 05.05.3. An MST and 704-circuit logic chip surrounded by salt crystals

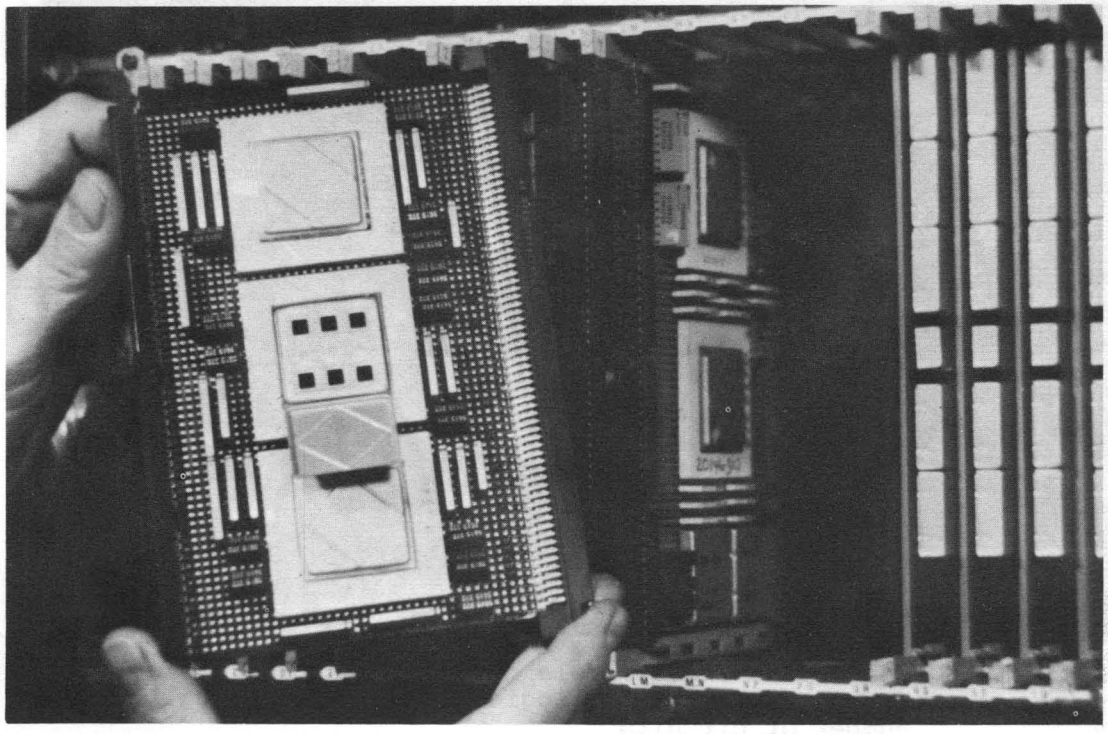
The logic module used in the 4331 Processor is called an MLC 50 module. This module is also used in the 4341 Processor. The MLC 35 module that is used in the 4341 Processor, which is 35 by 35 millimeters (approximately 1.5 inches square), and the MC 28 module (metalized ceramic module), which is 28 by 28 millimeters (slightly over one-inch square), are not used in the 4331 Processor.

The larger ceramic size and use of multilayer wiring within an MLC 50 substrate enable several logic chips to be packaged in a single logic module. For the 4331 Processor, up to nine logic or array 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 in the 4331 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 4331 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.

MLC 50 modules are mounted on circuit cards 11.32 by 17.46 centimeters (approximately 4.5 by 7 inches) in size. Logic cards in the 4331 Processor are mounted on a board that also contains processor storage. 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 (required to handle the higher circuit density per module). The MLC 50 module used in the 4331 Processor has 361 pins, which compares to 16 for an MST logic module.

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.



three mounted on an FRU

Figure 05.05.4. The MLC 50 modules used for instruction processing function logic in the 4331 Processor

The large space reduction that results from the use of the 704-circuit logic chip and the multi-chip, 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 4331 Processor. Assuming the maximum capacity of 704 circuits per chip 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 logic module and 700 MST logic modules that provide the same 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.

The advantages of the logic technology used in the 4331 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 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.

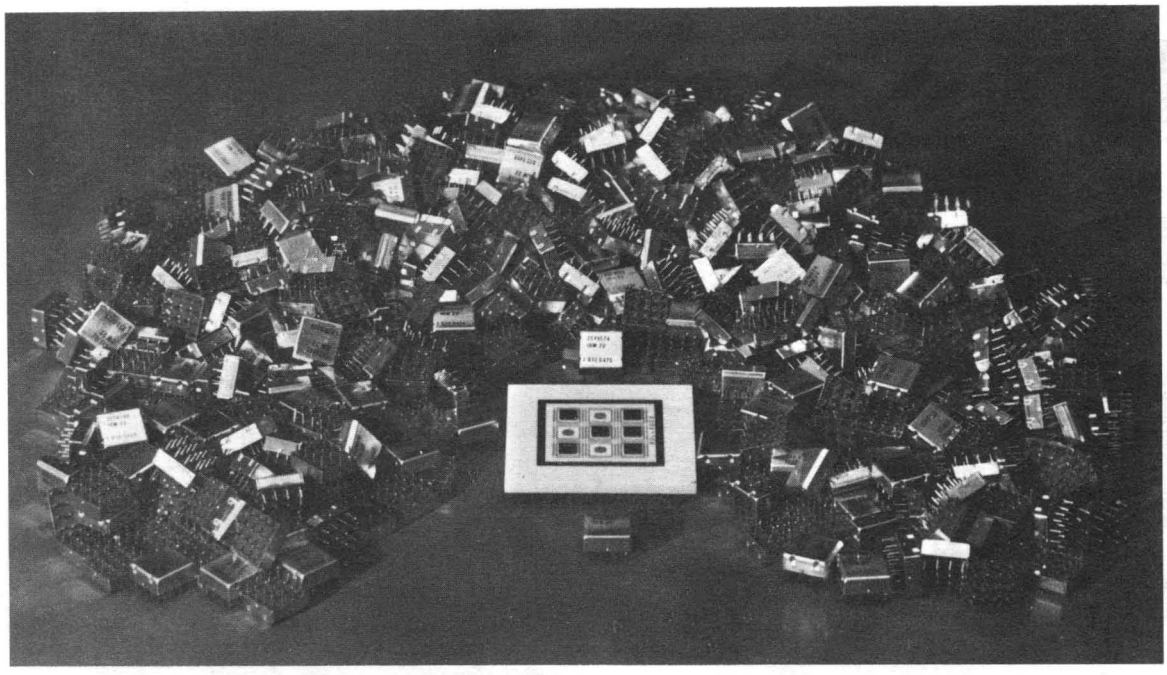


Figure 05.05.5. One MLC 50 logic module containing approximatley 4200 circuits and 700 MST logic modules containing the same number of circuits

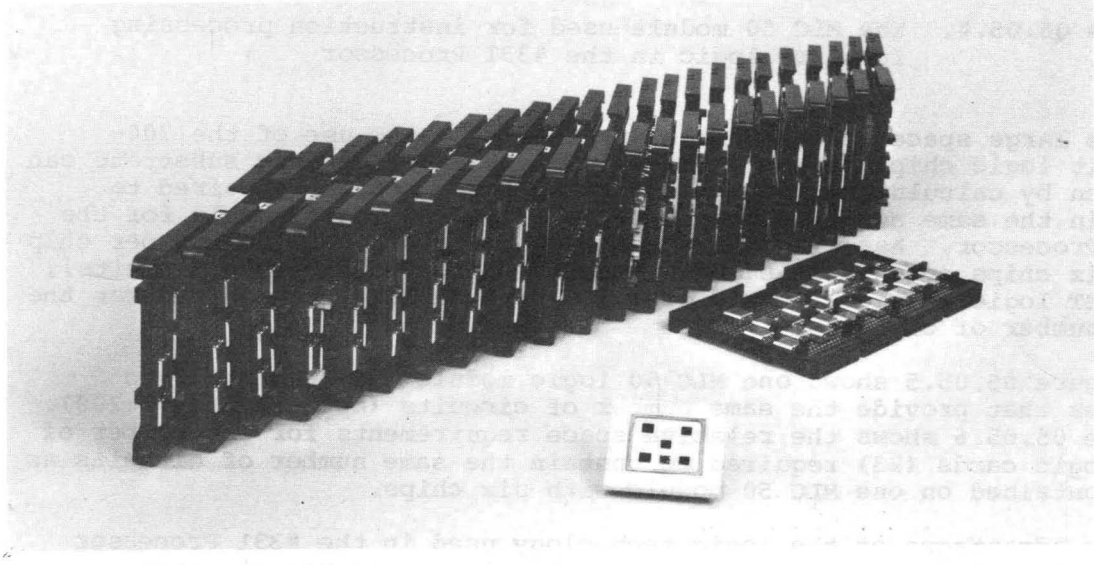


Figure 05.05.6. One MLC 50 logic module with six chips and 23 MST logic cards with the same number of circuits

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 chip 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 in the amount of wire required).

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 4331 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 instruction processing function of the 4331 Processor contains approximately 60,000 logic circuits while the System/370 Model 115 contains approximately 25,000; however, logic circuitry in the 4331 Processor requires only three logic cards, compared to 43 logic cards in the Model 115.

- 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 4331 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 4331 Processor, no preventive maintenance is scheduled for 4331 Processors.

  Serviceability is improved because the diagnostic procedures for the 4331 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 it 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 an MLC logic module 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 used in the 4331 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 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 operation.

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, 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 a 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 4331 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 4331 and 4341 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 4331 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. Processor storage and control storage in the 4331 Processor are a dynamic type of monolithic storage, as opposed to the static type of storage implemented in System/370 processors.

The 64K (65,536) bit SAMOS chip that is used in processor storage in the 4331 Processor utilizes one transistor per storage cell, while the 18K (18,432) bit chip used in control storage utilizes two transistors per storage cell. Processor storage in the 4331 Processor is sometimes referred to as a single-cell storage while control storage is referred to as a twin-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) and the 18K-bit chip is 4.8 by 5.5 millimeters (approximately 6/32 by 7/32 of an inch). Six transistors per bit are used on a 2K-bit storage chip for System/370, which is 3.88 by 4.52 millimeters (approximately 5/32 by 6/32 of an inch). 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 storage implemented in the 4331 Processor, a storage cell is implemented as one or two capacitors 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".

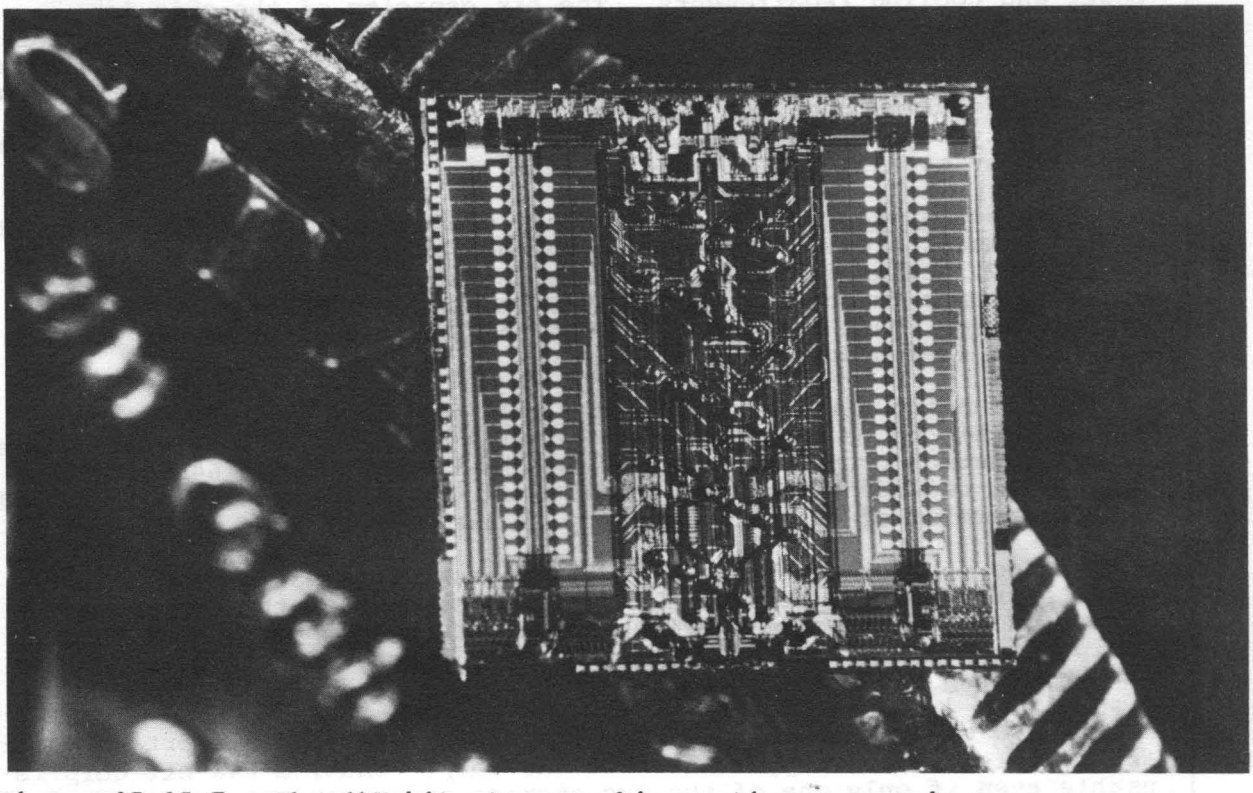


Figure 05.05.7. The 64K-bit storage chip resting on a coin

A portion of the processor storage installed in a 4331 Processor is automatically refreshed every 11.2 microseconds. For a 512K-byte processor storage, a different 1K bytes of storage is refreshed every 11.2 microseconds such that in 5.73 milliseconds (or 512 refresh operations) all processor storage has been refreshed. For a 1024K-byte processor storage, a different 2K bytes of storage is refreshed every 11.2 microseconds.

The refresh cycle for 1K or 2K bytes of processor storage is one microsecond in length. If a refresh request occurs while processor storage is busy, the refresh cycle is delayed until the end of the storage cycle. If the instruction processing function attempts to access processor storage during a refresh cycle, it must wait until the refresh cycle completes (maximum delay of one microsecond).

Control storage in the 4331 Processor is automatically refreshed every 50 microseconds and the duration of the refresh cycle is 400 nanoseconds. Control storage need not be refreshed as frequently as processor storage because the components on an 18K-bit chip are larger than those on the 64K-bit chip and the capacitors retain their charge for a longer interval of time.

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 the chip is improved by the use of fewer components per storage 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 4331 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 the reading of storage bits.

The two storage areas each have their own dedicated support circuitry 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 circuity 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 implementation of the eight-bit register, 405 nanoseconds would be required to access the first bit while 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 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 4331 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 4331 Processor is 2.54-centimeters (one-inch) square and contains one or two substrates. A substrate always contains four storage chips. The use of one or two substrates per module enables one or two independent 32K-bit storage arrays per chip to be utilized.

A storage array module for the 4331 Processor contains either (1) two substrates with 8 chips containing 16 storage arrays or 8 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 4331 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 4331 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 4331 Processor contains one or two processor storage cards for 512K or 1024K bytes of processor storage, respectively. A processor storage card containing 512K bytes is shown in Figure 05.05.8. doesn't compute = JMB.

When the 2K-bit chip is used for processor storage in System/370, 18 and 36 cards 10.8 by 17.78 centimeters (4.25 by 7 inches) in size are required for 512K and 1024K bytes, respectively. The volume of space required for 1024K bytes of processor storage implemented in 2K-bit chips is 30 times greater than the requirement for 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 cards required for a given storage size makes locating the failing FRU faster.

The new card and board design used for logic in the 4331 Processor is also used for processor storage. Processor storage array cards in the 4331 Processor are held in place by screws at each end to prevent them from coming loose and, thus, aid reliability.

Note that the 512K-byte processor storage array card for the 4331 Processor contains spare storage bits. For every 39 bits (one word plus ECC bits), one additional bit is present. If a storage bit fails, the customer engineer can assign one of the spare bits to replace it (see discussion of the defect dictionary in Section 50:10).

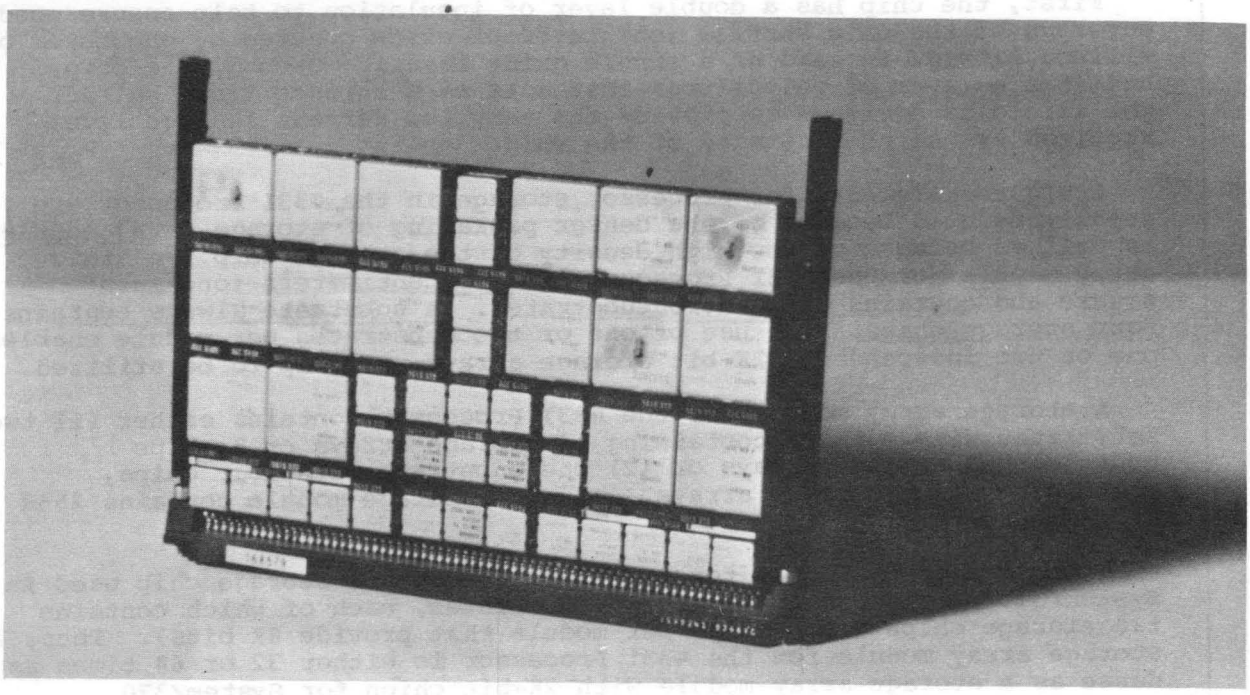


Figure 05.05.8. A processor storage card containing 512K bytes

The reduction in space requirements that results from the use of large-scale integrated logic technology and the high-density SAMOS storage technology in the 4331 Processor is illustrated in Figure 05.05.9. This figure shows the space requirement for one megabyte of 2K-bit-chip processor storage and the logic in the Model 115 on the left and one megabyte of 64K-bit-chip processor storage and the logic in the 4331 Processor on the right. Note that the maximum amount of processor storage for the Model 115 Model 0 is 192K bytes. One megabyte is shown only for comparative purposes.

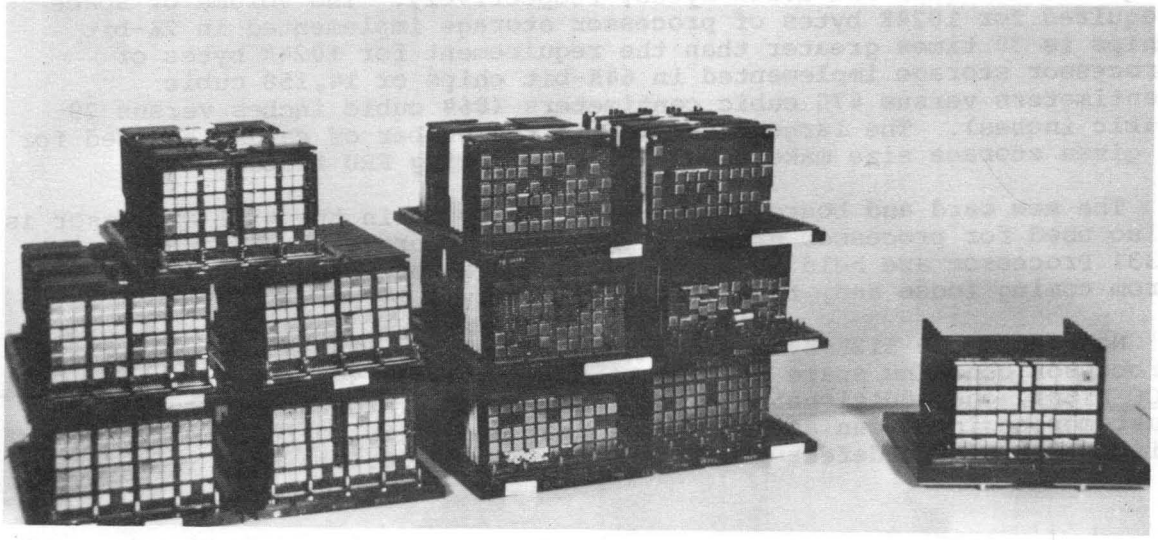


Figure 05.05.9. Logic and one megabyte of processor storage for the Model 115 and the 4331 Processor

Control storage. The SAMOS 18K-bit chip is used in control storage in the 4331 Processor because its design is more suited to the needs of

a control storage. First, the access time of the 18K-bit chip is faster than that of the 64K-bit chip because the larger array signals allow improve sensing. Second, the twin cell design provides increased noise rejection, which is important in a higher performance storage chip.

Like the 64K-bit chip, the 18K-bit chip contains redundant storage cells (about 500) that can be substituted for nonfunctional storage cells during chip fabrication. However, three storage array areas, instead of two, are implemented on the 18K-bit chip. A chip is usable if two or three of the storage array areas are functional.

A control storage array module for the 4331 Processor is 2.54-centimeters (one-inch) square and always contains two substrates. A substrate will contain either two chips with three functional array areas each (36K bits) or three chips with only two functional array areas each (36K bits). A control storage array card for the 4331 Processor provides 64K bytes of storage. Thus, a 4331 Processor will contain one or two control storage array cards for 64K or 128K bytes of control storage, respectively.

Summary

The significant impact made on the physical and environmental characteristics of the 4331 Processor by the use of large-scale integrated logic technology and the high-density SAMOS storage technology can best be shown by comparison with the same characteristics of other System/360 and System/370 intermediate-scale processors.

The 4331 Processor requires less power than System/360 Models 30 and 40 despite its greatly increased function and larger processor storage sizes. Specifically, a 4331 Processor with 1024K bytes of processor storage and attached 3278 Model 2A Display Console and 3287 Printer requires approximately two-thirds the amount of power (measured in kilovolt amps) as a Model 30 processor (2030 Processing Unit) with 64K of processor storage or a Model 40 processor (2040 Processing Unit) with 256K bytes of processor storage. The power requirement for 3310 disk drives is also less than that for 2314 disk drives.

The power requirement for a 4331 Processor with 1024K bytes of processor storage and attached 3278 Model 2A and 3287 is approximately one-half that for a System/370 Model 115 processor (3115-0 Processing Unit) with 192K bytes of processor storage or a Model 125 processor (3125-0 Processing Unit) with 256K bytes of processor storage.

The reduction in power requirements for a 4331 Processor causes it to dissipate less heat, as measured in BTUs per hour. The 4331 Processor with a 3287 Model 2A and 3287 attached generates approximately 69 and 66 percent the amount of heat as Models 30 and 40, respectively, and approximately 45 percent the amount of heat as Models 115 and 125. Thus, less air conditioning is required for a 4331 Processor. The 3310 disk drives also generate less heat than 2314 or 3340 disk drives.

Space savings in square feet of floor space also result when a 4331 Processor replaces a Model 30 or 40. A 512K- or 1024K-byte 4331 Processor (not including the 3287 Model 2A and 3287) requires 46 percent less space than a 256K Model 40 and seven percent less space than a 64K-byte Model 30.

While the space saving for the 4331 Processor itself is not large for a Model 30 user, the total space requirement for a 4331 Processor configuration, taking I/O devices into account, will be less than for a Model 30 configuration when 3310 or 3370 disk drives replace 2311 or 2314 drives. The reduction occurs because (1) the DASD Adapter is contained within the 4331 Processor, while 2311 and 2314 disk storage

require additional space for the control function (2841 or 2314 frames) and (2) a 3310 or 3370 disk storage unit requires less space than a 2311 or 2314 disk storage unit. The Model 40 user will also achieve more space savings by utilizing 3310 or 3370 instead of 2311 or 2314 drives.

The space requirement for a 4331 Processor with 512K or 1024K of processor storage is approximately the same as for a Model 115 or 125 processor with 192K or 256K of processor storage, respectively. However, the height of the 4331 Processor is only one meter (39.36 inches) or about two-thirds that of the Model 115/125 processor. Space savings will result from the use of 3310 or 3370 instead of 231X disk storage and from the greater flexibility in the physical installation layout that is made possible by keeping each component physically separate from the other. For Models 115 and 125, for example, the console and line printer are bolted to the processor.

05:10 DESIGN OBJECTIVES AND ARCHITECTURES

DESIGN OBJECTIVES

The basic design objectives embodied in the 4331 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 4331 Processor provides System/360 users with many new functional capabilities, significant performance improvements, and features to enhance system availability and serviceability. This progress has been achieved under the following conditions:

- The architecture implemented in IBM 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 4331 Processor with either System/370 or ECPS:VSE mode in effect.

- The architecture implemented in IBM 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 4331 Processor with either System/370 or ECPS:VSE mode in effect.

- Programming systems support of the 4331 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 4331 Processor configuration. (See Section 20:05 for a list of the I/O devices that attach to the 4331 Processor.)

ARCHITECTURES IMPLEMENTED

Two architectures are implemented in 4300 Processors: System/370 and 4300 Processor architecture. The mode of processor operation selected during initial microcode load (IML) or initial program load (IPL) determines the architecture that is functional in the 4331 Processor. 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 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.

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 facilites
- 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 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 4331 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 4331 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 4331 Processor (subject to the compatibility constraints discussed later in this subsection and any appropriate operating system constraints).

SYSTEM/360 PROGRAM COMPATIBILITY WITH THE 4331 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 4331 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 4331 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 4331 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 4331 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 execute 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 4331 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

Hardware incompatibilities between the Model 20 and the 4331 Processor operating in BC mode include the above and the same incompatibilities that exist between Model 20 and System/360 Models 22 and up. These differences are discussed in Section 60.

SYSTEM/370 PROGRAM COMPATIBILITY WITH THE 4331 PROCESSOR

Control and processing programs written to operate in BC mode on System/370 can execute in a 4331 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 on System/370 with EC mode and dynamic address translation enabled can operate on a 4331 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 4331 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 4331 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 4331 Processor is shown in Figure 05.15.1. The physical components of a 4331 Processor configuration are the 4331 Processor, a 3278 Model 2A Display Console as the operator console, and I/O devices. The 4331 Processor is air cooled.

The functional components physically contained within the frames of the 4331 Processor are the instruction processing function, all processor storage, integrated channels (optional features), I/O adapters (one is standard), and the support processor subsystem (includes support processor 1). The 4331 Processor can also contain one additional support processor that is part of the optional I/O subsystem and one optional diskette drive.

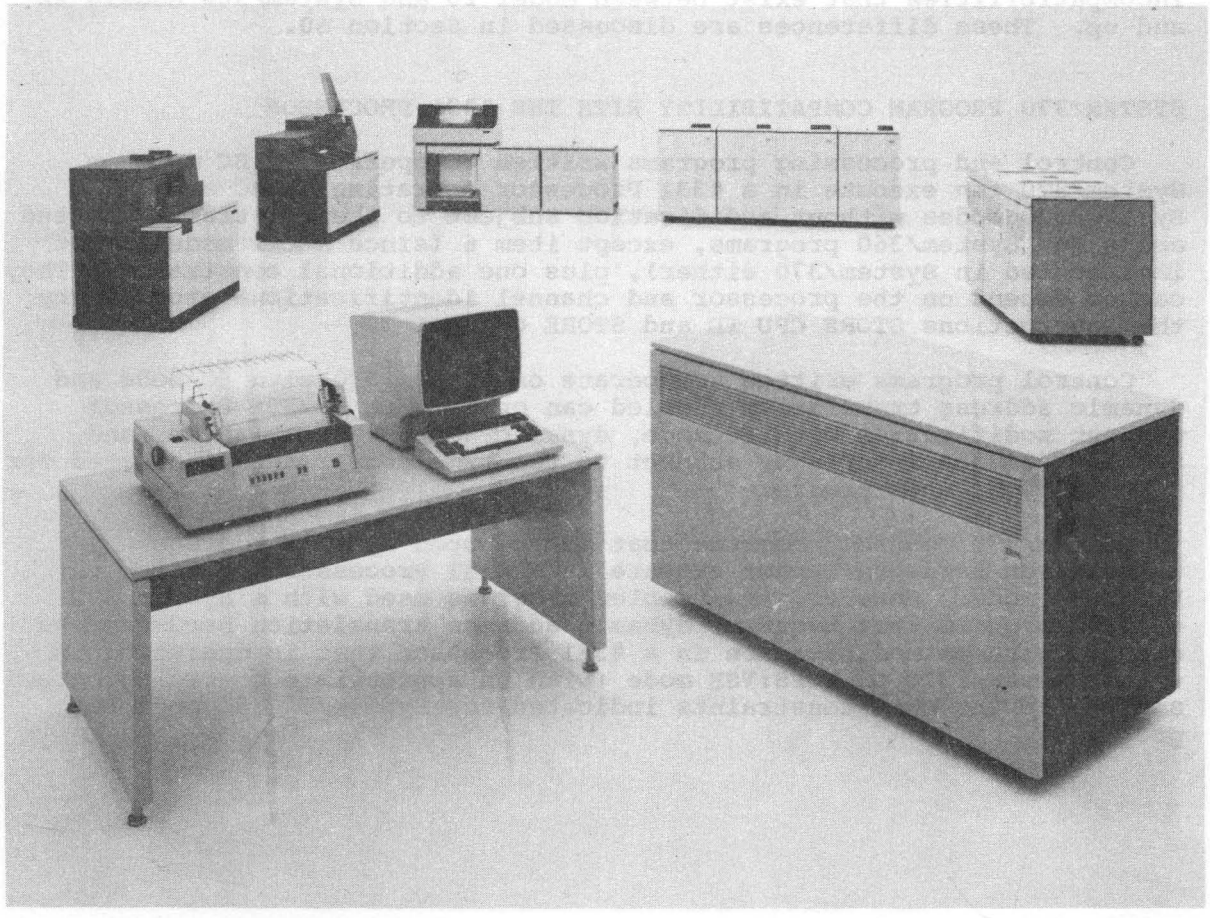


Figure 05.15.1. The 4331 Processor (design model shown right to left with 8809 Magnetic Tape Units, 3310 Direct Access Storage, the 3203 Model 5 Printer, the 3505 Card Reader, the 3525 Card Punch, the 3287 Printer, and the 3278 Model 2A Display Console)

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 4331 Processor. I/O instructions are partially processed by the instruction processing function and partially processed by the appropriate channel or I/O adapter.

All instruction execution functions and most channel operations are microcode controlled. Microinstructions are four bytes in length. Control storage for the residence of instruction processing function microcode is 64K or 128K bytes. The data path within the instruction processing function is four bytes wide. Extensive parity checking is done within the instruction processing function to ensure data validity.

Certain basic control and service functions are provided for the 4331 Processor by the support processor subsystem, instead of by the instruction processing function. The support processor subsystem is controlled by a microcoded controller (support processor 1) that contains its own control storage (see discussions in Sections 10:15 and 50:15).

The instruction processing function in the 4331 Processor has a variable-length cycle time. Cycle time varies from 200 nanoseconds to 1600 nanoseconds in 100-nanosecond increments, depending on the instruction.

Elements included in the instruction processing function to perform instruction execution are a four-byte-wide arithmetic logic unit, a four-byte-wide shifter, fullword working registers, and semiautomatic hardware facilities designed to speed up instruction execution. An instruction buffer and control storage buffer are also present.

Instructions are fetched from processor storage and placed in a 16-byte instruction buffer from which they are fetched for execution. The use of such a buffer avoids most delays that could be caused by the refreshing of processor storage. Instruction buffer loading requires 2.4 microseconds.

Instructions are fetched for execution from the 16-byte instruction buffer at a rate of 200 nanoseconds per halfword. Instructions are fetched from the buffer as long as the next required instruction is contained in the buffer. If the required instruction is not currently in the buffer or only partially contained in the buffer, the entire buffer is reloaded, beginning with the required instruction.

The instruction buffer is also reloaded for each successful branch, whether or not the instruction required is present in the instruction buffer. This is done to properly handle self-modifying code. There is no overlap of instruction buffer filling and instruction execution (that is, no instruction prefetching).

The microcode that controls instruction processing function operations is partially resident in reloadable control storage and partially resident in processor storage (see discussion of processor and control storage in Section 10:10). To avoid most delays caused by the

refreshing of control storage (and processor storage), a 64-byte control storage buffer is present in the instruction processing function. This buffer can contain 16 four-byte microinstructions at a time.

Any microinstruction in the control storage buffer can be accessed directly. The fetching of one microinstruction and its placement in an operation register for execution requires 100 nanoseconds. When the next required microinstruction is not present in the control storage buffer, the buffer is refilled with the 64-byte block of microinstructions that contains the needed microinstruction. The 16 microinstructions are fetched from control storage or processor storage, as required.

A buffer fill from control storage requires 300 nanoseconds while 2.6 microseconds are required to fill the control storage buffer from processor storage. Instruction execution waits until the control storage buffer is refilled.

The instruction processing function accesses a 2K-byte data local storage area as required during the execution of instructions. This data local storage contains the control registers, general registers, floating point registers, all the subchannels used by the byte multiplexer channel and Communications Adapter, all unit control words (UCWs) used by the channels and I/O adapters, work registers, and various work areas. The work areas are used by certain I/O adapters and emulator routines.

The address translation facilities provided for System/370 and ECPS:VSE modes are discussed in Section 15. The ECPS:VM/370 feature is discussed in Section 18, which describes a virtual machine environment. Other significant new features of the instruction processing function of the 4331 Processor for Model 30 and 40 users are discussed in the remainder of this subsection. A complete discussion of hardware and I/O differences between the 4331 Processor and the Model 20 is contained in Section 60.

CONTROL REGISTERS

The program states in which the 4331 Processor is operating are reflected in the current program status word (PSW) and in processor status indicators called control registers, which are contained in data local storage in the 4331 Processor. 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 4331 Processor is considerably altered from its implementation in Models 30 and 40 (see Section 50). However, the other four interruption classes (I/O, SVC, program, and external) operate in the same manner on Models 30 and 40, and the 4331 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 4331 Processor. Imprecise interruptions do not occur in the 4331 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 4331 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 4331 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.

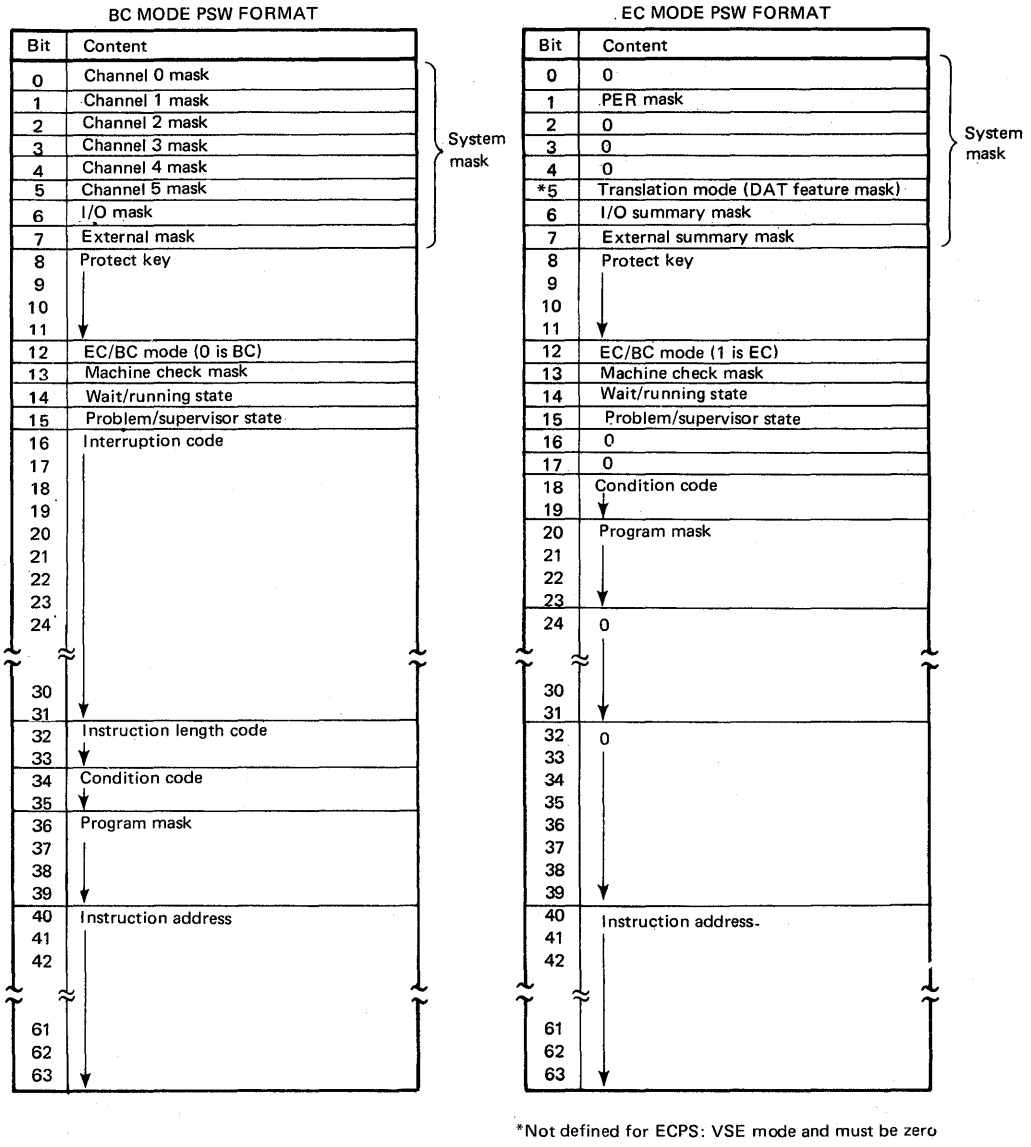


Figure 10.05.1. BC and EC mode PSW formats

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.

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 and disables the dynamic address translation facility for System/370 mode, must be zero for ECPS:VSE mode.

Change in Permanently Assigned Processor Storage Locations

When a 4331 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.

Channel Masking Changes

When a 4331 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 4331 Processor, bits 0 to 3 in control register 2 are assigned to control channels 0 to 3, 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 3 are controlled only by the channel mask bits (bits 0 to 3) in the current PSW.

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.

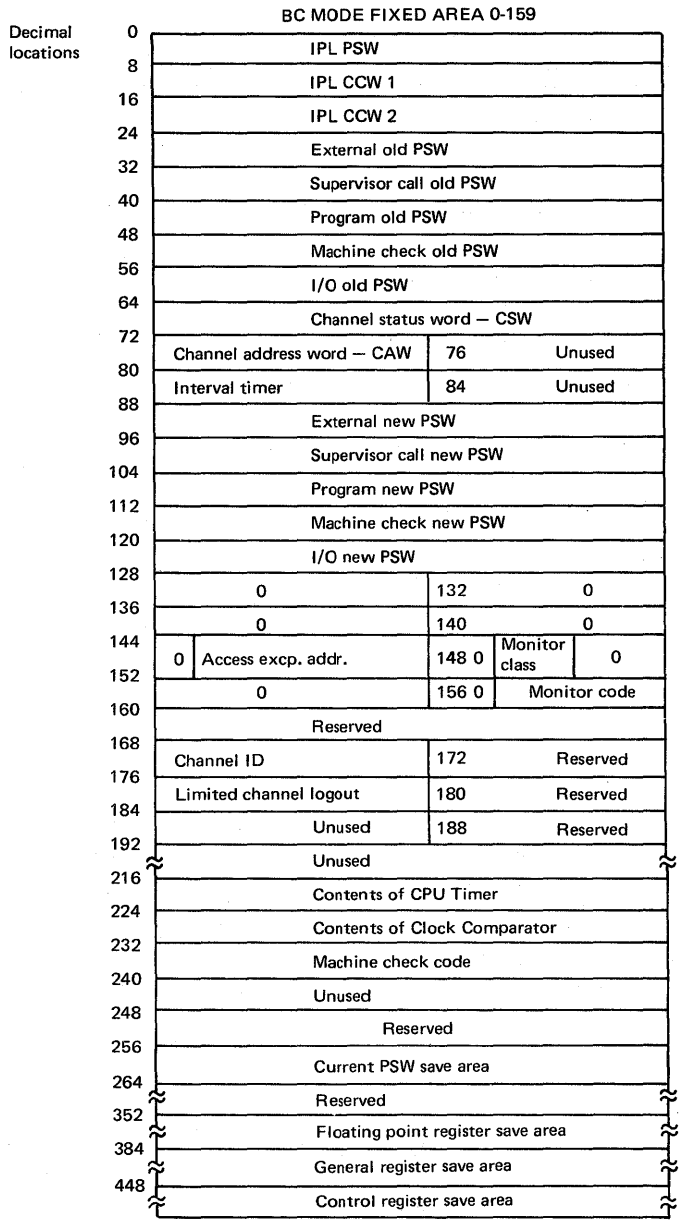


Figure 10.05.2. BC mode fixed processor storage locations 0 to 511

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.

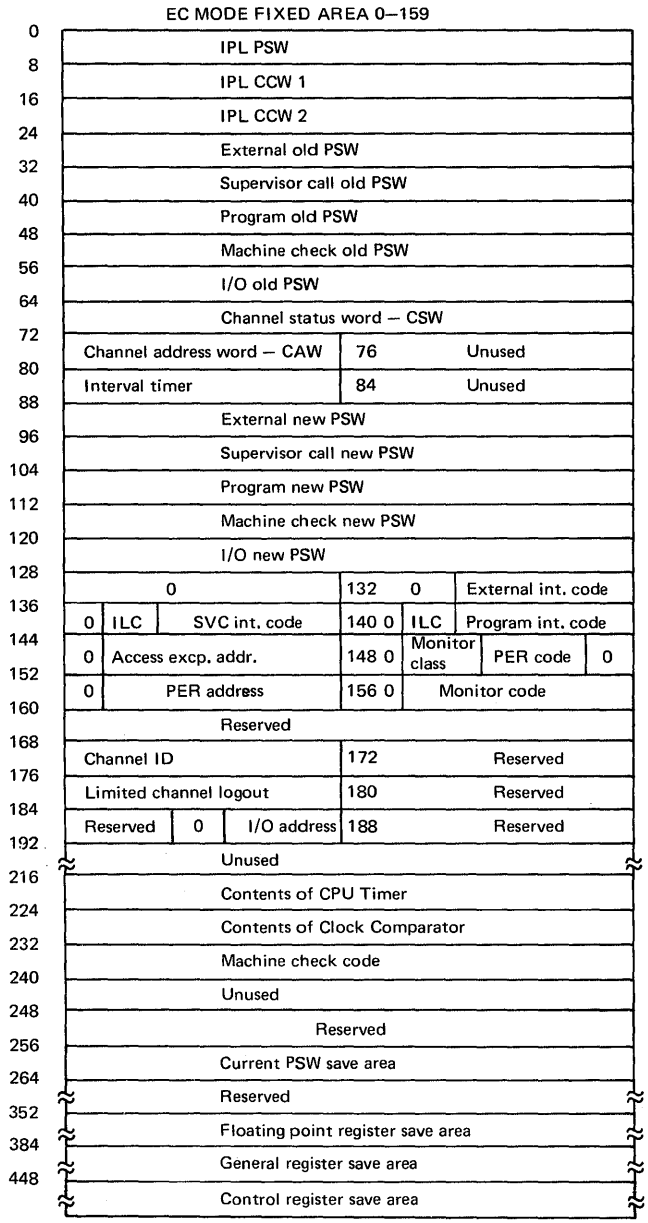


Figure 10.05.3. EC mode fixed processor storage locations 0 to 511

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	

Program Event Recording

Program event recording (PER), a standard feature for the 4331 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 4331 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 4331 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 4331 Processor instructions (no instructions are optional).

The standard instruction set for the 4331 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, 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 the 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 4331 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 4331 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).

Use of byte alignment in a program degrades instruction execution performance. However, 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.

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 on both the 4331 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 4331 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 4331 Processor during the execution of certain instructions common to both System/360 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 4331 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 4331 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 4331 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 10 milliseconds instead of the 16.6-ms resolution implemented for the interval timer provided for Models 30 and 40. 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 that are 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 16 microseconds. 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. In the 4331 Processor, these instructions operate only on bit positions 0 to 47 of the clock. Hence, the read-out resolution of the clock is 16 microseconds. 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.

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 typical switchover situations, such as midnight, New Year's eve, etc.

CLOCK COMPARATOR AND CPU TIMER

These timing facilities are a standard feature of the 4331 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. Bits 0 to 47 of the time-of-day clock and the clock comparator are compared in the 4331 Processor. 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 having 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 16 microseconds. 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 10 milliseconds can be more accurately measured because of the 16-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. This can result from the fact that 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

PROCESSOR STORAGE

The 4331 Processor is available with 512K bytes (Model I1) or 1024K bytes (Model J1) of processor storage. A Model I1 is field upgradable to a Model J1. Access to processor storage is controlled by the instruction processing function. A four-byte register is used to transfer data aligned on a fullword boundary between processor storage and the instruction processing function.

Processor storage has a read cycle time of .9 microseconds for a fullword and a write cycle time of 1.3 microseconds for a fullword when the write is followed by a read. The cycle time for successive fullword writes is 1 microsecond. A fullword can be fetched from processor storage and made available to the instruction processing function in .7 microseconds.

The cycle time for a fetch of a fullword followed by a store of the same fullword is 2 microseconds. This operation is used, for example, when an input I/O operation begins or ends on a byte that is not located on a fullword boundary.

In addition to fetch, store, and fetch/store operations involving a fullword, the instruction processing function can fetch 16 consecutive bytes (four fullwords). The cycle time for this operation is 1.6 microseconds. This operation is used to load the 16-byte instruction buffer with instructions from processor storage and to load the 64-byte control storage buffer with microinstructions contained in processor storage.

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

The processor storage actually present in a 4331 Processor is divided into sections for programming use and sections that are reserved for use by the instruction processing function. The sections reserved for use by the instruction processing function cannot be addressed by programs and reduce the amount of processor storage available for program execution. The minimum amount of processor storage that will be unavailable for program use is 16K bytes (assumes no optional microcoded features, the Control Storage Expansion feature is installed, 231X disk

only, and System/370 mode or one megabyte of virtual storage defined if ECPS:VSE mode is used). A minimum of 52K is required for processor use if Control Storage Expansion is not installed.

The layout of processor and program sections within processor storage is shown in Figure 10.10.1. Processor and program sections alternate in location and their sizes vary, depending on the optional features installed and the mode in effect, System/370 or ECPS:VSE. Each section is a multiple of 2K bytes. The program sections, although not physically contiguous, are considered to contain consecutively addressed locations, beginning with address 0.

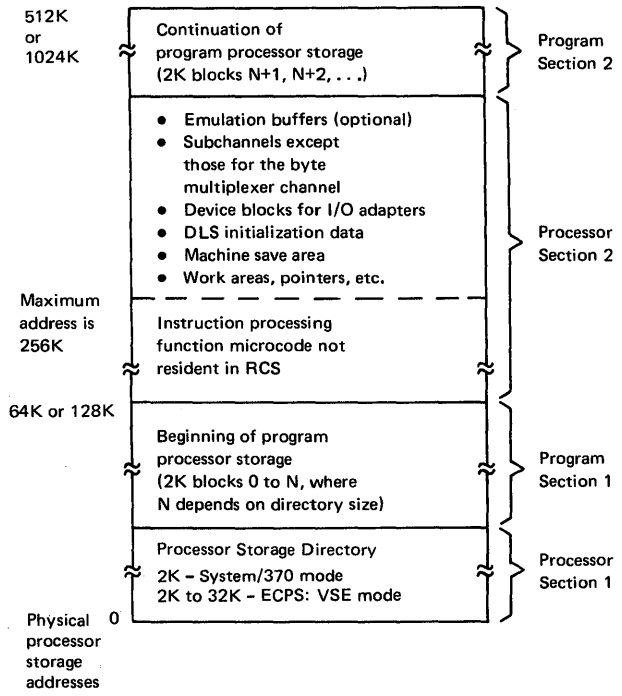


Figure 10.10.1. Processor and program sections of processor storage

The processor and program sections and their contents in ascending processor storage sequence are as follows:

- Processor section 1, which begins at physical processor storage location 0 and contains the processor storage directory. The entries in this directory point to 2K blocks of processor storage within program sections only. When ECPS:VSE mode is in effect, the processor storage directory is used for the translation of virtual storage addresses to processor storage addresses.

  When System/370 mode is in effect and dynamic address translation is enabled, the processor storage directory is referenced after address translation has been performed using the dynamic address translation facility in order to determine the location of the translated address in program processor storage. When System/370 mode is in effect without dynamic address translation enabled, the real addresses in instructions and CCWs are converted to processor storage addresses during instruction execution and channel operation using the directory.

  The processor storage directory must be referenced in System/370 mode (for both BC and EC mode operations) because program processor

storage is not contiguous and does not begin at physical location 0 in the installed processor storage. Details about the use of the processor storage directory for both System/370 and ECPS:VSE modes are given in Section 15.

For ECPS:VSE mode, the processor storage directory contains one four-byte entry for each 2K bytes of virtual storage defined. The size of the directory normally will be 2K, 4K, 8K, 16K, or 32K bytes for virtual storage sizes of 1, 2, 4, 8, or 16 megabytes, respectively. For System/370 mode, the processor storage directory contains one four-byte entry for each 2K bytes of processor storage installed. Thus, the directory is 2K bytes for a processor storage size of 1024K. The directory is also allocated 2K bytes for a 512K processor storage size, since each section of processor storage must be a multiple of 2K bytes in size.

While a program cannot directly address the processor storage directory, certain instructions cause the instruction processing function to set or modify appropriate entries within the directory (see Section 15:15).

- Program section 1, which begins at the next 2K block of processor storage after the processor storage directory. This section is the beginning of program processor storage. Its first 2K block is assumed to have program processor storage addresses 0 to 2047, the next block has addresses 2048 to 4095, etc. Thus, for System/370 mode, for example, the first entry in the processor storage directory (which is for the first 2K bytes of program processor storage) contains the processor storage address of the first 2K block in program section 1. The size of program section 1 is equal to 64K or 128K (the size of the reloadable control storage installed) less the size of the processor storage directory.

- Processor section 2, which begins at processor storage location 64K or 128K, depending on the size of control storage. This section contains an area for instruction processing function microcode residence, an area that contains work areas and data used by the instruction processing function and I/O adapters, and, optionally, emulation buffers allocated by the operator for 231X/3340 emulation functions.

  The size of the area for microcode residence depends on the optional features installed (see Table 10.10.2 later in this subsection for optional feature microcode requirements). The next area contains all subchannels except those for the byte multiplexer channel and Communications Adapter, data to initialize the data local storage area when a processor reset occurs after IML, various work areas (machine save area, for example), device blocks used by certain I/O adapters, and various pointers.

  Emulation buffers must be defined during IPL when 2311 or 2314/2319 devices are to be emulated on 3310 disk drives or 3340 direct access devices are attached to the 4331 Processor via the DASD Adapter. A maximum of 16 emulation buffers can be allocated by the operator (up to 8 for 231X emulation on 3310 devices and up to 8 for 3340 devices). See Table 10.10.2 for emulation buffer sizes.

- Program section 2, which consists of all the processor storage above processor section 2. Processor section 2 is considered to contain the next consecutive program processor storage address after the last address in program section 1 to form a consecutively addressed processor storage area for program use.

CONTROL STORAGE

Unlike System/360 Models 30 and 40, the 4331 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.

Functional capability is extended by the ability to more easily support different architectures and features in one system. IBM-supplied 1401/1440/1460 emulator, System/370 or 4300 Processor architecture, and optional feature microcode is quickly and easily loaded.

The 4331 Processor contains 64K of RCS for residence of the microcode for the instruction processing function. If required by the optional features selected, an additional 64K of RCS must be installed (Control Storage Expansion feature). Each standard and optional support processor contains its own reloadable control storage.

Microcode for the most frequently used standard functions of the instruction processing function are located in RCS while infrequently used standard function microcode routines (such as those for handling exception conditions) are located in processor storage in the second processor section. For most optional features, the required microcode is partially resident in RCS (the most frequently used portion) and partially resident in processor storage. For a few features, all the required microcode is totally resident either in RCS or processor storage.

The instruction processing function has the capability of addressing a 256K-byte area that contains microinstructions. The first 64K-byte or 128K-byte area for microinstructions is the reloadable control storage area. The second 192K-byte or 128K-byte area is in lower processor storage (in processor section 2) in locations 64K to 256K or 128K to 256K, respectively.

During microinstruction execution, the instruction processing function determines whether the next microinstruction to be executed is contained in control storage or processor storage by inspecting the high-order bit(s) of the microinstruction address. If the bit(s) indicate the address of the microinstruction is lower than the installed control storage size (64K or 128K), the microinstruction is in control storage. If the bit(s) indicate the address is higher than the installed control storage size, the microinstruction is in processor storage.

For configuration purposes, the microcode is organized into functional microcode groups and assigned a group number. Table 10.10.1 lists the standard and optional features of the 4331 Processor that require microcode and indicates the numbers of the microcode groups that are required to support the feature. Table 10.10.2 indicates microcode storage requirements for each group. It indicates the portion that must be resident in control storage (Column A), the portion that must be resident in processor storage (Column C) and the portion that can be resident in either control or processor storage (Column B).

When the microcode that must reside in control storage for a given 4331 Processor configuration (the total requirement from Column A for the needed groups) is less than 65,536 (64K) bytes, the Control Storage Expansion feature is not required. When the Column A requirement is more than 65,536 bytes, the Control Storage Expansion feature is required. Even when not required, the Control Storage Expansion feature can be installed to increase the amount of processor storage available for program use. The amount of the increase depends on the features installed.

Table 10.10.1. Functional microcode group requirements for standard and optional 4331 Processor features

Function or Feature Installed	Microcode Group
4331 Processor standard features	1
System/370 mode or virtual storage size for ECPS:VSE mode	2
3310 disk drives attached	3,4,5,6
3370 disk drives attached	3,4,6,17
8809 Magnetic Tape Units attached	3,4,6,7
3340 Direct Attach	3,6,8,15
System/3 Data Import	3,6,8,15
2311/2314/2319/3310 Direct Access Storage Compatibility	3,4,5,6,9,15
Communications Adapter Base	6,10
Binary synchronous lines attached	6,10,11
Start/stop lines attached	6,10,12
Synchronous data link control lines attached	6,10,13
1401/1440/1460 Compatibility	14
ECPS:VM/370	16

When the Column A requirement for a given 4331 Processor configuration is less than the required control storage size (65,536 or 131,072 bytes), all the available control storage is loaded with microcode that can reside in either control or processor storage (the requirement from Column B). The portion that does not fit in control storage is placed in processor storage along with the microcode that can reside only in processor storage (total requirement from Column C).

Table 10.10.2. Reloadable control and processor storage requirements for the 4331 Processor by functional microcode group number

Microcode Group Number	(A) Bytes Resident in Control Storage Only	(B) Bytes Resident in Control or Processor Storage	(C) Bytes Resident in Processor Storage Only	Notes
1	33,792	66,816	12,764	Includes requirement for microcode and that for processor data and work areas in processor section 2.
2	-	-	2048	Group is required once for System/370 mode operations. For ECPS:VSE mode, group is required once for each one megabyte of virtual storage defined. This is the processor storage directory requirement.
3	6144	24,320	3150	
4	5120	12,288	-	
5	-	4608	11,250	
6	-	-	10,250	
7	6144	9728	3060	Mutually exclusive with group 14 (1401/1440/1460 Compatibility)
8	9216	13,312	11,600 plus 8800 per 3340 emulation buffer defined (1 to 8) plus 1800 if two 3340 strings are attached to the DASD Adapter	
9	-	26,624	1300 plus 231X emulation buffers (1 to 8): 4096 for 2311 or 7680 for 2314/2319	

Table 10.10.2 (continued)

Microcode Group Number	(A) Bytes Resident in Control Storage Only	(B) Bytes Resident in Control or Processor Storage	(C) Bytes Resident in Processor Storage Only	Notes
10	8192	9216	2150	
11	6144	-	-	For groups 11,12, and 13, only two out of three can be selected.
12	5120	-	-	
13	12,288	-	1024	
14	14,336	-	1800	Mutually exclusive with group 7 (8809 adapter) and group 16 (ECPS:VM/370)
15	-	5760	200	
16	6656	-	-	Mutually exclusive with group 14 (1400 compatibility)
17	-	9316	10,000	

The amount of processor storage that will be required for processor use for a given 4331 Processor configuration is calculated using the following procedure:

1. Determine the microcode group numbers required to support the mode to be used (System/370 or ECPS:VSE) and the optional features installed using Table 10.10.1. Note that with the exception of group 2, each group is required only once even though the group is listed for more than one feature. The group 2 requirement can be included more than once for ECPS:VSE mode, as indicated in Table 10.10.2.

2. Using Table 10.10.2, determine the microcode needed for each required group from each of the three columns and add the requirements for each column.

3. The total for Column A determines the total amount of control storage required for the configuration. If this total exceeds 65,536 (64K) bytes, the Control Storage Expansion feature must be installed. If this total exceeds 131,072 (128K) bytes, an invalid feature combination has been configured. An invalid configuration also exists when the sum of the totals for Columns A and B exceeds 262,144 (256K) bytes (the maximum amount of microcode that can be addressed).

4. Subtract the total for Column A from 65,536 (if Control Storage Expansion is not required) or 131,072 (Control Storage Expansion is required) and then subtract the result from the total for Column B (to determine the microcode that will not fit in control storage and that must be made resident in processor storage).

5. Add the results of step 4 to the total for Column C and round up to the next multiple of 4096. This figure is the amount of processor storage that will be unavailable for programming use (that is, the amount of storage contained in processor sections 1 and 2).

A sample calculation of the processor storage unavailable for program use for a specific 4331 Processor configuration is given in Figure 10.10.2.

4331 Processor Configuration	Groups Required	Microcode Requirements		
		A	B	C
4331 Processor	1	33,792	66,816	12,764
System/370 mode	2	-	-	2,048
3310 drives	3	6,144	24,320	3,150
attached	4	5,120	12,288	-
	5	-	4,608	11,250
	6	-	-	10,250
2311/2314/2319/3310	9	-	26,624	1,300
Direct Access Storage	15	-	5,760	200
Compatibility (2314 - two buffers)				15,360
1401/1440/1460 Compatibility	14	14,336	-	1,800
Totals		59,392	140,416	58,122

Step 4 65,536 - 59,392 = 6,144 140,416 - 6,144 = 134,272

Step 5 134,272 + 58,122 = 192,394 = 192,512 = 188K bytes of processor storage required

Figure 10.10.2. Calculation of processor storage unavailable for program use for a sample 4331 Processor configuration

10:15 THE SUPPORT PROCESSOR SUBSYSTEM

COMPONENTS AND FUNCTIONS

The support processor subsystem provides basic operational functions for the 4331 Processor and is the primary maintenance tool for diagnosing hardware malfunctions. It is designed to maximize system availability and to provide rapid fault location and repair, where possible.

The components of the support processor subsystem are the support processor 1, system diskette drive, Display/Printer Adapter and attached devices, support bus adapter and support bus, power control interface, and common communications adapter.

Support processor 1 is a microcoded controller that controls the operation of the support processor subsystem, which is responsible for the following:

- System initialization (IML and IPL) functions including microcode loading for itself, the instruction processing function, and support processor 2, if it is present

- Control of the system diskette drive and the optional diskette drive, if it is installed
- Control of the I/O devices (up to 16 maximum with an optional feature) that attach to the 4331 Processor via the Display/Printer Adapter. These devices include the required operator console, 3278 displays, and certain 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
- Diagnostic program loading
- Power sequencing and power monitoring via microcode to detect under- and over-voltage conditions (utilizing the power control interface)
- Control of the Remote Support Facility (utilizing the common commmunications adapter)

Operation of the support processor is independent of, and overlapped with, operation of the instruction processing function for certain of its functions. While instruction execution occurs during system operation, support processor 1 interfaces with the integrated channel on behalf of the I/O devices attached to the Display/Printer Adapter. The support processor first resolves priority among the outstanding I/O requests from devices it handles and then contends with the channels and I/O adapters for access to processor storage via the integrated channel bus.

During system operation, while instruction execution occurs, support processor 1 also performs power monitoring and the writing of power logout data to the system diskette when necessary under microcode control.

Whenever a machine check or channel check condition occurs, the support processor receives control to perform error diagnosis and logging before a machine check interruption occurs. The instruction execution function does not process instructions during error diagnosis and logging.

Details about the last four functions listed for the support processor subsystem are covered in Section 50. The Display/Printer Adapter is discussed in Section 10:20. The other functions of the support processor subsystem are discussed in the remainder of this subsection.

SYSTEM INITIALIZATION

When power is turned on via the power-on switch on the 4331 Processor, power is turned on first in support processor 1 and it is initialized. A bootstrap routine resident in the support processor performs basic assurance tests on support processor 1 and its control storage. In addition, the path to the system diskette drive is checked and the first record is read.

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 power-on sequence for the balance of the 4331 Processor is initiated. If no errors occur, the 4331 Processor is then IMLed. The microcode for the instruction processing function is loaded into reloadable control storage and processor storage, microcode for support processor 2 is loaded (if it is

present), and data local storage is initialized as required based on the customized microcode.

The Power Interface feature provides the required power, power control, and instantaneous power-off interfaces to the 4331 Processor for eight control units attached to the byte and block multiplexer channels. The Power Interface, Additional feature provides the interfaces for eight additional control units. An interface feature is required for all control units that attach to the byte or block multiplexer channel of the 4331 Processor except the 3250 Graphics Display System.

I/O devices attached to the byte and block multiplexer channels with their power control switch set to the remote position are also powered on when the power-on switch on the 4331 Processor is turned on and powered off when the power-off pushbutton on the operator's control panel of the 3278 Model 2A console is pressed.

THE SYSTEM DISKETTE DRIVE

Control storage is loaded directly from a small read/write diskette drive, called the system diskette drive, which is a basic component of the 4331 Processor. The system diskette drive is located in the right-hand end of the 4331 Processor (as shown in Figure 05.15.1). It reads removable prerecorded disk cartridges (diskettes). Recording is done on both sides of the system diskette.

A power on of the 4331 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 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 program can execute to cause read or write operations to the system diskette drive.

Three diskettes are sent to each 4331 Processor installation. Two are identical system diskettes (one for backup) and the other is the service diskette. The system diskette contains all the microcode required for the configuration, areas for logout data from the 4331 Processor, and certain logout analysis programs (see additional discussion of the system diskette in Section 50).

The microcode on a system diskette is customized based on the optional features specified by the installation for the specific 4331 Processor configuration and contains only the features requested. The service diskette (which is not customized) contains additional service programs that are to be used by customer engineers. Storage space for diskettes is provided within the frames of the 4331 Processor.

When the system diskette is mounted on the system diskette drive, an IML occurs automatically after a power on of the 4331 Processor is performed. Microcode for support processor 1, support processor 2 if it is present, and the instruction processing function is loaded. If an IML is required thereafter, it can be performed using the IML pushbutton on the operator's control panel.

A hash total is taken during loading to ensure correct loading of the microcode. Parity checking is used for reloadable control storage during processor operation. In addition, while each microinstruction contains a parity bit, the instruction decoder also contains a validity check bit to determine whether the decoder misses a bit during instruction decoding.

A procedure exists that enables the customer engineer to temporarily patch the microcode in reloadable control storage or a 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 both processor and control storage is lost, so an IML is performed automatically when power is turned on again. Normally, the system diskette will stay mounted on the system diskette drive and diskette changing will occur only when diagnostics are to be performed.

A customized system diskette for a given 4331 Processor contains the processor serial number and is not portable from one 4331 Processor to another (since the serial number on the diskette is checked against the processor serial number during IML and a mismatch causes termination of the IML procedure). In addition, a given system diskette contains configuration data (such as I/O device addresses, UCW assignments, etc.) that is specific to its associated 4331 Processor.

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 4331 Processor installation on the service diskette. These routines can be loaded directly from the system diskette drive into the 4331 Processor and executed (see Section 50:15).

SUPPORT BUS ADAPTER AND SUPPORT BUS

The support bus adapter and support bus provide a direct path between support processor 1 and the instruction processing function for maintenance functions. They enable the support processor to obtain logout data from the instruction processing function when a machine check occurs and enable the customer engineer to access maintenance logic contained in the components of the 4331 Processor during maintenance operations (see discussion in Section 50:10).

When the optional 5424 Adapter is installed, support processor 2 is present. Then the support bus adapter also provides a communication path between support processors 1 and 2. This path is used to load microcode in support processor 2 and to provide support processor 1 with logout data related to support processor 2 and the 5424 MFCU and its adapter.

10:20 CHANNELS AND I/O ADAPTERS

GENERAL DESCRIPTION

While channel functions compatible with those available on Models 30 and 40 are provided, the 4331 Processor also offers additional facilities (such as block multiplexing), integrated I/O adapters, and faster burst mode channel data rates. In addition, faster I/O devices and a significantly larger amount of lower-cost direct access storage can be attached to the 4331 Processor. These capabilities enable the user to tailor a 4331 Processor configuration to I/O processing needs, on an improved price performance basis, to increase channel throughput.

I/O devices can be attached to the 4331 Processor via a channel and control unit or directly attached via an I/O adapter that functionally replaces a channel and control unit. The latter approach provides lower-cost attachment of I/O devices and reduces space requirements.

Conceptually, all I/O devices are channel attached and up to four channels (addressed 0 to 3) can be configured for a 4331 Processor.

The Display/Printer Adapter is standard. Both channels and all other I/O adapters for the 4331 Processor are optional. The following channels and I/O adapters are available for the 4331 Processor (all can be installed in one 4331 Processor):

- One byte multiplexer channel
- One block multiplexer channel
- One Display/Printer Adapter (and optional Display/Printer Adapter Expansion) for the operator console, certain display devices, and certain printers
- One DASD Adapter for the direct attachment of certain disk drives
- One 8809 Magnetic Tape Unit Adapter for the direct attachment of 8809 tape drives
- One 5424 Adapter for the direct attachment of one 5424 Multi-Function Card Unit
- One diskette drive and included adapter
- One Communications Adapter for the direct attachment of up to eight communications lines

Note that for certain I/O adapters, the processor is inoperative if the adapter is installed without any I/O devices attached. For the DASD Adapter, 5424 Adapter, and 8809 Magnetic Tape Unit Adapter, the adapter and attached devices must be installed/removed together. The Communications Adapter and Display/Printer Adapter Expansion can be installed without communications lines and devices attached, respectively.

The byte multiplexer channel for the 4331 Processor, like that for System/360 Models 30 and 40, can handle the concurrent operation of multiple slower speed devices when operating in byte interleave mode. The block multiplexer channel, not available for Models 30 and 40, is 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. The 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. The DASD Adapter functions like a block multiplexer channel and permits the concurrent execution of multiple disk channel programs.

Channels and I/O adapters for the 4331 Processor are integrated, as they are on Models 30 and 40. That is, they share with the instruction processing function the use of reloadable control storage, the instruction processing function and processor storage data flow, data local storage, and the instruction processing function arithmetic logic unit. When a channel/adapter and the instruction processing function simultaneously need to access processor storage, the channel/adapter is given priority over the instruction processing function.

Comprehensive error checking is incorporated in the basic design of the channel/adapter hardware. Checking is performed on the control logic in most areas, and standard parity checking is done on the data

flow between the integrated channel bus and channels/adapters. Improved error recovery data is provided by the channel/adapters (discussed fully in Section 50).

The standard instruction set for the 4331 Processor includes two I/O instructions not provided for System/360: HALT DEVICE and CLEAR I/O. 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 should always be used, instead of HALT I/O, 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 that could cause processor termination occur.

A Channel-to-Channel Adapter cannot be installed in a 4331 Processor nor can a channel in the 4331 Processor be attached to a Channel-to-Channel Adapter that is installed in another processor.

Integrated Channel

Logically, all I/O operations in a 4331 Processor configuration are controlled by a single integrated channel. The elements of the integrated channel are the instruction processing function, data mover, integrated channel bus, byte and block multiplexer channels, and I/O adapters. The basic data flow between processor storage and the channels and I/O adapters in the 4331 Processor is shown in Figure 10.20.1.

When required, the instruction processing function is dedicated to servicing an I/O operation for a short period. The data mover provides the hardware paths, address registers, and buffers required for multiplexing data between the integrated channel bus and processor storage. The integrated channel bus connects the installed channels and I/O adapters to the data mover via a two-byte-wide data/address path and provides control functions.

The channels and I/O adapters adapt the integrated channel bus to a standard or special interface for the connection of control units or I/O devices, respectively. The channels and all I/O adapters provide the same interface to the integrated channel bus while providing a standard or special interface for I/O device attachment.

A cycle steal operation is implemented to transfer (via hardware control) data between channels and I/O adapters and processor storage via the integrated channel bus. The channels and I/O adapters can also transfer (via microcode control) two bytes at a time for sense/control operations. This two-byte sense/control transfer requires 400 nanoseconds.

A cycle steal is initiated by microcode and executed under hardware control. A cycle steal consists of the transfer of a fullword, in two successive halfword transfer operations, to or from processor storage over the integrated channel bus. Each halfword transfer requires 400 nanoseconds to perform. The instruction processing function does not process instructions during a cycle steal operation but it updates the data address and the byte count for the I/O during the data transfer.

Each channel and integrated I/O adapter except the DASD Adapter and 8809 Magnetic Tape Unit Adapter attaches to the integrated channel bus via a common type of subadapter. The common subadapter performs functions such as data collection for cycle steal operations, execution of sense and control operations, and the handling of appropriate request

and response signals for its attached channel/adapter. A common subadapter receives data from a channel or I/O adapter one byte at a time and collects the bytes in two sets of four one-byte buffers.

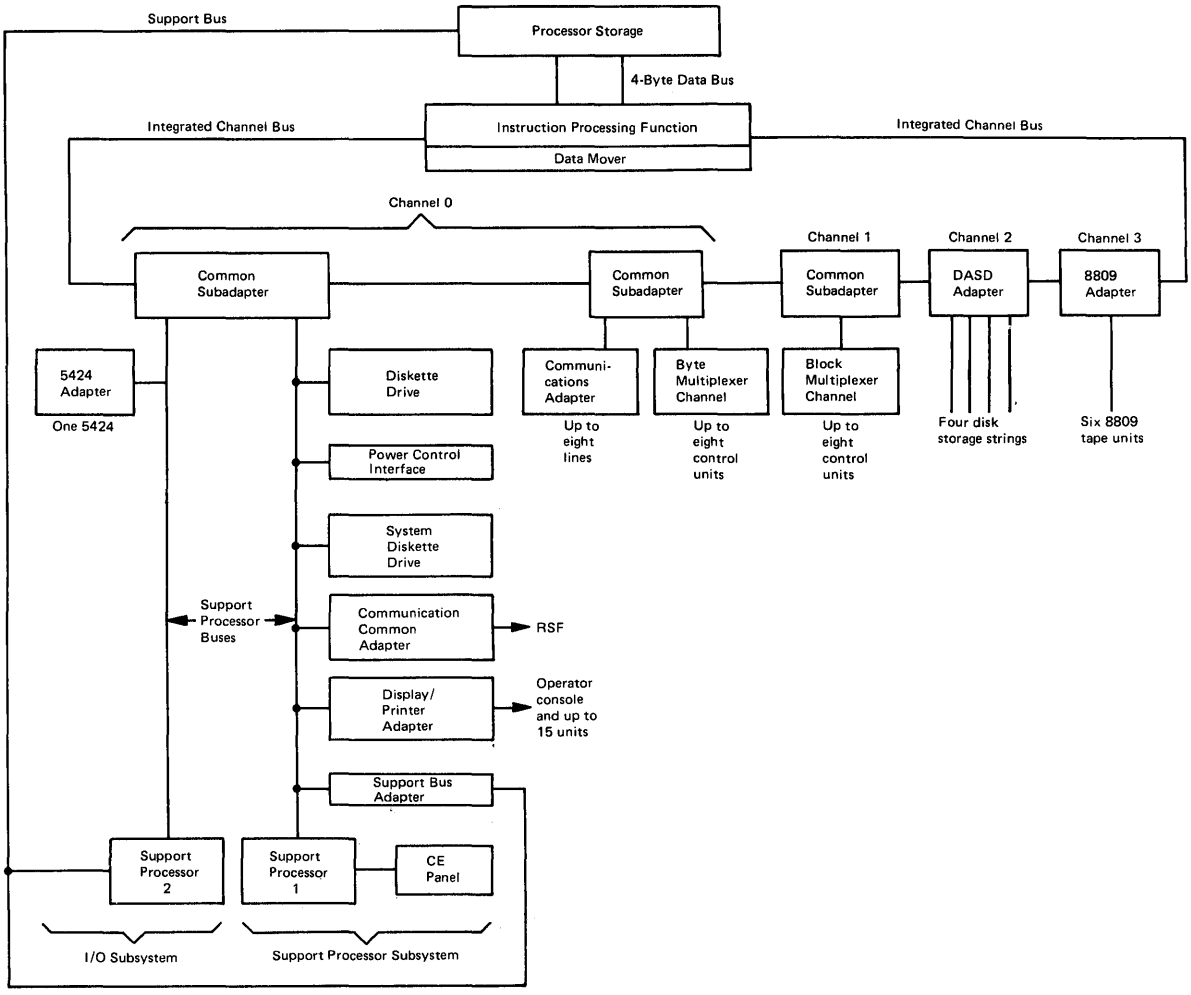


Figure 10.20.1. The basic data flow between processor storage and the channels and I/O adapters in the 4331 Processor

Data is transferred between the common subadapter buffers and the integrated channel bus on a word basis. Two sets of buffers are provided for each common subadapter so that data can be transferred to the integrated channel bus for placement in processor storage from one set of buffers while the other set is collecting data from the associated channel or I/O adapter one byte at a time. Similarly, data from processor storage can be received by the integrated channel bus and placed in one set of buffers for a common subadapter while data contained in the other set of buffers is being sent to its associated channel or I/O adapter one byte at a time.

The DASD Adapter and 8809 Magnetic Tape Unit Adapter attach directly to the integrated channel bus. They contain their own buffers and a small control storage (see discussions under "DASD Adapter" and "8809 Magnetic Tape Unit Adapter" later in this subsection.)

Once an I/O operation has been started on a channel or I/O adapter, it contends with other operating channels and I/O adapters for cycle

steal requests. The high-to-low priority for handling simultaneous cycle steal requests is the following:

- Block multiplexer channel
- Byte multiplexer channel
- DASD Adapter
- 8809 Magnetic Tape Unit Adapter
- Support processor subsystem and I/O subsystem (5424 Adapter). Priority is allocated between the two subsystems on an alternating basis. The I/O scheduling microcode for devices attached to the Display/Printer Adapter dynamically determines the cycle steal priority among the devices attached to this adapter.

The high-to-low priority for handling simultaneous I/O interruptions from the channels is: channel 0, channel 1, channel 2, channel 3.

Subchannels and Unit Control Words

I/O operations in the 4331 Processor are controlled using subchannels and unit control words (UCWs) for both the channels and the I/O adapters. A subchannel is a storage area that contains information required to start and stop an I/O operation on a device. It contains the device address, protect key from the channel address word, the channel command word (CCW) for the I/O operation, and status information (subchannel is available, working, or has an I/O interruption pending). The number of subchannels provided for a channel or I/O adapter determines the number of concurrent I/O operations possible for the channel/adapter.

A unit control word is the storage area that contains the information required for an actual data transfer operation for a device via cycle stealing. A UCW contains the data address from the CCW, count from the CCW, and residual count at the end of the data transfer. A channel or I/O adapter may have one UCW per subchannel provided or one or two UCWs that are shared by all the provided subchannels, depending on its type.

Subchannels for the 4331 Processor are located in the reserved processor storage area (processor section 2) and are four words each in size or in data local storage and are eight words each in size. UCWs for the 4331 Processor are always located in data local storage and consist of eight words each. The number and location of the subchannels and UCWs for each channel and I/O adapter type are indicated in the specific discussion of that channel/adapter.

A subchannel can be shared or nonshared. A shared subchannel can be used by a set of devices, one device at a time. A shared subchannel generally is assigned to a control unit or I/O adapter that has multiple devices attached, only one of which can be in operation at a time. A nonshared subchannel is one that is assigned to only one device. A nonshared subchannel is designed for use with a control unit that has only one I/O device attached or that has multiple I/O devices attached and is capable of using the block multiplexing facility.

Channel Configuration and Device Addresses

The channels and I/O adapters available for the 4331 Processor can be configured to provide a maximum of four channels, which are normally addressed 0 through 3. The device address ranges available for each type of channel and I/O adapter are fixed and certain addresses cannot

be assigned when certain adapters are installed. The components of channel 0 are fixed. Channel addresses 1, 2, and 3 are assigned to certain channels and adapters by default, but these assignments can be changed by the customer engineer, who also assigns addresses to disk devices, block multiplexer subchannels, and devices attached to the Display/Printer Adapter.

Channel 0 can consist of the support processor subsystem, byte multiplexer channel, Communications Adapter, and 5424 Adapter. The following I/O addresses can be assigned to the devices attached to channel 0:

09 to 1F	For the support processor subsystem. (Addresses 09 to 1F are available for assignment by the user to devices attached via the Display/Printer Adapter.)
30 to 37	For the communications lines attached via the Communications Adapter. (These addresses are fixed.)
4C	For the 5424 MFCU. (This address is fixed.)
24 to 3D and 3F	27 addresses for the devices assigned nonshared subchannels of the byte multiplexer channel
80 to BF	4 addresses for the devices assigned shared subchannels of the byte multiplexer channel

The number of subchannels available to the byte multiplexer channel is reduced when certain other I/O adapters are installed and certain of the device addresses listed above are not available. The following device addresses and associated subchannels cannot be used, depending on the adapter installed:

30 to 37	When the corresponding communications lines addressed 30 to 37 are attached
3D	When the block multiplexer channel is installed
3C and 38	When the DASD Adapter is installed
3A and 39	When the 8809 Magnetic Tape Unit Adapter is installed

Channel 1 is the default address for the block multiplexer channel. Addresses 20 through 3F can be assigned to the I/O devices attached to nonshared block multiplexer subchannels, while addresses 80 to FF can be utilized by the devices assigned to shared subchannels.

Channel 2 is the default address for the DASD Adapter. Functionally, channel 2 operates as a block multiplexer channel with a storage control function attached that can have certain types of direct access devices connected to it. Addresses 00 to 7F can be utilized for the I/O devices assigned to nonshared subchannels of this adapter, while addresses 80 to FF can be assigned to the 231X direct access devices that are emulated on 3310 drives. Within these ranges, standard addresses are automatically assigned and these addresses can be changed by the customer engineer.

Channel 3 is the default address for the 8809 Magnetic Tape Unit Adapter. The standard addresses range for 8809 tape units is 00 to 05 and nonshared subchannels are provided for these tape units. The standard range of addresses for 8809 tape units can be changed by the customer engineer; however, several rules must be observed when selecting an alternative range of addresses.

DISPLAY/PRINTER ADAPTER

The Display/Printer Adapter is standard to provide for the direct attachment of the required operator console and up to seven additional units. The optional Display/Printer Adapter Expansion feature can be installed to permit direct attachment to the Display/Printer Adapter of eight additional units (total of 15 units plus the operator console).

The following can be directly attached to the 4331 Processor via the Display/Printer Adapter:

- One operator console (required 3278 Model 2A Display Console). An address in the range of 009 to 01F can be assigned to the operator console.

- The 3278 Display Station Model 2 (1920-character screen with 12 lines of 80 characters each). When the Display/Printer Adapter Expansion feature is not installed, up to seven 3278 Model 2 displays can be directly attached to the 4331 Processor to be used as alternate and additional operator console devices (if supported by the operating system being used) and/or workstations accessed via problem programs. When the expansion feature is installed, up to fifteen 3278 Model 2 displays can be directly attached.

  The Display/Printer Adapter supports all the standard functions of a 3278 Model 2 attached to the 3274 Model 1B control unit except cursor select and the device cancel, print, print ident, keyboard numeric lock, and click keys. The following optional features of the 3278 Model 2 are supported by this adapter: audible alarm, security key lock, and switched control unit.

- The 3287 Printer Model 1 (80 characters per second) or Model 2 (120 characters per second) with 132 print positions. When the Display/Printer Adapter Expansion feature is not installed, up to seven 3287 Printers can be directly attached to the 4331 Processor. When the expansion feature is installed, up to fifteen 3287 Printers can be directly attached.

- The 3289 Line Printer Model 4 with 132 print positions and a 48-, 64-, 94-, 116-, or 127-character set. A maximum of two can be attached. The 3289 uses a forms control buffer for printer spacing control, instead of a carriage tape, and the universal character set feature is standard. The print speed varies depending on the character set used:

Number of Characters	Speed (lines per minute)
48	400
64	300
94	230
116/127	160

- The 3262 Line Printer Model 1 with 132 print positions and a 48-, 64-, 94-, 116-, or 127-character set. A maximum of two can be attached. A forms control buffer and the universal character set are standard. The 3262 is similar in construction to the 3289 but the 3262 provides faster print speeds than the 3289, as follows:

Number of Characters	Speed (lines per minute)
48	650
64	466
96	364
128	274

Any combination of 3278 Model 2 displays, 3287 Printers, and 3262 and 3289 Line Printers up to a maximum of 7 or 15 (with the expansion feature) can be attached to the Display/Printer Adapter as long as no more than two line printers, any combination of 3289 Model 4 and 3262 Model 1 printers, are attached.

The 3289 and 3262 printers are designed to be used as a system output device (SYSLST for DOS, DOS/VS, and DOS/VSE, for example) and for remote printing operations. The 3287 is a tabletop printer that is designed to be used as a hard-copy device for the operator console and 3278 displays attached via the Display/Printer Adapter. Additional printers, such as the 3203 Model 5 and 3211, can be channel attached to the 4331 Processor.

The devices attached via the Display/Printer Adapter are controlled by support processor 1. Data is transferred between the support processor and its attached I/O devices on a byte basis. The support processor transfers data to the common subadapter (via an intermediate adapter) using a two-byte cycle stealing procedure. Four-byte cycle stealing is then used to transfer data to and from processor storage over the integrated channel bus.

Sixteen nonshared subchannels are provided for the Display/Printer Adapter. They are contained in reserved processor storage (processor section 2). One UCW is provided in data local storage for the adapter. When a subchannel is required for a data transfer operation, it is fetched from processor storage and the UCW is assigned to it for the duration of the byte transfer.

BYTE MULTIPLEXER CHANNEL

The optional byte multiplexer channel for the 4331 Processor is functionally identical to the byte multiplexer channel for System/360 and System/370 processors. A maximum of eight control units can be attached to the byte multiplexer channel, which presents a standard I/O interface. The 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 high-speed device to operate.

The byte multiplexer channel is controlled by a combination of hardware and microcode. Data transfer operations are accomplished using cycle stealing. The maximum data rate for byte mode operation is 18 KB/sec. A maximum of 500 KB/sec is possible for a burst mode operation.

The byte multiplexer channel has 31 subchannels, 4 of which are shared. A shared subchannel can handle a maximum of 16 devices. The 31 subchannels provided are reduced by 2 when the DASD Adapter is installed, by 2 when the 8809 Magnetic Tape Unit Adapter is installed, by 1 when the block multiplexer channel is present, and by 1 for each communications line attached to the Communications Adapter, as previously indicated.

For the byte multiplexer channel, one UCW is provided for each subchannel. All the subchannels and UCWs for the byte multiplexer channel are located in data local storage so that they can be made available to the instruction processing function immediately (that is, without a fetch to processor storage).

Section 20:05 lists the I/O devices that can be attached to the byte multiplexer channel. Note that magnetic tape units cannot be attached to the byte multiplexer channel when 231X disk devices are attached to the block multiplexer channel.

BLOCK MULTIPLEXER CHANNEL

The optional block multiplexer channel for the 4331 Processor can operate in block multiplexer or selector mode. When operating in selector mode, the block multiplexer channel for the 4331 Processor is functionally equivalent to the selector channels for System/360 and System/370 processors. When the block multiplexer channel operates in block multiplexer mode, it is functionally equivalent to a System/370 block multiplexer channel.

The block multiplexer channel presents a standard I/O interface and can have a maximum of eight control units attached. It has a maximum data rate of 500 KB/sec and, therefore, permits attachment of I/O devices that cannot be attached to Model 30 and 40 selector channels, which have a 312 KB/sec maximum data rate.

Like the byte multiplexer, the block multiplexer channel has multiple subchannels, each of which can support one I/O operation. The block multiplexer channel for the 4331 Processor has 40 subchannels. When the 4331 Processor is ordered, these subchannels can be configured as (1) 32 nonshared and 8 shared with up to 16 devices each or (2) 32 nonshared and 4 shared with up to 32 devices each. Nonshared subchannels are designed to be used with devices that are capable of using block multiplexing.

The shared subchannel configuration (number of shared subchannels and number of device addresses per shared subchannel) can be altered from the two standard configurations by the customer engineer. Any combination of shared subchannels and device addresses is possible as long as (1) the number of device addresses per shared subchannel is a multiple of 8 and (2) the number of shared subchannels times the number of device addresses per shared subchannel equals 128.

Each shared subchannel for the block multiplexer channel is assigned to operate in one of three modes: selector, block multiplexer, or 231X. The mode for a shared subchannel is assigned by default based on its address and the specify code used when the 4331 Processor is ordered. For addresses 8X, AX, and CX (where X can range from 0 to F), selector mode is assigned by default. For addresses 9X, BX, and DX, 231X mode is the assigned default. For addresses EX and FX, block multiplexer mode is assigned by default.

The default assignments can be changed by the customer engineer at installation time or thereafter. When a shared subchannel operates in selector mode, there is no disconnection from the channel during command-chained channel programs. This mode should be assigned to shared subchannels associated with devices (such as magnetic tape units) that do not have disconnection capability.

When block mutliplexer mode is assigned to a shared subchannel, it can disconnect from the channel if the I/O control unit is capable of disconnection and block multiplexer mode is in effect for the block multiplexer channel. The 3272 Control Unit, for example, should be assigned a shared subchannel that operates in block multiplexer mode.

A 231X mode shared subchannel is designed to be assigned to a 231X device. This mode provides for temporary higher priority handling of interruption requests for microcode servicing of 231X I/O operations. The higher priority is designed to ensure that the chaining requirements in 231X channel programs are met. The subchannel also operates in selector mode.

Note that 231X disk devices cannot be attached to the block multiplexer channel if any one of the following is installed in the 4331 Processor: (1) 3340 Direct Attach, System/3 Data Import, or 8809

Magnetic Tape Unit Adapter; (2) magnetic tape units are attached to the byte multiplexer channel; (3) 3370 disks are attached to the DASD Adapter; or (4) a line attached to the Communications Adapter operates at a speed greater than 9600 bps. These devices would overrun because of the high priority assigned to handle servicing requests from 231X devices.

For block multiplexer subchannels that are assigned block multiplexer mode, the setting of a channel mode bit (0) in control register 0 determines whether the addressed subchannel operates in block multiplexer 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. Note also that the CLEAR I/O instruction is executed as a TEST I/O instruction when the mode bit is set to zero.

The subchannels for the block multiplexer channel are contained in reserved processor storage (processor section 2). One UCW is provided in data local storage for the channel. When a subchannel is required for a data transfer operation, it is fetched from processor storage and the block multiplexer UCW is assigned to it for the duration of the data transfer operation. Data transfer operations between the block multiplexer channel and the integrated channel bus are accomplished using cycle stealing.

The devices that attach to the block multiplexer channel in the 4331 Processor are listed in Section 20:05. Note that disk devices with the rotational position sensing capability (such as 3310 and 3370 direct access storage) attach to the 4331 Processor via the DASD Adapter only, not via the block multiplexer channel.

The block multiplexer channel in the 4331 Processor can be used for the attachment of unit record and display devices that are capable of block multiplexing (see later discussion of these devices), tape units with a higher data rate than 8809 drives (it is recommended that tape units not be attached to the byte multiplexer channel), and direct access devices that do not have block multiplexing capability (2314/2319, for example).

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 subchannel 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 a block multiplexer channel, 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, for the block multiplexer channel in the 4331 Processor, only 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 4331 Processor when attached to a nonshared subchannel (of the block multiplexer channel or DASD Adapter) are:

- 3310 Direct Access Storage. One subchannel per drive in each string is required.
- 3370 Direct Access Storage. One subchannel per logical device (two subchannels per 3370 drive) is required.
- 3340 Direct Access Storage. One subchannel per drive in each string is required (and more than one emulation buffer must be allocated).
- 2540 Card Read Punch. One subchannel for the reader and one for the punch is required.
- 3505 Card Reader and 3525 Card Punch. One subchannel for each reader and for each punch is required.
- 3203 Model 5 Printers. One subchannel per printer is required
- 1403 Printers attached to a 2821 Control Unit. One subchannel per printer is required.
- 3211 Printer. One subchannel per printer is required.

When attached to the 4331 Processor, magnetic tape units and direct access devices without rotational position sensing capability, such as 2311, 2314, and 2319, should be associated with a shared subchannel of the block multiplexer channel and 231X mode should be assigned to the shared subchannels to obtain high-priority chaining and selector mode for these devices.

Each 3272 Control Unit attached to the block multiplexer channel in the 4331 Processor should be assigned a shared subchannel 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 subchannels 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 the block multiplexer channel for the 4331 Processor when executing a command-chained channel program:

1. Direct access devices without rotational position sensing capability (2311, 2314, and 2319) assigned to a shared subchannel with block multiplexer mode assigned disconnect after a chained seek if the channel is operating in block multiplexer mode. When a 231X is assigned a shared subchannel that operates in 231X (or selector) mode, the channel and the disk control unit are busy during the entire time a command-chained 231X channel program is in operation and there is no disconnection after a chained seek.

2. All tape drives attached to a shared or nonshared subchannel operate in exactly the same way 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 Model 5, 3211, and 3800, disconnect during the mechanical motion of the device when assigned to a nonshared subchannel. Reconnection occurs later to fill or empty the associated buffer.

   For example, a 3203 Model 5 Printer assigned to a nonshared subchannel 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 subchannel of the block multiplexer channel for the 4331 Processor. Table 10.20.1 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 subchannel type when the device is attached to the 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 10.20.1. Channel attachment and subchannel mode for frequently used I/O devices for the 4331 Processor

I/O Device or Control Unit	Channel Attachment Capability	Recommended Channel Type	Recommended or Required Subchannel 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	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
Displays			
2840	Byte, Selector, Block	Selector	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	Selector, Block	Selector	Shared in selector mode
3411	Selector, Block	Selector	Shared in selector mode
3803	Selector, Block	Selector	Shared in selector mode
Direct Access Storage			
2314/2319	Selector, Block	Selector	Shared in 231X mode
2841 (with 2311)	Selector, Block	Selector	Shared in 231X mode

Table 10.20.1 (continued)

I/O Device or Control Unit	Channel Attachment Capability	Recommended Channel Type	Recommended or Required Subchannel Mode When Attached to a Block Multiplexer Channel
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

DASD ADAPTER

The optional DASD Adapter is designed to handle data transfer operations for direct access devices that transfer data in fixed-length blocks, that is, for fixed block architecture devices (such as 3310 and 3370 drives). It provides channel and control unit functions for the direct attachment of disk units that conform to its required interface.

The DASD Adapter is customized to handle a variable record length architecture (count, key, data) disk device (such as the 3340) through the use of appropriate functional adapter microcode. The DASD Adapter operates together with microcode resident in reloadable control storage to control data transfer operations.

The DASD Adapter connects directly to the integrated channel bus (without an intermediate common subadapter) to function like a block multiplexer channel. It uses cycle stealing for data transfer operations over the integrated channel bus. It has 32 nonshared subchannels. These subchannels are located in reserved processor storage (processor section 2). A group of 8 consecutively addressed subchannels is allocated for each string of devices attached to the DASD Adapter.

Two UCWs are provided in data local storage for use with the 32 subchannels. The second UCW is provided for the prefetching of one CCW (for 3310 and 3370 drives only) and to permit fast switching from one channel program to another during block multiplexing operations.

The DASD Adapter can have a maximum of four strings attached and strings of the three different supported disk types can be intermixed on the adapter. The following direct access devices can be attached to the DASD Adapter:

- A maximum of four strings of 3310 drives when no other direct access device types are attached to the adapter. Each 3310 string can have a maximum of four drives. The four standard device address ranges assigned to 3310 drives are 40 to 43, 50 to 53, 60 to 63, and 70 to 73 for the first to fourth strings, respectively.

- A maximum of four strings of 3370 drives when no other direct access device types are attached to the adapter. Each 3370 string can have a maximum of four drives, which provide a maximum of eight access mechanisms. The standard device address ranges assigned to the 3370 strings for the first and second strings, are 20 to 27 and 30 to 37.

Note that 3370 disk drives cannot be attached to the DASD Adapter if 231X disk drives are attached to the block multiplexer channel or a communications line attached to the Communications Adapter operates at a data rate greater than 9600 bits per second.

- A maximum of two strings of 3340 Model A2 and B drives. The other two strings attached to the DASD Adapter can be any combination of 3310 and 3370 strings. Each 3340 string can have a maximum of eight drives (the first unit in a string must be 3340 Model A2). The 3340 Direct Attach feature is required to attach 3340 disk storage to the DASD Adapter. The addresses assigned to 3340 drives can be in the range of 00 through 7F. The standard device addresses assigned to 3340 drives are 00 through 07 and 10 through 17 for the first and second strings, respectively.

The customer engineer must provide configuration data for the DASD Adapter when it is installed. This data is stored on the system diskette and consists of the types of direct access devices attached and their addresses. Configuration data can be changed by the customer engineer when required.

The DASD Adapter contains 512 bytes of control storage. This control storage is used to contain two fullword data buffers and the microcode that is used to control the operation of the DASD Adapter. The data buffers are used in a fashion similar to the two sets of buffers in the common subadapter. Microcode for the DASD Adapter is called picocode and portions are paged into control storage in the DASD Adapter, as required, from reserved processor storage, where the picocode is resident together with instruction processing function microcode.

The DASD Adapter picocode controls the operation of its direct access devices after I/O instruction interpretation and execution is done by the instruction processing function (until condition code setting occurs). The DASD Adapter prefetches one CCW during data chaining on input operations involving 3310 or 3370 drives in order to handle data chaining between sectors/blocks for these drives. No other CCW prefetching is performed by the DASD Adapter.

The optional 3340 Direct Attach feature provides functional adapter microcode that enables the DASD Adapter to handle the count, key, data format of 3340 drives. The DASD Adapter itself (without functional adapter microcode) can handle only the transfer of fixed-size blocks of data. The functional adapter microcode is required to interpret 3340 CCW lists and handle exceptional conditions. Note that the 3340 Direct Attach feature cannot be installed in a 4331 Processor when 231X disk devices are attached to the block multiplexer channel.

In order to access 3340 direct access devices, the operator must specify the number of emulation buffers that can be used for 3340 operations. These buffers are specified using the operator console (program load display) and are allocated during IPL (see additional discussion of 3340 command emulation in Section 20:15).

The optional System/3 Data Import feature can be installed together with the DASD Adapter to provide a conversion capability for System/3 users. This feature enables 3348 Data Modules created by a System/3 Model 12 or 15 to be read by 3340 Model A or B drives attached to the DASD Adapter in a 4331 Processor. This feature does not permit any writing of System/3-format 3348 Data Modules. Thus, the feature can be used to convert System/3-format files to disk types that are attachable to the 4331 Processor. Standard device addresses for these 3340 drives are 00 through 07 and 10 through 17.

When the System/3 Data Import feature is installed, System/3-format 3348 Data Modules can be read concurrently on one or both strings of

installed 3340 drives. In addition, System/3-format 3348 Data Modules and 3348 Data Modules created by a 4300 Processor or System/370 processor can be processed concurrently using the same 3340 string (assuming the 3340 Direct Attach feature is installed). The System/3 Data Import feature cannot be installed in a 4331 Processor if 231X disk devices are attached to the block multiplexer channel.

Note that while specification of the 3340 Direct Attach feature enables 3340 drives to be attached to the DASD Adapter, System/3-format 3348 Data Modules cannot be read on the 3340 drives unless the System/3 Data Import feature is specified. Both features must be specified if System/3- and 4300 Processor/System/370-format 3348 Data Modules are to be processed.

The optional 2311/2314/2319/3310 Direct Access Storage Compatibility feature can be installed together with the DASD Adapter to provide for the conversion of 231X disk devices to 3310 disk devices. This feature permits emulation of 2311 or 2314/2319 drives on a maximum of two of the 3310 strings attached to the DASD Adapter in a 4331 Processor. One shared subchannel is provided for use with 231X emulation I/O operations. The addresses assigned to the 231X devices that are emulated on 3310 drives can be in the range of 90 to CF.

Emulation of both 2311 and 2314/2319 drives on 3310 disk devices at the same time is not supported. As for 3340 drives, the operator must define the required emulation buffers using the operator console (program load display). Operation of the 2311/2314/2319/3310 Direct Access Storage Compatibility feature is discussed in detail in Section 20:10.

Note that emulation (231X and/or 3340) can be performed simultaneously on only two of the direct access storage strings attached to the DASD Adapter. Thus, when one string of 3340 drives is active, 2311 or 2314/2319 emulation can be performed on only one 3310 string attached to the DASD Adapter. When two 3340 strings are active, 2311 or 2314/2319 emulation cannot be performed on any attached 3310 strings. Conversely, when 231X emulation is being performed on one 3310 string, only one 3340 string can be active, and when 231X emulation is active on two 3310 strings, neither 3340 string can be active.

8809 MAGNETIC TAPE UNIT ADAPTER

Optionally, one 8809 Magnetic Tape Unit Adapter can be installed in the 4331 Processor to provide for the attachment of up to six 8809 Magnetic Tape Units. Like the DASD Adapter, the 8809 Magnetic Tape Unit Adapter connects directly to the integrated channel bus and is designed to transfer data in blocks. The 8809 adapter has 512 bytes of control storage that contains two fullword data buffers and the picocode required to control the 8809 adapter.

The 8809 adapter has eight nonshared subchannels located in reserved processor storage (processor section 2). Two UCWs are provided in data local storage for use by the nonshared subchannels. The second UCW is used during picocode loading operations.

The 8809 Magnetic Tape Unit Adapter cannot be installed with the optional 1401/1440/1460 Compatibility feature and cannot be installed if 231X devices are attached to the block multiplexer channel. Additional information regarding this adapter is contained in Section 20:25, which discusses 8809 tape units.

5424 ADAPTER

One optional 5424 Adapter can be installed to permit one 5424 Multi-Function Card Unit, Model A1 or A2, with address 04C to be attached to the 4331 Processor via a cable up to approximately 3 meters (9.7 feet) in length. The 4331 Processor Attachment feature must be installed on the 5424 to enable it to be attached to the 4331 Processor. The 5424 handles 96-column cards and is provided as a conversion feature for System/3 users and System/370 installations that have a 5425 MFCU installed.

The operation of the 5424 is controlled by support processor 2, which is identical to the standard support processor 1. The 5424 Adapter, 5424 MFCU, and support processor 2 and its I/O bus are called the I/O subsystem. One nonshared subchannel and one UCW are provided for this adapter. Note that the optional Adapter Power Prerequisite and Adapter Logic Prerequisite features are required for installation of the 5424 Adapter.

DISKETTE DRIVE

Optionally, one single-feed diskette drive (identical to that used as the system diskette drive) can be directly attached to the 4331 Processor for user use. The diskette drive unit contains its own adapter. This diskette drive is controlled by support processor 1 and can have an address in the range of 009 to 01F.

This diskette drive is capable of reading and writing removable diskettes that are created using IBM I/O devices and systems that have a diskette drive, such as the 3741 Data Station, 3741 Programmable Work Station, 3540 Diskette Input/Output Unit, 3747 Data Converter, 3770 Data Communication System, 3790 Communication System, 8100 Information System, Series/1, System/3, System/32, System/34, System/38, and the optional diskette drive in another 4331 Processor.

The diskette drive records EBCDIC code only. Data can be recorded on both sides of the diskette. The maximum capacity of a single diskette is 242,944 bytes when the diskette is organized for basic data interchange with other I/O devices and processors other than the 4331 Processor. The data interchange format utilizes tracks 1 through 74 to provide 1898 sectors of 128 bytes each for data. Track 0 is used as the label track and tracks 75 and 76 are the alternate tracks.

Diskettes that are to be used only for data interchange with other 4331 Processors that have the diskette drive installed have a maximum capacity of 246,272 bytes organized in 1924 sectors of 128 bytes each.

COMMUNICATIONS ADAPTER

The Communications Adapter optional feature, together with lower cost-per-byte direct access storage and lower-cost processor storage for the 4331 Processor, offers economical and flexible entry into communications-based applications. The Communications Adapter provides communications facilities compatible with those of Models 30 and 40 and thus also provides simplified transition from these processors to the 4331 Processor.

The Communications Adapter permits direct attachment to the 4331 Processor of up to eight low- and medium-speed communications lines without installation of a channel and communications controller. A power expansion feature is required when more than three lines are installed. Only one Communications Adapter can be installed in a 4331 Processor.

Lines connected via the adapter are addressed and logically operate as if attached to the byte multiplexer channel via a 2703 Transmission Control Unit. One nonshared subchannel is provided for each communications line installed (up to eight nonshared subchannels). The 4331 Processor can have communications lines and terminals connected via the byte and/or block multiplexer channel and 2701, 2702, 2703, 3704, and 3705 transmission control units in addition to, or instead of, via the Communications Adapter.

The adapter supports half-duplex and full-duplex communications lines (but not full-duplex data transmission). They can be private, leased, or switched public telephone lines. Binary synchronous control (BSC), start/stop, and synchronous data link control (SDLC) lines can be attached to the Communications Adapter; however, only two of the three supported types of line control can be utilized with a given Communications Adapter, in any combination. Up to two lines can have an autocall unit attached.

When multiple lines operate concurrently, for start/stop lines, data rates from 75 to 1200 bps can be handled and for BSC and SDLC lines, data rates of 600 to 9600 bps are supported. The maximum aggregate data rate that can be handled by the lines attached to the Communications Adapter is 64,000 bps, provided interference from other operating channels permits. In order to achieve the maximum rate, the highest-speed line must be installed in line position 1, the next highest speed line must be installed in line position 2, etc.

One high-speed BSC or SDLC line can be attached to the Communications Adapter. A high-speed line is defined as one that has a data rate over 9600 bps up to a maximum of 56,000 bps. One high-speed line can operate concurrently with other lines attached to the Communications Adapter as long as no other line operates at a rate higher than 9600 bps and the aggregate data rate of the lines does not exceed 64,000 bps.

The Communications Adapter Base feature contains common circuits and control for the lines attached. It incorporates the following capabilites: auto answer, autopoll operation, multipoint control station (primary) functions, EBCDIC transparent mode for BSC lines only, and EBCDIC and ASCII mode for BSC lines only. For each line attached, one of two different line attachment base features (Line Attachment Base for Clocked Modems or Line Attachment Base for Nonclocked Modems) must be installed also. The Autocall Unit Interface feature can be installed for two of the attached lines.

Each line also requires a modem feature unless the Digital Data Service Adapter or Local Attachment Interface feature is installed. The modem can be a standalone unit or a 1200-bps integrated modem. Processor-clocked and modem-clocked (self-clocking) modems can be used.

The Digital Data Service Adapter provides the circuits and controls for the attachment of one BSC or SDLC line and includes an internal Dataphone Digital Service Adapter that provides an interface to the Private Line Digital Data Service Network of AT&T via the AT&T Channel Service Unit. The Digital Data Service Adapter can operate at speeds of 2400, 4800, 9600, and 56,000 bps.

The Local Attachment Interface provides the circuits and controls for the local attachment of one BSC or SDLC terminal to the Communications Adapter without the use of any modems. Line speeds of 1200, 2400, 4800, and 9600 bps are supported.

A configuration table for the Communications Adapter is contained on the system diskette. It contains one entry for each installed line that describes the characteristics and features of the line.

The customer engineer can modify line configuration parameters at installation time or thereafter. Changes are written to the configuration table. The customer engineer can change the following parameters for BSC, start/stop, and SDLC lines using the operator console and a configuration display for the Communications Adapter:

- Nonswitched or switched
- Permanent request to send (duplex facility)
- EIA/V35 interface card wrap disabled (external modem wrap)
- Select standby for switched network backup
- Integrated modem answer tone select (2100 or 2025 hz)

The customer engineer can change the following additional parameters for both SDLC and BSC lines:

- New sync
- Data signal rate select (half-speed select for SDLC and low or high speed for BSC lines)
- High-speed operation (enables an extra transmit buffer to be used)
- DTR or CDSTL modem procedure
- Not NRZI (SDLC only)
- Error index byte mode and EBCDIC or ASCII transmission code (BSC only)

The customer engineer can change the following additional parameters for start/stop lines:

- Read interrupt
- Write interrupt
- Unit exception suppress
- Turnaround delay select
- Line speed

The operator can specify certain parameters for each individual line before transmission occurs or during transmission when retries are being performed to attempt to correct an error. Such changes override the parameters specified for the line in the configuration table but are effective only until a system reset, IPL, or IML is performed.

The following configuration parameters for an individual line can be specified by the operator using the operator console:

- Select standby for SDLC, BSC, and start/stop lines
- Data signal rate select (high or low speed) for SDLC, BSC, and start/stop lines
- NRZI mode for SDLC lines

- Write interrupt, read interrupt, and unit exception suppression for start/stop lines
- Half-speed operation for BSC lines (for clocked and nonclocked modems)
- Error index byte mode for BSC lines
- ASCII instead of EBCDIC code for BSC lines

The following can be attached to BSC lines of the Communications Adapter:

- 2770 Data Communication system
- 2780 Data Transmission Unit
- 3270 Information Display System
- 3650 Retail Store System
- 3660 Supermarket System
- 3735 Programmable Buffered Terminal
- 3741 Data Station and Programmable Work Station
- 3747 Data Converter
- 3770 Data Communication System
- 3780 Data Communications Terminal
- 5231 (equivalent to a 3741)
- A System/360, System/370, or 4300 Processor via a 2701, 2702, 2703, 2715, 3704, or 3705 transmission control unit
- 1130 System with the Synchronous Communications Adapter
- 1800 System with the Communictions Adapter
- System/360 Model 20 with the Binary Synchronous Communications Adapter
- System/360 Model 25 with the Integrated Communications Adapter and Synchronous Data Adapter
- System/370 Model 115, 125, 135, or 138 with the Integrated Communcations Adapter
- System/7 with Binary Synchronous Communications Control (as a System/3)
- System/3 with a Synchronous Communications Adapter
- System/32 with a Synchronous Communications Adapter
- System/34 with a Synchronous Communications Adapter
- 4331 Processor with a Communications Adapter
- Series 1 with a Synchronous Communications Adapter

The following can be attached to SDLC lines of the Communications Adapter:

- 3270 Information Display System
- 3600 Finance Communication System
- 3650 Retail Store System
- 3660 Supermarket System
- 3767 Communication Terminal
- 3770 Data Communication System
- 3790 Communication System
- System/32 as a 3770
- System/34
- System/370 processor or 4300 Processor via a 3705 Communications Controller as a secondary station in an SNA 3 (Systems Network Architecture) or SNA 4 network
- 4331 Processor with a Communications Adapter (as a primary or secondary station in an SNA 3 or SNA 4 network)
- 8100 Information System

The following (which use IBM Terminal Control - Type 1) can be attached to start/stop lines of the Communications Adapter:

- 2740 Communication Terminal
- 2741 Communication Terminal
- 3767 Communication Terminal as a 2740 or 2741
- System/7 (5010 Processor) as a 2740
- 5100 Portable Computer as a 2741

10:25 BLOCK MULTIPLEXING OPERATIONS WITH FIXED BLOCK ARCHITECTURE 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 33XX count, key, data format direct access devices to enable them to utilize block multiplexing more effectively. This function is also implemented in fixed block architecture devices, such as 3310 and 3370 Disk Storage.

While RPS devices cannot be attached to the block multiplexer channel in the 4331 Processor, the advantages discussed apply to fixed block architecture devices attached to the DASD Adapter, which has the block multiplexing capability.

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/adapter 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 than is possible when RPS and block multiplexing are not utilized.

Multiple requesting is implemented in a direct access device control unit to enable it to handle concurrent execution of multiple channel programs for devices with the rotational position sensing capability. The DASD Adapter, for example, can simultaneously control a maximum of 32 channel programs.

Shown below is the RPS channel program that is required to read or write one physical disk record from a CKD disk device. Commands shown in the sample channel program 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. The RPS channel program is shown for comparison with the FBA channel program that is illustrated later in this subsection.

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 (few hundred microseconds)
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/WRITE	(Processor storage address of input or output area)	Busy

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 the advantages of the RPS capability.

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 3310 disks, for example, an addressable block, which is called a sector, 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/integrated adapter.

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 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/adapter occurs as soon as the data specified by the LOCATE command is transferred to the storage control

function. The channel/adapter 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/adapter to execute the data transfer command after LOCATE. If the channel/adapter 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. For 3310 drives, for example, diagnostic commands are utilized to process the ID field of a data record and to read the data field of a data record whose ID is unreadable (see Section 20:10 for the specific subcommands implemented for 3310 drives).

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 at the beginning of 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. That is, disk designers can place the alternate blocks for a given FBA 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, the alternate block/track location during processing is automatically performed by the storage control function for both 33XX CKD and FBA 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. The storage control function for a disk device 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 (3310 to 3370 for the 4331 Processor, for example) without changing the operating system being used, processing programs that access the disks, or job control statements for these programs.

  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.

10:30 STANDARD AND OPTIONAL FEATURES

STANDARD FEATURES

The following are standard features of the 4331 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 4331 Processor that are not available for Models 30 and 40 are PURGE TLB and LOAD REAL ADDRESS (System/370-mode-only instructions), the ECPS:VSE-mode-only instructions, 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 by the 4331 Processor because it provides no advantage as a result of the utilization of the instruction processing function for I/O initialization.)

  *Privileged instruction

- BC and EC modes of operation and control registers
- Interval timer (10-ms resolution)
- Time-of-day clock (16-microsecond resolution)
- CPU timer and clock comparator
- Monitoring feature
- Program event recording
- Expanded machine check interruption class
- Interruption for SSM instruction
- Reloadable control storage for the instruction processing function (64K)
- ECC on processor storage
- Byte-oriented operands
- Store and fetch protection
- Reference and change recording
- Channel retry data in a limited channel logout area
- Support processor subsystem
- Display/Printer Adapter

The following are standard features that can operate only when System/370 mode is 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, capacity counts, and virtual storage size control
- 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 4331 Processor, all of which can be field-installed and can operate in System/370 or ECPS:VSE mode unless otherwise indicated, are:

- 3278 Model 2A Display Console (required)
- Display/Printer Adapter Expansion
- Control Storage Expansion (additional 64K of RCS)
- Printer-Keyboard Mode for the 3278 Model 2A Display Console
- ECPS:VM/370 (operates only in System/370 mode and mutually exclusive with 1401/1440/1460 Compatibility)
- 1401/1440/1460 Compatibility (mutually exclusive with the 8809 Magnetic Tape Unit Adapter and ECPS:VM/370)
- Byte multiplexer channel (one maximum)
- Block multiplexer channel (one maximum)
- DASD Adapter (one maximum)
- 3340 Direct Attach
- 2311/2314/2319/3310 Direct Access Storage Compatibility
- 8809 Magnetic Tape Unit Adapter (one maximum and mutually exclusive with 1401/1440/1460 Compatibility)
- System/3 Data Import
- Communications Adapter (one maximum)
- 5424 Adapter (one maximum and includes support processor 2)
- Diskette drive (one maximum)
- External signals (provides six external interruption lines)
- Remote Support Facility

SECTION 12: OPERATOR CONSOLE

12:05 GENERAL DESCRIPTION

A display console for system control and operator/operating system communication is required for the 4331 Processor. The 1052 Model 7 Printer-Keyboard (for System/360 processors) and 3210/3215 Console Printer-Keyboards (for System/370 processors) cannot be attached to the 4331 Processor. A display console provides faster display than a typewriter keyboard and the console for the 4331 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 4331 Processor, (2) communicate with the operating system, and (3) perform diagnostic operations. The cable connecting the operator console to the 4331 Processor can be a maximum of 6 meters (20 feet) in length.

The operator console for the 4331 Processor is a 3278 Model 2A Display Console (cathode ray tube for displaying data and keyboard for entering data). The operator console is a 3278 Model 2 with certain modifications. Specifically, the 3278 Model 2A contains an operator's control panel that is not present on a 3278 Model 2, the function of several of its keys is altered to support console-type operations, and the last four lines of the screen cannot be used for operator-to-operating system communication.

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 Display Console. The 3278 Model 2A is also used as the operator console for the 4341 Processor but has a different keyboard and operator control panel combination.

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.

The screen of the operator console can display 25 lines of 80 characters each simultaneously. 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. Additional displays are provided for customer engineer use.

Note that if the operator console is malfunctioning during processor operation, the cable connecting it to the 4331 Processor can be unplugged and plugged to another installed 3278 Model 2 display that is attached via the Display/Printer Adapter, if one can be made available for use. The address assigned to the malfunctioning 3278 Model 2A console is retained for use with the 3278 Model 2 display. The operator will be able to continue processor operation but cannot perform any functions (such as power off) that are controlled by the operator control panel, which is present only on a 3278 Model 2A Display Console.

OPERATOR/OPERATING SYSTEM COMMUNICATION MODES

The 4331 Processor operator console has two possible modes of operation for operator/operating system communication: display and printer-keyboard. The display mode of operation is standard. The

Printer-Keyboard Mode optional feature can be installed to enable the operator console to operate in printer-keyboard mode to provide emulation of 1052, 3210, and 3215 printer-keyboards. The operator determines the mode to be used during IPL.

Display Mode

For display mode, the operator console appears to be a 3278 Model 2 display attached to a 3272 Model 1B 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, the 3287 Model 1 Printer (80 characters per second) or 3287 Model 2 Printer (120 characters per second) can be attached via the Display/Printer Adapter for hard-copy output. When one or more 3287 Printers are attached to the Display/Printer Adapter, the only 3287 Printer or the 3287 Printer connected to the cable port with the lowest number is automatically assigned as the hard-copy printer during IPL. Cable ports for devices attached to the Display/Printer Adapter are numbered 1, 2, 3, etc.

The display/keyboard combination and hard-copy printer, if present, are addressed separately when display mode is in effect. While addresses in the range of 009 to 01F 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. The 3287 Printer is supported for hard-copy by DOS/VSE, OS/VS1, and VM/370.

For display mode, the first 20 lines of the screen are the system area while the last 5 lines are the machine status 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 machine status area is used to display certain status information about the processor, messages for the customer engineer, and status data regarding the screen and keyboard.

Printer-Keyboard Mode

For printer-keyboard mode, the display console appears as a 1052 Printer-Keyboard to a System/360 operating system, or as a 3210/3215 Console Printer-Keyboard to a System/370 operating system. The keyboard is used for input and the cathode ray tube and a required 3287 Model 1 or 2 Printer attached to the Display/Printer Adapter are used for output. When printer-keyboard mode is made effective, the 3287 Printer attached to the lowest numbered cable port among those 3287 Printers attached to the Display/Printer Adapter is automatically assigned as the output printer. Device address X'01F' or X'009' would normally be used for the display/keyboard.

The message area for operator-to-operating system communication consists of lines 1 to 18 for printer-keyboard mode. 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.

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

In printer-keyboard mode, the 3278 Model 2A Display Console is controlled using 1052 commands. The display/keyboard and 3287 Printer have the same address and the same data is printed on the 3287 as is displayed on the screen. This mode is supported by DOS Version 3 (Release 26), DOS/VS Release 34, DOS/VSE, OS/VS1, and VM/370. The Printer-Keyboard Mode feature must be installed when DOS Release 26 or a DOS/VS or OS/VS1 operating system that utilizes the 3210/3215 console is used with a 4331 Processor.

OPERATOR'S CONTROL PANEL

The operator's control panel is located on the keyboard of the operator console. It contains the following pushbutton controls and indicators:

- IML pushbutton. When this pushbutton is pressed, an IML of all required microcode for the 4331 Processor is automatically initiated. A clear reset is then performed but the time-of-day clock is not reset. After microcode loading is completed, the processor is placed in the stopped state and the program load display is shown on the operator console.

  A switch on the 4331 Processor is used to turn power on in the 4331 Processor, the 5424 MFCU, and up to 24 channel-attached I/O devices with their power control switches set to remote. An IML of support processor and instruction processing function microcode occurs automatically after such a power on. The IML pushbutton need be used only if an IML is required when power is already on. THe IML that occurs automatically after a power on includes a clear reset and time-of-day clock reset.

- Power-off pushbutton. This pushbutton is used to remove power from the 4331 Processor, the 5424 MFCU, and channel-attached I/O devices with their power control switches set to remote. Note that disk and tape units attached to the 4331 Processor via the DASD Adapter and 8809 Magnetic Tape Unit Adapter as well as all devices connected to the Display/Printer Adapter must be powered on and off individually, using the local on/off switch contained on the unit.

- Lamp test pushbutton. When pressed, this pushbutton causes all functional indicator bulbs on the operator's control panel and the indicator light on the 5424 MFCU (if installed) to be lit. This button is used for testing purposes.

- Power-in-process indicator. This light turns on as soon as power is turned on via the power on switch on the 4331 Processor. This indicator stays on until power-on sequencing of all system components is successfully completed, at which time it is turned off and an IML of the 4331 Processor occurs. This light is also lit during a power-off process.

- 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 power section of the 4331 Processor that requires customer engineer intervention. This indicator is similar in function to the machine check light for System/360 processors.

- System indicator. This indicator is lit whenever instruction processing is taking place and the usage meter is running (as in System/360 and System/370 processors).

- Wait indicator. This indicator is lit when instruction execution is not occurring because the processor is awaiting the completion of an event (such as an I/O operation) to continue instruction processing (as for System/360 and System/370 processors).

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 keys, 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. For normal mode, pressing the mode select key causes a mode selection display to be shown on the screen 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.

  For alternate mode, the mode selection display provided for customer engineer use is shown on the screen when the mode select key is pressed. This mode selection display indicates the manual functions the customer engineer can select for the purpose of loading and executing diagnostic programs and displaying processor indicators.

- CHG DPLY. This key causes a switch between operating system and manual modes and a switch in the display currently being shown. That is, operating system mode is made effective and the current contents of the system message buffer are displayed if manual mode was in effect and the contents of the manual operations buffer were being displayed (or vice versa).

  The two buffers are contained in control storage of the support processor and each is 80 by 25 bytes in size. The system message buffer is used to hold the contents of lines 1 to 25 of the screen during the time the operator console is being used for operator-to-operating system communication. That is, it contains the current messages from the operating system and executing processing programs and status information.

  The manual operations buffer is used to hold the contents of lines 1 to 25 of the screen when operator or customer engineer manual operations are being performed. The contents of one of these two buffers is always displayed.

- CNCL (PA2). When operating system mode is in effect, the current data being entered via the console is canceled (CNCL key would be pressed before the ENTER key). An attention interruption is generated and a special identifier character is available to indicate the CNCL key was pressed. When manual mode is in effect, the CNCL key causes the console to return to operator/operating system communication mode and a change in the current display (from a display of the contents of the manual operations buffer to a display of the contents of the system message buffer). The manual operations buffer is cleared.

- INTR/LINE DISC. For normal mode, this key causes an external interruption when pressed. For alternate mode, this key is used to terminate operation of the Remote Support Facility (see Section 50).

- ERASE EOF/SP M/O. The ERASE EOF (end-of-field) key erases the data at the cursor location and to the right of it until the end of the field is reached. The support processor manual operations (SP M/O) function is operative in alternate mode only. It is used to execute certain diagnostic functions in the support processor. Actual execution does not occur when the key is pressed unless a specific plug has been inserted in the support processor control board. This prevents accidental operator invocation of diagnostic functions that are for maintenance only or that require special hardware knowledge.

- REQ (PA1)/COMM REQ. For normal mode, when this key is pressed, an attention interruption is generated when operating system mode is active. The key is not active for manual mode. For alternate mode, this key is used to request communication between a local and remote customer engineer when the Remote Support Facility is active (see Section 50).

- COPY. When the key is pressed, the contents of lines 1 through 24 of the current display are written to the 3287 Printer attached via the Display/Printer Adapter that is designated as the hard-copy printer, if one is present. This key is functional only when manual mode is in effect.

- RESET. The keyboard is reset.

- Twelve program function keys. The function that is to be executed when a program function key is pressed is installation defined.

- Start and stop keys. These keys are used to start instruction processing and to stop instruction processing after execution of the current instruction, any I/O command, or any I/O command chain in progress has completed and any enabled interruptions are taken, respectively.

The above describes the operation of the keyboard for display mode operations. Operation of the keys is the same for printer-keyboard mode except that the CNCL, REQ, and twelve program function keys cannot be used by the operator during manual operations.

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 upper-case characters to be displayed on the screen

- A normal/test mode switch that establishes normal operating mode that is used for controlling system processing operations or a test mode that is used to execute problem determination procedures when the display console malfunctions

- 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

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 mode selection display. Each function has a single-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.

MODE SELECTION DISPLAY

The mode selection display is shown in Figure 12.10.1. The functions listed on this 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. The user diskette control, diskette device address, and Communications Adapter functions appear only if the associated facility (diskette drive or Communications Adapter) is actually installed.

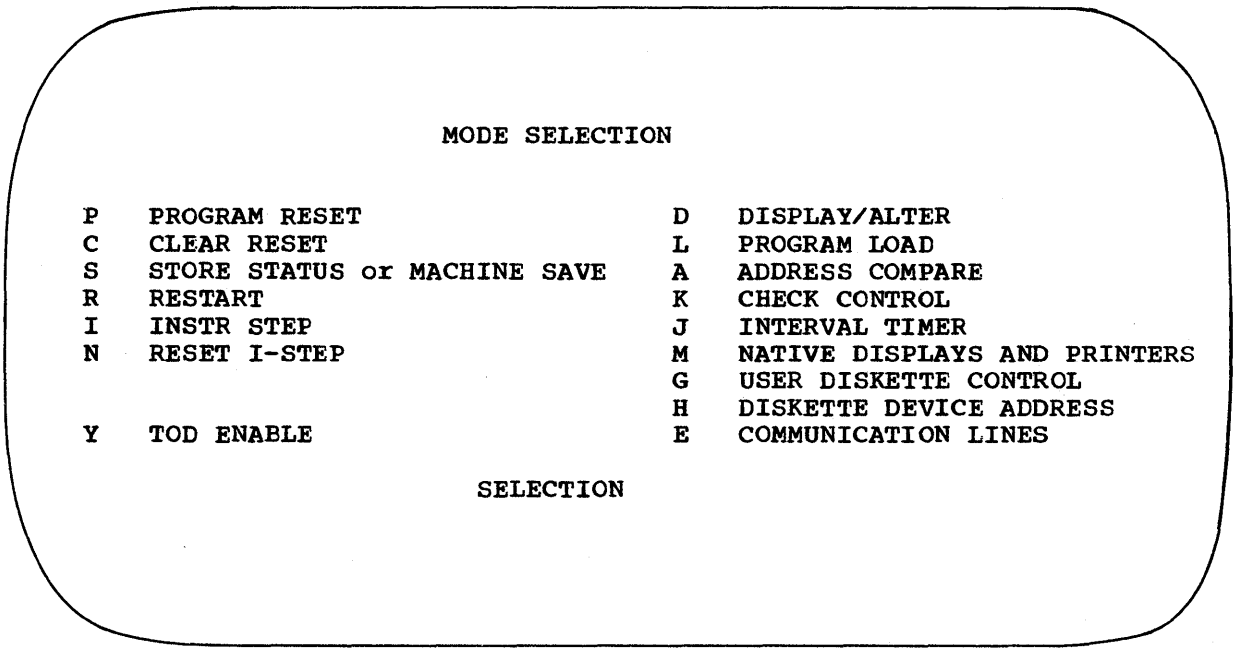


Figure 12.10.1. The mode selection display

The functions listed on the left-hand side of the mode 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 mode selection display on the screen by pressing the MODE SEL key on the operator console keyboard. Before the mode 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 one-character 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 mode selection display on the screen when the enter key is pressed.

The functions that have a display are those that require information from the operator in order to be executed. If the operator knows the data that must be supplied, this data can be entered together with the one-character identification of the function. When the enter key is pressed, the associated display is shown and the function is executed. This is called fast selection.

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

Note that execution of the display/alter, machine save, and store status functions require the processor to be stopped. However, the operator need not stop the processor before selecting one of these functions, as a stop occurs automatically upon selection.

Functions of the Mode Selection Display

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

The message "Reset Complete" appears on the mode 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) or press the start key 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. The current contents of the system message buffer are then displayed.

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 "status stored" appears on the mode selection display 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 system 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 program processor storage) are saved in reserved processor storage (processor section 2).

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 save indicator in the machine status area is turned on.

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.

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 the check-stop state.

The instruction step function is used to execute one instruction at a time each time the start key is pressed (for instruction tracing purposes). The operating system is given control of the display screen when this function is invoked. The machine status area shows the address of the instruction just executed (and contents of the instruction address) each time the start key is pressed. Another function (such as display/alter) can be selected while the instruction step function is operative. The reset I-step function terminates operation of the instruction step function and normal processing continues.

The TOD enable function permits the operator to enable the processor for time-of-day clock setting via the SET CLOCK instruction. After approximately two seconds, time-of-day clock setting is automatically disabled.

PROGRAM LOAD DISPLAY

The program load display is shown in Figure 12.10.2. It is used to specify the IPL device, indicate the type of reset to be performed (initial program reset without clear or a reset and clear), enable System/370 or ECPS:VSE mode for the processor, select the operator console mode, provide functional adapter (for 3340 emulation) and disk compatibility feature (for 231X emulation) specifications, and initiate

an IPL. The four items in the bottom row and rightmost two items in the second row do not appear unless the associated features (disk emulation or printer-keyboard mode) are installed.

The program load display appears automatically after an IML is successfully completed and can be selected thereafter using the mode selection display. The selections currently in effect are shown on the program load display.

The system diskette sent to a 4331 Processor installation will establish the following by default during IML: System/370 mode, display mode for the operator console, and no emulation. Any parameters specified for an IPL using the program load display are written to the system diskette. These parameters become effective during all successive IPLs unless changed again using the program load display. Only the parameter to be changed need be specified.

An IML occurs automatically whenever the mode (System/370 or ECPS:VSE) in effect or the virtual storage size for ECPS:VSE mode is changed using the program load display. The ECPS:VM/370 feature is automatically loaded during IML if it is installed. However, it cannot be utilized if ECPS:VSE mode is in effect.

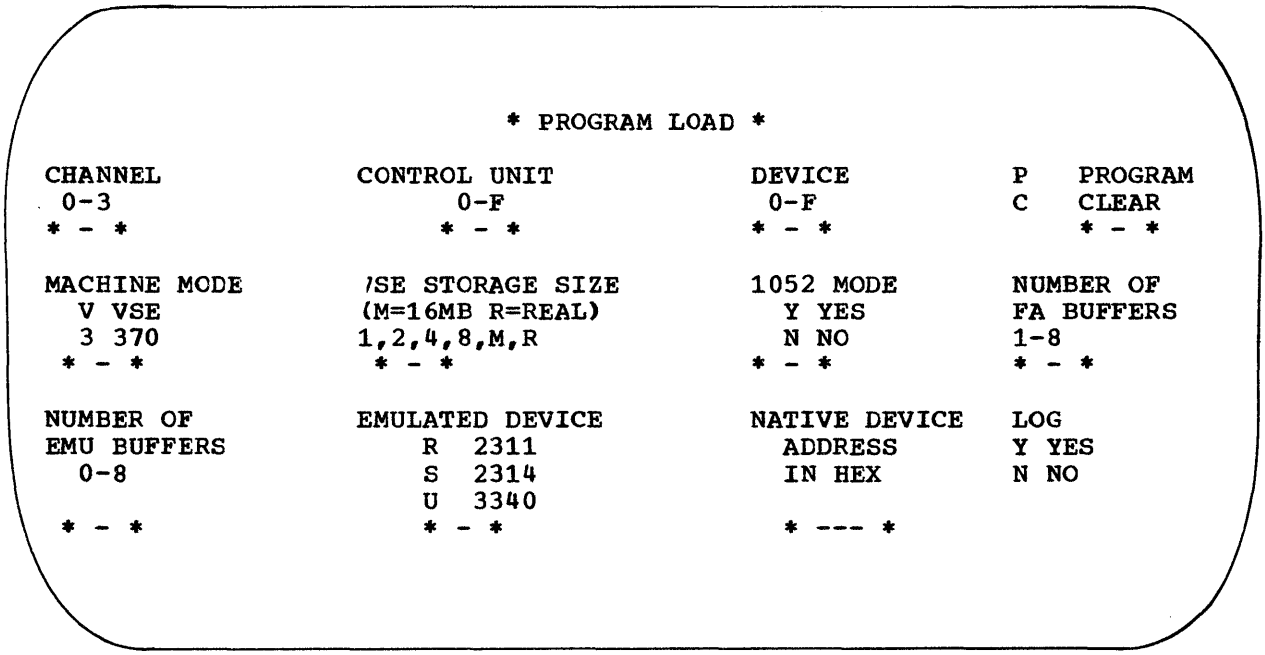


Figure 12.10.2. The program load display

DISPLAY/ALTER DISPLAY

The display/alter display enables the operator to display or alter the contents of several items, depending on the mode, System/370 or ECPS:VSE, in effect. Figures 12.10.3 and 12.10.4 show the display/alter displays for System/370 and ECPS:VSE modes, respectively.

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.

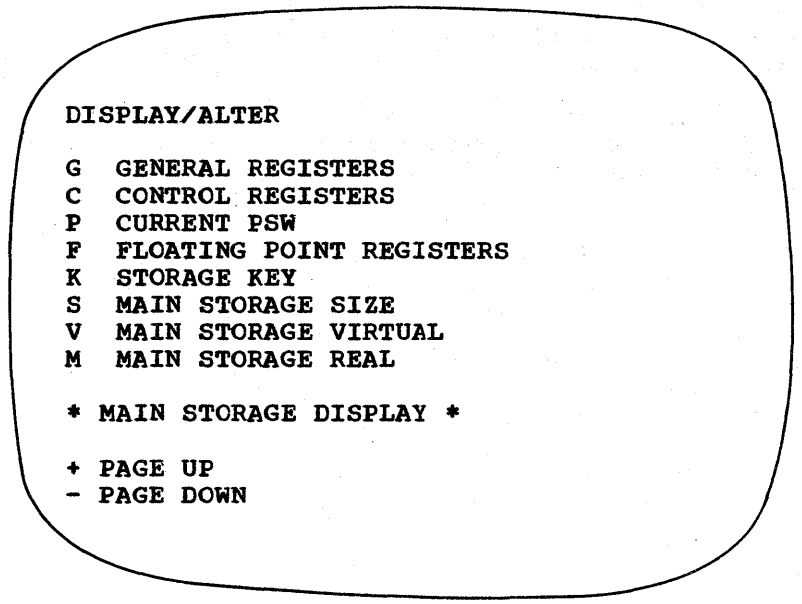


Figure 12.10.3. The display/alter display for System/370 mode

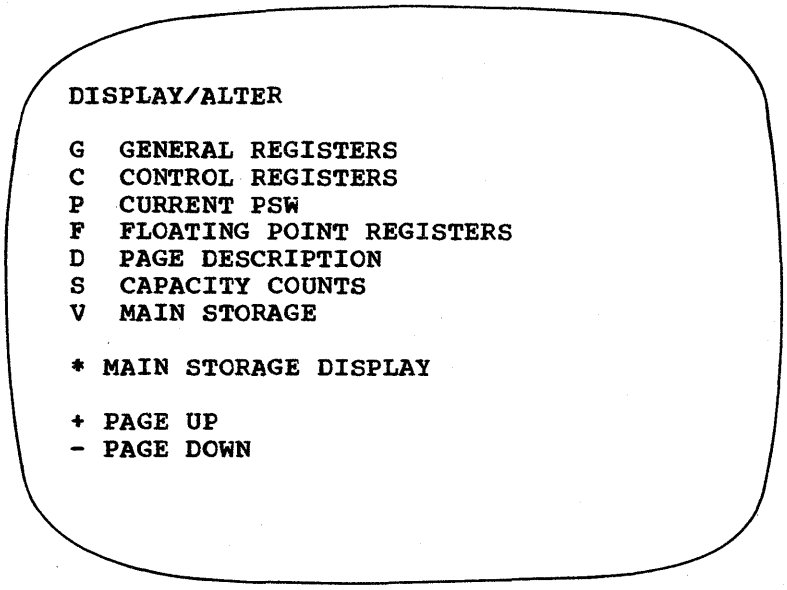


Figure 12.10.4. The display/alter display for ECPS:VSE mode

When the main storage size item is selected for System/370 mode operation, the amount of program processor storage available is displayed.

When the page description item is selected for ECPS:VSE mode operation, the page description bits in the processor storage directory can be displayed and certain of them can be altered. The bits displayed include the access control, programmable, reference, change, fetch protection, addressable, connected, and disconnected bits. Only the last three bits listed can be altered. The address of the assigned page frame and frame index value (which cannot be altered) are also shown.

For System/370 mode, 32 halfwords of virtual storage or real storage can be displayed in hexadecimal by selecting the main storage virtual or main storage real items, respectively. For ECPS:VSE mode, 32 halfwords of virtual storage can be displayed in hexadecimal by selecting the main storage item.

For both modes, the page up and page down functions enable the operator to display the next 32 halfwords after the current display or the 32 halfwords preceding the current display, respectively. A page up or page down can be requested by entering a + or - character or by pressing the page up or page down key on the keyboard.

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 CNCL or CHG DPLY key must be pressed to return control of the screen to the operating system.

ADDRESS COMPARE DISPLAY

The address compare display, shown in Figure 12.10.5, enables the operator to cause the processor to stop when the specified virtual storage address for ECPS:VSE mode or real storage address for System/370 mode is encountered during processing. The operator can limit the comparison to instruction addresses only, addresses used for storing in (but not fetching from) processor storage only, or addresses used for transferring data to and from I/O devices only. The address compare can also be set for any address that is used during instruction processing.

```
                    * STORAGE ADDRESS COMPARE *

FUNCTION            COMPARE TYPE                STORAGE ADDRESS

N NORMAL            A  ANY                      0-FFFFFF

S STOP              C  INSTRUCTION COUNT

Y SYNC              D  DATA STORE

                    I  I/O

* - *               * - *                       ADDRESS:
```

Figure 12.10.5. The address compare display

When the operator presses the enter key after keying in the required address stop specifications, processor operations resume and the address compare display is replaced by the current contents of the system message buffer. The entered specifications are displayed in the machine status area to indicate the address compare function is in effect.

CHECK CONTROL AND INTERVAL TIMER DISPLAYS

The check control display is used to specify the action to be taken when a machine check or channel check condition occurs. Normal or hard

stop mode can be set and the mode in effect is shown in the machine status area. When normal mode is established, an interruption and logout to the system diskette and processor storage occur after any type of machine check if the processor is enabled for the machine check type.

When hard stop mode is set, after a machine check or channel check (channel control, interface control, or channel data check) condition occurs, the processor enters the check-stop state. (See additional information in Section 50.)

The interval timer display is used to turn the interval timer on or off. The current state of the interval timer is shown in the machine status area.

NATIVE DISPLAYS AND PRINTERS DISPLAY

This display is provided to enable the operator to specify characteristics about the devices attached to the Display/Printer Adapter (such as keyboard type, language, and translation tables). The characteristics in effect are shown when the display is selected.

USER DISKETTE CONTROL AND DISKETTE DEVICE ADDRESS DISPLAYS

The user diskette control display is provided to show the status of the optional diskette drive and to enable the operator to control (start, stop, or reset) the device. The diskette device address display is used to set the address of the optional diskette drive.

COMMUNICATIONS ADAPTER DISPLAYS

These displays enable the operator to alter certain characteristics of the lines attached via the Communications Adapter (as discussed previously in Section 10:20).

MACHINE STATUS AREA

The last five lines of the operator console screen (21 to 25) are used as the machine 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, lines 23 and 24 are reserved for customer engineer use, and line 25 shows status information about the screen, keyboard, and hard-copy printer if it is attached. The reference code generated by the support processor when a machine check occurs is displayed on line 23 (see reference code discussion in Section 50:10).

Line 21 displays the following information:

- MAN (when the processor is stopped), CHECKSTOP (when the processor is in the check-stop state), IPL-ERROR (when an IPL could not be performed successfully), or TEST. TEST is displayed when instruction step mode is in effect, address compare has been set, check control mode is in effect, or an inline test is operating.
- LOAD during the time an IPL is in progress
- SAVE when a machine save completes successfully
- Virtual storage size established by the operator when ECPS:VSE mode is in effect

- The mode, System/370 or ECPS:VSE, in effect
- TIMER ON or TIMER OFF for the state of the interval timer
- SYSDSK when the system diskette drive or both the system diskette drive and optional diskette drive require the attention of the operator
- DISK when the optional diskette drive requires operator attention
- COMP when an address compare has been set. When the processor stops because of an equal address compare, the address at the next instruction to be executed and the contents of the first halfword of the instruction are also displayed.

Line 22 displays the following:

- RATE: I-STEP when instruction step mode is in effect
- CHK-CTL: HARD when check-stop mode is in effect
- TOD SEC or TOD ENBL to indicate the time-of-day clock cannot or can be set, respectively
- ADDR-COMP together with the specifications made when an address compare was established.

Lines 23 and 24 are reserved for displays and messages for the customer engineer. Information can be displayed on line 25 of the screen only by the support processor. This line shows status information about the screen (such as display or printer-keyboard mode in effect, manual operations in progress) and the console printer, if one is installed. This line is also used when the Remote Support Facility is active.

12:15 MAINTENANCE

A problem determination guide for the 3278 Model 2A Display Console will be provided with each 4331 Processor. When the operator console malfunctions, the operator can take the steps outlined in the guide before calling the customer engineer. The procedures include the execution of certain offline tests on the operator console. This approach is designed to increase 4331 Processor availability by reducing the time required to locate a malfunction.

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 also 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 4331 Processor are discussed. Other hardware items associated with dynamic address translation, such as reference and change recording, are discussed as well.

Subsection 15:15 describes in detail the implementation of address translation for ECPS:VSE mode. The internal mapping function and processor storage directory are described together with the page control instructions.

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 4331 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, 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 4331 Processor, for example, which uses a 24-bit binary address, a virtual storage as large as 16,777,216 bytes can be 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 4331 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.

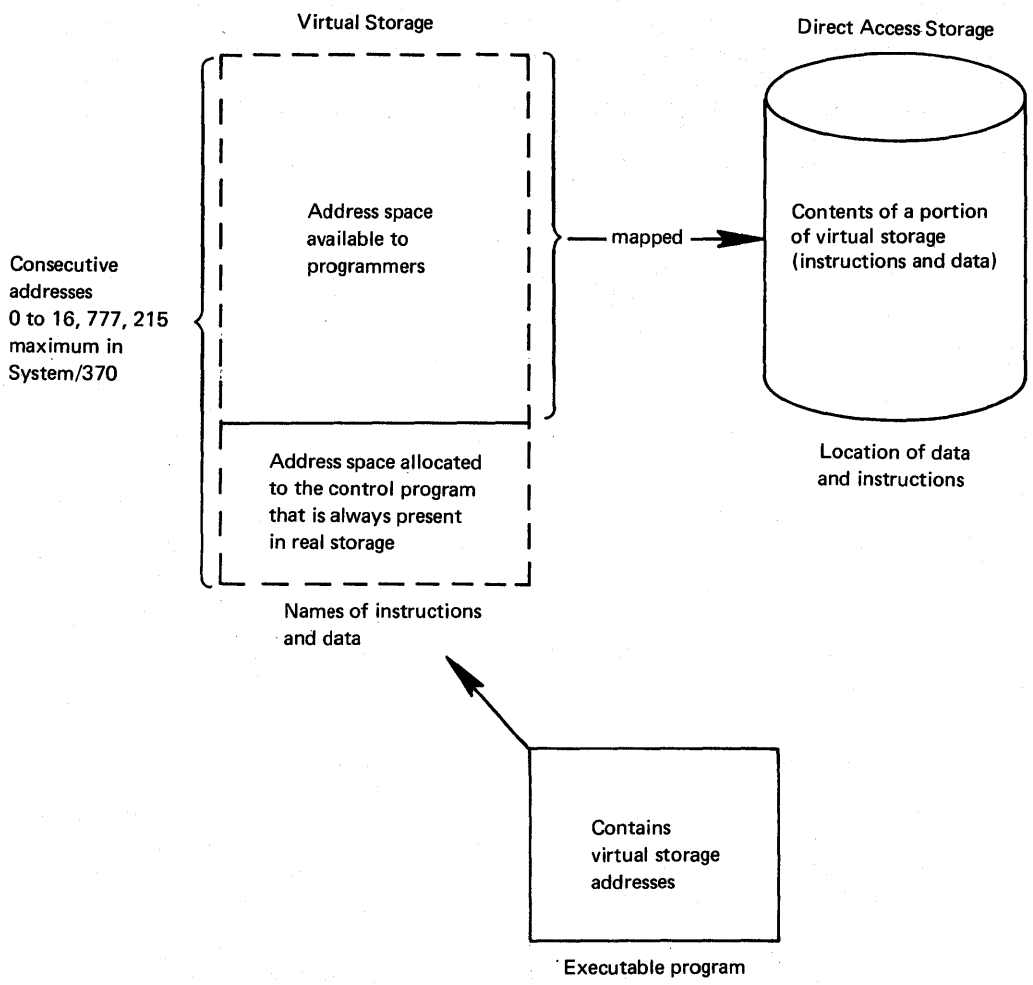


Figure 15.05.1. Names and location of instructions and data in a virtual storage environment

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.

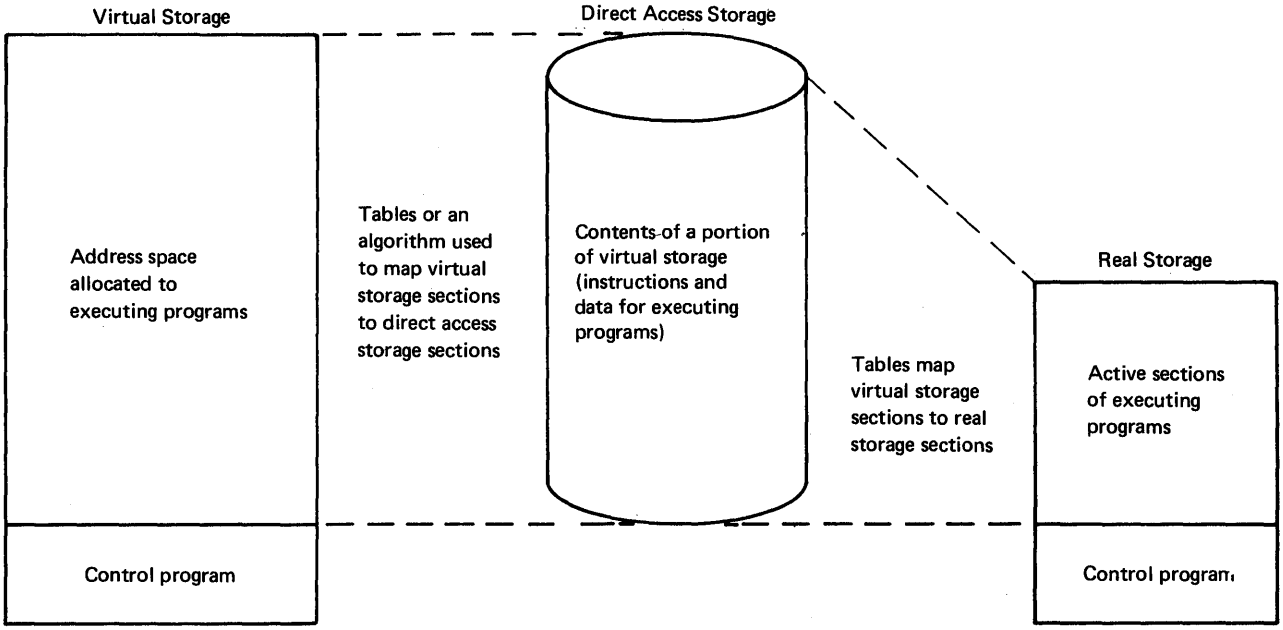


Figure 15.05.2. Relationship of virtual storage, direct access storage, and real storage

While a virtual storage up to 16 million bytes in size can be addressed by any 4331 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 processor storage directory. 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.

For System/370 mode, 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. This real storage address is then converted to a processor storage address using the processor storage directory. For ECPS:VSE mode, the translation process produces a processor storage address (see discussions in Sections 15:10 and 15:15).

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. DOS Version 3 or 4 supports program relocation only at link-edit time. MFT supports 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 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. (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 MFT and DOS Version 3 or 4 environments in order to use real storage effectively can 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, 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 4331 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.

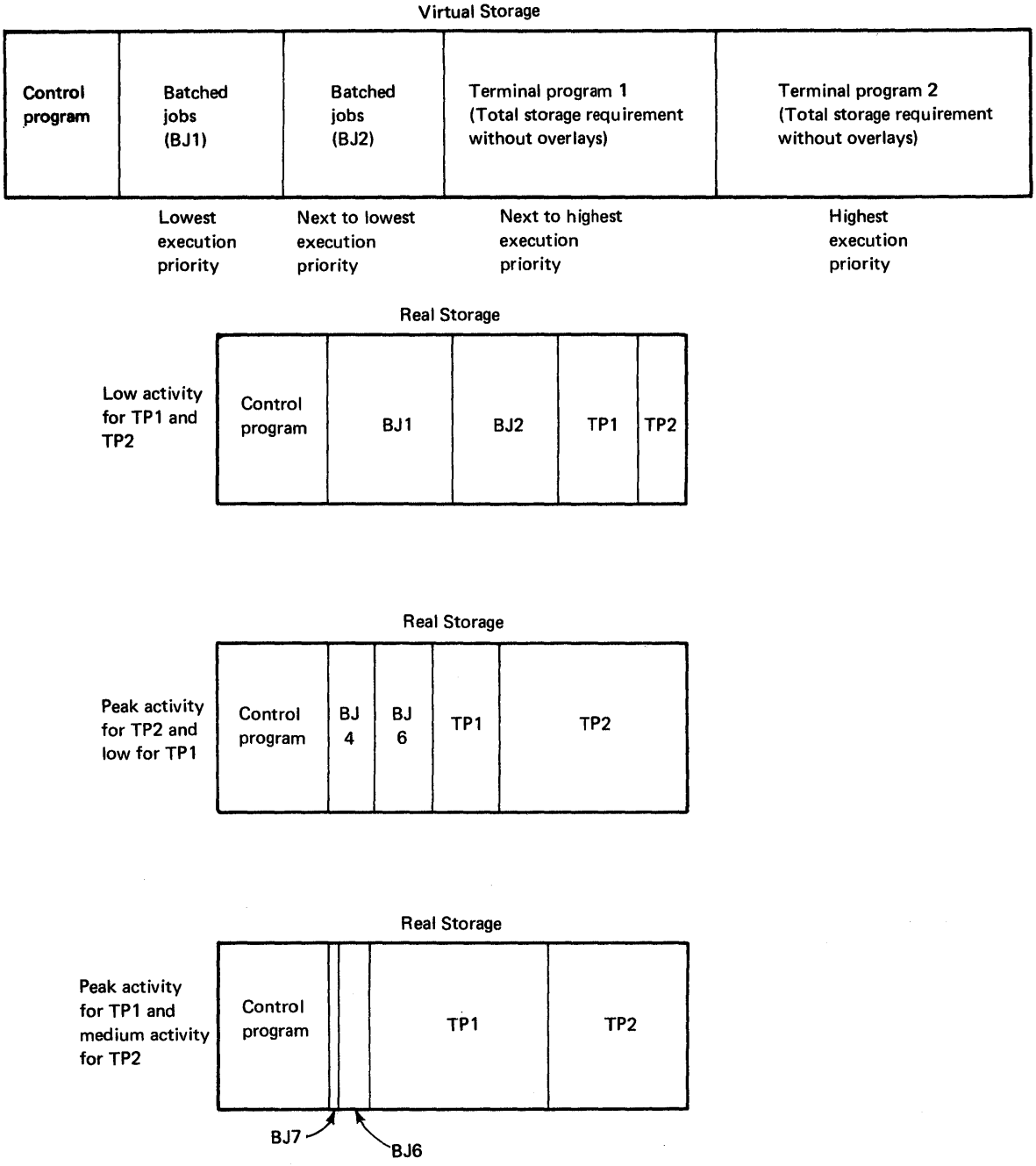


Figure 15.05.3. Conceptual illustration of real storage utilization in a mixed batch and online virtual storage environment

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).

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 4331 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 4331 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.

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 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.

For System/370 mode in the 4331 Processor, the processor storage directory is utilized to convert the real storage addresses in the segment and page tables and the real storage address obtained from the translation process to program processor storage addresses (see discussion in Section 15:10).

The internal mapping function provided for ECPS:VSE mode uses one table, called the processor storage directory, to perform virtual storage address to processor storage address translation. This directory indicates which pages are currently in processor storage and the processor storage addresses of these pages. The size of the directory indicates the amount of virtual storage defined by the operator. A program cannot directly address the processor storage directory. However, instructions provided for page handling in ECPS:VSE mode cause the processor storage directory 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 DOS (Versions 3 and 4) environments, real storage is, in effect, still allocated on a priority basis.

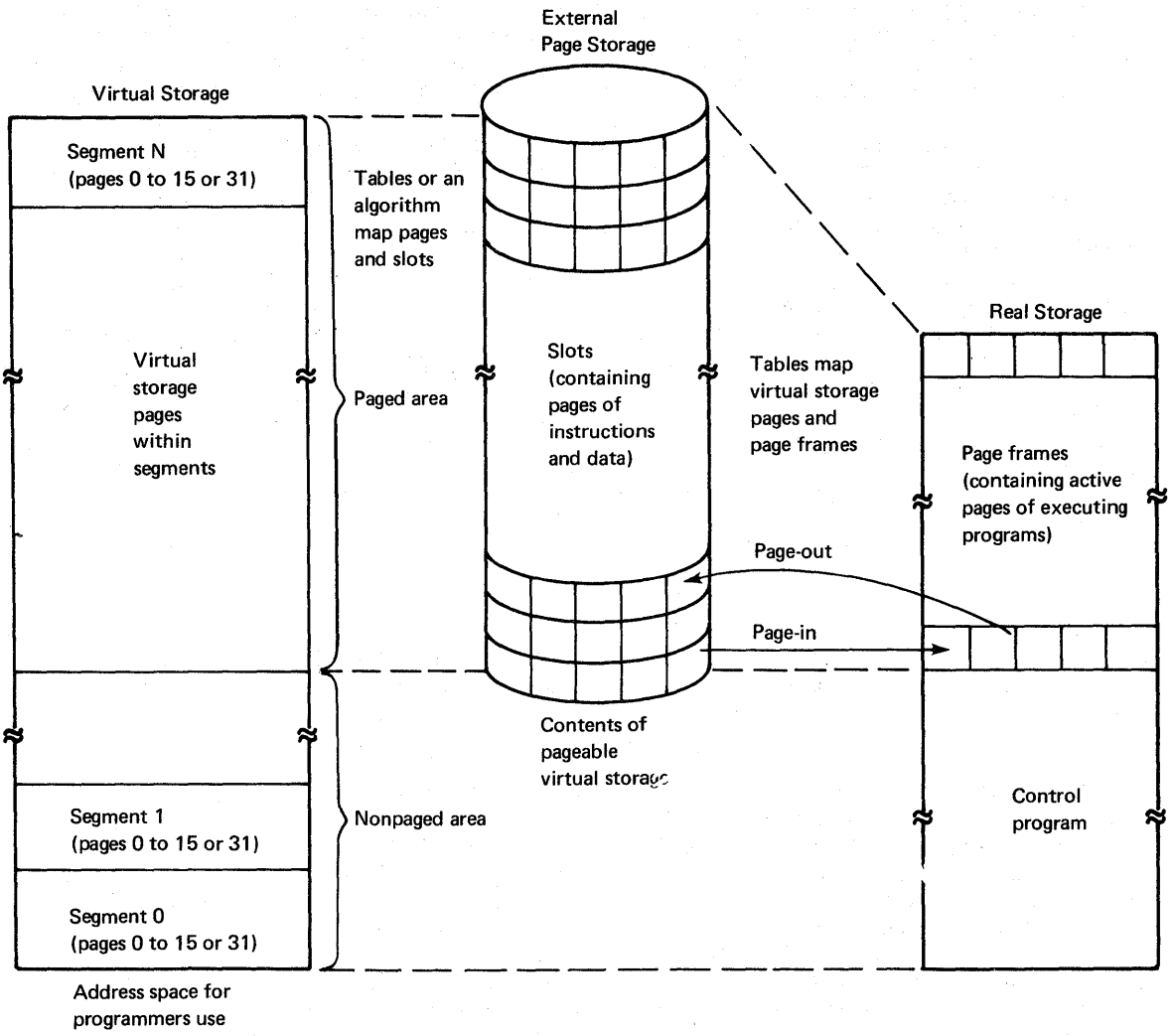


Figure 15.05.4. Layout of virtual storage, external page storage, and real storage

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 required also 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 processor storage assigned (a page that is not addressable). The processor storage directory 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 4331 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 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 4331 PROCESSOR OPERATING IN SYSTEM/370 MODE

When the 4331 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 4331 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 4331 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 4331 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 4331 Processor configuration 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 requirements for microcode, the processor storage directory, etc., are subtracted).

For System/370 processors with dynamic address translation hardware, there are conceptually two levels of storage: virtual and real. The size of real storage is identical to the size of the processor storage installed and there is no differentiation between a real storage address and a processor storage address (real and processor storage addresses are identical). However, for the 4331 Processor operating in System/370 mode, there are conceptually three levels of storage: virtual, real, and processor. The additional level exists because all the processor storage present in a given 4331 Processor is not available for program use.

For a 4331 Processor operating in System/370 mode, the size of real storage is equal to the size of the processor storage available for programming and real storage consists of a contiguous set of addresses beginning with address 0. However, a real storage address is not the same as a processor storage address, since the contiguous set of real storage addresses available are mapped to the program sections in processor storage that are not contiguous and do not begin at processor storage address 0.

The mapping of real storage addresses to processor storage addresses is reflected in the processor storage directory. Thus, for System/370 mode, each real storage address must be converted to a processor storage address, using the processor storage directory, before an actual reference to a processor storage location is made. A contiguous set of real storage addresses is required for System/370 mode because of the way in which the table lookup procedure used for address translation operates.

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 4331 Processor reside in program processor 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 4331 Processor is identical to the process used in System/370 except that references to the processor storage directory are required during the translation process in the 4331 Processor.

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. The required real storage address from the segment table must be converted to a processor storage address during the translation process, using the processor storage directory.

There is one page table for each segment in the virtual storage defined in a DOS/VSE and 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.

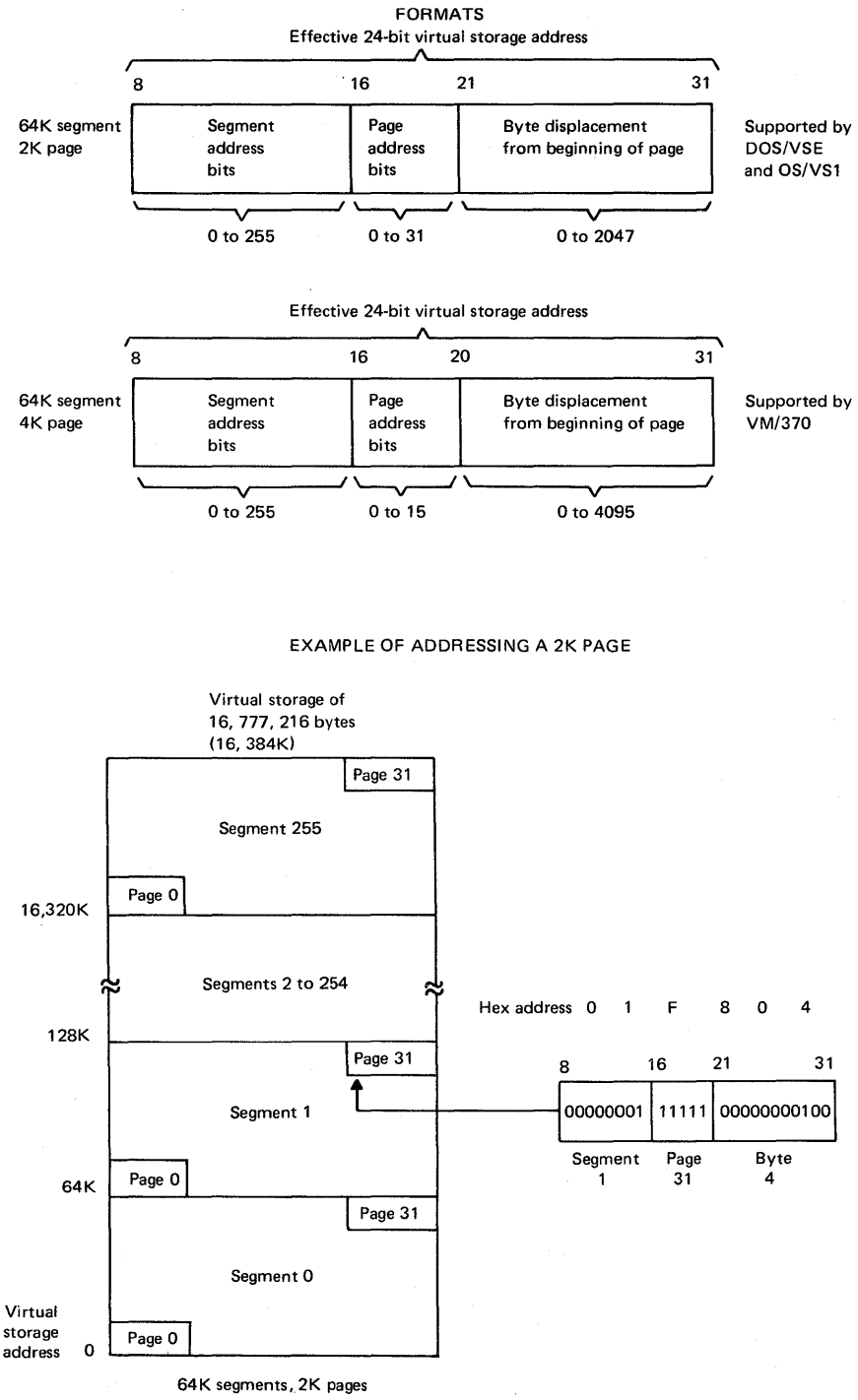


Figure 15.10.1. Virtual storage address fields for a 64K segment

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. The real storage address in a page table entry is used to reference the processor storage directory to obtain the appropriate processor storage address.

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.

Segment and page table formats and entries used for address translation are shown in Figure 15.10.2. 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.

For System/370 mode, the processor storage directory is 2K bytes for processor storage sizes of 512K and 1024K. Each four-byte entry in the directory represents a 2K page frame of real storage. The first directory entry is associated with page frame 0, the second entry with page frame 1, etc.

During an IPL procedure in which the 4331 Processor is initialized for System/370 mode of operation, the entries in the processor storage directory are set to point to the location of page frames in the program sections of processor storage. That is, each directory entry contains the high-order 13-bits of the processor storage address of the 2K block of storage in a program processor storage section that is assigned to the page frame associated with the directory entry.

Each entry also contains the access control key, fetch protection bit, and reference and change bits (discussed later) for the 2K block of program processor storage whose address it contains. Additional bits in each directory entry are utilized primarily for ECPS:VSE mode operations. The exact layout of a processor storage directory entry is described later in Section 15:15.

Figure 15.10.3 shows the three levels of storage for System/370 mode. It illustrates the way in which real storage is mapped to the processor storage available for programs via the processor storage directory when System/370 mode, EC mode, and DAT are enabled.

A specific virtual storage size (eight megabytes), processor storage size (1024K), mode (System/370), processor storage directory size (2K), and reserved processor storage requirement (188K) are used in Figure 15.10.3 so that actual page, page frame, and processor storage block numbers can be used. This is the configuration described in Section 10:10. The program processor storage available in the example is 836K, which is also the real storage size.

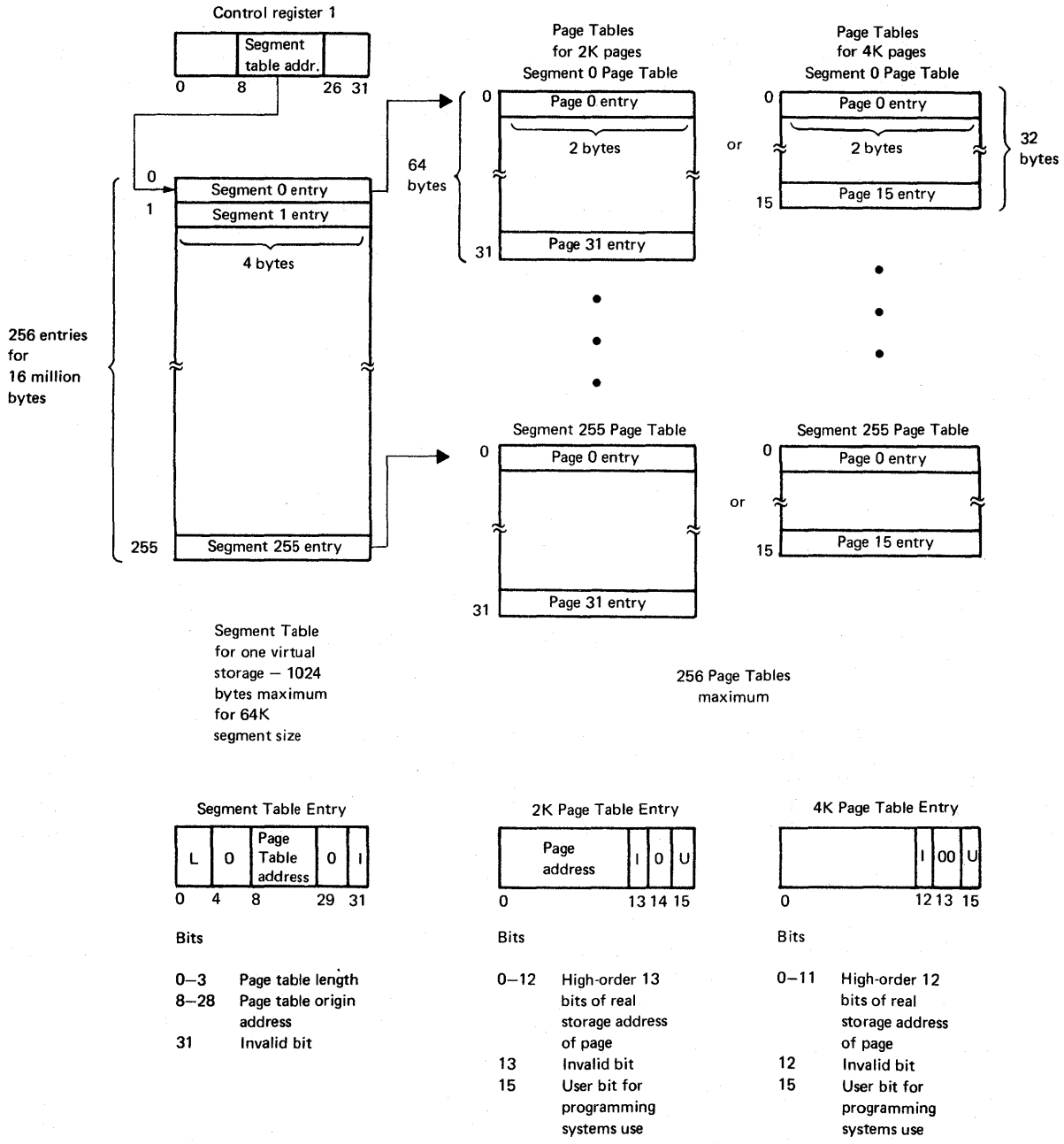


Figure 15.10.2. Segment table and page tables used for dynamic address translation

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. For System/370 mode operations, the processor storage directory is initialized by hardware

during IPL to reflect the mapping of real storage to the processor storage available for programming use. This mapping is not changed during processor operation.

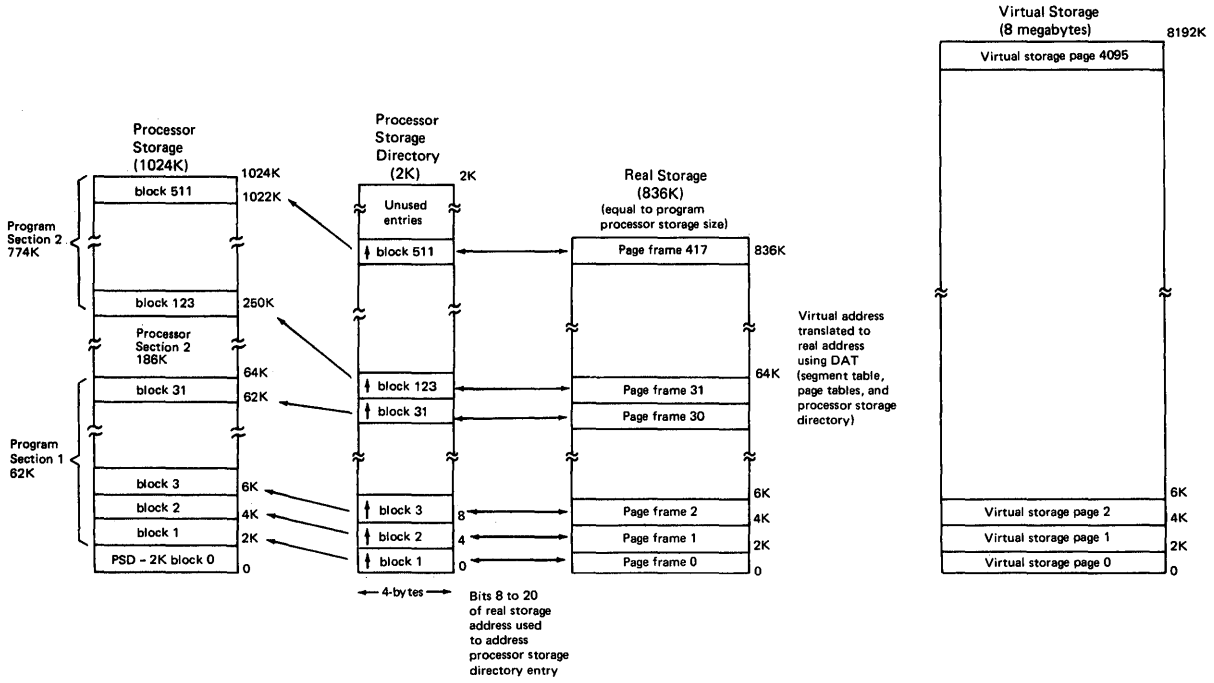


Figure 15.10.3. Storage levels and real-to-processor storage mapping for System/370 mode in the 4331 Processor

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 15.10.4. The procedure consists of a two-level, direct address table lookup operation (which includes two references to the processor storage directory). This process produces a real storage address, followed by a processor storage directory reference that converts the translated address to a processor 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.

The time required for address translation utilizing the DAT hardware in System/370 mode (including the table lookup procedure and the processor storage directory references) is 29 microseconds. This translation time is eliminated if the translation lookaside buffer can be used for the translation. (See discussion below.)

The details of the translation process are as follows (refer to Figure 15.10.4):

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.

3. The 13 high-order bits of the calculated real storage address are used to address a processor storage directory entry. The 13 processor storage address bits in the directory entry replace the 13 high-order bits of the calculated segment table (real storage) address. This processor storage address is used to access the segment table entry. If the invalid bit is on in the selected entry, a segment translation exception is indicated.

4. 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. 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.

5. The 13 high-order bits of the calculated real storage address are used to address a processor storage directory entry. The 13 processor storage bits in the directory entry replace the 13 high-order bits of the calculated page table (real storage) address. This processor storage address is used to access the page table entry. If the invalid bit is on in this entry, a page translation exception is indicated.

6. 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 high-order 13 bits of this real storage address are used to address a processor storage directory entry. The 13 processor storage address bits in the directory entry replace the 13 high-order bits of the translated real storage address. The resulting processor storage address is used to access program processor storage.

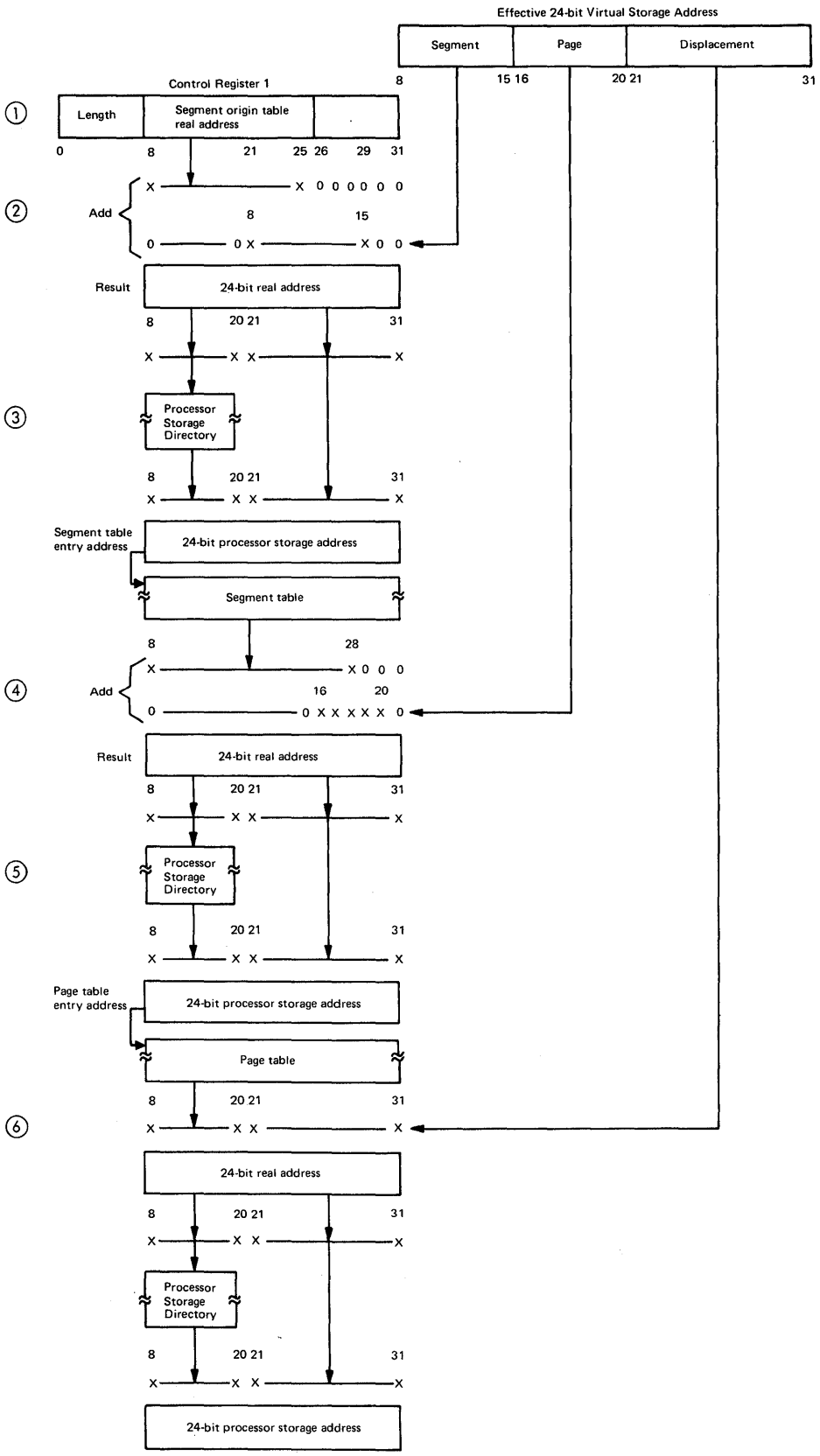


Figure 15.10.4. Dynamic address translation procedure for System/370 mode

Note that the processor storage directory is also utilized during System/370 mode operations when BC mode is in effect or EC mode is in effect without DAT enabled. For these two cases, the addresses specified in a program are real storage addresses and are used to address the processor storage directory to obtain the location of the addressed 2K page frame. Translation of a real storage address to a processor storage address, using the processor storage directory, requires 10 microseconds. This translation time is eliminated if the translation lookaside buffer can be used for the translation.

Translation Lookaside Buffer

When the 4331 Processor operates in System/370 mode with DAT specified, additional processor time is required to perform address translation using the segment and page tables and processor storage directory. Thus, a 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 4331 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, to avoid the time required to reference the processor storage directory when possible.

As shown in Figure 15.10.5, the TLB contains 32 rows and 2 columns. Each row contains two entries (one per column). Each entry contains one address translation. An entry has a pair of registers, one for a virtual storage address and one for a processor storage address. A virtual storage address register contains the 13 high-order bits of a 24-bit virtual storage address, while the associated processor storage address register contains the 13 high-order bits of the program processor storage location that contains the page frame allocated to the virtual storage page.

A validity bit is associated with the processor storage address register to indicate whether the entry contains a valid translation. 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.

Every time a virtual storage address is translated during instruction execution using the segment and page tables and processor storage directory, the processor storage address resulting from the translation procedure and the virtual storage address are placed in the TLB. The validity bit for the entry is turned on. Bits 17 to 20 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. The LRU bits are initialized to zero to indicate the right-hand column. 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.

At any given time, only half the TLB is actually utilized for address translation. The TLB is split into an upper and lower half of 16 rows each. A mode latch setting determines whether the upper or lower half is to be used. The latch is set based on the setting of certain mode control bits in the PSW. For System/370 mode, the TLB is used as follows:

- EC mode enabled and DAT not enabled - upper TLB half used
- EC mode and DAT enabled - lower TLB half used
- BC mode enabled and supervisor state in effect - upper half used
- BC mode enabled and problem state in effect - lower half used

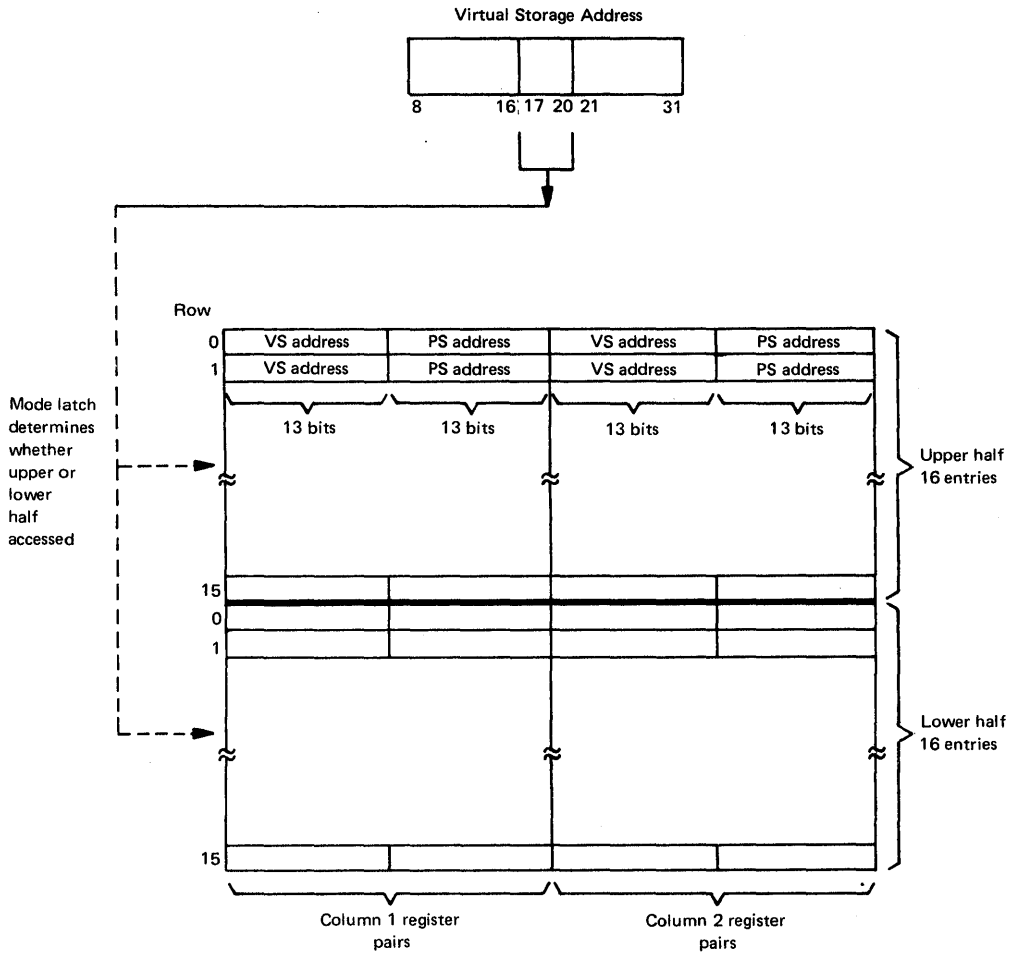


Figure 15.10.5. TLB for the 4331 Processor

The approach of using only half the TLB at a time eliminates the need to purge the entire TLB when a state change occurs. When a mode switch that affects the mode latch occurs (DAT is enabled or disabled or there is a switch between problem program and supervisor state), the latch setting is changed. No TLB purging occurs for a switch between problem and supervisor state. However, whenever DAT mode is enabled or disabled, the lower half of the TLB must be purged.

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 17 to 20 of the address in the instruction are used to select one of the 16 rows in the upper or lower TLB half, depending on the setting of the mode latch.

The address in the instruction is simultaneously compared with the addresess in the two virtual address registers in the selected row. If there is an equal comparison with either register and the valid bit for the processor storage register is on, the 13 processor storage address bits in the processor storage address register are combined with the 11 low-order bits of the address in the instruction to obtain the required 24-bit processor storage address. No further address translation processing occurs. When the required processor storage address can be obtained from the TLB, no address translation time is added to the time required for execution of the instruction.

If an equal comparison does not occur when the instruction address is compared with the two virtual storage registers in the addressed TLB row, the full translation process, using segment and page tables, and the processor storage directory is performed, as previously described, to obtain a processor storage address. The 13 high-order bits of the virtual storage address in the instruction and the 13 processor 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 translation, no address translation time is added to instruction processing time. If the TLB does not contain the required processor storage address, the processor storage directory is then referenced and the processor storage address obtained and its corresponding real storage address are placed in the TLB.

All virtual and processor storage registers in the TLB are purged (set to zero) during system reset, when a switch between BC and EC mode occurs, and whenever a LOAD CONTROL for control register 0 or 1 is issued during System/370 mode operations, regardless of whether DAT is operative. Thus, no TLB match can occur until new values are loaded as a result of the translation process.

The PURGE TLB instruction provides the capability of clearing all virtual and processor storage TLB registers in the currently active TLB half during processing, 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.)

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 4331 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 and the processor storage directory. Table 15.10.2 also indicates those storage addresses that are real or not used to reference processor storage. The real addresses must be converted to processor storage addresses using the TLB or processor storage directory. 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 4331 Processor operating in System/370 mode with DAT enabled

Virtual Storage Addresses Explicitly Designated by the Program (translated using the segment table, page tables, and processor storage directory)

- 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 (converted using the processor storage directory)

- 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 (converted using the processor storage directory)

- 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
- CCW address in the CSW

Reference and Change Recording Facility for Processor Storage Blocks

A hardware recording facility is standard in the 4331 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 4331 Processor has four access control bits (for store protection), one fetch protection bit, one reference bit, and one change bit. In the 4331 Processor, the storage keys are located in the instruction processing function and a copy of the key settings is maintained in the processor storage directory.

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/adapter 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. Testing is accomplished in the 4331 Processor by additional microcode routines that are executed before normal instruction execution microcode.

Instructions that do not need pretesting are those whose operation is such that when the operands they reference are on integral storage boundaries that are a multiple of the implied operand length, only one page can be involved. For example, a store halfword instruction that addresses a two-byte data field aligned on a halfword boundary cannot cross a page boundary during execution. The aligned data will always be totally contained in one page. This instruction is allowed to execute without pretesting as soon as it has been determined that the addressed data field is on an integral boundary.

Similarly, if a store halfword instruction addresses a two-byte field that is not on a halfword boundary, a pretest is required to determine whether the two bytes are actually in processor storage. The pretest microcode for this instruction issues a fetch to the highest addressed byte in the two-byte data field (virtual storage address in the instruction plus 1). The absence of a page translation exception during translation of the virtual storage address indicates that (1) if the data field spans two pages, at least the second of the two pages is present in processor storage or (2) the data field is totally contained in one page, which is present in processor storage. Hence, the instruction is allowed to proceed without nullification.

If the data field actually does span two pages and the first page is not in processor storage, this fact will be indicated by a page fault during translation of the address of the high-order byte of the field. Instruction nullification will occur and will cause a page-in of the first page to be initiated by the control program as usual.

If the pretest fetch operation does cause a translation exception, the store halfword instruction is nullified and the control program gains control to load the missing page. Once again, the page-in caused by the pretest may have brought in the second of two pages spanned by the data field or the only page containing the data field. After the page-in, the instruction is reexecuted.

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 processor 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, block multiplexer channel, and all I/O adapters. As shown in Figure 15.10.6, 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.

Note that for all System/370 mode operations (BC mode and EC mode with or without DAT enabled), the channel programs specified in START I/O instructions contain real storage addresses. Thus, the processor storage directory is always referenced during channel program execution in System/370 mode to convert the real storage addresses in CCWs (and IDAWs if present) to processor 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 15.10.6 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.

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 real storage address in the CCW is translated to a processor storage address, using the processor storage directory, and the I/O operation is started to this address.

If bit 37 in the first CCW is a one, the real storage address in the CCW is translated to a processor storage address, using the processor storage directory, and the first IDAW is fetched from this address. The real storage address in the first IDAW is translated to a processor storage address, utilizing the processor storage directory, and the I/O operation is begun using this address. Assuming that the I/O operation is not a read backward, ascending processor 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 (which must be translated to a processor storage address), and the data transfer operation continues. The channel continues using the IDAL until the operation indicated by

the CCW completes (CCW count reaches zero, IRG 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.

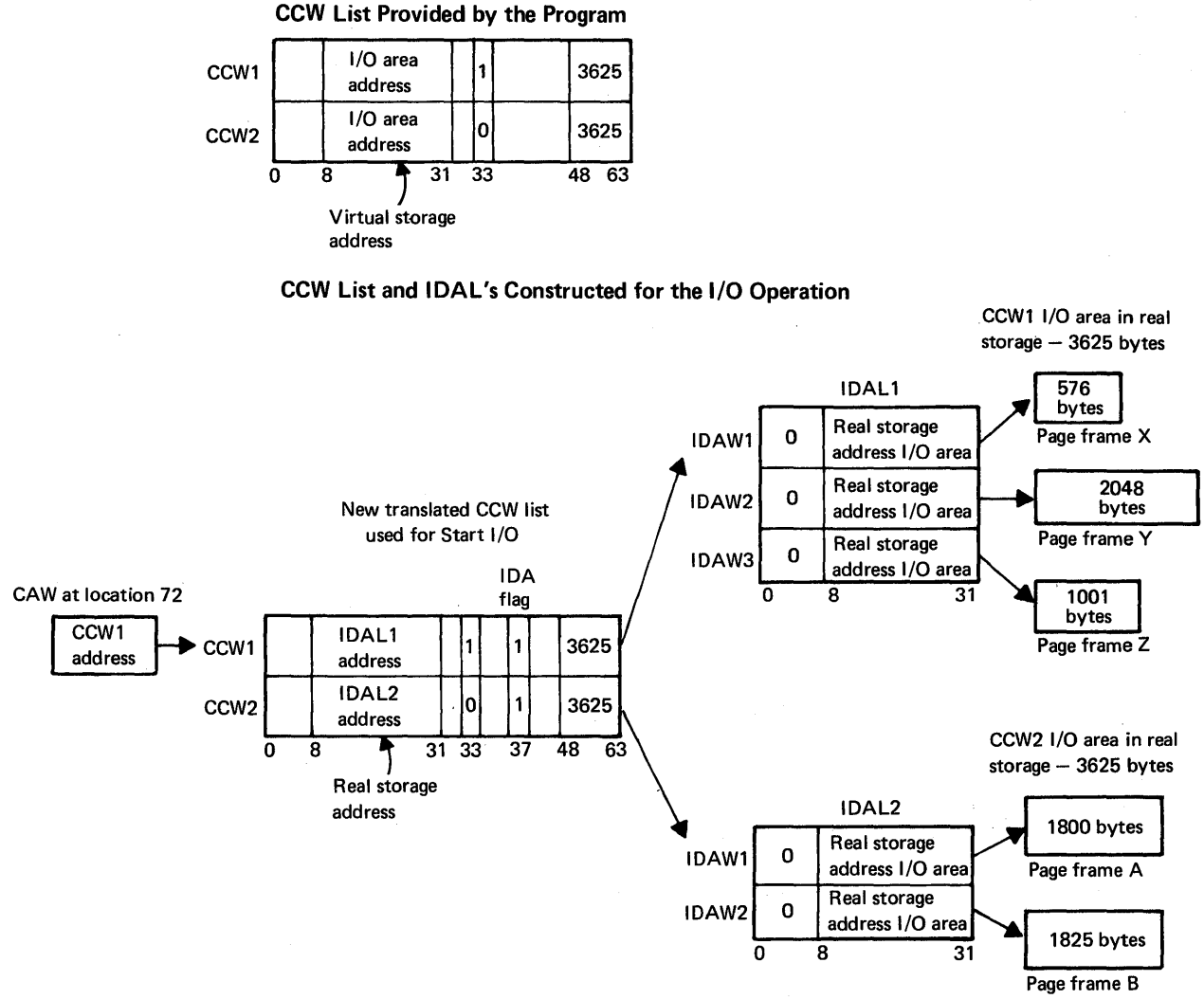


Figure 15.10.6. Example of IDALs required for a CCW list when page size is 2K

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 processor 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 4331 PROCESSOR OPERATING IN ECPS:VSE MODE

The address translation facility for ECPS:VSE mode is an internal mapping function that utilizes the processor storage directory. 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 (1) virtual storage addresses in instructions and the instruction counter to processor storage addresses during program execution and (2) virtual storage addresses in CCW lists to processor storage addresses during channel program operation. 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 LOAD REAL ADDRESS and PURGE TLB instructions are not valid for ECPS:VSE mode. 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

The maximum amount of virtual storage that can be supported in the 4331 Processor when ECPS:VSE mode is in effect is 16,777,216 bytes. The actual amount to be supported for a given IPL is controlled by the operator using the operator console (program load display). There is no programmed way to establish virtual storage size for ECPS:VSE mode.

The amount of processor storage required by the processor storage directory depends on the virtual storage size specified, and more program processor storage is available when a size other than the maximum of 16 million bytes is specified. The virtual storage size set should be a function of the amount of virtual storage the operating system to be used is generated to support.

During IPL, the virtual storage size specified by the operator for the IPL, if any, is established and written to the system diskette. If no specification is given, the virtual storage size specified for a previous IPL, if any (saved on the system diskette), is established.

The only virtual storage sizes supported for ECPS:VSE mode are 1, 2, 4, 8, and 16 megabytes and program processor storage size. Setting the virtual storage size equal to the size of program processor storage enables an operating system that does not support ECPS:VSE mode or use dynamic address translation hardware to operate with ECPS:VSE mode in effect (in BC or EC mode).

The virtual storage defined is divided into 2K-byte virtual storage pages that are located on 2K-byte address boundaries. These pages are addressed 0 to 8191 maximum. 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.

When ECPS:VSE mode is in effect in the 4331 Processor, there are conceptually, only two levels of storage: virtual and processor. For this mode, all addresses in all programs are considered to be virtual storage addresses and program processor storage cannot be directly addressed (in either BC or EC mode). The processor storage directory is utilized to translate a virtual storage address directly to a processor storage address. A contiguous set of real storage addresses (which is required for translation using segment and page tables) is not required.

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 are 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/adapters.

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 processor storage directory). A page state 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 or I/O adapter attempts to access a disconnected page.

A connected virtual storage page has a page frame assigned. It can be accessed by the channels/adapters, 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/adapters. 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 4331 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 an available 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 connected state. If a page frame is available, 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 available 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 processor storage directory 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 4331 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 processor storage address of the assigned page frame and is the frame index of the 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. Each count is a 16-bit unsigned binary integer that is initialized during IML and updated by the processor during processor operation in ECPS:VSE mode. These four counts are located in reserved processor storage (processor section 2) in the 4331 Processor. The four counts can be placed in processor 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 the following:

- Page capacity count (PCC), which is the number of virtual storage pages in the virtual storage size defined for this IML. This count is set during IML and also IPL if the virtual storage size is changed during IPL.

- 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 4331 Processor, the AFCC is set to the same value as the EFCC. Since the 4331 Processor stops after a double- or mutliple-bit processor storage error and the operating system does not receive control, the DECONFIGURE PAGE instruction is not issued during 4331 Processor operation and the AFCC count does not change.

- 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. The PCC is set equal to the AFCC when the operator, using the operator console (program load display), sets the virtual storage size to equal the available processor storage size.

The four capacity counts are also established as part of System/370 mode initialization procedures. The PCC, EFCC, and AFCC are set equal to the number of page frames in program processor storage. The FFCC is set to zero. The capacity units are not changed by the processor during processor operation and cannot be placed in program processor storage by programming while System/370 mode is in effect.

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 available for allocation to virtual storage pages. This list is contained in the available page frame stack in the processor storage reserved for processor use (processor section 2). The available page frame stack consists of one halfword entry location per page frame of program processor storage, the number of entry locations in the stack is thus equal to the AFCC less one (for the page frame assigned to page 0).

Each halfword in the available page frame stack can contain the frame index of a page frame that is available for assignment. After processor initialization, the available page frame stack is empty, since all program processor storage is allocated to the first N virtual storage pages. Frame indexes are added to and deleted from the stack as DISCONNECT PAGE and CONNECT PAGE instructions are issued during processing.

A first-in, first-out queuing technique is used for the available page frame stack. A pointer is maintained in data local storage that indicates the next entry in the stack to be used. It points to the next available entry location in the stack after the last entry containing the frame index of an available page frame. The pointer is initialized to indicate the first entry location.

When a frame index is added to the stack, as a result of the execution of a DISCONNECT PAGE instruction, it is placed in the entry

location indicated by the pointer and the pointer value is incremented by two. When a CONNECT PAGE instruction is issued, the pointer value is decremented by two. The frame index in the entry indicated by the decremented pointer value is used for the connection and, in effect, is removed from the available page frame stack.

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 processor storage directory indicates the referenced page is not in the addressable state. The CONNECT PAGE instruction should then be issued to cause the assignment of an available page frame.

If the available page frame stack is not empty (FFCC is not zero), the CONNECT PAGE instruction causes the instruction processing function to subtract two from the available page frame stack pointer and assign the available 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 (disconnect bit in the associated processor storage directory entry is turned off). 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 available 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 frame index of the page frame the virtual storage page was assigned is placed in the available page frame stack and the FFCC is increased by one. The page frame is not cleared by the processor.

The preceding procedure makes a page frame available 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.

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

Processor Storage Directory

The processor storage directory is used in ECPS:VSE mode for translation of virtual storage addresses in instructions and channel programs. For ECPS:VSE mode, there is one four-byte entry in the processor storage directory for each virtual storage page in the defined virtual storage. Thus, the size of the processor storage directory is 2, 4, 8, 16, or 32K for virtual storage sizes of 1, 2, 4, 8, or 16

megabytes, respectively. When the operator sets the virtual storage size equal to the program processor storage size, the directory is 2K bytes for processor storage sizes of 512K and 1024K.

A 32-bit processor storage directory entry contains the following:

- Addressable bit (0). This bit indicates whether the virtual storage page associated with this entry is addressable (bit is 0) or unaddressable (bit is 1). 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.

- Disconnect bit (1). This bit indicates whether the virtual storage page associated with this entry is connected (bit is 0) or disconnected (bit is 1). This bit is set by microcode when CONNECT PAGE and DISCONNECT PAGE instructions are issued. When a page is addressable, its disconnect bit is zero to indicate it is also connected.

- Compare bit (2). This bit is one when an address compare stop has been established for this virtual storage page.

- Programmable bits (3, 4, and 5). 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.

- Size-exceeded bit (6). This bit is used to identify unused entries at the end of the processor storage directory that are associated with invalid program processor storage addresses. When System/370 mode is in effect, the directory is always 2K bytes in size. Thus, for a 1024K-byte processor there are always unused entries that are associated with 2K processor storage blocks that are not available for program use. For a 512K-byte processor, entries for uninstalled, as well as unavailable, processor storage blocks exist and must be marked unused. For ECPS:VSE mode, unused entries occur when the virtual storage size is set equal to the program processor storage size (since the processor storage directory is 2K bytes in size).

- Page frame address bits (8 to 20). This address is the frame index 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, 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 available page frames in program processor storage to the first N virtual storage pages and initializes the processor storage directory as follows:

- All addressable bits and disconnect bits are set to zero 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 one.
- All programmable bits in all entries are set to zero.
- The size-exceeded bit is on for all entries associated with virtual storage pages that cannot be accessed (occurs only when virtual storage size is set equal to program processor storage size).
- 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.

When the virtual storage size is set to equal the program processor storage size, the above initialization ensures that all defined virtual storage pages have page frames assigned and no page access exceptions will occur during translation operations. If an address above the program processor storage size is specified, an addressing exception occurs. This initialization enables a control program that operates in BC mode (the default established during IPL) and does not support ECPS:VSE mode to execute with this mode in effect (such as a System/360 operating system).

Translation Process

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 processor storage directory. Bits 8 to 20 of the virtual storage address are the page address and are used to select the appropriate four-byte directory entry. The addressable bit is inspected. If it is off (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 six microseconds.

If the addressable bit is on, 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.

Address translation for channel programs is performed as follows for I/O operations that utilize cycle stealing. When the channel program is started, the microcode inspects the first (or only) CCW to determine whether a 2K boundary will be crossed while using the CCW. If so, the buffer can span two noncontiguous page frames. If a boundary will be crossed, the virtual storage address in the first CCW and the one that will address the second page frame are translated before the I/O operation is actually begun.

The first translation is accomplished using bits 8 to 20 of the virtual storage address in the CCW to access the appropriate processor storage directory entry. The disconnect bit is inspected. If it is off (page is connected), the 13 page frame address bits in the entry replace the 13 virtual storage bits from the CCW and will be used in the actual

I/O operation. This lookup procedure requires six microseconds. The second virtual storage address is then translated using the processor storage directory and held in the UCW to be used for the I/O operation. The I/O operation is physically started.

When the first 2K boundary is reached during operation of the first CCW, an internal pointer is set to point to the second translated address and a page frame boundary crossing internal interruption occurs. The page frame crossing internal interruption is enabled automatically by the processor for ECPS:VSE mode operations. The interruption is internal in that it is not reflected to any program (PSWs are not switched). The internal interruption causes the microcode to translate the third page frame address, if necessary. This translation operation is overlapped with cycle steal operations.

After the first 2K boundary is reached, the I/O operation continues, using the previously translated second address. When the second 2K boundary is reached, the internal pointer is set to point to the third translated address, the I/O operation continues, using the third translated address, and a page frame boundary crossing interruption occurs to cause translation of the next (fourth) page address, if necessary. The process continues until the I/O operation completes. This technique ensures that there are always two translated addresses available, one for the page being accessed and one for the next page to be accessed.

For I/O operations that use byte transfer instead of cycle steal operations, such as those involving the Communications Adapter, pretranslation of the virtual storage address following the first to be used is not done. When an internal interruption occurs as a result of the crossing of a 2K boundary in processor storage, the next required translation is done using the directory and the operation is then continued.

Translation Lookaside Buffer

The translation lookaside buffer is also utilized to speed up address translation during ECPS:VSE mode. However, it is used only for the translation of virtual storage addresses in instructions and not for the virtual storage addresses in 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 processor storage directory is accessed to perform the translation, as previously described.

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. 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 and if a switch occurs between BC and EC modes while ECPS:VSE mode is in effect. Thereafter, the full TLB or half the TLB is never purged during ECPS:VSE mode operations. Since address translation is always operative, TLB entries never become invalid as a result of switching between translation mode and nontranslation mode, as for System/370 mode.

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, when page replacement occurs and a page table entry is invalidated (since the invalidated entry could be in the TLB), and when DAT mode is enabled or disabled.

The utilization of the TLB in ECPS:VSE mode is the same as its utilization in System/370 mode with BC mode active. That is, when the current PSW indicates supervisor state, the upper half of the TLB is used. When problem state is in effect, the lower half is used. The method of addressing a TLB entry and use of the LRU bits are the same for ECPS:VSE and System/370 modes.

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 nonvirtual storage environments, and (3) provide performance equal to or better than that achieved with a nonvirtual storage operating system using the same system hardware configuration. Attainment of the last objective may not be possible for all 4331 Processor configurations.

In addition, some of the new functions a virtual storage environment provides cannot be achieved in a nonvirtual 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 4331 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 4331 Processor is affected by which address translation hardware is used and the number of times the address translation hardware procedure must be performed. The 4331 Processor has a 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 4331 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 TLB.

- For System/370 mode operations in the 4331 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 4331 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 (processor storage directory) 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 nonvirtual 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 4331 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 nonvirtual 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 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 4331 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 4331 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 job stream 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.

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 nonvirtual 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.

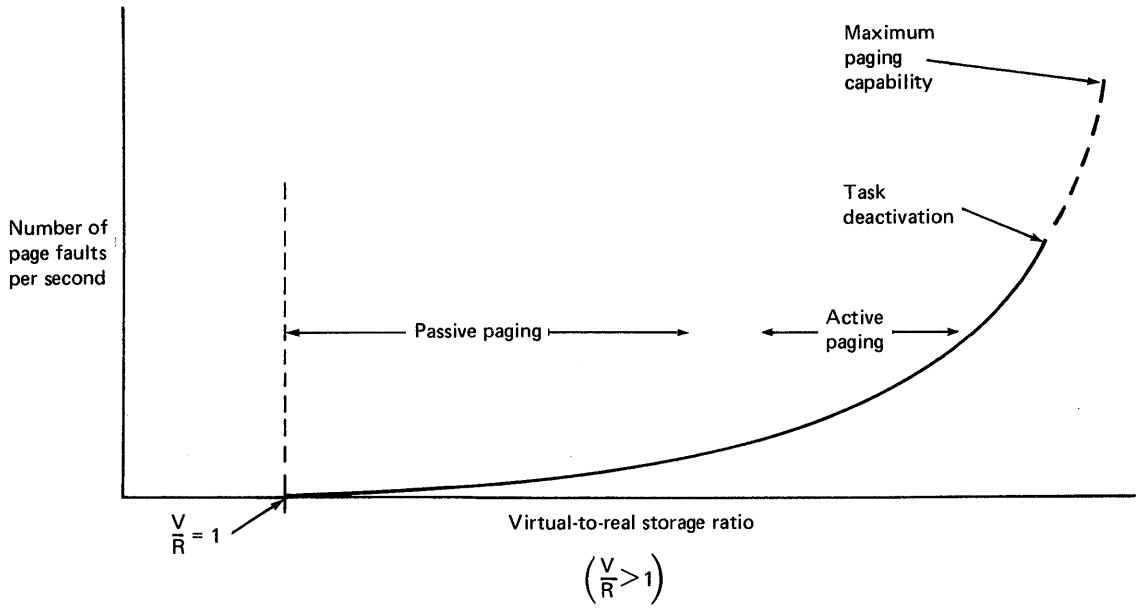


Figure 15.20.1. General effect on page faults of increasing the ratio of the virtual storage used to the real storage present in the system

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.

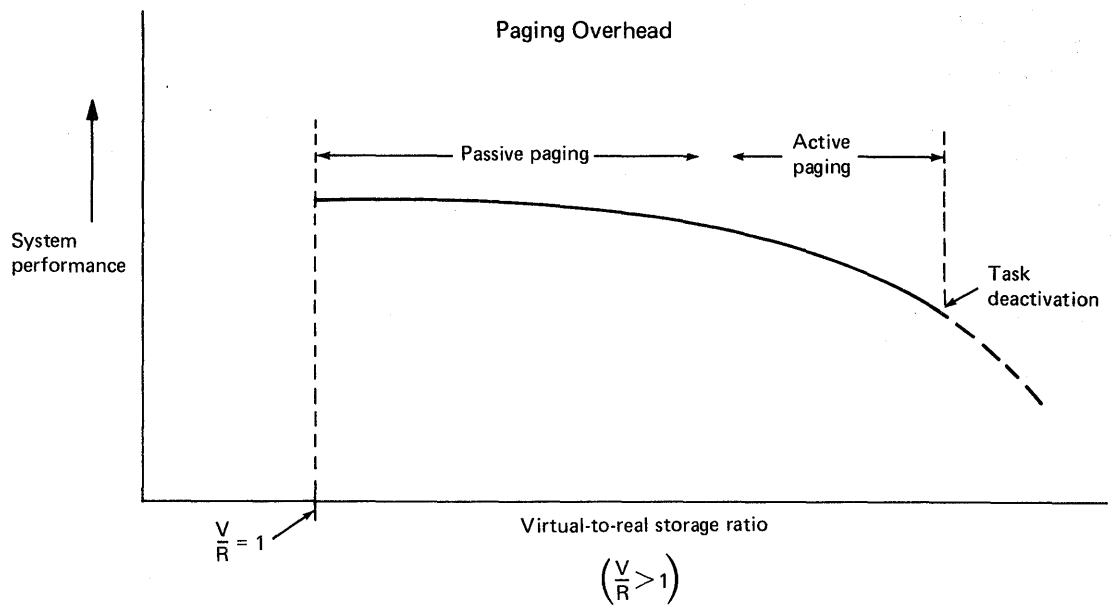


Figure 15.20.2. General effect on system performance of the paging factor only

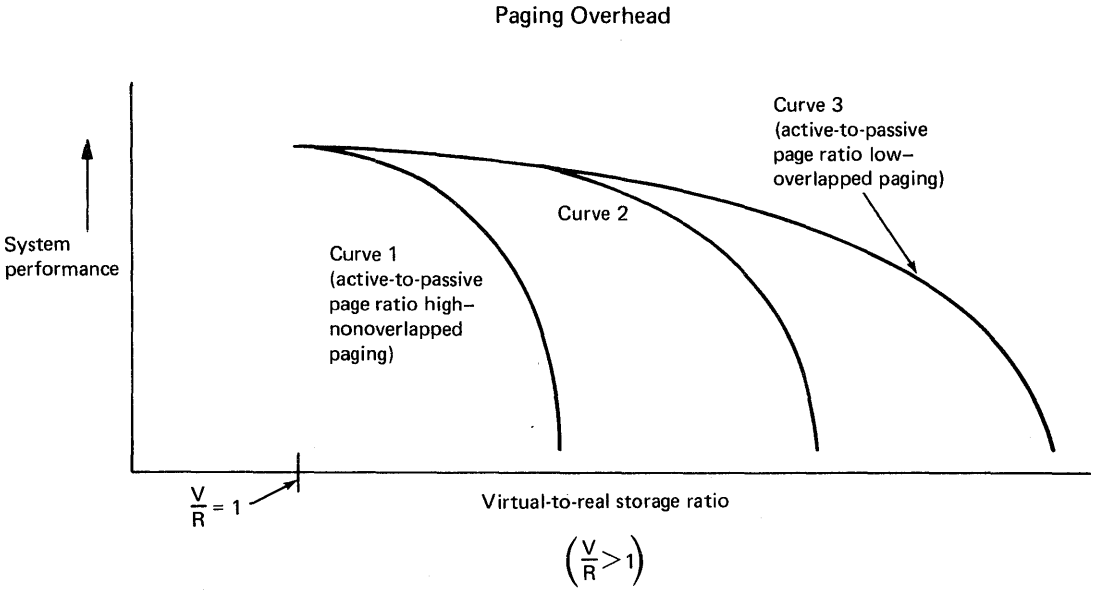


Figure 15.20.3. General effect of the paging factor on system performance for various active-to-passive page ratios

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.

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.

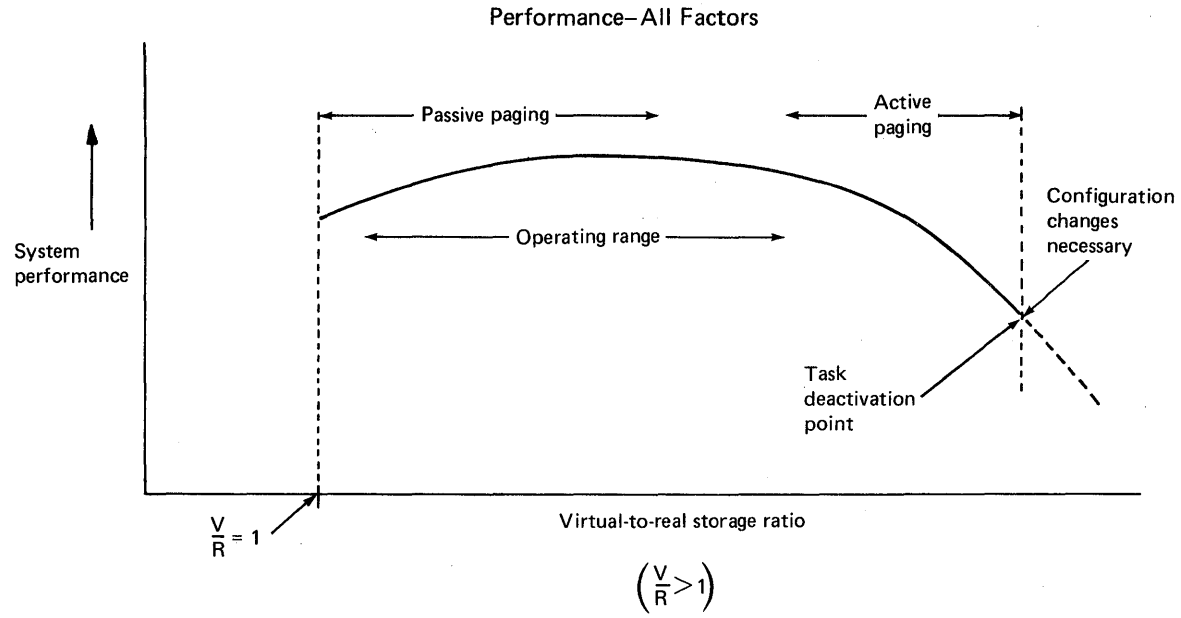


Figure 15.20.4. General system performance curve for a virtual storage environment

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 nonvirtual 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 4331 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 under DOS/VSE), (2) operating more partitions concurrently, or (3) increasing the use of multitasking (structuring an ACF/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 4331 or 4341 Processor or a 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 operating in System/370 mode, 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 4331 or 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.

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 4331 Processor instruction 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.

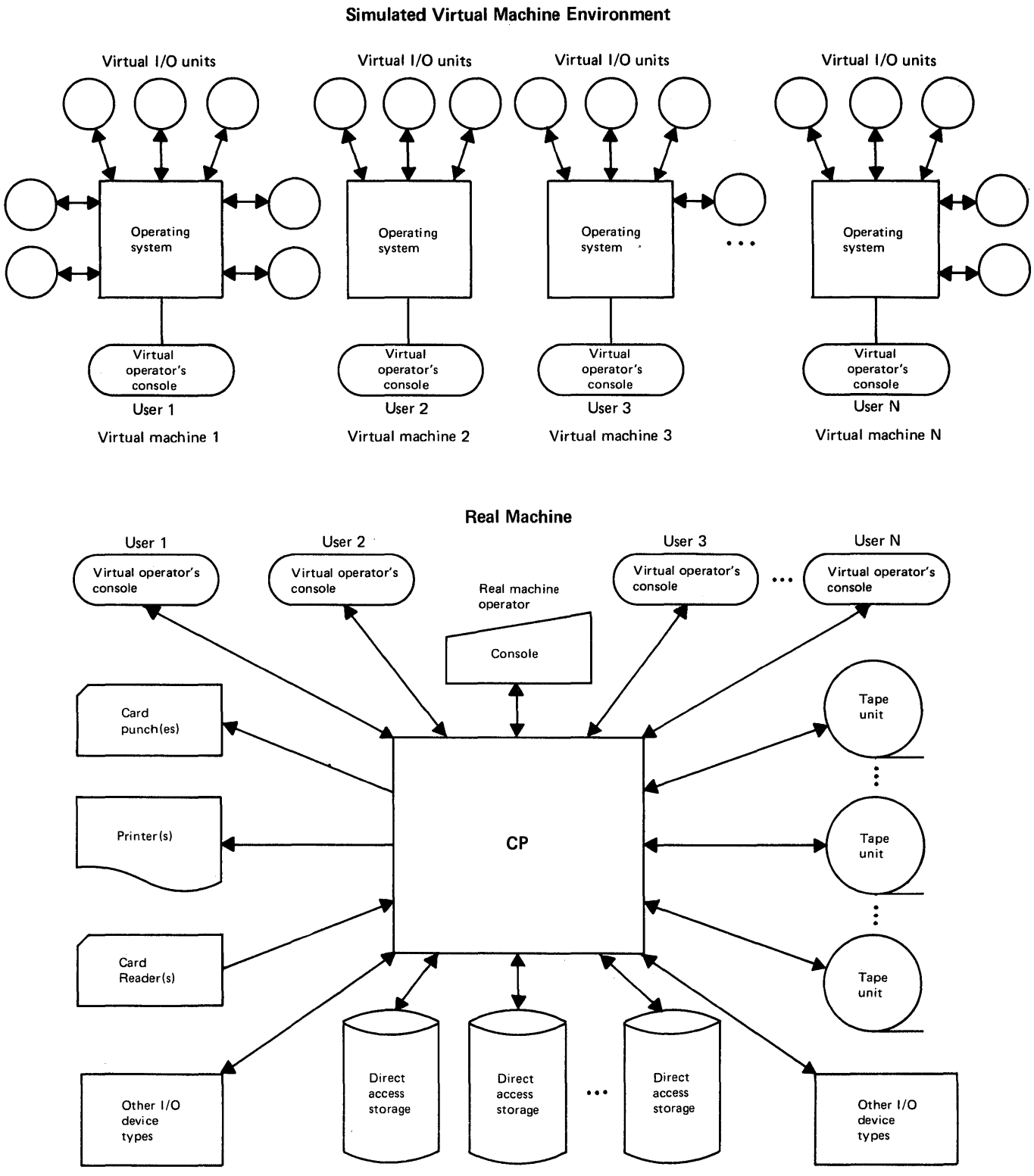


Figure 18.05.1. Conceptual illustration of the real and virtual machine environment that is supported by VM/370

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 maximum virtual storage size for the 4331 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).

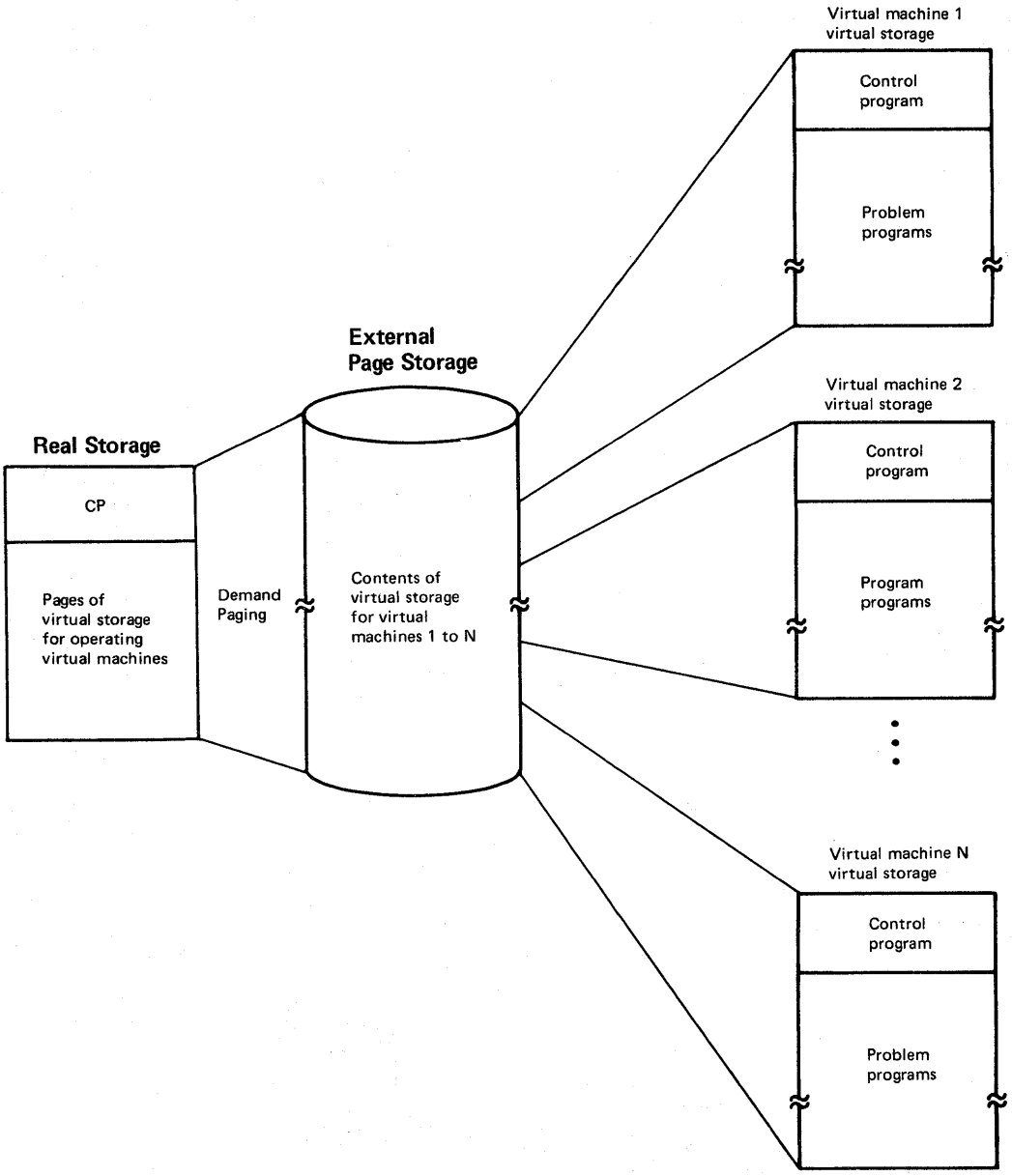


Figure 18.05.2. Conceptual illustration of the implementation of virtual storage in a virtual machine environment

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, 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.

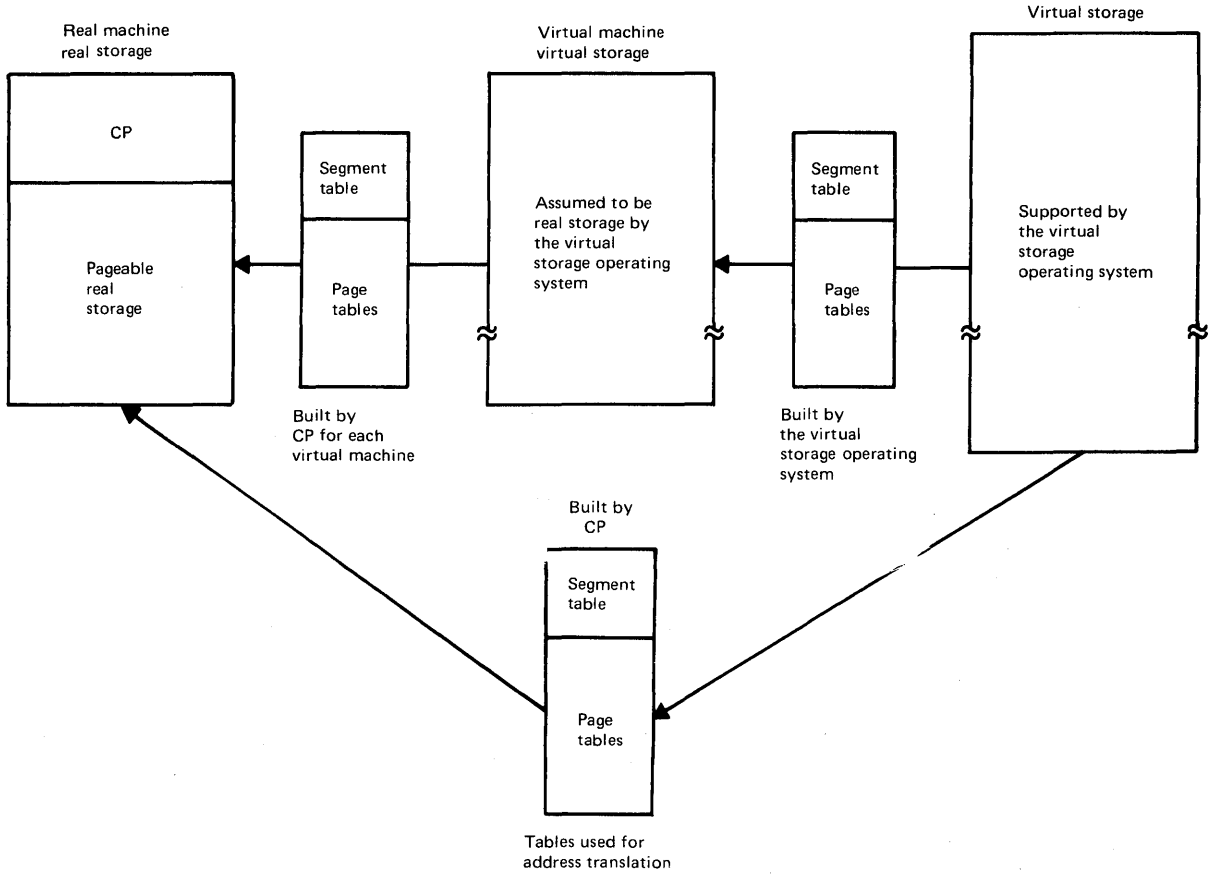


Figure 18.05.3. Segment tables and page tables built when a virtual storage operating system executes in a virtual machine

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. In the 4331 Processor, these real storage addresses are then translated into program processor storage addresses in the real machine using the processor storage directory, as discussed previously.

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 effecient 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 the POWER component of DOS, DOS/VS, and DOS/VSE, 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 FEATURE

General Description

The optional ECPS:VM/370 feature is designed to improve the performance of a 4331 Processor operating in System/370 mode under VM/370 control by providing for the execution of certain CP routines and functions in hardware to reduce the amount of processor time used by CP. With ECPS:VM/370 enabled, a reduction of 82 percent in supervisor processor-busy time was measured in a representative DOS/VSE Advanced Functions batch environment under VM/370 when compared to operation of the same workload and VM/370 control program without the use of ECPS:VM/370.

The ECPS:VM/370 feature for the 4331 Processor consists of the Virtual Machine Assist, Expanded Virtual Machine Assist, Control Program Assist, and Virtual Interval Timer Assist components. Operation of the four components of the ECPS:VM/370 feature is enabled and disabled using three bits (0, 6, and 7) in control register 6. System/370 mode must be in effect to enable any component of ECPS:VM/370.

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 system 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 Assist
1	1	1	Control Program, Virtual Machine, Expanded Virtual Machine, and Virtual Interval Timer Assist

Note that 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.

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.

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 as well as System/370 and 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
- DIAGNOSE

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 the VM/370 hardware assist function, 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 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 instructions. These Control Program Assist instructions are included in the CP of a VM/370 system that is generated to support the ECPS:VM/370 feature.

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 component is enabled and supervisor state is in effect.

A Control Program Assist instruction is executed as an NOP instruction if the ECPS:VM/370 feature is present in a 4331 Processor but the Control Program Assist component is disabled and supervisor state is in effect. This enables a VM/370 system that includes ECPS:VM/370 feature support to execute correctly in a 4331 Processor that has the ECPS:VM/370 feature when the feature is not enabled (because it is malfunctioning, for example). If a Control Program Assist instruction is issued in a 4331 Processor that does not have the ECPS:VM/370 feature installed, an operation exception program interruption condition occurs.

The five new instructions defined for the Control Program Assist function are six bytes in length, 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. In addition, two supervisor call instructions are defined for link and return purposes.

The following are the Control Program Assist instructions and the CP function they perform:

- SHARED PAGE - Obtain space from the free storage area
- SCNVU - Locate virtual I/O control blocks
- ZAPSEGS - Invalidate a segment table
- ZAPPAGE - Invalidate a page table
- STECPSVM - Store ECPS:VM/370 feature level identification. This instruction stores a value to indicate the architecture level of the ECPS:VM/370 feature. It is used by CP during IPL to determine whether ECPS:VM/370 is installed in the processor and, if so, the level of the support.
- LINK/RETURN - SVC instructions

Virtual Interval Timer Assist Component

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. Essentially, CP is a simulator. Traditionally, simulators have been used to provide a desired function at the expense of 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, enable 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 in a 4331 Processor 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 the possibility of 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 the possibility of 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 4331 Processor installations in which DOS, DOS/VS, DOS/VSE, or OS/VS1 is used as the primary programming system.

20:05 I/O DEVICE SUPPORT

Table 20.05.1 lists the I/O devices that can be attached to the 4331 Processor and indicates the possible methods of attachment: I/O adapter or byte and/or block multiplexer channel. Specific I/O configurations should be evaluated for the possibility of overrun.

Table 20.05.1. I/O devices attachable to the 4331 Processor

I/O Device	Attachment
Card Readers and Punches	
1442 Card Read Punch, Models N1 and N2	Byte, block
2501 Card Reader, Models B1 and B2	Byte, block
2520 Card Read Punch Model B1	Byte, block
2520 Card Punch, Models B2 and B3	Byte, block
2540 Card Read Punch	Byte, block
3505 Card Reader	Byte, block
3525 Card Punch	Byte, block
5424 Multi-Function Card Unit, Models A1 and A2	5424 Adapter
Printers	
1403 Printers, Models 2, 7, and N1	Byte, block
1443 Printer, Model N1	Byte, block
3203 Printer, Model 5	Byte, block
3211 Printer	Byte, block
3262 Line Printer, Model 1	Display/Printer Adapter
3287 Printer, Models 1 and 2	Display/Printer Adapter
3289 Line Printer, Model 4	Display/Printer Adapter
3800 Printing Subsystem	Byte, block
Paper Tape	
2671 Paper Tape Reader	Byte, block
Magnetic Tape	
2401 Magnetic Tape Unit, Models 1 to 6 and 8	Byte, block
2415 Magnetic Tape Unit and Control, Models 1 to 6	Byte, block
2420 Magnetic Tape Unit, Models 5 and 7	Block
3410 Magnetic Tape Unit, Models 1 to 3	Byte, block
3411 Magnetic Tape Unit and Control, Models 1 to 3	Byte, block
3420 Magnetic Tape Unit, Models 3, 5, and 7	Byte, block
3420 Magnetic Tape Unit Model 4	Block
8809 Magnetic Tape Unit	8809 Magnetic Tape Unit Adapter
Diskette	
3540 Diskette Input/Output Unit Diskette drive	Byte, block (Includes own adapter)

Table 20.05.1 (continued)

I/O Device	Attachment
Direct Access Storage	
2311 Disk Storage Drive	Block
2314 Disk Storage, Model A1	Block
2319 Disk Storage, Models B1 and B2	Block
3310 Direct Access Storage	DASD Adapter
3340 Direct Access Storage	DASD Adapter
3370 Direct Access Storage	DASD Adapter
Character Recognition	
1255 Magnetic Character Reader	Byte, block
1287 Optical Reader	Byte, block
1288 Optical Page Reader	Byte, block
1419 Magnetic Character Reader	Byte, block
3881 Optical Mark Reader	Byte
3886 Optical Character Reader	Byte, block
3890 Document Processor	Byte, block
3895 Document Reader/Inscriber	Byte, block
Communications Controllers	
2701 Data Adapter Unit	Byte, block
3704 Communications Controller	Byte
3705-I, 3705-II Communications Controllers	Byte,block
3791 Controller	Byte, block
Displays	
2250 Display Unit	Byte, block
3250 Graphics Display System	Block
3270 Information Display System	Byte, block
Audio Response	
7770 Audio Response Unit, Model 3	Byte

The following subsections discuss the following I/O devices for the 4331 Processor: 3310 Direct Access Storage, 3340 Direct Access Storage, the 3803/3420 Magnetic Tape Subsystem, the 8809 Magnetic Tape Unit, and the 3203 Model 5 Printer. The differences between and the advantages of the preceding devices and their predecessor devices are covered.

20.10 3310 DIRECT ACCESS STORAGE

The 3310 Direct Access Storage unit is low-cost, intermediate capacity, modular, high-performance direct access storage. The 3310 utilizes fixed block architecture for track formatting and attaches to the 4331 Processor only via the DASD Adapter. It has nonremovable disks that are accessed by movable heads.

The 3310 offers several advantages over 2314 and 3340 direct access storage, such as faster data transfer, lower cost, improved reliability, and enhanced physical characteristics. These advantages are discussed in detail at the end of this subsection.

STRING CONFIGURATIONS AND CAPACITIES

Four models of 3310 units are available: 3310 Disk Storage and Control Models A1 and A2 and 3310 Disk Storage Models B1 and B2. The 3310 Model A1 unit contains one drive and a controller function that provides power and device-oriented control functions for the drive. The Model A1 cannot have any Model B unit attached.

The 3310 Model A2 unit contains two drives and a controller function for the two drives and up to two additional drives. The Model A2 can have one Model B1 or one Model B2 attached. The 3310 Model B1 unit contains one drive and no controller while the 3310 Model B2 unit contains two drives and no controller.

A 3310 string can consist of from one to four disk drives, in one-drive increments. The string can consist of one 3310 unit or two 3310 units that are bolted together. The four drives in a 3310 string can have device addresses 0 to 3 or 4 to 7. The following are the only permissible 3310 string configurations for the 4331 Processor:

- Model A1 alone (for a single-drive string)
- Model A2 alone (for a two-drive string)
- Model A2 with a Model B1 attached (for a three-drive string)
- Model A2 with a Model B2 attached (for a four-drive string)

A 3310 Model A1 can be field upgraded to a Model A2. A 3310 Model B1 can be field upgraded to a Model B2. Model changes between Model A and B units cannot be made. A four-drive 3310 string is shown in Figure 20.10.1.

Each drive in a 3310 string has a capacity of 64,520,192 bytes (approximately 64.5 million bytes). Thus, a 3310 string has a maximum capacity of 258 million bytes while a 4331 Processor configuration with four full 3310 strings has 1032 million bytes of online disk storage.

A 3310 disk drive has a data rate of 1.031 MB/sec, average seek time of 27 milliseconds, maximum seek time of 46 milliseconds, and average rotational delay of 9.6 ms. Cylinder-to-cylinder seek time is 9 milliseconds and head-switching time between consecutively addressed heads is 4.8 milliseconds.

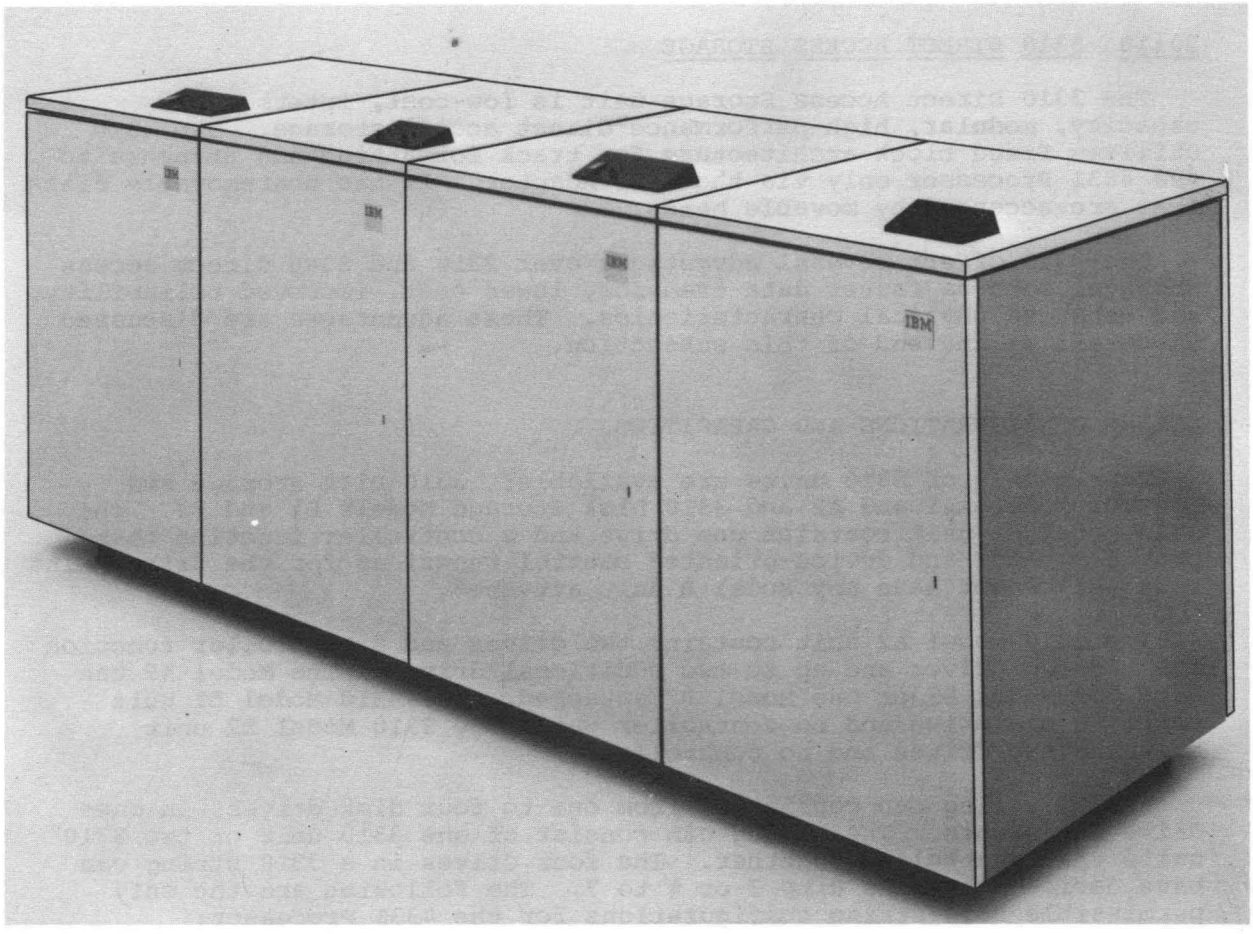


Figure 20.10.1. Two four-drive 3310 strings (design models)

POWERING

Each 3310 frame has an operator panel. For each drive present, this panel contains a start/stop switch and a ready indicator. The 3310 Model A2 operator panel contains an AC power on/off switch. With this switch in the on position, each drive can be controlled with its own start/stop switch. With the AC power switch in the off position, power is removed from all disk drives attached. The AC power on/off switch can be used as the emergency power-off switch for the 3310 string, since 3310 units cannot be powered on or off remotely from the 4331 Processor. The ready indicator for each drive indicates that power is on and that the drive is in the ready condition.

TRACK FORMATTING AND DATA ORGANIZATION

Each track in a 3310 drive is physically divided into fixed-length addressable blocks of equal size called sectors. A sector consists of an identification (ID) field that uniquely identifies the sector and a data field. The two fields are separated by a gap. A sector, therefore, corresponds to an addressable block as defined in fixed block architecture (see Section 10:25). The sectors on a given track are separated from each other by gaps and are numbered consecutively, beginning with sector number 0.

The ID field of a sector is four bytes in length. It contains (from left to right) one flag byte and three sector address bytes. Two cyclic redundancy check (CRC) bytes are appended to each ID field. The flag

byte indicates the type (primary or alternate) and condition (nondefective or defective with an alternate assigned, for example) of the sector.

The three sector address bytes contain the two-byte head and cylinder number and one-byte sector number required to access the sector. The CRC bytes are used for error detection within the ID field. The ID field in a sector is read before the data field is accessed to ensure that the correct sector has been positioned before the selected read/write head during execution of the LOCATE command.

The data field consists of 512 bytes for data recording. Four bytes of error checking and correction (ECC) code that are used for error detection and correction are appended to each data field. A single burst of errors of three bits maximum that occurs within the 512-byte data field are corrected automatically by the control function without programming assistance (see discussion later). The ECC code also detects single error bursts of up to 14 bits span.

The cylinder concept utilized in 23XX and 33XX direct access devices is also utilized in 3310 disk drives. The access mechanism contains one head per track in a cylinder and, thus, can access one full cylinder of data at any access (cylinder) position. The first sector on a track in a cylinder can be accessed after the last sector on the preceding track in the cylinder without a rotation loss.

A 3310 drive has 358 primary cylinders, one alternate cylinder, and one CE cylinder. Each cylinder contains 352 sectors. Thus, cylinder capacity is 180,224 bytes.

DISK INITIALIZATION

The tracks in a 3310 drive are formatted with sectors at the plant of manufacture. Surface analysis tests (that are not available for installation use) are executed on each drive during the initialization procedure and alternate sectors are assigned for any primary sectors that are found to be defective. During initialization, the ID field of each sector is written with the appropriate flag and sector address bytes and the data field is written as ones.

Note that there is one alternate sector on each primary track in a 3310 drive. When a 3310 drive is initialized at the plant of manufacture, the first defective primary sector on a track is assigned the alternate sector on the same track. This eliminates seeking to and from an alternate cylinder.

If two or more primary sectors on the same track are found to be defective, alternate sectors from the alternate cylinder (located on the track addressed by the same head) are assigned for all but the first defective primary sector. When an alternate sector is assigned in the installation, it is always assigned from the alternate cylinder.

COMMAND OPERATION

All the commands defined for an FBA device are implemented for 3310 drives (see discussion in Section 10:25). This discussion indicates details about their implementation by the DASD Adapter in the 4331 Processor and the controller function in the 3310 Model A unit, as appropriate. If a command or subcommand is not discussed, it operates as described in Section 10:25.

The DASD Adapter has the capability of controlling the concurrent operation of up to 32 disk channel programs. The DASD Adapter

disconnects from the controller function in the Model A unit in a 3310 string during record positioning (seek and sector locate) operations when a LOCATE command is executed. This disconnection permits another disk channel program to be started on the DASD Adapter or an operational disk channel program to be resumed.

Disconnection during LOCATE command execution does not occur when the subcommand specified is Format Defective Block nor for any record positioning operations that occur during the execution of READ and WRITE commands.

LOCATE Command

Head positioning during the operation of a LOCATE command is accomplished using a rotational position sensing facility that is similar to that implemented in CKD devices with the RPS feature. The controller in the 3310 Model A drive contains one sector counter and one target register for each of the drives in the string. Each sector counter is constantly altered as the disks rotate to reflect the rotational position of the disks relative to the read/write heads. The target register is used during positioning operations.

A sector pulse indicates the beginning of each sector, just as the index pulse indicates the beginning of each track. The sector pulse indicates the beginning of a gap that precedes the ID field in the sector. The sector counter for a drive is incremented each time a sector pulse is recognized. The sector counter contains the sector number of the sector that is about to pass under the read/write heads.

When a LOCATE command is issued, the DASD Adapter calculates the ID (three-byte sector address) of the required sector from the specified relative block number of the first sector to be accessed. The adapter then issues a seek control command to the controller that contains the calculated cylinder and head number. The controller initiates the seek operation.

When the seek is completed, the controller is notified and a set sector control command utilizing the calculated sector number is issued by the controller. This causes the specified sector number to be placed in the target register and a comparison between the target register and sector counter values to be made every time the sector counter value changes. The controller is notified when an equal comparison occurs.

Execution of the LOCATE command is terminated when an equal sector condition arises. If the DASD Adapter is available, the controller reconnects to execute the next command in the channel program. The READ or WRITE command that follws the LOCATE command causes the controller to execute a verify ID operation on the sector that is passing under the read/write heads.

The controller reads the ID field of the sector. If a read error does not occur, the controller checks the flag byte for an alternate sector indication and, if this is a good primary sector, compares the three-byte sector address read with the calculated sector address to ensure correct positioning. If the two are equal, the data field of the sector is read/written.

Format Defective Block Subcommand. When a permanent error occurs in a data field or when an ID field is unreadable, the Format Defective Block subcommand can be used to assign an alternate sector from the alternate cylinder.

When a LOCATE command that specifies the Format Defective Block subcommand is received by the DASD Adapter, it executes the following alternate sector assignment procedure:

1. Seek to the alternate cylinder, specifying the head address of the defective primary sector.

2. Search for an available nondefective alternate sector on the addressed track only. If none is found, execution of the LOCATE command is terminated with intervention required and operation incomplete indicated in the sense bytes generated.

3. When an alternate is found, rewrite the ID to contain the sector address of the defective primary and a flag byte with the alternate bit on.

4. Reread the ID of the alternate sector to verify the write, and rewrite the ID if an error occurs. When the retry limit is reached, the ID of the defective alternate sector is rewritten with the defective and alternate flag bits on in the flag byte. Steps 2 to 4 are repeated until the ID in an alternate sector is correctly written or alternate sectors on the addressed track are exhausted.

5. Reposition to the defective primary sector. Rewrite the ID of the defective primary sector to contain the sector address of the assigned alternate sector and a flag byte that specifies an alternate is assigned.

6. Read the ID of the defective primary sector for write verification and retry the write if an error occurs. When the retry limit is reached, rewrite the ID in the displaced position (offset 64 bytes to right of the normal ID location).

The DASD Adapter handles switching to and from an alternate sector during a data transfer operation without interruption of the channel program. When a defective primary sector with an alternate assigned in the alternate cylinder is encountered during a data transfer operation, the DASD Adapter automatically repositions the access mechanism to the alternate sector that is specified in the ID of the defective primary sector. Once the alternate sector is read/written, if the block count is not zero, the DASD Adapter automatically repositions the access mechanism to the sector after the defective primary sector and continues the data transfer operation.

READ Command

When a data check occurs during the reading of the data portion of a sector, the DASD Adapter determines whether the error is correctable. If so, the correction code generated is applied to the affected data field (exclusive OR operation is performed). The corrected data is presented to processor storage without interruption of the channel program in process.

WRITE Command

When a Write and Check Data subcommand has been specified as the write operation, the DASD Adapter automatically initiates a readback check operation after writing is completed. The DASD Adapter repositions the access mechanism to the first sector written and reads all the sectors just written. No data is transferred to processor storage. The ECC bytes are generated during the readback check of each

sector and these bytes are compared with the ECC bytes just written for the sector. If they are equal, the data was written correctly.

The DASD Adapter does not initiate a retry procedure when correctable errors are encountered during verification. Such errors will be corrected the next time the sectors are read. However, if an uncorrectable error is detected during the verify read operation, the DASD Adapter repositions the access mechanism to the first sector written and repeats the readback check. The readback check operation is repeated each time an uncorrectable error condition occurs, up to a maximum of eight retries. If the uncorrectable error still exists after the retries, the write operation followed by a readback check (retried up to eight times) is repeated.

This write and readback check procedure is repeated four times if uncorrectable errors continue to occur. When the error persists throughout the entire retry procedure, sense bytes indicating a check data permanent error condition are generated and operation of the WRITE command is terminated.

Diagnostic Commands

For 3310 drives, the DIAGNOSTIC CONTROL and DIAGNOSTIC SENSE commands can be utilized to execute three subcommands: Format ID, Read ID, and Space ID and Read Data. The Format ID subcommand is used to rewrite the ID field of a sector. It specifies the flag byte to be written and the sector address of the sector to be accessed. The flag and sector address are written in the ID field of the specified sector. Only one ID can be written at a time using the Format ID subcommand.

The Read ID subcommand enables the ID field in one sector or in multiple consecutively addressed sectors to be read. Track and cylinder boundaries are not crossed during the execution of a Read ID subcommand even if too many sectors are specified. This subcommand can be used to verify Format ID write operations, since no automatic readback check is performed after a Format ID operation is performed.

The Space ID and Read Data subcommand enables the data field of one sector to be read without any reading of its associated ID field. This command can be used to recover the data field of a sector when a permanent error condition occurs when reading its ID field.

If an error occurs during the reading of an ID field (via the Read ID subcommand) or the data field (via the Space ID and Read Data subcommand), the automatic error retry procedures that are invoked during the execution of READ and WRITE commands when an ID or data field read error occurs are executed (discussed below). When a Read ID subcommand is used to verify a Format ID operation and an uncorrectable error persists after all retries have been performed, rewriting of the ID, using the Format ID subcommand, must be programmed. Error retry procedures are also performed for overruns and seek malfunctions that occur during diagnostic command execution.

ERROR RETRY FUNCTIONS

The DASD Adapter automatically performs retry procedures during channel program execution without the occurrence of an I/O interruption after certain types of errors occur. Error retry procedures are executed for seek malfunctions, read errors, and data overruns.

Seek Error Retry

The DASD Adapter verifies that a seek has been executed successfully when a seek operation is performed as a result of a LOCATE command and when implicit seeks or head switches are executed to cross cylinder or track boundaries during the execution of READ and WRITE commands. A seek error occurs when the three-byte sector address of the first sector read during a READ/WRITE command operation does not match the seek address utilized during LOCATE command execution.

When a seek error is detected, the DASD Adapter executes a recalibration operation (which moves the access mechanism to cylinder 0 and selects head 1) and then retries the seek operation that failed. If the seek error persists after ten retries of the recalibrate/reseek procedure, sense bytes with a permanent error indicated are generated and channel program execution is terminated. When the retry procedure is successful in correcting the seek error, the channel program (LOCATE command execution) continues normally.

Seek retry is not performed after seek errors that result from a track unavailable (invalid seek address) or no home found condition.

Data Check Retry

When an error occurs in the reading of an ID field (for a Format ID subcommand, READ command, or WRITE command), as indicated by the CRC bytes, or when a data field is read and an uncorrectable error occurs, a data check condition exists. The DASD Adapter repositions the access mechanism to the sector in error and rereads the ID or data field. If the error condition persists, seven additional rereads are attempted (eight retries total). If the rereading procedure is not successful, sense bytes with a data check permanent error condition indicated are generated and execution of the channel program terminates.

Data Overrun Retry

A data overrun condition occurs during a read operation when the buffer in the DASD Adapter is full and the DASD Adapter could not obtain access to the integrated channel within a specified time interval to empty the buffer. For a write operation, data overrun occurs when the buffer is empty and the integrated channel does not respond to a data request to fill the buffer within a specified time interval.

When the overrun condition is detected, data transfer is stopped and for a write operation the remainder of the sector being written is padded with zeros. The read/write operation is then retried up to 32 times. If the overrun condition persists, sense bytes with a data overrun permanent error condition indicated are generated and the channel program is terminated.

USAGE, ERROR, AND OVERRUN STATISTICS

One set of usage and one set of error counters for each drive in a 3310 string are present in reserved processor storage in the 4331 Processor (processor section 2). These counters are contained in the drive information block that is maintained for each 3310 drive. One overrun counter that is shared by all disk drives attached to the DASD Adapter is also present. These counters are maintained by the DASD Adapter.

Three usage counters are provided for each 3310 drive. One accumulates the number of sectors read (up to 16,777,215), one

accumulates the number of seeks issued (up to 65,535), and one accumulates the number of sectors written (up to 16,777,215).

The four error counters for each drive accumulate the number of errors that are retried by the error retry procedures executed by the DASD Adapter. Four counters are provided to accumulate the number of retries after (1) data checks for reads (up to 255), (2) check data errors that occur during readback checks for write verification (up to 255), (3) seek errors that result from ID mismatches (up to 255), and (4) correctable data checks that are eliminated by the error correction function (up to 65,535).

The one-byte overrun counter is incremented any time an overrun occurs on any drive attached to the DASD Adapter, except when the overrun occurs during the overrun retry procedure or when overruns are unavoidable (data chaining within a block is attempted, for example). When the counter reaches 255 and the next overrun occurs, the counter is not incremented and no more recording occurs until the counter is unloaded.

When a usage or error counter for a 3310 drive overflows, a flag is set for the drive. The next time a START I/O instruction is issued to that drive, a unit check condition is presented to cause the error recovery routine to issue a SENSE command to the drive. The contents of the usage and error counters for the drive and those of the overrun counter are placed in the sense bytes presented. The counters are then reset to zero. The contents of these counters can also be obtained using the READ AND RESET BUFFERED LOG command.

ADVANTAGES OF 3310 DIRECT ACCESS STORAGE FOR 2314/2319 DISK STORAGE USERS

The 3310 provides several advantages over 2314/2319 disk storage. When compared with 2314/2319 disk storage, 3310 Disk Storage provides significantly larger capacity per drive (29.1 million bytes maximum versus 64.5 million bytes) at a significantly reduced cost per bit. The 3310 also offers additional functions, significantly improved performance, enhanced physical characteristics, and RAS improvements, as follows.

Additional Functions

The 3310 has the rotational position sensing capability to increase DASD Adapter throughput and automatic error correction facilities provided by the DASD Adapter. The 3310 also offers the advantages of fixed block architecture discussed in Section 10:25.

Improved Performance

The 3310 has a much higher data transfer rate than the 2314/2319 drive (almost three times faster) as well as faster average and maximum seek times. Average seek time for the 3310 is less than half that of the 2314 while maximum seek time is approximately one-half that of the 2314. Cylinder-to-cylinder seek time for the 2314 is two-and-one-half times that for the 3310. Improved seek time also results from the fact that more data is covered per minimum, average, and maximum seek time for a 3310 drive than for a 2314/2319 drive.

Enhanced Physical Characteristics

A four-drive 3310 string requires significantly less floor space, power, and cooling than a four-drive 2314 string (composed of a 2314 and a 2313 unit). The 3310 string requires less than half the floor space while it contains more than twice the capacity (258 million bytes versus 116.4 million bytes maximum for the 2314 string). In addition, the weight of the 3310 string is about one-third that of the 2314 string (700 pounds versus 2325).

The 3310 string requires about one-fourth the amount of power and dissipates about one-tenth of the amount of heat as the 2314 string. Thus, less cooling is required for the 3310 string.

Improved Reliability

Reliability of the 3310 is improved over that of the 2314 by the removal of head-to-disk alignment problems that occur for removable media disk storage. Each read/write head within a 3310 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 3310 drives is not required, the critical alignment tolerances that are normally necessary for a disk pack are not needed for the 3310. There is little chance of a misaligned head causing an error because the 3310 servo system actually uses information from each data head to position itself. The 3310 provides accurate tracking and data transfer with minimal errors for far greater data densities than before.

Reliability is also improved because the chance of damaging read/write heads through operator mishandling is eliminated and exposure of disk data to outside contamination is nearly eliminated by the fixed-media design of the 3310. In addition, the possibility of head crashes is minimized by the improved flying characteristics of the read/write heads in a 3310 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 3310 permits a bit with a weaker than normal signal to be read correctly.

The recording density in bits per inch for a track and tracks per inch in a 3310 drive is several times greater than the track recording and track densities of a track in 2314/2319 drives. The 3310 has a higher recording density than any IBM 23XX or 33XX disk storage device except the 3370. However, the advanced head design used for the 3310 enables greater density to be achieved, together with improved reliability.

Reliability of 3310 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 a 3310 drive contains more electronics than a 2314/2319 drive, higher density logic cards are used in the 3310, which results in significantly fewer logic cards.

The power design of the 3310 further aids reliability as it results in fewer components, fewer power regulators, and improved power isolation. As a result of the reliability features of the 3310, no preventive maintenance is scheduled for 3310 drives.

Improved Serviceability

Serviceability improvements for the 3310 are designed to help the customer engineer find and correct failures more quickly. The following are the major serviceability improvements of the 3310 over the 2314:

- An enhanced fault symptom index is provided to aid faster problem isolation. It contains a list of "possible causes", an action/checklist, and cross references.
- Maintenance information manuals for the customer engineer are 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 are improved over those available for 2314 disk storage and are more powerful in fault isolation capability. In addition, they are much faster and execute with less customer engineer intervention.

ADVANTAGES OF 3310 DIRECT ACCESS STORAGE FOR 3340 DISK STORAGE USERS

The 3310 offers the following advantages over 3340 disk storage:

- Comparable capacity per drive and comparable maximum capacity per DASD Adapter at a significantly lower cost (64.5 million bytes for a 3310 drive versus 70 million bytes maximum, assuming full track records, for a 3348 Model 70 Data Module). The 3310 may provide larger capacity per drive than a 3340, depending on the size of the physical data blocks written on the 3340.
- A faster data transfer rate (1031 KB/sec versus 885 KB/sec)
- Reduced space requirements. A four-drive 3310 string requires approximately half the space required by a four-drive 3340 string and has less than half the weight. The recording density in bits per inch of a track and the number of tracks per inch on a 3310 disk are significantly greater than those of the tracks in 3348 Data Modules. This results in a smaller disk size and thus a reduced drive size.
- Improved reliability via implementation of a more reliable access mechanism and because the exposure of the disks to outside contamination is reduced. The access mechanism implemented in the 3310 is a swing arm voice coil mechanism instead of a linear voice coil mechanism, which can move in and out only. The swing-arm mechanism is simpler and requires fewer parts.

  While the 3340 offers improved reliability over the 2314/2319 as a result of the removal of head-to-disk alignment problems, the fixed-media design of the 3310 offers reliability advantages over the removable media design of the 3348 Data Module. Specifically, the exposure to outside contamination is less for the disks in a 3310 drive (the disks in a 3340 module are exposed to outside air during loading of the module) and the air filters in a 3310 drive are contained within the head disk assembly enclosure, instead of within the drive, and need to filter less air.
- Advantages of FBA over CKD track formatting (discussed in Section 10:25)
- Significantly reduced power and cooling requirements. The power requirement for a 3310 drive is approximately one-fifth that for a 3340 drive and it dissipates about one-tenth the amount of heat as a 3340 drive. A 3310 string can also operate in an environment with a wider range of temperatures and humidity than a 3340 string.

THE 2311/2314/2319/3310 DIRECT ACCESS STORAGE COMPATIBILITY FEATURE

The optional 2311/2314/2319/3310 Direct Access Storage Compatibility feature can be installed in a 4331 Processor to aid in the conversion from 2311/2314/2319 drives to 3310 drives when DOS Release 26, DOS/VS Release 34, or DOS/VSE is used. It enables an installation to utilize the greater speed and capacity of 3310 drives without program modification.

The compatibility feature enables programs written to access files on 2311/2314/2319 volumes to access these same files on 3310 drives attached to the DASD Adapter without any change to 2311/2314/2319 programs (that is, 2311/2314/2319 I/O macros, DTFs, and logic modules) once the 2311/2314/2319 files have been placed on 3310 drives.

The compatibility feature emulates all 2311/2314/2319 commands except RESERVE and RELEASE, which apply to the Two-Channel Switch feature for the 2311/2314/2319 facility. Therefore, multitrack, track overflow, and file scan features are supported by the compatibility feature. The RESTORE command, which is not used with the 2314, is treated as an NOP by the compatibility feature if it is issued to an emulated 2311/2314/2319 device.

Any 2311/2314/2319 program that is not time-dependent, does not use the PCI flag in its 2311/2314/2319 channel programs, and does not issue RESERVE/RELEASE commands can access 2311/2314/2319 files emulated on 3310 drives.

The compatibility feature permits the emulation of 231X drives on a maximum of two of the 3310 strings attached to the DASD Adapter. When 231X emulation is operative on two 3310 strings, 3340 strings attached to the DASD Adapter cannot be operative also. When 231X emulation is operative on only one 3310 string, one 3340 string attached to the DASD Adapter can be operative also.

The compatibility feature (which is loaded during IML) can be used to emulate 2311 or 2314/2319 drives as specified by the operator during IPL. Both type drives cannot be emulated at the same time.

When an operating system that does not support fixed block architecture devices is used (DOS Release 26 or DOS/VS Release 34) only 231X-format files can be accessed on 3310 drives. This is called the compatibility-only mode of operation and can be utilized when System/370 or ECPS:VSE mode is in effect.

When an operating system that supports fixed block architecture devices (DOS/VSE) is used, the 3310 drives within the two emulation strings that are not used for 231X emulation and the other two strings of 3310 drives, if installed, can be accessed in native (3310) mode during 231X emulation. In addition, a single program can access both emulated 231X and native mode 3310 files. This is called the mixed mode of operation and can be utilized only when ECPS:VSE mode is in effect.

A 3310 drive can be used to emulate a maximum of seven full or partial 2311 volumes or one or two full or partial 2314/2319 volumes. Unused space on the 3310 drive after the seventh 2311 or second 2314/2319 volume can be used to emulate one additional 231X minidisk. Up to four 3310 drives in a string can be used for 231X emulation. Thus, up to 28 full volume 2311 drives plus four 2311 minidisks can be emulated on one 3310 string while up to 8 full volume 2314/2319 drives plus four 2314 minidisks can be emulated per 3310 string.

Both 2311 and 2314/2319 disk volumes cannot be placed on the same 3310 drive. However, if a 3310 drive is not filled with emulated volumes, the unused space can contain native mode 3310 files and a

maximum of one emulated 231X minidisk. When ECPS:VSE mode is in effect, 231X and native format files contained on the same 3310 volume can be accessed concurrently by a single program during mixed-mode operations.

3310 and Emulated 2311/2314/2319 Device Addresses

An emulated 2311/2314/2319 volume appears to a 2311/2314/2319 program as a device with its own device address. Each 2311/2314/2319 drive to be emulated on a 3310 drive must be assigned a dummy device addess. This dummy address identifies the 2311/2314/2319 drive as an emulated device and determines the specific 3310 drives that can be used in the emulation process.

The standard range of addresses available for the 3310 drives that are to be used for 231X emulation is X'240' through X'273' while the range of device addresses for emulated 231X drives is 90 to CF. The high-order digit of the 3310 address is the channel number and 2 is the default channel address for the DASD Adapter. The middle digit is the 3310 string number, which can range from 4 to 7. However, two consecutive 3310 string numbers must be utilized for the two emulation strings. The low-order digit is the 3310 device number, which can range from 0 to 3.

Each 3310 drive used for emulation will have a fixed range of eight associated 231X device addresses. Shown below are the eight permissible emulated device address ranges and how they would be associated with eight 3310 drives assuming the first 3310 drive used has address 40.

3310 Drive	Emulated Device Addresses
40	90 - 97
41	98 - 9F
42	A0 - A7
43	A8 - AF
50	B0 - B7
51	B8 - BF
52	C0 - C7
53	C8 - CF

The address of the first 3310 drive used for emulation can be any one of drives in the selected string. The first emulated 231X file in the system always has address 90 assigned, the next has address 91 assigned etc, regardless of the address of the 3310 drive on which the first emulated 231X file resides. Therefore, if the first emulated 231X device resides on a 3310 drive with the address 50 (to use an example), this 3310 drive has the address range 90-97 associated with it for the emulated 231X devices. If a 3310 drive with address 51 is the next drive that contains emulated 231X devices, this 3310 drive would have the address range 98-9F associated with it, and so forth.

Whether all of the eight 231X addresses associated with a 3310 drive can be used depends on the number and type of emulated devices. For example, seven full 2311 files and one 2311 minidisk can be emulated on a 3310 drive. Thus all eight addresses in the range can be used. However, only two full 2314 volumes and one 2314 minidisk can be emulated on one 3310 drive. Thus, the remaining addresses in the range cannot be used.

If a 231X minidisk is contained on a 3310 volume, it must have the highest possible address in the address range used (for example, address 97 for a 2311 minidisk in the 90 to 97 address range or address 92 for a 2314/2319 minidisk in the 90 to 97 address range).

The operator must specify the first (or only) device address of the first emulation string at IPL, using the program load display. Thereafter, when an emulated device address is specified in a program the compatibility feature automatically selects the associated native (3310) device address.

Mapping of Emulated 2311/2314/2319 Volumes to 3310 Drives

Each 231X track is emulated on as many consecutive 512-byte sectors on a 3310 drive as are required. Each 2311 track is mapped to 8 sectors while each 2314/2319 track is mapped to 15 sectors. The mapping includes the home address record, track descriptor record, and count, key, and data areas of each physical record on the track written as a string of data without any gaps.

When a full seven 2311 volumes are emulated on a 3310 drive, 12,336 sectors are unused and are available for native format files or one 2311 minidisk of a maximum of 151 cylinders. When two full 2314 volumes are emulated on a 3310 drive, 4216 sectors are available for native format files or one 2314 minidisk of a maximum of 11 cylinders. The available sectors are the lowest addressed sectors, since emulated volumes begin with the highest addressed sectors.

The starting sector for each emulated 231X volume, whether full or partial, is fixed and allows enough space for the mapping of full volumes. However, to allow for the mapping of partial 231X volumes, the alternate cylinders of a 2311/2314/2319 volume (cylinders 200 to 202) are placed in front of the prime cylinders. Unused sectors between the last emulated 231X cylinder and the beginning sector of the next emulated 231X volume can contain native format files but not a 231X minidisk. The one minidisk permitted on a 3310 volume also has a fixed starting sector.

The 2311/2314/2319 files that are emulated on 3310 drives can have SAM, ISAM, DAM, or VSAM data organization and contain fixed blocked or unblocked, variable blocked or unblocked, or undefined format logical records, as appropriate for their data organization.

General Operation

The 2311/2314/2319/3310 Direct Access Storage Compatibility feature performs all I/O operations on the 3310 volumes that are emulating 2311/2314/2319 volumes. The compatibility feature processes all the 2311/2314/2319 channel programs that are issued to emulated 2311/2314/2319 devices by the executing programs.

Compatibility feature microcode is entered whenever a START I/O, TEST I/O, HALT I/O, or HALT DEVICE instruction is issued that specifies one of the device addresses that are reserved for emulated 2311/2314/2319 drives. It is also entered when a TEST CHANNEL instruction is issued that specifies the channel address of the DASD Adapter.

When the compatibility feature is entered as a result of the issuing of a START I/O instruction, it determines whether the 3310 device that is emulating the addressed 2311/2314/2319 is busy with another I/O operation. If so, a busy condition code is generated by the compatibility feature. If the 3310 drive is not busy, the feature fetches the 2311/2314/2319 channel program to be started, checks each channel command word (CCW) for validity before it is processed, and when necessary, initiates the 3310 channel program that is required to emulate the indicated I/O operation on the 3310 drive that is emulating the specified 2311/2314/2319 drive.

Emulation Buffers

The compatibility feature uses from one to eight emulation buffers for the read and write operations it performs on 3310 drives that are emulating 2311/2314/2319 volumes. An emulation buffer is 4096 bytes for 2311 emulation and 7680 bytes for 2314/2319 emulation. The number of emulation buffers to be used for the duration of an IPL is specified by the operator using the program load display.

If the operator specifies zero, the compatibility feature will not be used for this IPL. If the operator presses the enter key without entering any number, the number of buffers used for this IPL is the same as was used for the last IPL. A not-operational condition code is returned by the compatibility feature if a START I/O instruction for an emulated 2311/2314/2319 device address is issued and no emulation buffers are allocated.

The emulation buffers allocated by the operator are located in reserved processor storage to prevent any program and the operator from accessing the emulation buffers. The emulation buffers allocated are not specifically assigned to any partition and are available to be used for the I/O operations to all the 3310 drives that are involved in 2311/2314/2319 emulation.

The first (or only) emulation buffer that is allocated contains a predefined 3310 channel program. This channel program is used for all the I/O operations between the allocated emulation buffers and emulated 2311/2314/2319 volumes on 3310 drives. The 3310 channel program consists of the following chained commands: DEFINE EXTENT, LOCATE, and READ/WRITE DATA. This 3310 channel program reads an emulated 2311/2314/2319 track (string of eight or fifteen 3310 sectors) into a specified emulation buffer or writes an emulated 2311/2314/2319 track from a specified emulation buffer to a string of eight or fifteen 3310 sectors.

The 3310 channel program in the first emulation buffer is modified as required by the compatibility feature to locate the required 3310 sector, address the particular emulation buffer to be used for this I/O operation, and read or write the required 3310 sectors.

The first (or only) emulation buffer also contains other control information, such as the following, that is not present in the other allocated emulation buffers:

- A logical arms position table containing one entry for each of the possible emulated devices. This table is used to keep track of the location of the simulated 2311/2314/2319 access mechanism for each emulated 2311/2314/2319 drive. Whenever a 2311/2314/2319 seek command is issued to an emulated 2311/2314/2319 device, the specified 2311/2314/2319 seek address is placed in the logical arms position table entry for the emulated 2311/2314/2319 device specified in the START I/O instruction.

- A buffer contents table of eight entries, one for each of the possible emulation buffers. An entry in this table is used to hold the address of the first byte of the emulation buffer it describes, indicate the address of the 3310 device to which the emulation buffer is currently assigned, if any, indicate the last used 3310 relative block address, and indicate track orientation within the emulated 2311/2314/2319 track record in the emulation buffer.

  The track orientation field points to the 2311/2314/2319 record following the last 2311/2314/2319 record to be accessed in the emulation buffer. It can point to an ID, key, or data area. Since there are no gaps between 2311/2314/2319 records in an emulated

2311/2314/2319 track record, the track orientation pointer is used to define the starting point within an emulation buffer for emulating 2311/2314/2319 search, read, and write commands.

- 3310 DEFINE EXTENT and LOCATE command parameters that are used by the predefined 3310 channel program to seek, read, and write emulated 2311/2314/2319 track records on emulating 3310 drives

- Eight entries that are used by the buffer assignment algorithm in determining the emulation buffer to assign for an operation to an emulating 3310 drive when multiple emulation buffers are assigned

The compatibility feature updates the required control information in the first emulation buffer as 2311/2314/2319 commands are received and emulated.

The allocation of more than one emulation buffer can be used to improve the performance of the compatibility feature when multiple 2311/2314/2319 volumes are being emulated at the same time or if only a few 2311/2314/2319 volumes with high activity are emulated concurrently. In a virtual storage environment, the performance improvement gained by allocating each additional emulation buffer must be weighed against the possible performance loss that may occur from a reduction in the amount of processor storage that is available for paging operations.

Emulation of 2311/2314/2319 Seek Commands

When the compatibility feature receives a valid 2311/2314/2319 seek command, it fetches the specified seek address from processor storage and checks its validity. If it is valid, the compatibility feature places the address in the logical arms position table entry for the 2311/2314/2319 device specified in the START I/O instruction.

A seek command causes the required emulated 2311/2314/2319 track to be read into an emulation buffer. The compatibility feature calculates the required 3310 sector address using the 2311/2314/2319 seek address stored in the logical arms position table entry for the addressed 2311/2314/2319 device and stores the 3310 sector address in the first emulation buffer. The compatibility feature then modifies the predefined 3310 data transfer channel program in the first emulation buffer to perform a read operation into the emulation buffer it assigned for this I/O operation.

If only one emulation buffer has been allocated by the operator, the emulated 2311/2314/2319 track record is read into this buffer (overlaying any other 2311/2314/2319 track record the buffer might have contained). If more than one emulation buffer has been allocated, a least-recently-used algorithm, utilizing data in the first emulation buffer, is used to determine the emulation buffer to assign. In either case, the buffer contents table is updated to reflect the new contents of the emulation buffer used, once the emulated 2311/2314/2319 track has been successfully read in. The track orientation field for the emulation buffer is also updated.

Subsequent search, read, and write commands for the same emulated 2311/2314/2319 track operate on the emulated 2311/2314/2319 track in the emulation buffer for as long as the emulation buffer remains assigned to this 2311/2314/2319 track record.

Emulation of 2311/2314/2319 Search Commands

The compatibility feature emulates a valid 2311/2314/2319 search command by comparing the specified search argument with the same fields

in the emulated 2311/2314/2319 track in an emulation buffer. That is, the SEARCH and TIC*-8 command loop is simulated in processor storage by the compatibility feature.

The buffer contents table indicates which emulation buffer to use when more than one emualtion buffer has been allocated. For a search on ID, key, or key and data, the comparison begins at the location specified in the track orientation field of the buffer contents entry for the emulation buffer.

The compatibility feature generates a pseudo-index marker for the comparison operation. For a single-track search, if the search is not satisfied after the index marker is passed twice or if the search is satisfied, the search operation terminates. In either case, the track orientation field for the emulation buffer is updated to reflect the record after the last record searched.

If the multitrack bit is on in the search command and the search is not satisfied after processing the current 2311/2314/2319 track, the next 2311/2314/2319 track (string of 3310 sectors) is read into the same emulation buffer and the comparison continues. This process is repeated until the search is satisfied or end of emulated 2311/2314/2319 cylinder is reached.

Emulation of 2311/2314/2319 Read Commands

If a read command is the very first command to be issued to an emulated 2311/2314/2319 device (that is, no seek command preceded the read command), a read operation to the 3310 drive is not performed. The read command accesses the emulated 2311/2314/2319 track record from track 0 of cylinder 0 of the emulating 3310 drive that is contained in an emulation buffer. This occurs because the compatibility feature issues a seek command to logical cylinder 0 track 0 chained to a read command as part of the initialization process for each 3310 drive that is used for 2311/2314/2319 emulation. This causes the first track of the 3310 volume to be read into an emulation buffer so that it is available in case a seek is not issued before the very first read (or write) command for the emulated 2311/2314/2319 device.

The compatibility feature emulates a 2311/2314/2319 read command by moving the requested data (home address, track descriptor record, count area, data area, key and data areas, or count, key, and data areas) from an emulation buffer to the processor storage location specified in the 2311/2314/2319 read command. The buffer contents table indicates the emulation buffer to use.

For commands other than READ HOME ADDRESS and READ TRACK DESCRIPTOR RECORD, the track orientation pointer in the buffer contents table entry for the buffer involved normally indicates the location of the desired data within the emulated 2311/2314/2319 track record. The pointer is set by the 2311/2314/2319 search command from which the 2311/2314/2319 TIC and read commands are chained.

Emulation of 2311/2314/2319 Write Commands

The compatibility feature emulates a 2311/2314/2319 write command by moving the data from the processor storage location indicated in the write command to the appropriate location within the 2311/2314/2319 track in an emulation buffer. The buffer contents table indicates the emulation buffer to use. The track orientation pointer in the buffer contents table entry for the buffer involved indicates the location within the emulated 2311/2314/2319 track that is to receive the data, when necessary. The table entry pointer is set by the 2311/2314/2319

search command, from which the 2311/2314/2319 TIC and write commands are chained.

After the specified data has been moved to the emulation buffer, only those sectors within the emulated track that were changed are written back to the 3310 drive being used to emulate the addressed 2311/2314/2319 device. Note that when several write commands are chained in one channel program, the compatibility feature issues only one write channel program to the 3310 drive when it finishes processing the channel program to write back the altered sectors. A write to the 3310 drive also occurs when head switching occurs within a channel program.

When the compatibility feature finds a read command with the skip bit on in the same 2311/2314/2319 channel program in which writing is done, the 3310 channel program is issued with the Write and Check Data subcommand specified so that write verification is performed. Such verification is suggested, since a read error in any sector within an emulated 2311/2314/2319 track causes the entire emulated track (not just the affected 2311/2314/2319 record) to be considered invalid.

Emulation of I/O Instructions

The compatiblity feature emulates a START I/O instruction for an emulated 2311/2314/2319 device by determining whether the 3310 device that is emulating the specified 2311/2314/2319 is busy and whether the compatibility feature is busy with another emulated 2311/2314/2319 I/O operation.

Block multiplexing mode can be in effect for the DASD Adapter when the 2311/2314/2319/3310 Compatibility feature is in use but only one emulated 2311/2314/2319 I/O operation can be active at a time. Emulated 2311/2314/2319 devices share one subchannel. One emulated 2311/2314/2319 I/O operation can operate concurrently with I/O operations on multiple other 3310 drives that are being used in 3310 native mode or 3340 drives.

When the required 3310 device is not busy and no other emulated 2311/2314/2319 I/O operation is in progress on the DASD Adapter, the specified 2311/2314/2319 channel program is processed by the compatibility feature and a condition code of 0 is generated by the compatibility feature in response to a START I/O instruction. Condition code 1 is presented when the required 3310 device is not ready, the compatibility feature has pending status from a previous operation, or the compatibility feature is busy with a previously started 2311/2314/2319 I/O operation to the required 3310 drive. The 2311/2314/2319 channel program is not processed and the channel status word must be inspected.

When the required 3310 drive is busy with a native I/O operation or the compatibility feature is busy with an emulated 2311/2314/2319 I/O operation on another 3310 drive, condition code 2 is presented and the 2311/2314/2319 channel program is not processed. Condition code 3 is generated if the compatibility feature is not operational because the operator assigned zero emulation buffers or the specified emulation address is not being utilized by this 4331 Processor.

The compatibility feature emulates a TEST I/O instruction by obtaining status information about the specified emulating 3310 device. The condition codes generated by the compatibility feature indicate the required 3310 device and the compatiblity feature are available (code 0), the channel status word is stored (code 1), the required 3310 device or the compatibility feature is busy (code 2), or the compatibility feature is not operational (code 3).

When a TEST CHANNEL instruction that specifies the address of the DASD Adapter is issued, the compatibility feature checks the status of all 3310 drives attached to the DASD Adapter. The condition code setting indicates the compatibility feature is available, an interruption is pending for a 2311/2314/2319 emulation operation, an emulation operation is in process, or the compatibility feature or DASD Adapter is not operational.

The HALT I/O, HALT DEVICE, STORE CHANNEL ID, and CLEAR I/O instructions are also emulated by the compatibility feature. The CLEAR I/O instruction is executed as a TEST I/O instruction if the DASD Adapter is operating in selector mode.

Converting 2311/2314/2319 Volumes to 3310 Drives

The steps required to transfer the contents of the 2311/2314/2319 volumes that are to be emulated to 3310 drives in the format required by the compatibility feature are as follows:

- Dump the 2311/2314/2319 volumes to magnetic tape using the DOS-BPS Utility program 208 or the OS or OS/VS standalone Dump/Restore Utility. This should be done on the System/360 or System/370 processor (assuming the 4331 Processor does not have 231X drives attached). If the installation does not have magnetic tape drives, the contents of the 2311/2314/2319 volumes can be punched into cards, using the appropriate card utility.

- Execute the standalone Initialize Emulated 231X on 3310 utility program on the 4331 Processor to initialize emulated tracks on the 3310 drives that are to be used for 2311/2314/2319 emulation. The compatiblity feature identification (2311 or 2314), the emulated home address, and the emulated R0 record are written for each emulated 231X track. If any sectors are found to be defective during the initialization process, alternates are assigned. The clear value X'FF' is written in the balance of the emulated track after R0.

- Execute the 2311/2314 initialize disk program (INITDSK or any other 2314 initialize program) on the 4331 Processor to initialize as many emulated 2311/2314/2319 volumes on 3310 drives as are required. Before executing this step, the operator must activate the compatibility feature by assigning at least one emulation buffer.

- Execute the previously used dump/restore program to restore the dumped 2311/2314/2319 volumes from tape or cards to the initialized 3310 drives. The compatibility feature must be activated for this step also.

Once the conversion process is accomplished, normal job processing can be performed. At IPL time, the operator must specify the device to be emulated (2311 or 2314), the address of the first native (3310) drive used for emulation, the number of emulation buffers, and whether or not the compatibility feature is to perform error logging. For compatibility-only mode, LOG=YES must be specified. For mixed mode, LOG=NO must be specified because the operating system will perform error logging.

3340 DISK STORAGE DRIVES AND THE 3348 DATA MODULE

The 3340 Direct Access Storage unit is an intermediate-capacity, modular, high-performance direct access storage device 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 4331 Processor via the DASD Adapter with the 3340 Direct Attach feature installed.

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 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 20.15.1 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 Disk Pack that is the storage medium for 2314 disk storage, 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 20.15.2. The access mechanism in a 3348 Data Module is an L-shaped carriage, which 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.

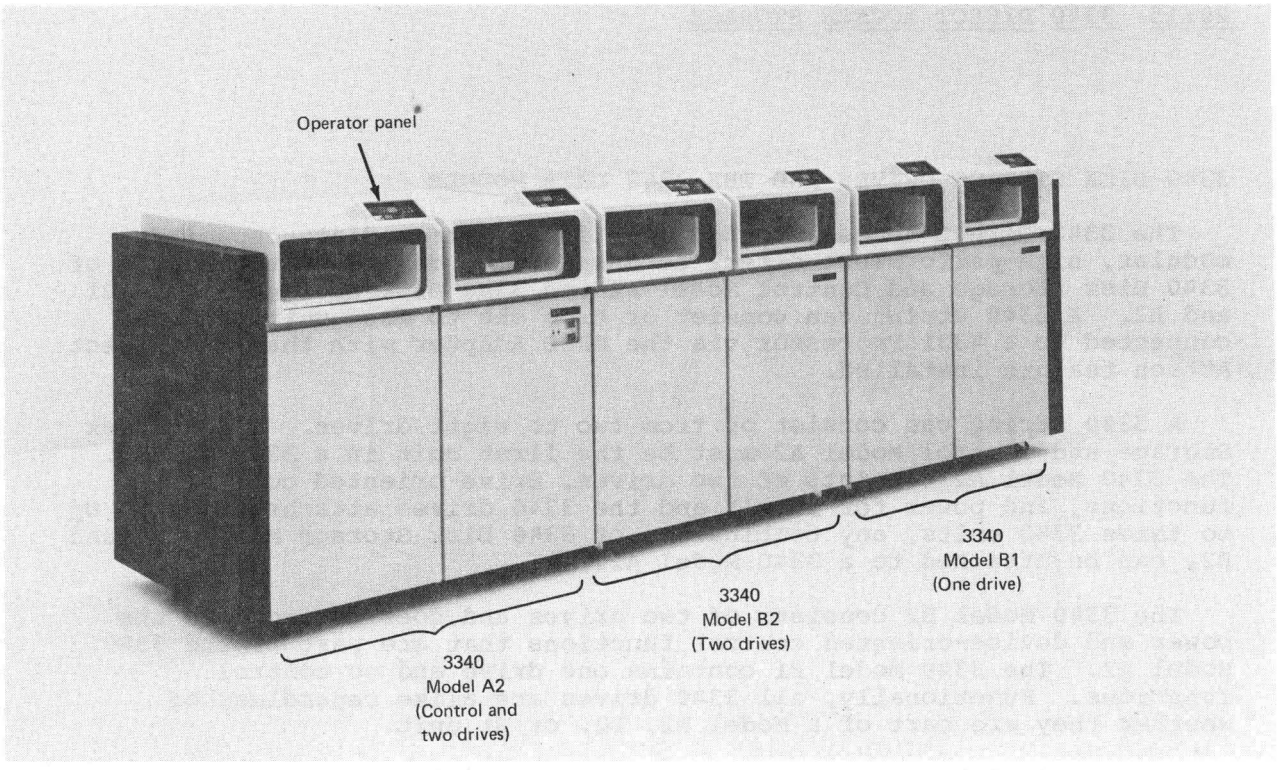


Figure 20.15.1. A five-drive 3340 string with 3340 Model A2, B2, and B1 units

Figure 20.15.2. 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.

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 either in a DOS/VSE or 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 drive is unlike other System/370 direct access storage drives 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 3340 Direct Access Storage more reliable than IBM direct access storage devices with a removable disk medium, 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. The advanced head design used for the 3348 Data Module enables greater density to be achieved, together with improved reliability.

Reliability of 3340 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 3340 drive contains more electronics than the 2314, higher density logic cards are used in the 3340, which results in significantly fewer logic cards.

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. The read-only function is enabled for a 3348 Data Module by turning an inset in the handle of the 3348 (see Figure 20.15.2) 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 3348 Data Module and the relative position of the read/write heads for a data surface are shown in Figure 20.15.3. 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).

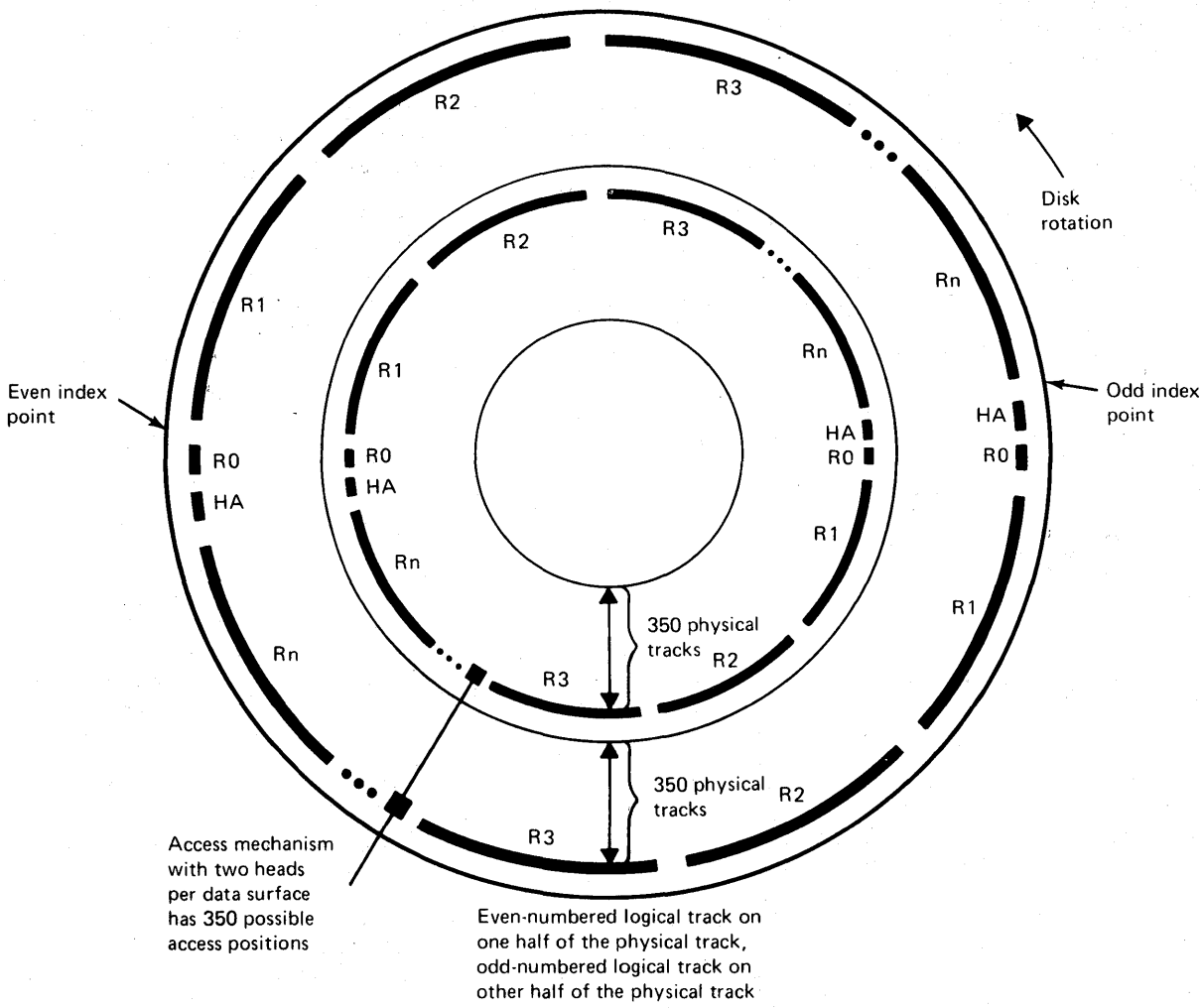


Figure 20.15.3. Location of physical and logical tracks and read/write heads on a data surface in a 3348 Data Module

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 20.15.3. During starting and stopping of 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.

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.

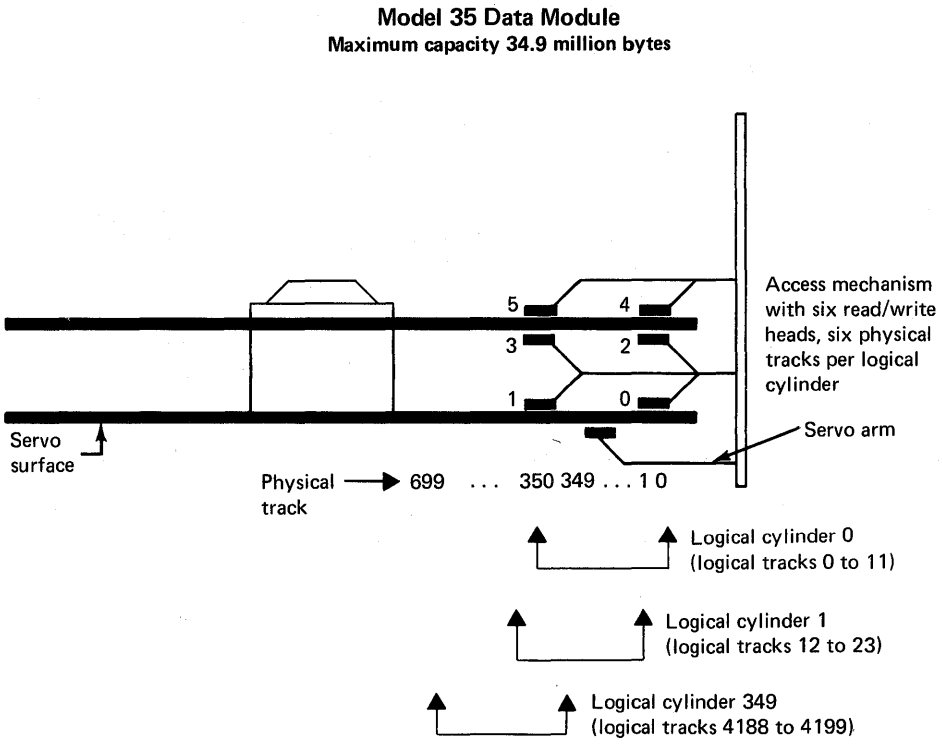
The access mechanism in a 3348 is driven by a voice-coil motor. This motor and the servo system provide fast, precise access mechanism positioning, which minimizes head-settling time.

Figure 20.15.4 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 20.15.4, 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.

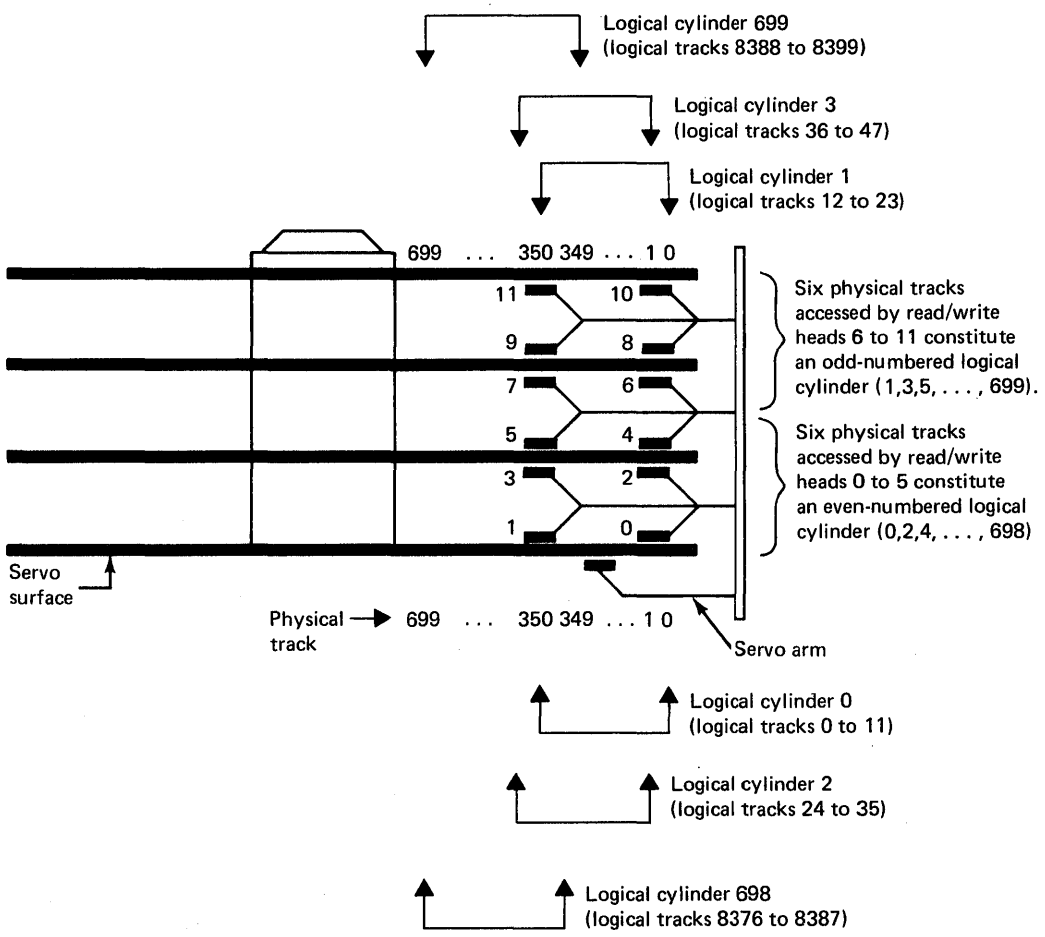
Figure 20.15.5 shows the layout of cylinders and read/write heads for the 3348 Model 70. A Model 70 contains four recording disks. Six data surfaces on the four recording disks, each of which is accessible by two read/write heads, are used for data recording in the Model 70. As for the Model 35, the 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 20.15.4. 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 20.15.5. 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 20.15.6 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 20.15.6. 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.

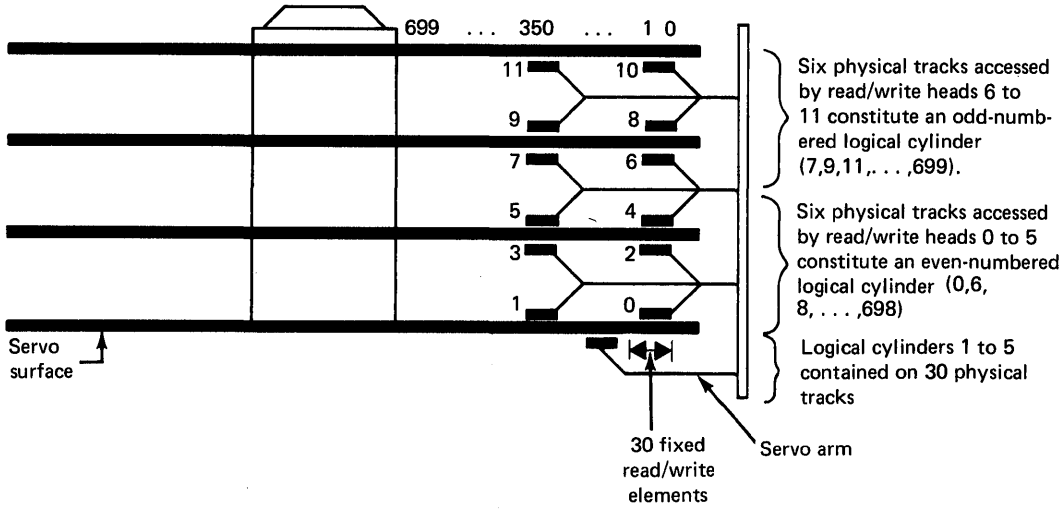
The physical tracks that contain logical cylinders 1 to 5 in a Model 70 are not used in a Model 70F and cannot be accessed by the user or a customer engineer because of the way in which head selection is performed. Hence, the data capacity of Models 70F and 70 is the same. Seek time for logical cylinders 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/files (such as the page data set, system catalog, TCAM message queue), small active user data sets/files, 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 Model 70F data module is a user responsibility. 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/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 20.15.6. 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 the Fixed Head feature in the ASSGN statements for files that are to be located on a Model 70F data module.

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.

Note that the defective track testing capability of OS/VS IEHDASDR and IBCDASDI is not provided by any DOS/VSE utility. DOS/VSE users can obtain IBCDASDI, the standalone utility, by ordering the OS/VS1 system.

The physical and capacity characteristics of 3348 Data Modules and the 2316 Disk Pack are given in Table 20.15.1. Table 20.15.2 gives the timing characteristics of 3340 and 2314 disk storage.

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 DASD Adapter. 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 is performed by the DASD Adapter, as discussed later.

Table 20.15.1. 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	148,880
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 20.15.2. 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 DOS/VSE Initialize Disk or 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 logical cylinder and track address specified by the seek command are converted to a physical cylinder and track address that is actually used by the drive in the seek operation. This physical address is saved by the DASD Adapter for later use in seek verification.

The physical address bytes are automatically written and read by the DASD Adapter 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 DASD Adapter 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 DASD Adapter 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 DASD Adapter 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 when the last seek command was issued. If the two

physical addresses are not equal, a seek retry procedure is executed to attempt to correct the error.

ATTACHMENT VIA THE DASD ADAPTER

One or two 3340 strings with device addresses in the range of 00 to 07 and 10 to 17 can be attached to the DASD Adapter in a 4331 Processor when the 3340 Direct Attach feature is installed. The 3340 Direct Attach feature is functional adapter microcode that enables the count, key, data track format of 3340 disk storage to be handled by the DASD Adapter, which is oriented to handle fixed block architecture devices.

When the 3340 Direct Attach feature is installed, the DASD Adapter handles full logical track read and write operations for 3340 drives. At least one 3340 emulation (functional adapter) buffer must be defined during IPL for use with 3340 drives. Up to eight 3340 emulation buffers (8800 bytes each) can be defined. The allocation of more than one emulation buffer will improve performance primarily for sequential processing of files/data sets on 3340 drives.

Once a 3340 track is in an emulation buffer, the functional adapter microcode executes the 3340 commands issued by programs on the contents of the 3340 emulation buffer. Data is moved between the program and the 3340 emulation buffer instead of between the program and the 3340 drives. Thus, programs communicate with the functional adapter microcode, rather than with the DASD Adapter microcode, when accessing 3340 drives. The functional adapter issues the required read and write requests to the DASD Adapter.

All 3340 commands except DIAGNOSTIC LOAD, DIAGNOSTIC WRITE, and READ DIAGNOSTIC STATUS 1 are supported by the functional adapter. The RESTORE command is treated as an NOP.

The DASD Adapter supports block multiplexing for 3340 drives when more than one 3340 emulation buffer is allocated. Eight nonshared subchannels are provided for each 3340 string to enable block multiplexing to be used for 3340 drives. The DASD Adapter can handle concurrent operation of one channel program on the attached 3340 drives for each emulation buffer defined.

Channel program disconnection occurs for 3340 drives during the positioning (seek and record locate) operations required to read and write 3340 logical tracks. This disconnection occurs whether or not the 3340 channel programs being emulated issue standalone or chained seeks or utilize SET SECTOR commands. Note that when a rotational position sensing (sector) command is encountered in a 3340 channel program, it is treated as an NOP. A full 3340 logical track is always read or witten.

The DASD Adapter and functional adapter microcode provide the error detection, correction, and logging features like those provided for 3340 strings when they are channel-attached to a System/370 processor via the 3830 Model 2. In addition, automatic error retry and alternate track positioning facilities like those provided for 3310 drives are provided for 3340 drives. The following facilities that are not implemented for System/360 direct access devices are supported for 3340 drives attached to the DASD Adapter:

- DASD Adapter correction of recoverable errors in the data area during read operations. When the DASD Adapter 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 and corrects the data sent to the emulation buffer without interrupting channel program operation.

- Statistical usage recording by the DASD Adapter. Statistical usage counters for each drive in each 3340 string are continuously maintained. These counters are contained in the drive information blocks that are maintained in reserved processor storage in the 4331 Processor (processor section 2). There is one drive information block for each 3340 drive attached to the DASD Adapter.

  The usage 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 DASD Adapter 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.

- Seek error retry by the DASD Adapter. The DASD Adapter verifies that a seek has been executed successfully when a seek operation is performed as a result of a LOCATE command and when implicit seeks or head switches are executed to cross cylinder or track boundaries during the execution of READ and WRITE commands. The verification is done as explained previously.

  When a seek error is detected, the DASD Adapter executes a recalibration operation (which moves the access mechanism to cylinder 0 and selects head 1) and then retries the seek operation that failed. If the seek error persists after ten retries of the recalibrate/reseek procedure, sense bytes with a permanent error indicated are generated and channel program execution is terminated. When the retry procedure is successful in correcting the seek error, the channel program continues normally.

  Seek retry is not performed after seek errors that result from a track unavailable (invalid seek address) or no home found condition.

- Data check retry by the DASD Adapter. When an error occurs in the reading of a count field, as indicated by the CRC bytes, or when a data field is read and an uncorrectable error occurs, a data check condition exists. The DASD Adapter repositions the access mechanism to the record in error and rereads the count or data field. If the error condition persists, seven additional rereads are attempted (eight retries total). If the rereading procedure is not successful, sense bytes with a data check permanent error condition indicated are generated and execution of the channel program terminates.

- Data overrun retry. A data overrun condition occurs during a read operation when the buffer in the DASD Adapter is full and the DASD Adapter could not obtain access to the integrated channel within a specified time interval to empty the buffer. For a write operation, data overrun occurs when the buffer is empty and the integrated channel does not respond to a data request to fill the buffer within a specific time interval.

  When the overrun condition is detected, data transfer is stopped and for a write operation the remainder of the sector being written is padded with zeros. The read/write operation is then retried up to ten times. If the overrun condition persists, sense bytes with a data overrun permanent error condition indicated are generated and the channel program is terminated.

- Automatic positioning to an alternate track. When the DASD Adapter determines the primary 3340 logical track to be processed has an alternate track assigned, it automatically repositions the access

mechanism to the alternate track and performs the required operation.

- Inline diagnostic testing of a malfunctioning drive (also provided for 2314 disk storage). The DASD Adapter can execute diagnostic tests on a malfunctioning 3340 drive while normal operations take place on the remaining drives in the string. This can be done in an online environment using OLTEP. 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.

SYSTEM/3 DATA IMPORT FEATURE

The System/3 Data Import feature consists of functional adapter microcode (similar to that of the 3340 Direct Attach feature) that enables 3348 Data Modules created by System/3 processors to be read but not written on 3340 Model A2 and B drives attached to the DASD Adapter. This feature enables the System/3-formatted volumes to be written to disks of the type that attach to the 4331 Processor. A conversion utility program product that operates under DOS/VSE control is provided to read System/3-formatted volumes and write them on disks attached to the 4331 Processor (see Section 30:05).

The functional adapter microcode to support System/3-format 3348 Data Modules operates like the 3340 functional adapter microcode in that it requests the DASD Adapter to read System/3 records into an emulation buffer and then processes System/3 commands on the data in the buffer. However, only one emulation buffer can be defined for reading System/3-format 3348 Data Modules.

SUMMARY

The hardware features of 3340 and 2314 direct access storage are summarized in Table 20.15.3. When compared with the 2314, the 3340 offers the following major advantages:

- Faster access to data
  - Data transfer rate almost three times that of the 2314
  - Seek times approximately 40% of 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

- Operational improvements
  - Cover tightening/untightening and removal/replacement operations are eliminated, which speeds 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 DASD Adapter
  - 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 are possible because the 3340 contains fewer circuit cards
  - Expanded microdiagnostics can test more than 95% of the circuits in a 3340

Table 20.15.3. Summary of hardware features of 3340 and 2314 disk storage

Feature	3340 Attached to the DASD Adapter	2314 (A-Series)
Number of drives per string	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 per control function	One or two	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

Table 20.15.3 (continued)

Feature	3340 Attached to the DASD Adapter	2314 (A-Series)
Attachment of a string to two control functions in the same or a different processor	No	Yes via 2844 Auxiliary Storage Control. Two concurrent data transfer operations per facility permitted.
Two-Channel Switch	Not applicable	Optional
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 but sector commands are executed as NOPs by the DASD Adapter for 3340 drives	Not available
Automatic correction of correctable errors in the data field by control function	Yes	No
Error retry functions performed by the control function	Yes	No
Automatic positioning to an alternate track	Yes	No
Surface defect skipping	Yes	No
Statistics logging by the control function	Yes	No
Inline diagnostics executed under OLTEP	Yes	Yes

20:20 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 (bytes per inch), 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. Models 6 and 8 of the 3420 cannot be attached to the 4331 Processor when 6250-BPI density is used because of their data rates.

This tape subsystem, which embodies a completely new control unit technology as compared to that in 2400-series tape 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 20.20.3 at the end of this subsection compares 3420, 2420, and 2401 tape unit characteristics.)

The tape commands, status responses, and basic sense data of the 3803/3420 tape subsystem are compatible 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 4331 Processor (subject to 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. Model 4 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, 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, and 7, respectively. In addition, less time is required to rewind less than a full reel on a 3420 as compared with 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, which minimizes 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-centimiter (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, and 5. For the 3420 Model 7, 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 20.20.1 summarizes 3803 Model 1 control unit capabilities.

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 20.20.1. 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	1. Nine-track, 1600-BPI PE tape on 3420 Models 3, 5, and 7 with Single Density	1. Nine-track, 1600-BPI PE tape on 3420 Models 3, 5, and 7 with Single Density
	2. Nine-track, 800-BPI NRZI tapes and nine-track, 1600-BPI PE tapes on 3420 Models 3, 5, and 7 with Dual Density	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 20.20.1 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.

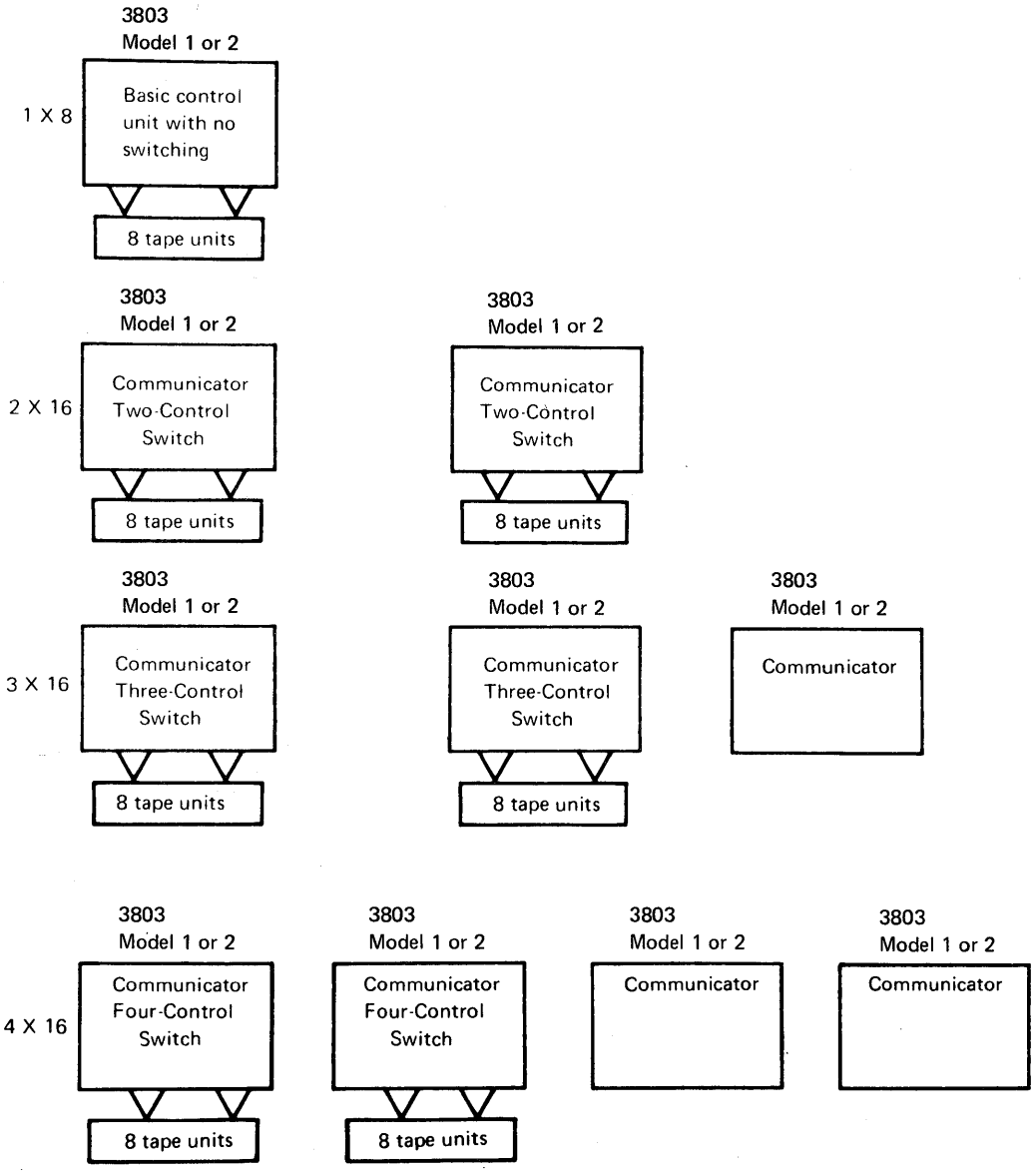


Figure 20.20.1. Tape-switching configurations for the 3803/3420 Magnetic Tape Subsystem

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. Previously, 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 MODEL 4 OF THE 3420 MAGNETIC TAPE UNIT

For the 4331 Processor, Models 3, 4, 5, and 7 of the 3420 can be attached to a 3803 Model 2. Up to eight 3420 units, in any combination of Models 3, 4, 5, and 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.

The 3803 Model 2 attaches only to the block multiplexer channel in a 4331 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 can be converted to a Model 4.

The 3803 Model 2 and 3420 Model 4 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
- A data rate of 470 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 the 3420 Model 4 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 the 3420 Model 4 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 the Model 4 (inverted tape path, single-capstan drive, in-column rewind, and optical tachometers). However, the capstan tachometer in the Model 4 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 the 3420 Model 4 provides the more precise motion and control that are needed. Nominal read and write access times for the Model 4, which are faster than access times for the Model 3, even at 1600-BPI density, are shown below.

Read access (ms) 1600 BPI	4.0
Write access (ms) 1600 BPI	3.0
Read access (ms) 6250 BPI	2.3
Write access (ms) 6250 BPI	2.1

Tape rewind speed for Model 4 is the same as that for the Model 3. 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-, 556-, and 800-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 required 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 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. Model 4 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 a 3420 Model 4). Table 20.20.2 summarizes 3803 Model 2 control unit configurations.

Tape mode setting for a 3420 Model 4 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 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.

Table 20.20.2. 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 the 3420 Model 4 with the 6250 Density or the 6250/1600 Density feature 2. Nine-track 1600-BPI tapes on the 3420 Model 4 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 Seven 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 3803 Model 2.

Tape-switching configurations can contain a mixture of 3803 and/or 3420 models. 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 20.20.2 illustrates a switching configuration with mixed 3803 and 3420 models and indicates the control units by which each tape unit can be accessed.

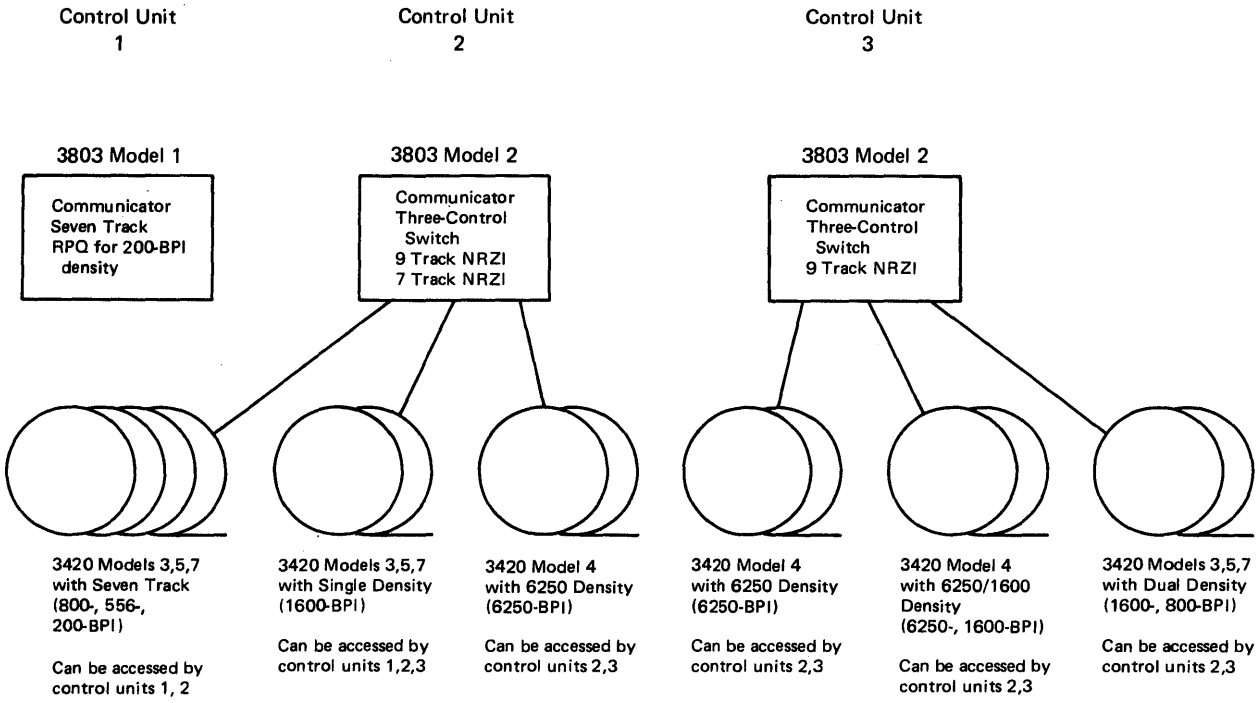


Figure 20.20.2. Sample tape-switching configuration with mixed 3803 and 3420 models

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 the 3420 Model 4, namely, automatic read amplification and a cleaning mechanism. When reading a 6250-BPI-density tape on a 3420 Model 4, 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 the 3420 Model 4. However, the 3420 Model 4 also has another cleaning mechanism that is designed to protect the read/write head when possible and to remove contaminants from the tape media.

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--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 attached to a 4331 Processor 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 the Model 4 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 20.20.3. 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 5 Model 7	Model 2 Model 5	Model 3 Model 6
Density (bytes/inch)	1600	1600	1600	6250	1600 1600	800 1600	800 1600
Data rate (KB/sec)	120	200	320	470	160 320	60 120	90 180
Tape speed (inches/sec)	75	125	200	75	100 200	75 75	112.5 112.5
Recording technique	PE	PE	PE	Group coded	PE PE	NRZI PE	NRZI PE
Nominal interblock gap size in inches (nine-track)	.6	.6	.6	.3	.6 .6	.6 .6	.6 .6
Nominal read/write access to data (ms)	4.0	2.9	2.0	2.3	3.9 2.5	8 8	5.3 5.3
In-column rewind	Yes	Yes	Yes	Yes	Yes Yes	No No	No No
Nominal rewind and unload time (secs)	76	66	51	76	78 66	90 90	66 66
Nominal rewind to ready status--full 2400-foot reel (secs)	70	60	45	70	72 60	84 84	60 60
Automatic threading	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	- -	- -

Table 20.20.3. 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 5 Model 7	Model 2 Model 5	Model 3 Model 6
Cartridge loading (10.5-inch reels only)	Standard	Standard	Standard	Standard	Standard Standard	Not available Not available	Not available Not available
Automatic reel latch	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 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 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	Automatic single-track. Automatic single-track	Programmed. Automatic single-track	Programmed. Automatic single-track
Longitudinal redundancy check	No	No	No	No	No No	Yes No	Yes No
Automatic read amplification	No	No	No	Yes	No No	No No	No No

*Trademark of E. I. Dupont deNemours & Co. (Inc.)

Table 20.20.3. 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 5 Model 7	Model 2 Model 5	Model 3 Model 6
Additional cleaning web mechanism	No	No	No	Yes	No No	No No	No No
Number of sense bytes	24	24	24	24	6 6	6 6	6 6
Microdiagnostics in control unit	Yes	Yes	Yes	Yes	No No	No No	No No
Separate erase head	Yes	Yes	Yes	Yes	Yes Yes	Yes Yes	Yes Yes
Data security erase command	Yes	Yes	Yes	Yes	No No	No No	No No
Seven Track feature	Optional	Optional	Optional	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 20.20.3. 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 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
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	Not available. Not available	Not available. Not available	Not available. Not available
Data rate (KB/sec) at 1600 BPI	-	-	-	120	- -	- -	- -
Recording technique at 1600 BPI	-	-	-	PE	- -	- -	- -
IBG size at 1600 BPI	-	-	-	.6	- -	- -	- -
Nominal read/write access to data (ms)	-	-	-	4.0/3.0	- -	- -	- -
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.	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	Serial Serial	Serial Serial	Serial Serial

Table 20.20.3. 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 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	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	Not available Not available	Optional on 2803 Model 1 for Model 67 and MP65 systems only. Not available	Same as Model 2 Not available

20:22 THE 3410/3411 MAGNETIC TAPE SUBSYSTEM

The 3410/3411 Magnetic Tape Subsystem extends the new dimension of price/performance and reliability, availability, and serviceability provided by the 3803/3420 Magnetic Tape Subsystem to users of half-inch tape who require data rates of less than 120 KB/sec. The 3410/3411 subsystem offers phase-encoded data recording and data rates of 20, 40, and 80 KB/sec at 1600-BPI density for Models 1, 2, and 3, respectively. In addition, the totally new, compact physical design of this subsystem minimizes floor space requirements while still permitting some flexibility in the arrangement of the units within the subsystem.

The 3410/3411 subsystem attaches to the 4331 Processor via the byte or block multiplexer channel. It offers significant advantages to Model 30 and 40 users with 2415 or 2401 Models 1, 2, and 4 in their tape configuration. The tape commands, status responses, and basic sense data of this tape subsystem are compatible with those of 2400-series and 3420 tape units. Therefore, existing Model 30 and 40 programs for 2400-series tape units will operate without change on the 4331 Processor (subject to the restrictions stated in Section 05:10) to handle operations on 3410/3411 subsystems with comparable features.

The 3410/3411 subsystem consists of Models 1, 2, and 3 of the 3411 Magnetic Tape Unit and Control and Models 1, 2, and 3 of the 3410 Magnetic Tape Unit. A 3410 consists of a single tape unit so that a subsystem can include an odd number of units (a 3410 advantage when compared to 2415 configurations). One 3411 must be part of every subsystem since the 3411 contains the tape control functions and power supply required to operate its own tape unit and the 3410 tape units that can be attached to it.

Each model of the 3411 can control only one tape speed. Therefore, different models of a 3410 tape unit cannot be attached to a given 3411 model. A 3410/3411 Model 1 subsystem can consist of from one to four tape units--one 3411 Model 1 unit, and up to three 3410 Model 1 units. A 3410/3411 Model 2 or Model 3 subsystem can include from one to six tape units--one 3411 unit, and up to five 3410 units, all of the same model as the 3411. The Additional Tape Units feature is required for the 3411 when more than three 3410 tape units are attached. A 3410 Model 1 can be field-upgraded to a Model 2 or 3 while a Model 2 can be field-upgraded to a Model 3. Field upgrades of 3411 models are possible also.

A significant advantage of the 3410/3411 tape subsystem is the relatively smaller amount of floor space it requires when compared with 2400-series tape units. This reduction is achieved by the integration of the control unit with one of the tape units, the compact physical size of units themselves, and a new unit design that reduces the amount of floor space required for service clearance.

The 3410 tape unit is shown in Figure 20.22.1. The 3411 and 3410 have the same dimensions: 31 inches across the front, 32 inches high in the front, 39 inches high in back, and 27 inches deep. A pair of units in a 3410/3411 subsystem (including tape control) is approximately the same width across the front as a pair of 2415 tape units with tape control (2415 Model 1 or 4) and as two 2401 tape units, which require a standalone control unit. The depth of a 3410 is about the same as that of a 2401 and a 2415 but the 3410 is slightly more than half the height of 2400-series units.

The 3411 and 3410 units have been designed such that normal servicing can be accomplished by access to the front of the unit. Thus, less clearance space is required, and the rear of any unit or of the

subsystem itself can be located as close as six inches to a wall. No clearance is required on the right or left side of the subsystem.

Figure 20.22.1. The 3410 Magnetic Tape Unit

Flexibility in subsystem arrangement is provided. Units in the subsystem must be physically attached to one another at the front corner. The 3411 can have 3410 units attached to either or both of its sides. Any two units can be placed side by side or can have an angle of up to 90 degrees between them. Figure 20.22.2 illustrates some of the permissible subystem layouts.

As in a 3803/3420 tape subsystem, 3410 tape units are radially, rather than serially, attached to the tape control unit so that a malfunctioning tape unit can be disconnected from the subsystem for servicing, parts replacement, etc., without disturbing the other tape units. A feature unique to the 3410/3411 tape subsystem, however, is the use of internal cabling to connect 3410 units to the 3411, instead of under-the-floor cabling.

The 3410 tape unit is designed to minimize tape-mounting time and reduce wear on the tape medium. A tape reel is mounted on the left-hand side of the tape transport. A new push-pull type of quick release latch is used. As shown in Figure 20.22.1, the tape path in a 3410 has been simplified so that the operator need thread the tape only across the top of the transport, instead of under two rollers as well as the read/write head. A single-capstan drive similar to that of the 3420 is used, which

minimizes tape wear. An optical tachometer provides more precise control of tape motion, as compared to previous tape units, to ensure accurate tape starts and stops and interblock gaps of correct size.

Dual Density and Seven Track optional features are available for 3410/3411 Models 1, 2, and 3. The Dual Density feature enables a drive to handle both nine-track 1600-BPI PE and nine-track 800-BPI NRZI operations. Seven-track 200/556/800-BPI NRZI operations can be handled on a drive with the Seven Track feature installed. These features are mutually exclusive and field-installable. If either the Dual Density or Seven Track feature is installed in a 3410 tape unit or the tape unit portion of the 3411, the companion feature must be installed in the tape control portion of the 3411. The translate and data convert functions are included with the Seven Track feature.

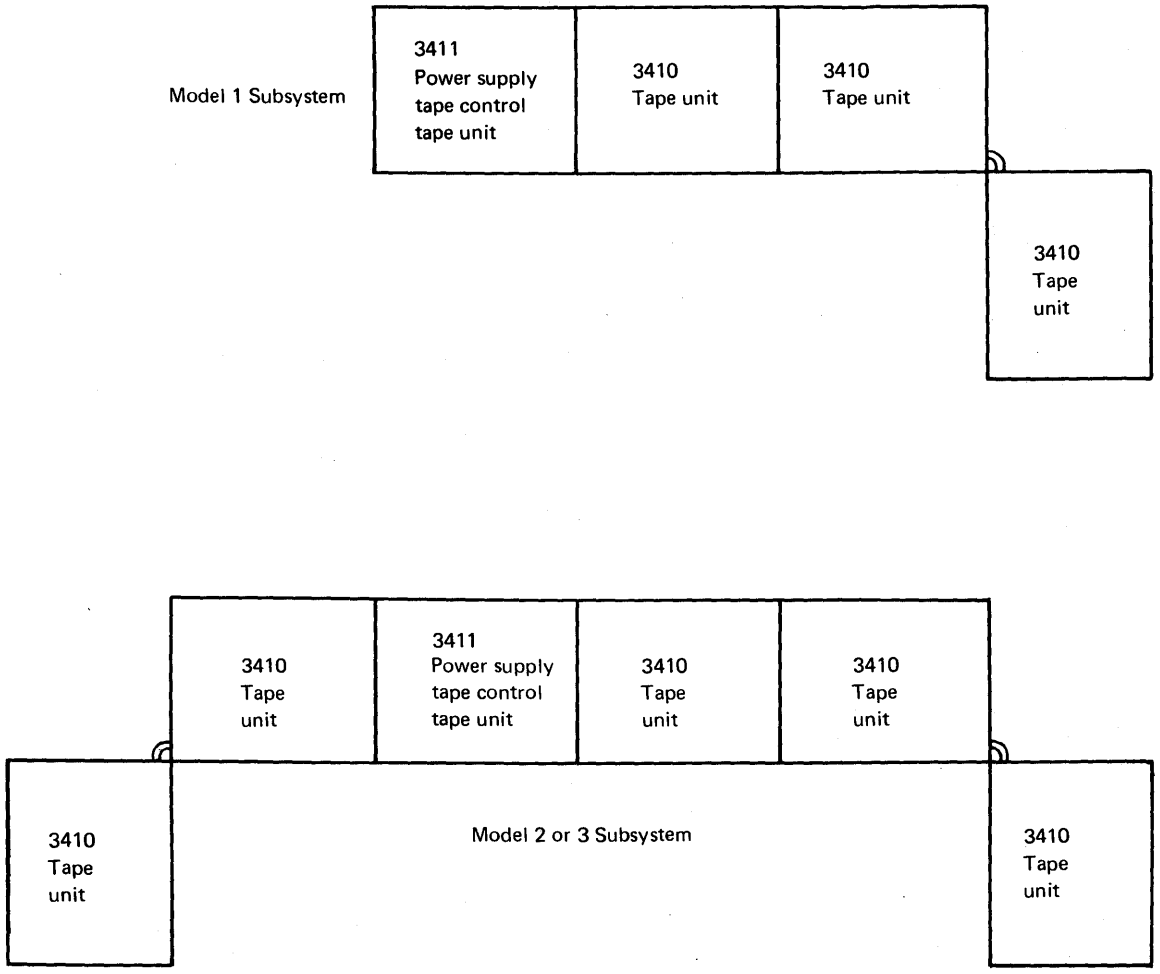


Figure 20.22.2. 3410/3411 tape subsystem physical layouts

A data security erase command is provided for the 3410/3411. 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.

The 3410/3411 tape subsystem has many of the reliability, availability, and serviceability features offered by the 3803/3420

subsystem. Phase-encoded recording and radial tape unit attachment have already been mentioned. (Phase-encoded recording is explained in Section 20:20.) The 3411, like the 3803, also uses digital signals, instead of analog, to transfer data across the I/O interface, which increases subsystem reliability by reducing its sensitivity to noise caused by power fluctuations, static, etc.

The tape control portion of the 3411 is implemented in monolithic technology and is a smaller version of the 3803 control unit that offers many of the same advantages of 3803 technology. The 3411 provides more sense bytes than the 2400-series tape units (nine instead of six) and improved serviceability via microdiagnostics contained in the control unit. Tape motion microprograms can be executed by the customer engineer when the 3411 is switched to offline status.

Table 20.22.1 compares the features of 3410, 2415, and 2401 tape units.

In summary, users of 2401 Models 1, 2, and 4 and 2415 tape units will find the 3410/3411 tape subsystem offers the following advantages:

- Improved price performance
- Minimized floor space requirements
- Simplified tape threading and a push-pull hub to minimize tape setup time
- Improved RAS features
- Compatibility with 2400- and 3400-series tape units

Page of GC20-1878-0
Added March 19, 1979
By TNL GN20-3935

Table 20.22.1. 3410, 2401, and 2415 Magnetic Tape Unit characteristics

Characteristic	3410 Tape Units			2401 Tape Units			2415 Tape Units	
	Model 1	Model 2	Model 3	Model 1	Model 2	Model 4	Models 1,2,3	Models 4,5,6
Data rate (KB/sec.)	20	40	80	30	60	60	15	30
Density (bytes/inch)	1600	1600	1600	800	800	1600	800	1600
Tape speed (inches/sec.)	12.5	25	50	37.5	75	37.5	18.75	18.75
Recording technique	PE	PE	PE	NRZI	NRZI	PE	NRZI	PE
Nominal interblock gap size in inches (nine-track)	.6	.6	.6	.6	.6	.6	.6	.6
Nominal gap passing time (ms)	48.0	24.0	12.0	16	8	16	32.0	32.0
Nominal read access (ms)	15.0	12.0	6.0	14.7	7.9	16.4	11.5	12.8
In-column rewind	Yes	Yes	Yes	No	No	No	Yes	Yes
Nominal rewind time 2400-foot reel (secs.)	180	180	120	132	90	132	240	240
Reel latch	Push-pull hub	Push-pull hub	Push-pull hub	Quick release latch	Quick release latch	Quick release latch	Quick release latch	Quick release latch
Recording medium (1/2-inch magnetic tape)	IBM Series/ 500, IBM Dynexcel, IBM Heavy Duty, or competitive formulations that meet the tape reel criteria in Tape Specifications (GA32-0006). IBM tapes other than the above do not provide adequate reliability and should not be used. 10.5", 8.5", 6.5" reels.	Same as Model 1	Same as Model 1	Same as 3410 plus Mylar*	Same as Model 1	Same as 3410	Same as 3410 plus Mylar*	Same as 3410

---

*Trademark of E. I. Dupont de Nemours & Co. (Inc.)

Table 20.22.1. 3410, 2401, and 2415 Magnetic Tape Unit characteristics (continued)

Characteristic	3410 Tape Units			2401 Tape Units			2415 Tape Units	
	Model 1	Model 2	Model 3	Model 1	Model 2	Model 4	Models 1,2,3	Models 4,5,6
Single-capstan drive and optical tachometer	Yes	Yes	Yes	No	No	No	No	No
Error checking								
Single-track corrections during reading (PE mode)	Automatic	Automatic	Automatic	Programmed	Programmed	Automatic	Programmed	Automatic
Vertical redundancy check	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Longitudinal redundancy check	PE mode-No NRZI-Yes	PE mode-No NRZI-Yes	PE mode-No NRZI-Yes	Yes	Yes	No	Yes	No
Number of sense bytes	9	9	9	6	6	6	6	6
Microdiagnostics in control unit	Yes	Yes	Yes	No	No	No	No	No
Separate erase head	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Data security erase command	Yes	Yes	Yes	No	No	No	No	No
Seven Track feature	Optional	Optional	Optional	Optional	Optional	Not available	Optional	Optional
Densities (BPI)	800 556 200	800 556 200	800 556 200	800 556 200	800 556 200	- - -	800 556 200	800 556 200
Data rate (KB/sec.)								
800 BPI	10	20	40	30	60	-	15	15
556 BPI	6.9	13.9	27.8	20.8	41.7	-	10.4	10.4
200 BPI	2.5	5.0	10.0	7.5	15	-	3.7	3.7
Recording technique	NRZI	NRZI	NRZI	NRZI	NRZI	-	NRZI	NRZI
IBG size (inches)	.75	.75	.75	.75	.75	-	.75	.75
Translator	Standard	Standard	Standard	Standard	Standard	-	Standard	Standard
Data Converter	Standard	Standard	Standard	Optional	Optional	-	Optional	Optional

Page of GC20-1878-0
Added March 19, 1979
By TNL GN20-3935

Table 20.22.1. 3410, 2401, and 2415 Magnetic Tape Unit characteristics (continued)

Characteristic	3410 Tape Units Model 1	Model 2	Model 3	2401 Tape Units Model 1	Model 2	Model 4	2415 Tape Units Models 1,2,3	Models 4,5,6
Dual Density feature (800-1600 BPI)	Optional	Optional	Optional	Not available	Not available	Optional	Not available	Optional
Data rate (KB/sec.) at 800 BPI	10	20	40	-	-	30	-	15
Recording technique at 800 BPI	NRZI	NRZI	NRZI	-	-	NRZI	-	NRZI
IBG size at 800 BPI (inches)	.6	.6	.6	-	-	.6	-	.6
Control unit	3411 includes one tape unit. One to three 3410 tape units can be added. Dual Density is optional.	3411 includes one tape unit. One to five 3410 tape units can be added. Dual Density is optional.	Same as Model 2	2803, 2804 (RWW) Model 1. Up to 8 tape units (Models 1,2, 3) can be attached. Seven-Track Compatibility feature is optional. 2803, 2804 Model 2 with optional Seven-Track, Nine-Track, or Seven- and Nine-Track Compatibility feature.	Same as Model 1	2803, 2804 (RWW) Model 2. Up to eight tape units (Models 4, 5, 6) can be attached. Seven-Track Nine-Track, or Seven- and Nine-Track Compatibility feature is optional.	Control integrated with tape units. Model 1 has 2 units and control, Model 2 has 4 units and control, Model 3 has 6 units and control.	Control integrated with tape units. Model 4 has 2 units and control, Model 5 has 4 units and control, Model 6 has 6 units and control.
Attachment of tape units to control	Radial	Radial	Radial	Series	Series	Series	Series	Series
Tape switching	Not available	Not available	Not available	2 x 16 3 x 16 4 x 16 (Requires one or two 2816 units)	Same as Model 1	Same as Model 1	Not available	Not available
Two-Channel Switch	Not available	Not available	Not available	Optional on 2803 Model 1 for Model 67 and MP65 systems only.	Same as Model 1	Not available	Not available	Not available

This page intentionally blank

20.25 THE 8809 MAGNETIC TAPE UNIT

The 8809 Magnetic Tape Unit is shown in Figure 20.25.1. It reads and writes nine-track, 1600-BPI, half-inch magnetic tape. The 8809 can process most industry standard reel sizes of 15.9 centimeters (6.25 inches), 17.8 centimeters (7.0 inches), 21.6 centimeters (8.5 inches), and 26.7 centimeters (10.5 inches).

Phase-encoded recording, like that utilized by certain models of IBM 2400-series and 3400-series magnetic tape units, is utilized by 8809 tape units and provides in-flight single-bit error correction during read operations. Phase-encoded tapes recorded at 1600-BPI density are interchangeable among 8809, 2400-series, and 3400-series magnetic tape units.

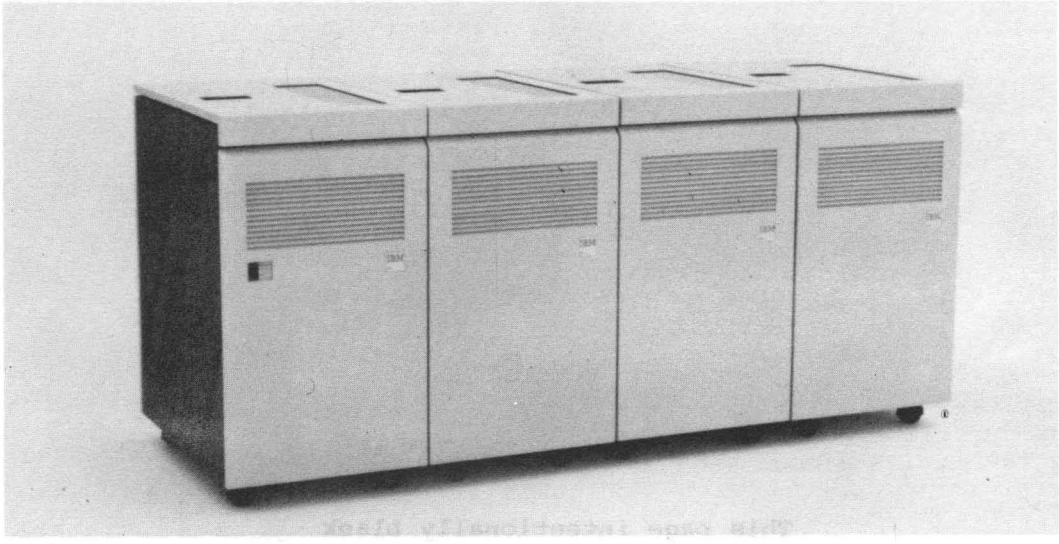


Figure 20.25.1. A string of four 8809 Magnetic Tape Units

Up to six 8809 tape units can be attached to a 4331 Processor via the 8809 Magnetic Tape Unit Adapter. An 8809 tape unit cannot be attached to a channel in the 4331 Processor. For 4331 Processor installations, the 8809 tape unit provides a high-speed save/restore facility for disk drives, and can be used for the other traditional tape functions (production processing, interchange with other systems, journaling, and archiving).

MODELS

Three models of the 8809 tape unit are available for 4331 Processor installations: 1A, 2, and 3. The Model 1A is a single-drive unit that contains power for itself and one additional tape unit, which must be a Model 2. A Model 1A must be the first 8809 unit attached to the 8809 Magnetic Tape Unit Adapter.

The Model 2 is a single-drive unit that does not contain any power. It attaches to a Model 1A or 3. The Model 3 is a single-drive unit with power for itself and one additional tape unit. The Model 3 attaches to a Model 2 and can have one Model 2 attached to it. For multi-unit 8809 strings, drives are cable attached and physically bolted together.

Thus, a six-unit 8809 string consists of the following models connected in the sequence listed: Model 1A, Model 2, Model 3, Model 2, Model 3, Model 2. A Model 1A or 3 unit can be powered on or off without affecting the other units in the string except for the attached Model 2 unit that derives power from it.

MODES OF OPERATION

An 8809 tape unit can operate in low-speed (start/stop) mode or high-speed (streaming) mode. In start/stop mode, the tape travels at .3175 meters (12.5 inches) per second while in streaming mode the tape travels at 2.54 meters (100 inches) per second.

Start/stop mode is established when an 8809 tape unit is powered on or a system reset (normal or clear) occurs in the 4331 Processor. The mode in effect is not altered by changing a tape reel. Thereafter, the mode is set using the SET LOW SPEED and SET HIGH SPEED commands. The mode set commands can be issued to an 8809 tape unit when (1) a tape reel is not loaded or (2) a tape reel is mounted, the cover is closed, and unit is not busy.

When operating in start/stop mode, an 8809 tape unit stops the tape in the interblock gap (IBG) after reading or writing each tape block, as do 2400- and 3400-series magnetic tape units. The instantaneous data rate for start/stop mode is 20 KB/sec.

Streaming mode is designed to be used for high-speed save/restore to direct access storage operations. When streaming mode is in effect, the tape is not stopped after a tape block is processed. The tape velocity is maintained while the IBG is passed. If the next command is received before the IBG is passed and is consistent with the current tape direction and mode setting, the command is processed without stopping the tape. If the next command is received after the IBG is crossed (a command overrun condition) or if no command is received (end of channel program condition), the tape unit stops the tape and automatically repositions the tape to process the next tape block.

The commands for which streaming operations are effective are READ BLOCK, WRITE BLOCK, WRITE TAPE MARK, FORWARD SPACE BLOCK, and BACKSPACE BLOCK. For streaming operations, the instantaneous data transfer rate is 160 KB/sec. This rate is achieved when no command overrun occurs. The effective data rate achieved is reduced when command overrun occurs. The format of a tape written by an 8809 tape unit is the same whether start/stop or streaming mode was in effect when the tape was written.

The size of the IBG created during tape writing, 3.048 or 1.524-centimeters (1.2 or .6 inches), is controlled by the SET LONG GAP and RESET LONG GAP commands. Either gap size can be used for start/stop and streaming mode. Long gap mode provides a longer time interval during which successive commands can be accepted to support streaming operations. During the power on of an 8809 tape unit or when a system reset (normal or clear) occurs in the 4331 Processor, the tape unit is set to write 1.524-centimeter (.6-inch) gaps. The gap size in effect is not altered when a tape reel change occurs. The SET LONG GAP command can be issued to establish long gap mode for which 3.048-centimeter (1.2-inch) gaps are written.

In order to maintain streaming operations when streaming mode is in effect, successive commands must be received within a certain time interval. This interval is the time from completion of a command execution to the latest point in the gap at which the 8809 tape unit can accept another command without repositioning. The time intervals between successive commands to maintain streaming operations for an 8809

tape unit attached to the 8809 adapter of the 4331 Processor are as follows:

- 2.7 ms between write block, write tape mark, or space block commands for short gap mode
- 8.7 ms between write block, write tape mark, or space block commands for long gap
- 4.2 ms between read block commands for short gap mode
- 8.7 ms between read block commands for long gap mode

If a command overrun occurs during streaming operations, the 8809 tape unit internally issues a stop request to the tape. As soon as the tape stops, it is moved backward toward the IBG after the last tape block processed. The tape is stopped when it reaches the correct position and is then ready to execute another command. The nominal time for this procedure, which is referred to as positioning time, is 870 ms while the worst case time is 1035 ms.

If no command has been issued during the positioning time, the tape unit awaits a command after it stops. If another command is received during the positioning time, the command is accepted and executed as soon as the tape is stopped at the correct position. The tape is moved forward and the first byte of data is transferred (read or written) in 295 ms.

The interval between the command overrun point in time and the transfer of the first byte of data for a command received during the positioning procedure is called the repositioning time. If a read or write command is received immediately after the command overrun time, the repositioning time (access time to first byte of data for that command) is 1165 ms nominal and 1405 ms worst case.

Note that when a mode set or gap size set command is issued while streaming operations are in process, the streaming operation is interrupted. Thus, such commands should be issued before data transfer operations are begun to avoid interrupting streaming operations.

COMMANDS

The data transfer and spacing commands for 8809 tape units are similar to those for 3400-series magnetic tape units. New commands for 8809 tape units are those that set the gap size and tape passing speed. When the DATA SECURITY ERASE command is executed, erasing is done at high-speed regardless of the mode (start/stop or streaming) currently in effect for the tape unit. When the SENSE I/O command is issued, the number of the 4331 Processor will be stored as the control unit type.

Read backward operations must be simulated by programming support (backspace followed by a forward read and another backspace) and the 8809 must be in start/stop mode.

Command chaining is supported for 8809 tape units. However, if data chaining is specified for any command other than SENSE or SENSE I/O, an overrun condition occurs and the command is terminated.

TRANSPORT DESIGN

As shown in Figure 20.25.2, the 8809 Magnetic Tape Unit has a tape transport design that is different from that of 2400- and 3400-series tape units. The 8809 design is simplified in that tape is transported

directly from reel to reel without capstans, vacuum columns, and associated components that are usually used in tape transports. Tape tension and velocity are controlled electronically. The simplified transport design significantly reduces the need for maintenance and there is no scheduled preventive maintenance for 8809 tape units.

Once a tape reel is mounted on the 8809 unit, the operator must manually thread the tape to the machine reel. The reel-to-reel design simplifies the threading process. When the LOAD REWIND pushbutton is pressed, the tape moves in the forward direction and the load time required depends on the location of the beginning-of-tape marker relative to the beginning/end-of-tape sensor located between the two reels. If the marker is on the mounted reel side of the sensor, forward tape movement is stopped as soon as the marker reaches the sensor and 10 seconds are required for the load.

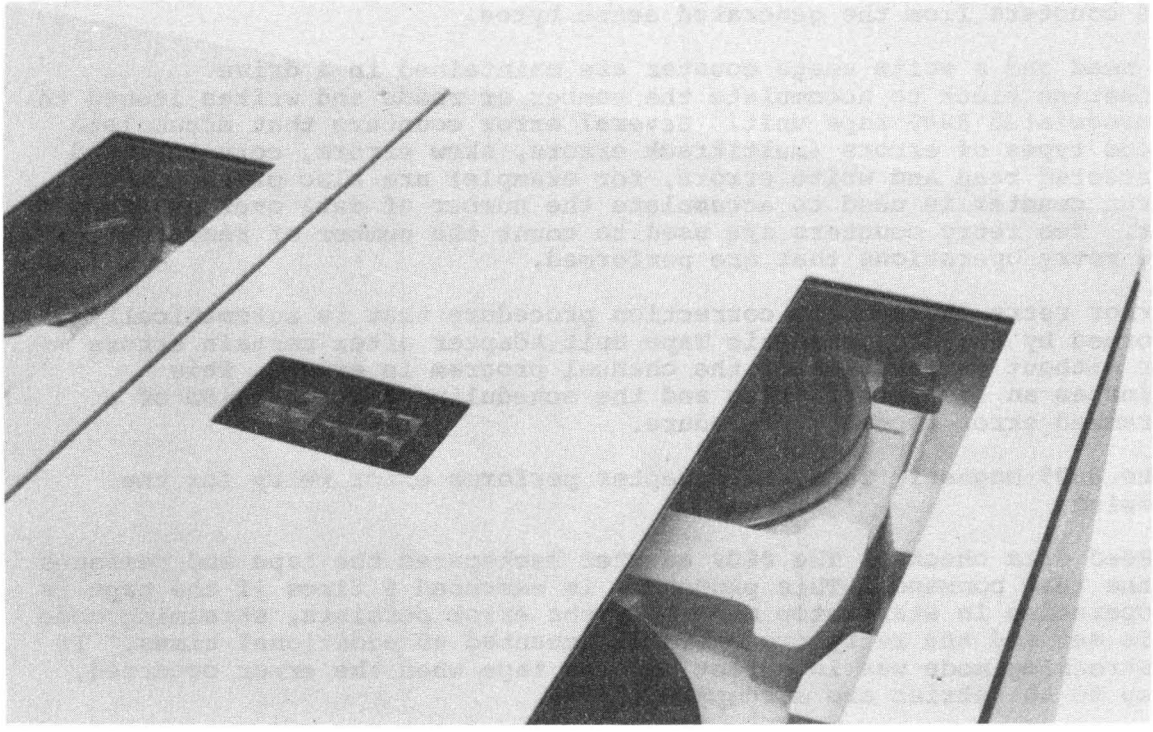


Figure 20.25.2. Top view of an 8809 tape unit showing the reel-to-reel transport design

If the marker is on the machine reel side of the sensor (marker passed the sensor during the threading procedure), the load requires 75 seconds. The additional time is required because the tape moves forward for an interval of time, and when the marker is not found in this interval, the tape is stopped and moved backward until the marker is located. Rewind time for a 731.5-meter (2400-foot) reel is 2.6 minutes.

SERVICEABILITY FEATURES

Like 3400-series tape units, the 8809 tape unit generates more sense bytes than 2400-series tape units to aid in diagnosing errors. The 8809 tape unit provides 32 sense bytes, the first 24 of which are compatible in format with the 24 sense bytes provided by a 3420 tape unit. The last 8 sense bytes provide information that is specific to 8809 units.

To aid the customer engineer in faster problem determination after an error occurs, the error recovery routine for the 8809 tape unit generates a symptom code for certain types of errors from the status and sense information provided. This code is stored in two of the additional eight sense bytes defined for 8809 tape units and written to the logout file (SYSREC in DOS/VSE). The code is an index to the Fault Code Dictionary, which indicates the maintenance analysis procedures that should be performed. For 2400- and 3400-series tape units, the customer engineer must execute a program that generates the symptom code from logout data contained in SYSREC.

The 8809 Magnetic Tape Unit Adapter maintains a drive information block in reserved processor storage in the 4331 Processor (in processor section 2) for each attached 8809 tape unit. Among other things, this block contains usage, error, overrun, and retry counters for its associated drive. The 8809 adapter accumulates certain information in these counters from the generated sense bytes.

A read and a write usage counter are maintained in a drive information block to accumulate the number of reads and writes issued to the associated 8809 tape unit. Several error counters that accumulate various types of errors (multitrack errors, skew errors, corrected and uncorrected read and write errors, for example) are also present. One overrun counter is used to accumulate the number of data overruns that occur. Two retry counters are used to count the number of read and write retry operations that are performed.

Error retry is an error correction procedure that is automatically performed by the 8809 Magnetic Tape Unit Adapter after certain errors occur without termination of the channel program in error. This eliminates an I/O interruption and the scheduling and execution of a programmed error recovery procedure.

The 8809 Magnetic Tape Unit Adapter performs error retry for the following:

- Read data checks. The 8809 adapter backspaces the tape and reissues the read command. This procedure is executed 5 times if the tape is operating in start/stop mode. If the error persists, streaming mode is set and the retry procedure is executed 40 additional times. If streaming mode was in effect for the tape when the error occurred, up to 40 retries are attempted.

- Write data checks during the execution of a write command or write tape mark command. The 8809 adapter backspaces one block, issues an erase gap command, and reissues the write command that failed. This procedure is attempted up to 15 times. When the failing command is not WRITE TAPE MARK, if the retry procedure does not correct the error, the LOOP-WRITE-TO READ command is issued to determine the location of the failure and the exact sense byte data to post to identify the location.

- Data overruns. The tape is backspaced one record and the command that caused the overrun is reissued. This retry procedure is executed up to 15 times.

If the retry procedure corrects the error, the channel program continues execution normally or terminates normally, as appropriate. The error retry counters are updated for reads and writes that occur during the retry procedures. Error and usage counters are not updated for errors and read/write operations that occur during error retry procedures.

Whenever any counter for an 8809 tape unit becomes full, a counter overflow flag is set. The next time a START I/O is issued to the tape

unit, a unit check condition is presented. When the operating system then issues a SENSE command, the data in all counters is placed in the appropriate sense bytes and the counters are set to zero. The READ AND RESET BUFFERED LOG command can also be issued to obtain counter data and reset the counters to zero (such as at end of processing for the day).

20.30 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 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 offers higher reliability than the 1403 Printer, quieter operation, and a more compact design.

The 3203 Model 5 Printer contains its own control function and is a standalone version of the 3203 Model 2 Printer that is available for attachment to System/370 Models 115 and 125. The 3203 Model 5 Printer attaches to a 4331 Processor via the byte or block multiplexer channel. Figure 20.30.1 shows the 3203 Model 5 Printer.

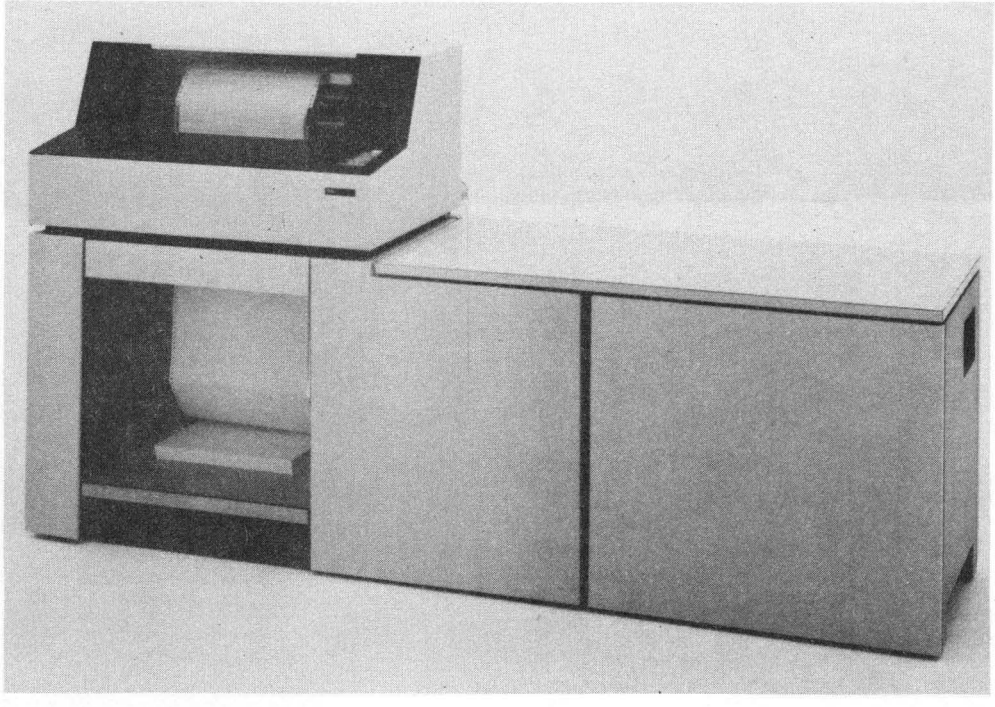


Figure 20.30.1. The 3203 Model 5 Printer (design model)

The 3203 Model 5 has a 132-print-position line and can print eight or six lines per 2.54 centimeters (1 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 (1 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. The additional status data presented can be stored for later analysis (by the EREP program) and should speed the diagnosis of hardware malfunctions.

30:05 DOS/VSE

DOS/VSE FUNCTIONS

The 4331 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 (4331 and 4341 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 in order 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 (3310 and 3370) 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. The 1400 emulator program will operate under DOS/VSE on System/370 processors but not on 4300 Processors. Emulation of 1400 programs on the 4331 Processor is provided via the IBM Systems 1401/1440/1460 Emulator Program (program product). A simulator, but not an emulator, program generated using the 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 4331 and 4341 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 (for the 4331 Processor), 3370 (for the 4331 and 4341 Processors), 8809, 5424, 3284, 3286, 3287, and 3289 I/O devices. (The 3880 Storage Control unit that attaches to the 4341 Processor is also supported.)
- 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-ES 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 4331 PROCESSOR FEATURES AND I/O DEVICES

The following discusses DOS/VSE support of specific 4331 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 4331 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 4331 Processor operating in System/370 mode (utilizing 3340 direct access storage) is a minimum of 84K bytes and operates with System/370, EC, and DAT modes in effect in the 4331 Processor. A DOS/VSE supervisor generated to support the 4331 Processor operating in ECPS:VSE mode (utilizing FBA direct access storage) is a minimum of 82K 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 for the 4331 Processor except the MOVE INVERSE instruction, including those instructions that operate only with ECPS:VSE mode in effect.

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.

1401/1440/1460 Compatibility feature. This feature is supported by the IBM Systems 1401/1440/1460 Emulator Program Release 1 program product. See discussion in Section 40.

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 3287 Model 1 or 2 Printer also attached to the Display/Printer Adapter of the 4331 Processor can be used for hard-copy backup of the 3278 Model 2A when it is operating in display mode. Printer-keyboard mode of operation, which requires a 3287 Printer, is also supported for the 3278 Model 2A.

Channels. The byte multiplexer channel and block multiplexer channel 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 use of 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. Use of either form of the IORB macro improves performance for ECPS:VSE mode operations.

DASD Adapter. This adapter is supported as a block multiplexer channel.

Communications Adapter. The teleprocessing access methods available for DOS/VSE support terminals attached to the Communications Adapter in the same way as if the terminals were attached via a 2703 transmission control unit with the same features. Synchronous data link control lines are supported only by ACF/VTAME.

ACF/VTAME does not support binary synchronous lines (except non-switched binary synchronous lines with a 3270 attached) or start/stop lines. However, BTAM-ES, which supports both binary synchronous and start/stop lines, and ACF/VTAME can operate in the same DOS/VSE system to provide the required mixture of line type support.

System/3 Data Import. This feature is supported by the VSE/IBM System/3-3340 Data Import program product. The program product reads System/3-format 3348 Data Modules on 3340 drives, converts the files to DOS/VSE SAM or VSE/VSAM files, and writes the files to 3310 drives (native or 231X emulation mode format) or 3348 Data Modules on 3340 drives. The VSE/VSAM program product must be installed (as well as VSE/Advanced Functions) in order to use the VSE/IBM System/3-3340 Data Import utility.

The import utility can operate in only one partition at a time. Any number of System/3-format files can be converted during execution of the

utility as long as one set of control statements per file is provided. Multivolume and multiversion System/3-format files are supported.

231X and 3340 Disk Storage. The 2311, 2314/2319, and 3340 are supported by all the disk access methods for all system files. A 2311 cannot be the system residence device.

3310 Direct Access Storage. The 3310 is supported in native mode by SAM (and VSE/VSAM) and for all system files. The same functions are supported as for other direct access devices (3340, for example). ISAM programs can access files on 3310 drives using the ISAM Interface Program of VSE/VSAM.

The 3310 Surface Analysis Utility (a standalone system utility) performs surface analysis on 3310 tracks and permits reclamation of alternate sectors, except those flagged at the plant of manufacture. This is a separately orderable utility. See Figure 30.05.1 for FBA device support by the program products shown.

The 3310 operating in native mode is supported by a System/370 mode supervisor only when the latter 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 program product). System/370 mode supervisors always support 231X emulation on 3310 drives. The 3310 is always supported by ECPS:VSE mode supervisors for both native mode and 231X emulation operations. Analysis Program-1 (AP-1) has been updated to include support of the 3310 in native mode format.

2311/2314/2319/3310 Direct Access Storage Compatibility feature. DOS/VSE supports emulation of 231X devices on 3310 drives using the compatibility feature (see discussion in Section 20:10) for both System/370 and ECPS:VSE modes. The Initialize Emulated 231X on 3310 utility (a standalone system utility) is available to preformat 3310 drives for 231X emulation. This is a separately orderable utility.

8809 Magnetic Tape Unit. The 8809 is supported by SAM in start/stop and streaming modes for all the same functions as 2400-series and 3400-series magnetic tape units. The READ BACKWARD command is simulated by the DTFMT logic module for start/stop mode of operation. The additional sense bytes provided by the 8809 (32 instead of 24) are supported. The ADD IPL command, ASSGN job control command, and SETMOD attention command support specification of long or short gap and start/stop or streaming mode for 8809 tape drives.

The Backup System, Restore System, and Fast Copy Disk system utilities are updated to support the streaming mode of operation of 8809 tape drives when 3310 disk storage in native mode or 3370 disk storage is used. Streaming mode operations can be achieved using one of these utilities when the 4331 Processor is dedicated to the utility operation.

The actual percentage of time that streaming operations occur during execution of one of these utilities in a multiprogramming environment depends on the other operations that are in process in the system at the same time. Best performance of the utility in a multiprogramming environment can be achieved by operating the 8809 in long gap mode and executing the utility in the highest priority batch partition.

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 sectors, except those flagged at the plant of manufacture.

This is a separately orderable utility. See Figure 30.05.1 for FBA device support by the program products shown.

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 3370 devices.

5424 Multi-Function Card Unit. The 5424 is supported, like the 5425 Multi-Function Card Unit, for System/370 Models 115 and 125 with extensions to provide error logging.

3289 Model 4 and 3262 Model 1 Line Printers. The 3289 is supported as an I/O device and as a system (SYSLST) printer of the PRT1 class. The 3262 is supported by VSE/Advanced Functions Release 2 and VSE/POWER Release 2.

3203 Model 5 Printer. This printer is supported as an I/O device and 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).

Diskette drive. This diskette drive is supported for the same functions as the 3540 Diskette Input/Output Unit.

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 given 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 (3310 and 3370) support in native mode 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 4331 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:

- Support of the 3262 Model 1 Line Printer as a system (SYSLST) printer of the PRT1 class. Since the error signaling of the 3262 differs from that of the currently supported line printers, user-written error recovery procedures must be modified for correct operation with the 3262.
- 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 (3310 and 3370) 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, 3400-series, and 8809 magnetic tape units.
- 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

- Support of the 3262 Model 1 Printer

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 and the 8809 Magnetic Tape Unit. Streaming and start/stop modes are supported by Access Method Services.

- 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 data 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 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 4331 Processor operating in System/370, EC, and DAT modes. Problem programs that execute under the Release 34 supervisor can operate in the 4331 Processor without change subject to the hardware constraints listed in Section 05:10.

The 4331 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) 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 optional Printer-Keyboard Mode feature must be installed in the 4331 Processor to enable the 3278 Model 2A to be used in printer-keyboard mode.

The 1400 emulator programs that operate under DOS/VS Release 34 control cannot be used in a 4331 Processor because of differences in the implementation of the 1400 compatibility feature in 4331 Processors and System/370 processors. However, if the IBM Systems 1401/1440/1460 Emulator Program is obtained, a 1400 emulator program that can execute under DOS/VS Release 34 control can be generated (see discussion in Section 40) and used in the 4331 Processor.

DOS/VS Release 34 does not support FBA devices or the 8809 Magnetic Tape Unit. However, the 3203 Model 5 Printer is supported.

USING DOS RELEASE 26

A DOS Release 26 supervisor generated to support a System/360 processor can execute in the 4331 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 4331 Processor are not present in a System/360 supervisor and the fixed logout area in the 4331 Processor is larger than in System/360 processors, the 4331 Processor should also be set to hard stop mode using the check control display of the operator console so that the processor will enter the check stop state whenever a machine check condition occurs.

Logouts to the system diskette and processor storage do not occur before the 4331 Processor enters the check stop state. The operator must then set the processor to normal mode for machine checks, which will cause logging to the system diskette and processor storage. A re-IPL is required to continue processor operation.

In order to use DOS Release 26, which supports the 1052 Printer-Keyboard as the operator console device, the Printer-Keyboard Mode feature must be installed in the 4331 Processor to enable the 3278 Model 2A Display Console to be used as a 1052 Printer-Keyboard. In addition, only I/O devices and hardware features of the 4331 Processor that are supported by DOS Release 26 can be utilized.

If 1400 emulation is to be performed, the IBM Systems 1401/1440/1460 Emulator Program must be obtained. A 1400 emulator program that supports the 1401/1440/1460 Compatibility feature of the 4331 Processor

or that uses a simulator program, if the 1400 compatibility feature is not installed in the 4331 Processor, can be generated to operate with DOS Release 26 so that emulation of a 1400 system can be performed (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 4331 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 4331 Processor. Table 30.05.1 lists the 4331 Processor hardware features and major I/O devices that DOS Release 26 does not support.

Table 30.05.1. 4331 Processor features and major I/O devices not supported by DOS Release 26

- Dynamic address translation and channel indirect data addressing and ECPS:VSE mode. (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 attach to the Display/Printer Adapter
- Block multiplexing. (The block multiplexer channel will operate in selector mode.)
- 3310 in native mode, 3370, and 3340 direct access storage (2311, 2314/2319, and 2321 devices are supported). The 3310 in 231X emulation mode is supported.
- 3400-series and 8809 Magnetic Tape Units. The 2400-series support can be utilized for these devices but the extended sense data is not recorded and only start/stop mode and forward (not backward) read operations are supported for 8809 drives.
- 5424 Multi-Function Card Unit
- 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.)
- Diskette drive and the 3540 Diskette Input/Output Unit

30:10 OS/VS1 SUPPORT

As of Release 7, OS/VS1 supports 4331 Processors and provides several additional enhancements that are not specifically related to these processors. The 4331 Processor operating in System/370 mode will be supported by OS/VS1 Release 7. OS/VS1 does not support the 4331 Processor operating in ECPS:VSE mode. Modifications to OS/VS1 to support the 4331 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 4331 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 4331 Processor, and I/O load balancing routines are set to use the 4331 Processor values.

OS/VS1 RMS routines need only be initialized to support the subset of logout provided by the 4331 Processor. The same level of support is provided for the machine check types and channel logouts that are implemented in the 4331 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 4331 Processor.

The following discusses OS/VS1 support of specific 4331 Processor features and I/O devices.

Instructions. The Assembler Language translator supports all the instructions for the 4331 Processor except MOVE INVERSE and those instructons 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 (GTF).

1401/1440/1460 Compatibility feature. This feature is not supported.

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's Console Support (DIDOCS) is

required to support primary and secondary display consoles. MCS is also required to support the 3287 Printer.

Channels. The byte multiplexer channel and block multiplexer channel are supported.

DASD Adapter. This adapter is supported as a block multiplexer channel (only for 3340 disk storage).

Communications Adapter. The teleprocessing access methods available for VS1 support terminals attached to the Communications Adapter in the same way as if the terminals were attached via a 2703 Transmission Control Unit with the same features. Only start/stop and binary synchronous lines are supported.

System/3 Data Import. This feature is not supported.

231X Direct Access Storage. The 2314/2319 is supported. The 2321 Data Cell Drive and 2311 Disk Storage are not supported.

3340 Direct Access Storage. The 3340 is supported as an I/O device and for all system data sets.

3310 and 3370 Direct Access Storage. The 3310 and 3370 are not supported.

2311/2314/2319/3310 Direct Access Storage Compatibility feature. This feature is not supported.

8809 Magnetic Tape Units. The 8809 tape unit is not supported.

5424 Multi-Function Card Unit. This I/O device is not supported.

3262 Model 1 and 3289 Model 4 Line Printers. These printers are not supported.

3203 Model 5 Printer. This printer is supported.

Diskette drive. The diskette drive is supported in the same manner as the 3540 Diskette Input/Output Unit.

30:15 VM/370 SUPPORT

Support of the 4331 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 4331 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. The ECPS:VM/370 feature and SDLC lines attached to the Communications Adapter of the 4331 Processor are also supported. The latter enables a DOS/VSE system with ACF/VTAME active to operate in a virtual machine. 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 4331 Processor is provided by Release 2 of VM/Basic System Extensions (VM/BSE), which also provides other support not included in 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 3310 and 3370 for the 4331 Processor) for the same functions as other supported direct access device types. (The 3880 attached to a 4341 Processor is also supported.)
- Support of the 8809 Magnetic Tape Unit in start/stop mode for the same functions as other supported tape units. The standalone dump/restore (DDR) utility supports the streaming mode of the 8809.
- Support of the 3289 Model 4 Line Printer for the same functions as other supported printers. (The 3262 Model 1 Line Printer, when available, will be supported by the then current level of VM/BSE).
- 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 4331 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, and System/370 processors with and without the Virtual Machine Assist feature 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 4331 Processor without any ECPS:VM/370 component enabled.

40:05 THE IBM SYSTEMS 1401/1440/1460 EMULATOR PROGRAM

GENERAL OPERATION

The 4331 Processor continues the advantages of integrated emulation available to Model 30 DOS CS/30 and Model 40 DOS CS/40 users and 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 emulator that executes as a problem program under DOS Releases 26, DOS/VS Release 34, or DOS/VSE with VSE/Advanced Functions installed on a 4331 Processor with the optional 1401/1440/1460 Compatibility feature installed (which is mutually exclusive with the 8809 Magnetic Tape Unit Adapter and ECPS:VM/370 features).

A 1400 emulator program can be generated with a 1400 simulator to operate on a 4331 Processor that does not have the 1401/1440/1460 Compatibility feature installed. Better performance is achieved when the compatibility feature is utilized.

The generated 1400 emulator 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 4331 Processor programs. Additionally, 1400 emulator jobs and DOS, DOS/VS, or DOS/VSE jobs can be intermixed in a single jobstream.

The 1401/1440/1460 Compatibility feature implementation in the 4331 Processor is different from its implementation in System/370 processors to increase its performance. In the 4331 Processor implementation, more functions are implemented in microcode, as opposed to simulation routines. Thus, a 1401/1440/1460 emulator program generated for the 4331 Processor will not execute on a System/370 processor and vice versa. The use of microcode instead of simulation routines also reduces the processor storage requirements for the emulator program.

The 1401/1440/1460 emulator for the 4331 Processor consists of the following basic parts:

- The 1401/1440/1460 Compatibility feature (emulator microcode). This microcode decodes and executes all 1401/1440/1460 instructions except EDIT, HALT, input/output, and invalid instructions. Note that the 1401/1440/1460 Compatibility feature for System/370 processors consists of a set of seven special instructions that handle a subset of the total 1400 instruction set.

- 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, DOS/VS, or DOS/VSE, that is to be used with the 1400 emulator 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 4331 Processor users the following advantages:

- Emulators 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 emulator 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 emulator 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 4331 Processor unit record devices, magnetic tape units, or direct access storage devices.

For the Model 30 CS/30 user or the Model 40 CS/40 user, the 1401/1440/1460 emulator for the 4331 Processor continues the advantages of integrated emulation and provides additional advantages, such as:

- Emulators operating concurrently in all partitions
- Support of CS/30 and CS/40 disk and tape files and emulator control statements
- 4331 Processor data formats for emulator tape and disk files
- Added emulator control available to the user at execution time
- DOS, DOS/VS, or DOS/VSE data management facilities and standard disk and tape label processing

Emulator Program Generation and Execution

The specific 1401/1440/1460 emulator program required by a given 4331 Processor installation must be generated. A 1401/1440/1460 emulator program can be generated to execute on the 4331 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 emulator program must be generated to support each of the three environments listed above. The emulator program product package consists of a set emulator program modules, a sample 1400 program with control cards to be used to verify correct operation of the generated emulator 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 emulator modules in the appropriate DOS, DOS/VS, or DOS/VSE libraries.

An emulator is assembled by the use of macro instructions. The macro instructions describe the 1400 CPU, input/output devices, special features, data files, emulator 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 emulator object module by the linkage editor for cataloging in a core image library. Any number of emulators can be assembled and cataloged in a core image library to run in any partition.

The emulator generation macros used for a 1400 emulator for the 4331 Processor are the same as those used to generate a 1400 emulator for a System/370 processor except for the addition of a MODE macro (to indicate in which of the three environments the emulator 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 emulator for the 4331 Processor.

The emulator 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.

An emulator program is handled by DOS, DOS/VS, or DOS/VSE in the same manner as any problem program. When using the 1401/1440/1460 emulator, 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 an emulator job. The EXEC job control statement causes the specified emulator program to be loaded and control is passed to the emulator program.

Emulator control statements are read by the emulator from the card reader or can be entered via the operator console. CS-format control cards are also accepted by the emulator.

Emulation with the 1401/1440/1460 emulator 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 emulator, information specified at emulator 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 emulator until an end-of-job condition is recognized. The 1400 program can be either executed or converted into 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 emulator program are provided by the exchange of emulator commands and messages between the operator and the emulator program. The emulator provides messages to inform the operator of errors or other conditions that require his attention or a response. Emulator 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 emulator program to support the following:

- 1400 operation codes not supported by the emulator (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 emulator or the 1400 program being emulated 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 emulator. Such a routine can be used to avoid having the operator respond to emulator program messages.

Note that since the interface to the 1400 emulator for the 4331 Processor is different from the interface to the 1400 emulator for System/370 processors, user-written exit routines that operate with a System/370 1400 emulator must be inspected to determine whether they will execute correctly with the 1400 emulator for the 4331 Processor.

A set of diagnostic commands (like those provided for System/370 1400 emulators) is provided that can be used during emulator program execution to obtain information about the job step in progress or to modify the 1400 program being emulated. In addition, the DOS/VS or DOS/VSE serviceability aids can be used without restriction for emulator 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 either 4331 Processor storage in hexadecimal format using the PDUMP macro or 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 emulator or the tape formatting utility.

The emulator accepts as input and produces as output two tape file formats:

1. 1400 format, which is produced by a 1400 system, a standalone emulator, CS/30, CS/40, the tape pre-postprocessor utility, a System/370 1400 emulator, 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, 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 emulator programs in a multiprogramming system.

- Files in spanned variable-length record format can be used by 4331 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 on 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 emulator. Tape I/O operations are executed using the EXCP macro.

The character codes supported by the 1400 emulator 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 emulator accepts three disk formats: CS emulator; count, key, data; and fixed block architecture. The DOS, DOS/VS, or DOS/VSE Clear Disk utility program is required to preformat 4331 Processor count, key, data type disk volumes that are used to emulate 1400 disk files. Each 4331 Processor disk record represents one 1400 disk track. Each 4331 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 emulator package must be used to format fixed block architecture devices used for 1400 file emulation. Each 1400 disk track is mapped to a consective group of fixed blocks. The number of blocks required is calculated at emulator generation time based on the 1400 device type.

A 1400 disk file can occupy one or more extents on 4331 Processor disk volumes but only one extent per volume. Extents must be allocated complete cylinders. When a file requires more than one 4331 Processor disk volume, the 4331 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 emulator 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 emulators.)

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 4331 Processor before emulation. Disk files created under CS/30 or CS/40 can be processed by the 1401/1440/1460 emulator if the CS option is specified at emulator 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 4331 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 emulator 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 4331 Processor disk volume, using a 1400 tape-to-disk or card-to-disk utility program under control of a 4331 Processor 1401/1440/1460 emulator.

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 emulator 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 Release 26 (which does 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 4331 Processor unit record devices.)

3. Generating the emulator 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 emulator is equal to the combined sizes of:

- Simulated 1401/1440/1460 storage. Each position of 1400 storage is simulated in one byte of 4331 Processor storage (for example, 8000 positions = 8000 contiguous bytes). This storage is marked permanently fixed during emulator program execution. Note that the communication region for the emulator 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 emulator program processor storage requirements for emulation 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 4331 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 4331 Processor devices

1401/1440/1460 Device and Features	Corresponding 4331 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, 3203 Model 5, 3262, or 3289 Model 4 Printer 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 Magnetic 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 (density) • Seven-Track feature is required if processing seven-track tapes. 8809 Magnetic Tape Unit (start/stop mode only)
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, 3340, 3310, 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 4331 Processor is designed to operate in such an environment, extensive availability and serviceability features are implemented for the 4331 Processor and its reliability is improved.

The objectives of the RAS features of the 4331 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 30 and 40, 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 System/370 intermediate-scale processors are provided in addition to commonly used diagnostics. 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:

- ECC validity checking on processor storage to correct all single-bit and detect all double-bit and many multiple-bit errors
- Defect dictionary to assign alternate bits to malfunctioning bits in processor storage via a special maintenance program
- I/O operation retry facilities, including channel retry data provided in the limited channel logout area, channel/control unit command retry, and I/O adapter error retry procedures to correct failing I/O operations
- Expanded machine check interruption facilities to facilitate 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 is impaired
- Microcode-controlled power sequencing and power 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 4331 Processor are provided in DOS/VSE. Machine check handler (MCH) and CCH routines for the 4331 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

- 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 I/O device errors for most devices that attach to the 4331 Processor

- Inline tests for the Communications Adapter and all I/O devices that attach to the 4331 Processor via I/O adapters

- Microdiagnostics for the components of the 4331 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 obtain assistance from 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 4331 Processor also represent improvements in the facilities available for intermediate-scale System/370 processors, such as:

- More extensive diagnostics with better problem isolation capabilities

- More checking circuits included in the basic design of the processor

- Microcode-controlled power sequencing and power monitoring

- Reference code generation to aid in faster malfunction diagnosis and repair
- Remote Support Facility to aid in diagnosing problems

50.10 RECOVERY FEATURES

Additional hardware, which provides correction of single-bit processor storage errors without programming assistance, is included as a basic part of the 4331 Processor. In addition, the control program can be notified, via an interruption, of various types of errors so that error recording and recovery procedures can be performed, depending on the error type. In addition, retry data is provided for channel errors.

ECC VALIDITY CHECKING AND DEFECT DICTIONARY FOR 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 in the 4331 Processor is handled on a four-byte basis, using a seven-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 the 4331 Processor that is not contained in processor storage. Models 30 and 40 use parity checking for main storage data verification.

As data enters and leaves processor storage in the 4331 Processor, ECC logic performs validity checking on each full word. When a word (39 bits, as shown in Figure 50.10.1) is fetched from processor storage, the seven-bit ECC code is checked to validate the 32 data bits. If the data is correct, the appropriate parity bit for each of the four data bytes is generated and the word is reformatted to look as shown in Figure 50.10.2.

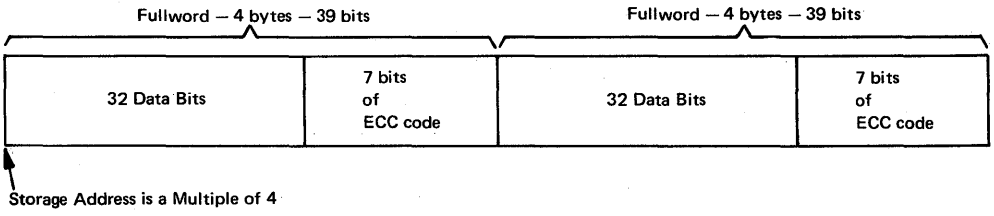
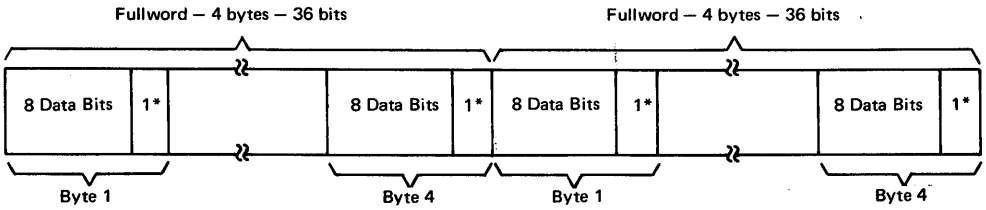


Figure 50.10.1. Data representation used in processor storage in the 4331 Processor

If a single-bit error is detected, the identified data bit in error is corrected automatically by ECC logic with no increase in instruction execution time. A single-bit correction in processor storage does not cause a machine check condition and a count of the number of single-bit corrections is not maintained.

When a word is to be placed in processor storage by a program, the four parity bits are removed and the seven-bit ECC code is generated and appended to the 32 data bits. The 39 bits are then stored as shown in Figure 50.10.1.



* Parity Bit

Figure 50.10.2. Data representation used in Model 30 and 40 processor storage and in the 4331 Processor in other than processor storage

If a double- or multiple-bit processor storage error occurs during instruction execution, a logout to the system diskette is written and the 4331 Processor enters the check stop state. There is no machine check interruption and logout to processor storage.

A defect dictionary is implemented for the 4331 Processor to provide recovery when a processor storage bit fails. This directory is not implemented in System/370 processors. Processor storage in the 4331 Processor contains one spare (additional) bit for every 39 bits (four bytes) of processor storage installed. These spare bits can be assigned to replace malfunctioning bits.

For every 39 consecutive bits (beginning at any bit location), one malfunctioning bit can be replaced within every 64K-byte block of storage for a 512K-byte processor storage or within every 128K block for a 1024K-byte processor storage. Only one malfunctioning bit within a 64K- or 128K-byte block can be replaced, for a total of eight replaced bits in the processor storage installed.

In order to assign spare bits, the operator must execute a program that locates the malfunctioning bits, builds a defect dictionary to identify the bit replacements, and writes the defect dictionary on the system diskette. During the next IML, the defect dictionary is utilized to set internal registers with the assigned alternate bits. Thereafter, when a malfunctioning bit is referenced, its alternate is automatically accessed instead.

The ECC function increases the availability of the 4331 Processor by permitting system operation to continue normally after single-bit processor errors occur and are corrected. Any processor storage errors on Models 30 and 40 necessitate at least termination of the processing program involved, since neither hardware nor programmed retry of processor storage errors is provided for these systems.

The defect dictionary permits the installed processor storage size to be utilized even when certain storage bits are malfunctioning. Without this facility, the 4331 Processor could not operate at all if a malfunction occurred in certain portions of the processor storage sections that are reserved for processor use. The defect dictionary also avoids the necessity of deleting an entire 2K block of processor storage from program processor storage when one bit within the block is malfunctioning.

I/O OPERATION RETRY

Channel retry and command retry features like those for System/370 processors and adapter error retry features 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 4331 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 4331 Processor, the block multiplexer channel has the command retry capability.

The DASD Adapter and 8809 Magnetic Tape Unit Adapter implement automatic error retry procedures during channel program execution (to eliminate programmed recovery) for the devices attached to them for certain errors (as discussed in Sections 20:10, 20:15, and 20:25).

EXPANDED MACHINE CHECK FACILITIES

Implementation of the machine check class of interruption in the 4331 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 4331 Processor differ from those of Models 30 and 40 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 fixed area in lower processor storage is increased to accommodate the storing of additional machine status information when a machine check interruption occurs.
- Error conditions are defined that cause the 4331 Processor to stop functioning immediately (enter the check-stop state) because the nature of the machine malfunction prevents valid processing from continuing.

The 4331 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 4331 Processor after an error has occurred that does not prevent continued successful execution of instructions. 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 a failure in the instruction processing function) occurs. Exigent conditions are those that prevent the successful execution of the current instruction.

For 4300 Processors, the fixed 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 and 4341 Processors are not identical, for example).

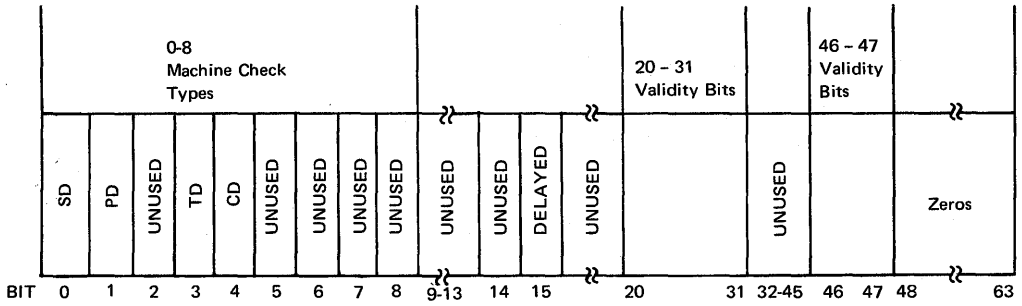
This approach is different from that implemented in System/370 in which processor-dependent data is stored in certain fields in locations 0 to 511 and a processor-dependent extended logout is also stored when a machine check interruption occurs (usually beginning at location 512). The length of the processor-dependent 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 4331 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 232-239). 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 the CPU timer and clock comparator values.

Figure 50.10.3 shows the layout and contents of the eight-byte machine check code 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. The delayed bit is set when machine check interruptions are disabled at the time a repressible machine check condition occurs. The bit indicates at least one instruction was executed after the machine check condition was recognized.

Table 50.10.1 lists the machine check types defined for the 4331 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 warning (W) 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.



Bit	Interruption Type
0	SD — System Damage
1	PD — Instruction Processing Damage
3	TD — Timer Damage
4	CD — Timing Facilities Damage
8	W — Warning
15	Delayed

Bit	Valid Fixed Area Data
20-23	Machine Check Old PSW (48-55)
	20 EMWP
	21 Masks and Protect Key
	22 Program Mask and Condition Code
	23 Instruction Address
24	Unused
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. 4331 Processor machine check code

Table 50.10.1. 4331 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 W	Warning •Under or over voltage condition •Over temperature condition •Console error	Repressible
PSW 13	System Damage •Irreparable hardware malfunction	Exigent
PSW 13	Instruction Processing Damage •Instruction processing function error	Exigent

Repressible Machine Check Interruptions

Repressible machine check interrptions 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 4331 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 timer damage and instruction processing 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.

- Warning. This interruption occurs to inform the operating system that processor operation will be terminated in 20 seconds as a result of an under- or overvoltage condition (see discussion under "Power System"), overheat condition, or failure in the console. This interruption is designed to enable the operating system to perform an orderly termination of I/O operations before processor operation stops.

Exigent Machine Check Interruptions

Exigent machine check interruptions are the following:

- Instruction Processing Damage. This interruption is enabled when 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.

- System Damage. This interruption is enabled when 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 4331 Processor to operate in normal or hard stop mode after a machine check error condition occurs. When the 4331 Processor is set to operate in normal mode, a logout to the system diskette followed by a machine check interruption and logout to program 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 are possible 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 4331 Processor operation.

In quiet mode, the processor is disabled for all or certain repressible machine check interruptions. Quiet mode can be used to permit system operation without error recording for certain repressible conditions when the 4331 Processor is operating under the control of an operating system without machine check handling routines for the 4331 Processor included.

A check-stop state is defined for the 4331 Processor (formerly called hard-stop state). If a check-stop condition occurs when the 4331 Processor is set to normal mode for machine checks, the 4331 Processor ceases all operations after logging out to the system diskette. There is no logout to the fixed area in program processor storage (locations 0 to 511) when a check stop occurs. 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 hardware 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.

The following conditions cause a check stop for the 4331 Processor when it is in normal mode for machine check conditions:

- An error occurs in the TLB.
- A double- or multiple-bit processor storage error occurs.
- An access key failure occurs.
- A system damage or instruction processing damage condition occurs and PSW bit 13 is zero.
- A system damage or instruction processing damage condition occurs while an interruption is being taken for a previous system damage or instruction processing damage condition.
- A machine check code cannot be stored or a new PSW cannot be fetched after a machine check condition.
- A machine check cannot be taken because of an error in the storage block assigned to program processor storage page 0.
- An error occurs while the page description bits for a virtual storage page are being updated and the page description is left in an inconsistent state.
- A microcode routine cannot branch to, or continue execution at, a required address. (For example, a protection check occurs when a saved return address in reserved processor storage is fetched.)
- Support processor-to-instruction processing function communication fails (as indicated by the elapse of a timeout interval).

When the 4331 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, in some cases, an IML is performed.

When the 4331 Processor is set to operate in hard-stop mode, after any type of machine check or channel check condition occurs, the 4331 Processor enters the check-stop state. There is no logout to the system diskette and no machine check interruption and logout to program processor storage occur. The suppressed log is kept until the 4331 Processor is set to operate in normal mode after machine checks, at which time the suppressed logs are written to the system diskette and program processor storage.

When the 4331 Processor is set to I/O stop mode using the check control display, the processor enters the check-stop state after a channel data, channel control, or interface control check occurs. The check-stop state is entered after any required logout to the system diskette is performed and the limited channel logout data is stored in program processor storage.

The state of the 4331 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. Warning 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 30 AND 40

A machine check situation in Models 30 and 40 results from hardware detection of a processor malfunction or a parity error. Bad parity can occur in main storage, local storage, a register, an adder, etc. Error correction is not attempted by Model 30 and 40 hardware when a machine check occurs (except for some instruction retry capability in the Model 30). If the machine check mask bit in the current PSW (bit 13) is on, a machine check causes an interruption and a diagnostic scan-out occurs, starting at location 128. The number of bytes logged is processor-dependent.

If the DOS machine check recording and recovery (MCRR) routine for the Model 30 or 40 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. (OS does not provide an SER routine for the Model 30.)

MACHINE CHECK ANALYSIS AND LOGGING TO THE SYSTEM DISKETTE

While 4300 Processors have expanded machine check capabilities similar to those implemented in System/370, 4300 Processors perform a machine check analysis function before writing logout data 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 machine check or check-stop condition arises during processor operation, the support processor receives control and a logout of parity

check data occurs in support processor 1. The parity check data is sent to the support processor via the support bus and support bus adapter.

The support bus provides a path to the maintenance logic in the instruction processing function (implemented on maintenance interface logic chips). This maintenance logic is required, for example, to access scan rings during a machine check or when maintenance functions are being performed. A scan ring is a latch that holds status information for a specific set of logic circuits. When a machine check occurs, status information is obtained from the scan rings and sent to the support processor. This is the processor-dependent logout information that is written to the system diskette.

After the logout, the support processor loads its log analysis program into its transient control storage area from the system diskette. The log analysis program inspects the parity check information, contents of certain registers, status information, etc., as required and generates an eight-digit reference code to identify the malfunction. This reference code is a pointer to a specific chapter and page of a customer engineer maintenance analysis procedure (MAP) guide.

The page identified by a reference code contains information about the malfunction and the actions that should be taken to repair it. The customer engineer may be directed to replace a card or execute certain diagnostic procedures. Output from the diagnostics is either another reference code that indicates the card to be replaced or the additional diagnostic actions that are to be taken to locate the malfunction.

The generated reference code is written to the system diskette and also displayed on line 23 of the operator console display so that reference codes can be communicated to the customer engineer when he is called. This may enable the customer engineer to bring any required spare parts. The reference code remains on the console display until the operator invokes the mode selection display.

An error recording program is loaded into the transient control storage area of the support processor to perform logging to the system diskette. In addition to the reference code, symptom data that includes the information utilized by the log analysis program to develop the reference code is also written to the system diskette. Logouts contained on the system diskette can be displayed on the operator console by the customer engineer using the Log Display Program.

Logouts are categorized by type (instruction processing function, block multiplexer channel, byte multiplexer channel, etc.) and the number of each type of logout (one or two) maintained on the system diskette depends on the type. Two logout areas are maintained. One contains logouts only for the Communications Adapter. The other contains logouts for all the rest of the functional units in the 4331 Processor.

To keep account of duplicate logouts, the reference code of the error to be logged is compared with the one or two reference codes and symptoms of the already recorded log(s) of the same type. If the current logout is the same as an existing log, the new log is not written but a counter associated with the reference code is incremented by one. Up to 255 logs of the same type can be represented in each counter. Counters can be reset by the customer engineer using the operator console (a log reset manual operation).

Note that logging to the system diskette does not occur during the time the operator or customer engineer is performing a manual operation using the operator console or when microdiagnostics contained on the service diskette are being executed.

Once the support processor has completed the recording of the reference code and associated processor-dependent logout data to the system diskette, a machine check condition is presented to the instruction processing function. If machine checks for this condition are enabled, a machine check interruption occurs 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. The reference code is not written to the operating system logout data set/file.

The MAP guides for the 4331 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 MAP guide indicates the steps to take to attempt to locate the failure.

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 data contained on the system diskette, rather than the operating system logout data set/file, to diagnose errors.

The processor-independent logouts contained in the operating system logout data set/file can be used in a 4300 Processor installation for the following:

- As a history of the machine check conditions that occurred. The system diskette keeps logout data for only the latest two machine check conditions of each type. The EREP program can be used to print the logout data set/file periodically.
- To inform the customer engineer of exactly what program was executing at the time the failure occurred (which is not recorded on the system diskette) if this information is required
- To enable the customer engineer to determine the type of error that has occurred before actually obtaining dedicated use of the 4331 Processor. The EREP program can be executed concurrently with normal customer processing to print the logout data.

POWER SYSTEM

The power system of the 4331 Processor was specifically designed to reduce power consumption from that required by intermediate-scale System/370 processors. In particular, the technology utilized has low power requirements, bleeder resistors are not utilized where possible, convenience outlets with associated transformers are removed, and complex AC-distribution systems for I/O devices are removed.

In addition, approaches to power control not implemented in intermediate-scale System/370 processors are utilized in the 4331 Processor, such as microcode-controlled power sequencing, power monitoring, and problem determination procedures. The advantages of these facilities are presented at the end of the power discussion.

The power system in the 4331 Processor consists of the following functional units:

- A hard-wired sequence that provides conventional sequencing for power to the support processor, system diskette, and operator console
- Power controller hardware that provides digital and analog sense points to permit power on/off control and power monitoring via microcode
- The power controller program, which consists of power-on/off sequencing, power-on test, power monitor, power log and reference code, and partial power-down microcode
- Power maintenance programs that consist of the Voltage Measurement Program, Power Log and Display Program, and Power Status Display
- A CE service panel that enables the customer engineer to power the 4331 Processor on and off and make power adjustments

Once the support processor is powered on via conventional hard-wired sequencing, the power on sequence for the rest of the 4331 Processor system is microcode controlled. During a power on, the power on test program is executed to test power controller hardware. 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.

The power monitor microprogram 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 256 milliseconds. Execution time for the program is approximately 20 milliseconds and a power off terminates any further initiation of the program.

The power monitor microprogram reads all the analog and digital sensor points and compares the results against predetermined tolerance limits for overvoltage and undervoltage conditions. The tolerance limits used depend on the mode in effect: normal mode (tight tolerance mode), switch-off mode (less tight tolerance mode), or switch-on mode (a CE mode to permit the customer engineer to make power adjustments).

Initially, the normal mode is made effective. If a tolerance limit violation is detected during an execution of the power monitor program, switch-off mode is made effective for all subsequent executions of the power monitor program. A power logout record that includes a reference code to identify the specific power failure is written to the system diskette. The reference code is displayed on the operator console. Processing continues normally.

If a tolerance limit violation is detected when the switch-off mode is in effect, a power logout record is written to the system diskette and a warning machine check interruption condition is presented to the processor. The power-off microprogram receives control after the warning interruption is taken and will power off the processor after approximately 20 seconds have elapsed.

The Power Log and Reference Code program is invoked in the support processor whenever a power failure is detected in the 4331 Processor. This microprogram determines the type of error (power-on error, power monitoring program detected error, or power-off error), determines the required reference code, and logs the failure to the system diskette.

The power log area on the system diskette contains space for four detailed power logouts that include a reference code and four additional reference codes. The power logout area contains the latest four power

logouts. The Power Log Display program can be utilized to display power logout data on the operator console.

The voltage measurement program can be run to cause all the analog sense points to be displayed simultaneously on the operator console. Nothing is displayed for voltages that are correctly set. A plus (+) or minus (-) is displayed for each voltage that is over or under the nominal voltage for the sense point.

The power system implemented in the 4331 Processor offers serviceability and availability advantages. Microcode-controlled power sequencing, versus hard-wired 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 also 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 4331 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 4331 Processor. RMSR performs all error recording functions. It replaces the following provided in DOS Version 3 (Release 26): 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 4331 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 4331 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 operations continue.

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. 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 4331 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 SYSREC or SYS1.LOGREC. These records have been constructed and/or written by machine and I/O error handling routines such as MCH, CCH, OBR, and SDR.

2. Accumulates a history of specified record types from SYSREC or SYS1.LOGREC by creating or updating an accumulation data set

3. Edits and prints a summary of selected records from SYSREC or SYS1.LOGREC or an accumulation data set

4. Clears all or certain types of error records

OLTEP AND OLTS - 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 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 4331 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 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 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 the procedures associated with the reference code for the error (if any).

IML TESTING AND INLINE TESTS

IML Testing

During IML, approximately 80 percent of the hardware in the 4331 Processor is implicitly or explicitly tested. If an error is found during IML testing, a message is written to the operator console that instructs the customer engineer to replace certain field replaceable units or run certain diagnostics.

Inline Tests

Inline tests are provided on the system diskette for the Communications Adapter and certain I/O devices that are attached to the 4331 Processor via I/O adapters. Inline tests are provided for the following: 5424, 3310, 3340, 3370, 8809, and the Communications Adapter. Inline tests in the 4331 Processor (maximum size 10K bytes) are loaded from the system diskette into reserved processor storage, from which they are executed.

The inline tests can be executed concurrently with normal processor operation. Since inline tests are located on the system diskette, when a malfunction occurs on a unit for which there is an inline test, the test can be loaded without diskette changing. The inline tests are invoked using the DIAG key on the keyboard via the maintenance and service selection display.

The operator has control of the operator console during the execution of certain inline tests after the customer engineer uses the console to initiate the test. The results of the tests are displayed on lines 23 and 24.

For some inline tests (Communications Adapter, for example), the customer engineer must use the operator console. In this case, when the

system message buffer becomes full, the audible alarm is sounded. The customer engineer can return the operator console to operating system mode for operating system use (using the CHG DPLY key, for example) to allow the operator to handle the messages. The operator console can then be returned to the customer engineer.

Communications Adapter Trace and Dynamic Display

The Communications Adapter Trace or Dynamic Display program can be invoked to operate concurrently with normal processing involving the Communications Adapter. The Communications Adapter Trace can be used to collect hardware and microcode information for a specified line attached to the Communications Adapter while the line is operating. After tracing is stopped, the collected information can be written to a printer to provide hard copy or can be displayed on the operator console display.

The Communications Adapter Dynamic Display program can be invoked to collect information about the operation of a communications line. This information is less detailed than that obtained using the Communications Adapter Trace. The collected information is displayed on the operator console and continuously updated.

MICROTESTS AND MANUAL OPERATIONS

Microdiagnostics that require the 4331 Processor to be dedicated to testing are provided on the service diskette. Microdiagnostics are provided for the instruction processing function, integrated channel components, processor storage, I/O adapters, power control hardware, and the support processor. The customer engineer utilizes a maintenance and service selection display to select the appropriate microdiagnostic.

Using the operator console, the customer engineer can also perform certain manual operations designed to aid in the diagnosis of problems. These operations are described in maintenance documentation.

REMOTE SUPPORT FACILITY

The Remote Support Facility (RSF) is an optional feature of the 4331 Processor (specify feature when the processor is ordered) 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 Models 158 and 168 and 303X Processors.

Via the Remote Console mode, RSF enables a remote customer engineer specialist to control operation of a malfunctioning 4331 Processor for diagnostic purposes utilizing a 3275 terminal. While in this mode, the remote specialist can perform online diagnosis as though he were at the customer installation. In addition, microcode patches can be applied to a system diskette using data link mode.

Operation of the Remote Support Facility is controlled by the service processor, and the 4331 Processor is dedicated to service functions while RSF is active. The RSF facility consists of support processor microcode to control the operation of RSF, a communications common adapter attached to the I/O bus of the support processor, and a 1200-baud modem in the 4331 Processor. These facilities handle communications functions for the Remote Support Facility. Autodial and autoanswer are not supported.

In order to utilize RSF, a 4331 Processor installation must install a point-to-point, switched communications line, which is connected to remote 3270 display devices that are operated by customer engineer specialists. This communications 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 Utilities Program Selection display and select the Remote Support option from the display. CE mode must be in effect in order to perform these selections. CE mode is established using a CE panel contained within the frames of the 4331 Processor. Using the Remote Support Selection display, the customer engineer selects the Console Support item to activate Remote Console Mode.

Remote Console Mode Functions

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 4331 Processor for diagnostic purposes.

When Remote Console mode is activated, the local customer engineer uses the Remote Console Support display to select the Remote Console entry and enter identifying information about the 4331 Processor (for example, machine type and serial number, associated branch office, and customer number). A security check is made to ensure that remote control becomes effective only when the installation has given permission for RSF to be used.

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 inline tests and 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 or support processor microcode
- Display system status data

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. Communication can be requested by the local or remote customer engineer by pressing the COMM REQ request 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 remote support selection display. The communication display appears on both consoles.

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 3275 terminal.

Installation and use of the Remote Support Facility is 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: HARDWARE AND I/O DIFFERENCES BETWEEN MODEL 20 AND 4331 PROCESSOR CONFIGURATIONS

The 4331 Processor is an attractive growth system for users of disk- and tape-oriented Model 20 systems who require greater job throughput or who want to expand the number and types of applications installed.

For greater throughput, the 4331 Processor offers significantly increased internal performance, new hardware functions, many times the amount of processor storage, significantly expanded channel capability, attachment of many more and faster I/O devices, and Disk Operating System/Virtual Storage Extended (DOS/VSE) as the programming support.

The 4331 Processor is designed to handle scientific as well as commercial applications and offers telecommunications capability that greatly exceeds that of the Model 20. Because considerably more direct access storage can be attached to a 4331 Processor than to a Model 20, the 4331 Processor is more suitable for applications that require large amounts of online data. A wide variety of IBM-supplied application-oriented programs that operate under DOS/VSE are also available, many more than are provided for the Model 20.

Because the 4331 Processor offers some facilities that are not compatible with the Model 20, more effort is involved in the transition to a 4331 Processor for a Model 20 user than for Model 30 or 40 users. The extent of this effort depends on whether applications are redesigned to take advantage of new hardware, new I/O devices, and additional programming systems features or merely modified to operate on a 4331 Processor under DOS/VSE.

Previous hardware sections of this guide have assumed knowledge of the Model 30 or 40. This subsection assumes knowledge only of the Model 20 (Submodel 5). However, it is not the aim of this discussion to give detailed explanations of concepts and hardware common to Models 22 and up and the 4331 Processor that are new to the Model 20 user. These differences are highlighted only and additional information can be found in the System/370 and 4300 Processor Principles of Operation publications.

60:05 HARDWARE DIFFERENCES BETWEEN THE MODEL 20 AND THE 4331 PROCESSOR

The hardware features of the System/360 Model 20 are identical to those of System/360 Models 22 and up and to the 4331 Processor (for System/370 mode) in many respects. Model 20 data formats, instruction formats, and all but a few instructions are a compatible subset of those for other System/360 models and the 4331 Processor.

The following are the significant feature differences between the Model 20 and the 4331 Processor. The 4331 Processor features described in Sections 05, 10, 15, and 18 that are new to Models 30 and 40 are new to the Model 20 as well but are not highlighted again in this section.

- In addition to the data formats for the Model 20, which are identical to their 4331 Processor counterparts, the 4331 Processor implements fullword (32-bit) binary data, instead of halfword only, and short, long, and extended form floating-point data formats. Hence, the 4331 Processor can process scientific applications that require the range of numbers that can be represented by floating-point notation.

- The Model 20 implements four of the five instruction formats used for 4331 Processor instructions. The register-to-storage (RS) format is not provided on the Model 20. The RS format permits reference to three operands, two in registers and one in storage. For example, in the 4331 Processor, two or more consecutively addressed general registers can be loaded with data from storage using a single LOAD MULTIPLE (RS format) instruction. Similarly, the contents of several registers can be stored with one STORE MULTIPLE instruction. These two instructions are very useful in an operating system environment because they permit rapid register unloading and loading when processor control is switched from one program to another (control program to problem program, for example).

- The Model 20 instruction set consists of 36 operation codes (including I/O instructions and excluding the DIAGNOSE instruction). More than 190 instructions are provided for the 4331 Processor. All Model 20 instruction operation codes are identical to their 4331 Processor counterparts except for Branch and Store, Set PSW, Halt and Proceed, and the three I/O instructions, none of which are part of the 4331 Processor instruction set.

  Some of the new programming capabilities offered by the 4331 Processor instruction set (in addition to those discussed in Section 10) are binary division, fullword binary arithmetic, floating-point arithmetic, additional logical operations (add and subtract logical instructions, register shifting, exclusive OR), decimal to binary or binary to decimal data conversion with a single instruction, single instruction programmed loop control, scanning for specific data characters, and additional operations involving the general registers. These instructions eliminate the need for writing subroutines to handle functions not supported by a single instruction, thereby saving programmer time and storage space and speeding up program execution. Table 60.05.1 lists the instructions for the 4331 Processor that are not provided for the Model 20.

- The Model 20 has 8 general registers (addressed 8 to 15) that are a halfword in size. The 4331 Processor has 16 general registers (0 to 15) that are a fullword in size. The extra registers are needed for use as base registers (to address the significantly greater processor storage size of the 4331 Processor) and for indexing. Additional registers can be used as intermediate work storage during arithmetic calculations and for loop control to eliminate loading and unloading operations. The registers are also used for subroutine linkage, for passing data between subroutines, and for communication between a control program and problem programs.

- During system operation, the 4331 Processor operates in either supervisor state or problem state. The state in effect is determined by a bit setting in the current PSW. In supervisor state, all 4331 Processor instructions can be executed. In problem state, certain instructions, those referred to as privileged, cannot be executed. If an attempt is made to execute a privileged instruction when the system is in problem state, an interruption occurs.

Table 60.05.1. 4331 Processor instructions not available on the Model 20. (Privileged instructions are identified by an asterisk.)

Arithmetic Operations

Add (fullword binary)
Subtract (fullword binary)
Multiply (fullword binary)
Divide (fullword binary)
Add Logical (fullword binary)
Subtract Logical (fullword binary)
All short, long, and extended precision floating-point instructions. (There are 44 instructions to handle short- and long-form data and 7 to handle extended precision.)

Logical Operations

AND (SS, RX, RR formats)
OR (SS, RX, RR formats)
Exclusive OR

Radix Conversion

Convert to Binary (from decimal)
Convert to Decimal (from binary)

Register Shifting Operations

Algebraic and logical shifting, to the left or the right of one register or a pair of registers

Miscellaneous

Branch and Link (similar to Branch and Store)
Edit and Mark
Execute
*Load PSW (similar to Set PSW on the Model 20)
*Set Program Mask
*Set System Mask
Supervisor Call
Test and Set
Translate and Test (for data scanning)

I/O Instructions

*Start I/O
*Halt I/O
*Test Channel
*Clear I/O
*Halt Device

Storage Protection

*Insert Storage Key
*Set Storage Key

Register Loading Operations

Load (fullword)
Load Address
Load and Test
Load Complement
Load Multiple (registers)
Load Negative
Load Positive
Insert Character (into a register)

Register Storing Operations

Store (fullword)
Store Multiple (registers)
Store Character (from a register)

Comparison Operations

Compare (fullword)
Compare Logical (RR, RX formats)

Loop Control Operations

Branch on Count
Branch on Index High
Branch on Index Low or Equal

New 4331 Processor Instructions (Not implemented in System/360)

Clear Storage Page
Compare and Swap
Compare Double and Swap
Compare Logical Characters Under Mask
Compare Logical Long
Insert Characters Under Mask
*Insert PSW Key
*Load Control
*Load Real Address
Monitor Call
Move Long
*Purge Translation Lookaside Buffer
*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
*Storage Then Or System Mask
*Page control instructions valid only in ECPS:VSE mode
Move Inverse

The resident control program operates in supervisor state, the problem program in problem state. This prevents a problem program from interfering with critical system-oriented functions, such as I/O operations, mode switching, storage protection, interruption masking, etc., that are handled by the control program. Privileged instructions for the 4331 Processor are identified by an asterisk in Table 60.05.1. The Model 20 does not implement these two states and, therefore, does not have any instructions that are privileged.

- User-written programs for the Model 20 can use direct addressing, that is, no base register, for addresses up to 16,383. The high-order bit of the B-field in a Model 20 instruction indicates whether direct addressing is used, and the low-order 14 bits of the combined B- and D-fields are used for the direct address.

  In the 4331 Processor, only addresses 0 to 4095 can be directly addressed. A value of zero in the B-field of a 4331 Processor instruction indicates direct addressing and the 12-bit D-field is used for the address. However, programs written to operate under DOS/VSE must use base registers for all addressing. The advantage of this approach is that a program can be relocated merely by adding a relocation factor to base register values and all address constants, either at link-edit or program execution time. Reassembly is not required.

- The PSW for the Model 20 is one word. The PSW for the 4331 Processor is a doubleword and different in format.

- A channel command word (CCW) for the Model 20 is six bytes and must be aligned on a halfword boundary. CCWs are used only for I/O operations on the Input/Output Channel. In the 4331 Processor, a CCW is eight bytes and must be aligned on a doubleword boundary. CCWs are used for all I/O operations in the 4331 Processor, regardless of the type of channel to which the I/O device is attached and whether or not the device is connected to the processor via an integrated adapter or a channel.

- The Model 20 implements only the I/O class of interruption. When a programming or a machine error occurs, the Model 20 halts. The 4331 Processor implements five interruption classes: I/O, external, program, supervisor call, and machine check. An old and a new PSW are provided for each class. The implementation of more interruption classes is designed to make the 4331 Processor hardware suited to an operating system environment in which total system halts are to be avoided when at all possible. (See Section 50 for a discussion of 4331 Processor recovery features.) Interruptions are also the means by which processor control is switched from problem programs to the control program.

- The reserved area in lower processor storage of the Model 20 is 168 bytes. Locations 0 to 143 are reserved for internal processor control and cannot be accessed by any program. There is no machine error logout area. The reserved area in lower storage in the 4331 Processor is 512 bytes. It contains fixed locations for specific data fields (PSWs, etc.) and logout areas into which the processor stores data during interruptions. These 512 bytes are program addressable. (See Section 10:05 for the layout of the 512 permanently assigned locations in the 4331 Processor.)

- The 4331 Processor includes storage and fetch protection as a standard feature. This feature, not available on the Model 20, is most useful in a multiprogramming operating system environment. Storage protection is used to prevent one program from destroying another by inadvertently storing data in that program's area of storage. Thus, the resident control program can be protected from

problem programs that are executing concurrently, and the problem programs are in turn protected from one another. Fetch protection can be used to prevent a program from fetching data from a storage area other than its own.

Storage protection is achieved by dividing program processor storage into 2048-byte (2K) blocks. The first address of each 2K block is a multiple of 2048. Each 2K block of storage is assigned a four-bit storage protect key, 0 to 15, using the SET STORAGE KEY instruction. (The protect key is not part of addressable processor storage.) Each time an attempt is made to store at any storage location during system operation, the system compares the storage protect key in the current PSW with the key in effect for the storage area in which the store instruction appears. If the protect keys match or the current PSW protect key is zero, the store operation is performed; if not, the instruction is not executed and a program interruption occurs.

Storage protection for I/O operations is also provided. When a read operation is initiated for a program, the storage protect key of that program is given to the channel, which then ensures that data read is placed only in storage locations with the same key.

In an operating system environment, the resident control program storage area is assigned protect key 0, and operates with protect key 0 in the PSW so that it can store in any storage area. A problem program storage area is assigned a key of from 1 to 15 and operates with its own key in the current PSW so that a problem program can store only in its own storage area. The assignment of storage keys and placing of the key in the current PSW are handled by the control program.

- There are no hardware timing facilities in the Model 20. The 4331 Processor has an interval timer (for compatibility with System/360 and System/370), CPU timer, clock comparator, and time-of-day clock (discussed in Section 10). These facilities are used for time of day, job accounting, interval timing, and time-stamping operations.

60:10 CHANNEL AND I/O DEVICE DIFFERENCES BETWEEN THE MODEL 20 AND THE 4331 PROCESSOR

CHANNELS

The 4331 Processor is basically a channel-oriented system and, therefore, offers considerably more capability for overlapping I/O operations with each other and with instruction processing than does the Model 20 Submodel 5. The integrated (native) adapters available for the 4331 Processor differ from those on the Model 20 in that they offer I/O overlapping functions comparable to those of channels rather than the limited overlap provided by the "time-sharing" mode of the Model 20. I/O devices connected to the 4331 Processor via integrated adapters are programmed exactly as if they were attached via a control unit and a channel. The advantage of integrated adapters is that they provide lower cost attachment of certain I/O units than would be the case if a separate control unit were used and reduce physical space requirements.

A block multiplexer channel in a 4331 Processor operating in selector mode (hereafter referred to as a selector channel) can best be compared in function and operation with the Input/Output Channel operating in overlap mode on a Model 20 Submodel 5. A selector channel in the 4331 Processor operates only in burst mode and, therefore, can handle only one data transfer operation at a time, as is true of the Model 20 Input/Output Channel. A selector channel is designed for the attachment of higher speed devices, such as tape and disk, but slower speed

devices, such as card and print units, display devices, etc., can be attached as well.

I/O operations on a selector channel can be overlapped with other I/O operations in progress on both integrated adapters and the byte multiplexer channel, and with instruction processing operations. A data transfer operation on a selector channel can also be overlapped with previously initiated non-data-transfer operations on the same channel, such as tape rewinding and direct access device arm positioning (seeking). Once a data transfer I/O operation is begun on a selector channel, the channel always operates asynchronously from other channels, integrated adapters, and instruction processing. There is no nonoverlap mode, as on the Model 20.

A selector channel causes instruction processing operations in the 4331 Processor to be temporarily suspended when the channel needs a component it shares with the instruction processing function (such as the arithmetic logic unit) and when the instruction processing function and the channel simultaneously require access to processor storage. The channel is given priority over the instruction processing function for storage accesses and the instruction processing function must wait until the channel stores data in, or fetches data from, processor storage. The channel informs the instruction processing function of the termination of an operation via an I/O interruption, during which channel and I/O device status information is stored in permanently assigned lower storage.

A selector channel can have up to eight control units of different types attached, and up to 256 I/O devices per channel can be addressed. Unlike Model 20 DPS, which supports only one control unit, DOS/VSE supports multiple control units on a selector channel.

There is no channel facility on the Model 20 comparable to the byte multiplexer channel in the 4331 Processor. This channel is provided to handle slower speed I/O devices, such as card readers, punches, printers, telecommunications terminals, and optical and magnetic character readers. While the byte multiplexer channel can operate in burst mode, like a selector channel, to handle one I/O operation at a time, it is designed primarily for operation in byte-interleaved mode. When operating in byte mode, the byte multiplexer channel can handle the concurrent execution of multiple data transfer operations for slower speed devices.

The byte multiplexer channel actually consists of multiple subchannels. Each subchannel can handle only one I/O operation at a time but can operate concurrently with all other byte multiplexer subchannels as long as the maximum aggregate data rate that can be sustained by the channel is not exceeded. A given subchannel is assigned to handle I/O operations for only one device or to handle I/O operations for multiple I/O devices attached to the same control unit. In the latter case, only one I/O device on the control unit can operate at any given time.

When the channel is in byte mode, concurrently operating subchannels share the facilities of the byte multiplexer channel (registers, microcode, etc.) on an as-needed, interleaved basis to transfer data between processor storage and the subchannels one byte at a time. When an I/O operation is started on a subchannel for a device, the device does not retain control of the channel facilities until the entire data record has been transferred, as is true in burst mode. Instead, requests from the device to transfer the bytes of a given data record are handled on an interleaved basis with data transfer requests of other operating devices.

Because byte multiplexer channel facilities must be shared when the channel operates in byte mode, each subchannel has its own unit control word (UCW), which is used as an intermediate storage area during each I/O operation on the subchannel. The UCW contains the information needed to continue the I/O operation when the subchannel again receives control, such as the number of bytes remaining to be transferred, the next storage location to be used, the storage protect key, etc.

Assuming several I/O operations have been started on the byte multiplexer channel, the following occurs. When any control unit is ready for a data transfer operation, it sends a request signal to the byte multiplexer channel. When free, the byte multiplexer channel responds by sending back a signal that inspects each control unit on the channel, in the physical sequence in which they are attached.

The first control unit found with a high-priority designation that is ready to transfer one byte of data is the one that will be serviced. (Physical sequence on the channel and high- or low-priority designation for I/O devices are established at installation time.) Control of the byte multiplexer channel facilities is given to the appropriate subchannel (the one associated with the device on the control unit whose data byte is ready for transfer). The contents of the UCW for that subchannel are loaded into the channel registers, the data transfer operation of one byte to or from storage is performed, the channel registers are updated, and the updated values are stored back in the UCW for the subchannel.

The byte multiplexer channel is then available to service another data transfer request. Thus single-byte data transfer operations for concurrently operating I/O devices are handled on an interleaved basis. This can be done because of the relatively long period of time between the availability of consecutive data bytes from any one device on the channel. During this interval other devices can be serviced.

The byte multiplexer channel, like a selector channel, can have up to eight control units of different types attached, and up to 256 I/O devices can be addressed. However, the actual number of I/O devices that can be attached is a function of the number of subchannels for the byte multiplexer channel, which in the 4331 Processor is 31, and the types of devices attached.

The byte multiplexer channel provides much more overlap of I/O operations than the "time-sharing" facility of the Model 20 that is implemented for slower speed, natively attached I/O devices. The block multiplexer channel offers significant advantages over a selector channel and is discussed in Sections 10:20 and 10:25.

I/O DEVICES

Model 20 users who install a 4331 Processor have a significantly wider variety of I/O device types from which to choose and can install a much larger I/O configuration that includes many faster I/O units. Table 60.05.2 shows the correspondence between Model 20 and 4331 Processor I/O devices and features. The following summarizes the contents of the table:

- The 2152 Printer-Keyboard, the 2203 Printer, and the 2560 MFCM cannot be attached to the 4331 Processor.
- Models of the 1442, the 2501, and the 2520 card units that attach to the Model 20 cannot themselves be attached to the 4331 Processor; however, these card units can be replaced with (but not physically converted to) appropriate models of the same units that do attach to the 4331 Processor. Alternatively, faster card readers and punches,

not attachable to the Model 20, can be installed in a 4331 Processor configuration to replace these card units.

- 2311 Models 11 and 12 cannot be attached to the 4331 Processor but they can be field-converted to the 2311 Model 1, which attaches to the 4331 Processor. Upgrades to larger capacity disk storage, such as the 3310 or 3370, are desirable. At least one non-2311 device type must be installed in order to use DOS/VSE, which does not support the 2311 as a system residence device.

- 2401 and 2415 Magnetic Tape Units as well as 1255, 1259, and 1419 Magnetic Character Readers attached to a Model 20 can be attached to a 4331 Processor. Alternatively, 2400-series tape units can be replaced with units ranging in speed from 20 KB/sec to 470 KB/sec and different tape types can be intermixed in the 4331 Processor tape configuration.

New device types, not available on the Model 20, include optical character readers, paper tape units, alphameric display units, audio response units, and start/stop and synchronous data link control telecommunications terminals. Table 60.05.3 lists I/O devices for the 4331 Processor that are supported by DOS/VSE but that cannot be attached to the Model 20.

Table 60.05.2. Model 20 and 4331 Processor I/O device correspondence

Model 20 I/O Device	4331 Processor I/O Device
• 2501 Card Reader A1 - 600 cpm read A2 - 1000 cpm read	Models A1 and A2 cannot be attached to the 4331 Processor or converted to B models. • 2501 Card Reader B1 - 600 cpm read B2 - 1000 cpm read • 2520 Card Read Punch B1 - 500 cpm read 500 cpm punch • 2540 Card Read Punch 1000 cpm read 300 cpm punch • 3505 Card Reader B1 - 800 cpm read B2 - 1200 cpm read
• 2520 Card Read Punch A1 - 500 cpm read 500 cpm punch	Model A1 cannot be attached to the 4331 Processor or converted to a Model B1. • 2520 Card Read Punch B1 - 500 cpm read 500 cpm punch • 2540 Card Read Punch 1000 cpm read 300 cpm punch • 3505 Card Reader and 3525 Card Punch B1, B2 - 800 or 1200 cpm read P1, P2, P3 - 100, 200, or 300 cpm punch • 2501 Card Reader and 2520 Card Punch B1, B2 - 600 or 1000 cpm read B2, B3 - 500 or 300 cpm punch

Table 60.05.2. (continued)

Model 20 I/O Device	4331 Processor I/O Device
• 1442 Card Punch Model 5 - 160 columns per second • 2520 Card Punch A2 - 500 cpm punch A3 - 300 cpm punch	The 1442 Model 5 and Models A2 and A3 of the 2520 Card Punch cannot be attached to the 4331 Processor or converted to N2 or B models. • 1442 Card Punch N2 - 160 columns per second • 2520 Card Punch B2 - 500 cpm punch B3 - 300 cpm punch • 3525 Card Punch P1, P2, P3 - 100, 200, or 300 cpm punch • 2540 Card Read Punch 300 cpm punch 1000 cpm read • 2520 Card Read Punch B1 - 500 cpm punch 500 cpm read
• 1403 Printer 2 - 600 lpm, 132 print positions 7 - 600 lpm, 120 print positions N1 - 1100 lpm, 132 print positions UCS and Selective Tape Listing features are supported.	• 1403 Printer Models 2, 3, 7, N1 Model 3 - 1100 lpm and 132 positions • 3203 Model 5 Printer 1200 lpm and 132 positions • 3211 Printer 2000 lpm and 132 or 150 positions The UCS feature is supported. The Selective Tape Listing feature cannot be installed on a printer attached to a 4331 Processor.
• 1255, 1259, or 1419 Magnetic Character Reader	The 1255 and 1419 attach
• 2415 Magnetic Tape Unit and Control Models 1, 2, 3 15 KB/sec (NRZI) Models 4, 5, 6 30 KB/sec (PE) • 2401 Magnetic Tape Unit Model 1 30 KB/sec (NRZI) Model 2 60 KB/sec (NRZI) Model 4 60 KB/sec (PE) 2401 and 2415 tape units cannot be intermixed in one system. A maximum of six tape units, all of the same type (2415 or 2401), can be attached.	• 2415 Magnetic Tape Unit and Control Models 1, 2, 3 15 KB/sec (NRZI) Models 4, 5, 6 30 KB/sec (PE) • 2401 Magnetic Tape Unit Model 1 30 KB/sec (NRZI) Model 2 60 KB/sec (NRZI) Model 3 90 KB/sec (NRZI) Model 4 60 KB/sec (PE) Model 5 120 KB/sec (PE) Model 6 180 KB/sec (PE) Model 8 15 KB/sec, 41.7 KB/sec, or 60 KB/sec (NRZI, 7-track only) • 3410 Magnetic Tape Model 1 20 KB/sec (PE) Model 2 40 KB/sec (PE) Model 3 80 KB/sec (PE) • 3420 Magnetic Tape Model 3 120 KB/sec (PE) Model 5 200 KB/sec (PE) Model 7 320 KB/sec (PE) Model 4 470 KB/sec (GCR) • 8809 Magnetic Tape Unit (20 KB/sec or up to 160 KB/sec) Different tape types can be intermixed in a system configuration.

Table 60.05.2. (continued)

Model 20 I/O Device	4331 Processor I/O Device
• 2311 Disk Storage Drive Model 11 - 5.4 million bytes Model 12 - 2.7 million bytes A maximum of four drives can be attached to a Submodel 5 for a total of 21.6 million bytes online.	Models 11 and 12 cannot be attached to a 4331 Processor but can be changed to a 2311 Model 1 in the field. • 2311 Disk Storage Drive Model 1 Up to 7.25 million bytes per drive. Up to eight drives on one 2841 control unit for a maximum capacity of 58 million bytes per 2841. 1316 Disk Pack used on 2311 Models 11 and 12 can be used on a Model 1 but data must be reformatted. • 2314/2319 disk storage facilities One to nine drives in a 2314B/2319B facility, one of which is a spare. Up to 29.1 million bytes per drive, 233 million maximum online per facility. 2316 Disk Pack used is larger than the 1316 pack. • 3340 disk storage (all models) One or two strings of from two to eight drives each. Up to 35 or 70 million bytes per drive, 560 million bytes maximum per string. The 3348 Data Module is used as the storage medium. • 3310 disk storage. From one to four strings of from one to four drives each. Drive capacity is 64.5 million bytes, string capacity is 258 million bytes. • 3370 disk storage. From one to four strings of from one to four drives each. Drive capacity is 571.3 million bytes, string capacity is 2.3 billion bytes.
• 2152 Printer-Keyboard	Cannot be attached. The 3278 Model 2A Display Console is required.

Table 60.05.2. (continued)

Model 20 I/O Device	4331 Processor I/O Device
• 2203 Printer Up to 350 lpm with a 52-character set, depending on the model, and 120 or 144 print positions. Only one printer (2203 or 1403) can be included in a Model 20 configuration.	Cannot be attached. The following can be substituted: • 1443 Printer Model N1 240 lpm and 120 or 144 print positions • 1403 Printer Models 2, 3, 7, N1 600 or 1100 lpm and 120 or 132 print positions • 3205 Model 5 Printer 1200 lpm and 132 print positions • 3211 Printer 2000 alphameric lpm and 132 or 150 print positions. UCS is standard. • 3262 Model 1 Printer Up to 650 lines per minute • 3289 Model 4 Printer Up to 400 lines per minute Multiple printers can be included in a 4331 Processor configuration. The Dual Feed Carriage special feature cannot be installed on printers attached to the 4331 Processor.
• 2560 Multi-Function Card Machine (MFCM) Card reading - 310 or 500 cpm Punching - 120 or 160 columns per second Card printing - on 25 lines, up to 6 lines per pass. Combined read, punch, print operations in each feed. Card selection into five stackers from either feed.	Cannot be attached. The following can be subsititued to handle some functions: • 2540 Card Read Punch Card reading - 1000 cpm Card punching - 300 cpm Combined reading/punching from punch side if Punch Feed Read installed. Selection into three pockets from read feed and into two pockets from punch feed (or vice versa). • 3505 Card Reader and 3525 Card Punch Card reading - 800 or 1200 cpm Card punching - 100, 200, or 300 cpm Card printing - 100, 200, or 300 cpm (for two-line feature, variable for multiline feature) Combined read, punch, print in 3525 if card read and card print options installed. Card selection into two pockets from 3505 (if optional stacker present) and two pockets from 3525.

Table 60.05.3. I/O devices for the 4331 Processor that are supported by DOS/VSE but not attachable to the Model 20

I/O Devices
3505 Card Reader
3525 Card Punch
1443 Printer
3203 Model 5 Printer
3211 Printer
3262 Line Printer
3289 Line Printer, Model 4
3800 Printing Subsystem
2671 Paper Tape Reader
1287 Optical Reader
3881 Optical Mark Reader
3886 Optical Character Reader
3277 Display Station
3410 Magnetic Tape Models 1,2,3
3420 Magnetic Tape Models 3,4,5,7
2401 Magnetic Tape Models 3,5,6,8
8809 Magnetic Tape Unit
2314/2319 Disk Storage
3310 Disk Storage
3340 Disk Storage
3370 Disk Storage
3540 Diskette Input/Output Device
2596 Card Read Punch
3890 Document Processor
3895 Deposit Processing System
5424 Multifunction Card Unit
7770 Audio Response Unit

SECTION 70: COMPARISON TABLE OF HARDWARE FEATURES - SYSTEM/360 MODELS 20, 30, AND 40 AND THE 4331 PROCESSOR

This table is included for quick reference. It compares the hardware features of Models 20, 30, and 40, and the 4331 Processor.

70:05 HARDWARE FEATURES - SYSTEM/360 MODELS 20, 30, AND 40 AND THE 4331 PROCESSOR

Hardware Feature	System/360 Model 20	System/360 Model 30	System/360 Model 40	4331 Processor
I. PROCESSOR				
A. Modes of operation	Only BC mode is implemented	Only BC mode is implemented	Only BC mode is implemented	BC and EC modes for System/370 and ECPS:VSE modes are standard
B. Instruction set				
1. Standard set (Binary arithmetic)	Standard (halfword operations only and no division)	Standard	Standard	Standard
2. Decimal arithmetic	Standard	Optional	Optional	Standard
3. Floating-point arithmetic	Not available	Optional	Optional	Standard
4. Extended precision floating point	Not available	Not available	Not available	Standard
5. New instructions (listed in Section 10:30)	Not available	Not available	Not available	Standard
C. Address translation hardware	Not available	Not available	Not available	Standard (dynamic address translation for System/370 mode and an internal mapping function for ECPS:VSE mode)
D. Channel indirect data addressing	Not available	Not available	Not available	Standard (for use in System/370 mode only)
E. Interval timer	Not available	Optional (16.6-ms resolution)	Optional (16.6-ms resolution)	Standard (10-ms resolution)
F. Time-of-day clock	Not available	Not available	Not available	Standard (16-microsecond resolution)
G. Clock comparator and CPU timer	Not available	Not available	Not available	Standard
H. Instruction retry by hardware	No	Limited	No	No
I. Machine check interruption	Not implemented. Machine errors cause a system halt.	Occurs on processor, main storage, and certain channel errors. One mask bit controls this interruption.	Same as the Model 30	Occurs after uncorrected errors. There are four types of machine check and two mask bits.

Hardware Feature	System/360 Model 20	System/360 Model 30	System/360 Model 40	4331 Processor
J. Fixed lower storage area size (including logout area for machine and channel errors)	168 bytes (There is no logout area. Locations 0-143 cannot be accessed by programming.)	139 bytes	324 bytes	512 bytes
K. Compatibility features (Optional unless otherwise indicated)	1. 1401/1440 (submodel 5 only)	1. 1401/1440/1460 2. 1620 (mutually exclusive features)	1. 1401/1460 2. 1410/7010 3. 1401/1440/1460 DOS Compatibility (for use with CS/40)	1401/1440/1460 Compatibility
L. Proccesor cycle time	Half-byte data flow (submodels 1-4). Two-byte data flow (submodel 5)	750 nanoseconds. one-byte data flow.	625 nanoseconds. Two-byte parallel data flow.	Variable from 200 to 1600 nanoseconds in 100-nanosecond increments. Four-byte parallel data flow
M. Direct Control feature or External Interrupt feature	Not available	Optional (mutually exclusive features)	Optional (mutually exclusive features)	External signals is optional, Direct Control is not available.
N. Monitoring feature	Not available	Not available	Not available	Standard
O. Program event recording	Not available	Not available	Not available	Standard
P. Interruption for SSM instruction	Not available	Not available	Not available	Standard
Q. 2311/2314/2139/3310 Direct Access Storage Compatibility	Not available	Not available	Not available	Optional
S. System/3 Data Import	Not available	Not available	Not available	Optional
T. ECPS:VM/370	Not available	Not available	Not available	Optional
II. STORAGE				
A. Processor (main) storage sizes	4K (submodels 1-4) 8K (all submodels) 12K (all submodels) 24K (submodel 5) 32K (submodel 5)	16K 24K 32K 48K	32K 128K 192K 256K	512K 1024K (less requirement for microcode use)

Hardware Feature	System/360 Model 20	System/360 Model 30	System/360 Model 40	4331 Processor
B. Processor storage cycle	3.6 microseconds for 1 byte (submodels 1-4). 2.0 micro-seconds for 2 bytes (submodel 5).	1.5 microseconds for 1 byte	2.5 micro-seconds for 2 bytes	.9 microsecond read and 1.3 microsecond write for 4 bytes
C. Processor storage validity checking	Parity checking on each byte. Errors are not corrected by hardware.	Same as Model 20	Same as Model 20	ECC checking on a fullword. Single-bit errors are cor-rected by hardware.
D. Control storage	Transformer read-only storage (sub-models 1-4). Reloadable core storage (submodel 5).	Card Capacitor read-only storage	Tape read-only storage	Reloadable monolithic (64K standard and 64K additional optional)
E. Byte-oriented operands	No	No	No	Standard
F. Storage and fetch protection	Not available	Storage protect is optional. Fetch protect is not available.	Same as Model 30	Standard
III. CHANNELS AND INTEGRATED I/O ATTACHMENTS				
A. Byte multiplexer channel (up to 8 control units)	Not available	Standard	Standard	One is optional
1. Subchannels provided		96 for all main storage sizes. A special feature permits systems with 32K or 64K to have 224 sub-channels.	32K-32 64K-64 128K-128 192K-128 256K-128	31 for either processor storage size
B. Selector channels (up to 8 control units)	Input/output channel for 2415 tape units (maximum of 1 on submodels 2, 4, and 5 only)	Optional, 1 or 2	Optional, 1 or 2	Selector mode provided for the optional block multiplexer channel
1. Channel data rate	60 KB/sec	312 KB/sec	312 KB/sec	500 KB/sec

Hardware Feature	System/360 Model 20	System/360 Model 30	System/360 Model 40	4331 Processor
C. Block multiplexer channels	Not available	Not available	Not available	One optional (500 KB/sec data rate)
D. Integrated attachments (for direct connection of I/O devices to the processor without a channel and control unit)	Attachments are provided for: 1. 2203 and 1403 Printers 2. 2560 MFCM 3. 1442, 2501, and 2520 reader punches 4. 1255, 1259, and 1419 MCR 5. 2415 tapes (up to 6 drives) for submodel 5 only 6. 2311 Model 11, 12 disk storage drives (maximum four drives on submodel 5) 7. Binary Synchronous Communications Adapter 8. Communications Adapter for attachment of STR devices	None available	None available	1. Display/Printer Adapter for attachment of operator console and up to seven additional displays and printers is standard. Display/Printer Adapter Expansion option permits another eight devices to be attached (total 16 devices). 2. One DASD Adapter for attachment of up to four disk strings is optional. 3. 3340 Direct Attach for attachment of one or two 3340 strings to DASD Adapter is optional. 4. System/3 Data Import for reading 3348 Data Modules written by a System/3 is optional. 5. One 8809 Magnetic Tape Unit Adapter for attachment of up to six 8809 tape units is optional. 6. One Communications Adapter for attachment of up to eight communications lines is optional. 7. One 5424 Adapter for the attachment of one 5424 is optional. 8. One diskette drive and included adapter is optional.
E. Channel retry data provided in a limited channel logout area after channel error	No	No	No	Yes
F. Channel-to-Channel Adapter	Not available	Optional	Optional	Not available and a channel in the 4331 Processor cannot be connected to an adapter installed in another processor.

Hardware Feature	System/360 Model 20	System/360 Model 30	System/360 Model 40	4331 Processor
IV. OPERATOR CONSOLE DEVICES	Console located on 2020. 2152 Printer-Keyboard optional.	1052-7 Printer-Keyboard--15 cps without alter/display mode	Same as Model 30	1. 3278 Model 2A Display Console is required (1052, 3210, and 3215 cannot be attached). Display mode is standard, printer-keyboard mode is optional. 2. Other displays (3278 Model 2 units) can be attached via the Display/Printer Adapter to be used as alternate or additional consoles, as supported by programming systems.
V. I/O DEVICES				
A. 3211 Printer	No	Yes	Yes	Yes
B. 3203 Model 5, 3262, and 3289 Model 4 printers	No	No	No	Yes
C. 3803/3420 Magnetic Tape Subsystem 1. Models 3, 5, and 7	No	Yes except Model 7. Models 3 and 5 cannot be attached to a byte multiplexer channel.	Yes except Model 7. Model 5 cannot be attached to a byte multiplexer channel.	Yes
2. Models 4, 6, and 8	No	No	No	Only Model 4 at 6250 BPI
D. 3410/3411 Magnetic Tape Subsystem	No	Yes	Yes	Yes
E. 8809 Magnetic Tape Unit	No	No	No	Yes (via 8809 Magnetic Tape Unit Adapter only)
F. Other tape units	2415 or 2401 Models 1, 2, and 4. (2415 and 2401 units cannot be intermixed in a system.)	All except 2420 Model 7	Same as Model [illegible]	Yes
G. Direct access devices (2311,2314,2319,2303 2301, and 2321)	2311 Models 11 and 12 only (maximum two drives on submodels 2 and 4, maximum four drives on submodel 5)	All except 2303 and 2301 drums. Only channel 1 can have 2314-type facilities present.	All except 2301 drums. Either channel 1 or channel 2 (but not both) can have 2314 facilities attached.	2311 and 2314/2319 only
H. 3310 disk storage	No	No	No	Yes (via DASD Adapter only)
I. 3330-series and 3350 disk storage	No	No	No	No
J. 3340 and 3344 disk storage	No	No	No	3340 only (via DASD Adapter only)

Hardware Feature	System/360 Model 20	System/360 Model 30	System/360 Model 40	4331 Processor
K. 3370 Disk Storage	No	No	No	Yes (via DASD Adapter only)
L. 3505 Card Reader and 3525 Card Punch	No	No	No	Yes
M. 5424 Multi-Function Card Unit	No	No	No	Yes
N. 3540 Diskette Input/Output Unit	No	No	No	Yes
O. 3600 Finance Communication System	No	No	No	Yes
P. 3650 Retail Store System	No	No	No	Yes
Q. 3660 Supermarket System	No	No	No	Yes
R. 3704, 3705-I, and 3705-II Communications Controllers	No	Yes, in 270X emulation mode only	Yes, in 270X emulation mode only	Yes
S. 3740 Data Entry System	No	Yes	Yes	Yes
T. 3767 Communication Terminal	No	Yes	Yes	Yes
U. 3770 Data Communication System	No	Yes	Yes	Yes
V. 3780 Data Communications Terminal	No	Yes	Yes	Yes
W. 3790 Communication System	No	No	No	Yes
X. 3800 Printing Subsystem	No	No	No	Yes
Y. 3850 Mass Storage System	No	No	No	No
Z. 3881 Optical Mark Reader	No	No	No	Yes
AA. 3886 Optical Character Reader	No	No	No	Yes
BB. 3890 Document Processor	No	No	No	Yes
CC. 3895 Deposit Processing System	No	No	No	Yes
DD. 3838 Array Processor	No	No	No	No
EE. 3250 Graphics Display System	No	No	No	Yes
FF. 3270 Information Display System	No	No	No	Yes

address space definition 88
architecture 15
  System/370 15
  4300 Processor 17
ASCII/EBCDIC mode 23
availability and serviceability features 260

basic control mode 18, 22
block multiplexer channel
  description 52
  general operation 53
  operation with RPS devices 64
  subchannels types by device 56
byte multiplexer channel 51
byte-oriented operands 31

CCH routine 274
change bit 124
channel indirect data addressing feature 125
channel logout 264
channel masking changes for EC mode 25
channel program translation
  ECPS:VSE mode 136
  System/370 mode 127
channel retry 264
Channel-to-Channel Adapter 46
channels 44
  block multiplexer 52
  byte multiplexer 51
  cycle stealing 46
  data flow between processor storage and the channels/adapters 47
  device addresses 49
  general description 44
  integrated channel 46
  interference 45
  maximum number 48
  priority 46
    cycle steal requests 48
    interruptions 48
  subchannels 48
  UCWs 48
check-stop 268
CLEAR I/O instruction 46
CLEAR PAGE instruction 129
clock comparator 33
command retry 264
Communications Adapter 60
  attachable devices
    for BSC lines 63
    for start/stop lines 64
    for SDLC lines 64
  configuration parameters
    specified by the customer engineer 62
    specified by the operator 62
  general description 60
  line speeds 61
COMPARE AND SWAP instruction 30
COMPARE DOUBLE AND SWAP instruction 30
COMPARE LOGICAL CHARACTERS UNDER MASK instruction 29

COMPARE LOGICAL LONG instruction 29
comparison table, Models 20, 30, and 40 and 4331 Processor hardware
  features 293
compatibility
  BC mode with System/360 18
  BC mode with System/370 19
  System/370 mode with System/370 19
  System/370 mode with ECPS:VSE mode 17
  1401/1440/1460 feature 250
  2311/2314/2319/3310 Direct Access Storage Compatability feature 177
CONNECT PAGE instruction 130
control registers 22
control storage buffer 22
control storage
  general description 37
  loading 77, 83
  microcode groups by feature 38
  requirements by microcode group 39
CPU timer 32
cycle time
  processor 21
  processor storage 34

DASD Adapter 58
data local storage 22
DECONFIGURE PAGE instruction 132
defect dictionary, processor storage 263
diagnostic functions 261
  IML tests 276
  Communications Adapter Trace and Dynamic Display 277
  inline tests 276
  microtests and manual operations 277
  Remote Support Facility 277
DISCONNECT PAGE instruction 131
Diskette drive 60
display console 75
Display/Printer Adapter 50
distributed application environments 14
DOS Release 26, using 243
DOS/VS Release 34, using 243
DOS/VSE
  compatibility with DOS/VS Release 34 232
  description 231
  differences from DOS/VS Release 34 231
  emulator support 250
  performance 2
  program products 236
  support of the 4331 Processor 233
  VSE/Advanced Functions 237
  VSE/Access Control-Logging and Reporting 242
  VSE/Fast Copy Data Set Program 242
  VSE/ICCF 241
  VSE/IPCS 242
  VSE/POWER 239
  VSE/VSAM 240
dynamic address translation facility
  addresses translated 122
  time to perform 118
  translation lookaside buffer 120
  translation process 117
  translation tables 113

ECC checking 262
ECPS:VM/370 159

ECPS:VSE mode
  address translation using the internal mapping function 136
  advantage 17
  architecture 17
  capacity counts 132
  channel program address translation 136
  compatibility with System/370 mode 18
  minimum program processor storage size 34
  page control instructions 130
  page frame assignment 133
  performance 2
  processor storage directory 134
  translation lookaside buffer 137
  virtual storage organization 129
  virtual storage page
    definition 129
    description 131
    states 130
EREP 275
ERPs 274
exigent machine check conditions 267
extended control mode 23
extended precision floating-point 30
external interruption masking 23
external page storage 166

features
  optional 74
  standard 72
fixed block architecture 65
  advantages 72
  basic channel program 67
  commands 66
  differences between CKD and FBA channel programs 71
  operation of commands 67
  subcommands
    Format Defective Block 69
    Read Data 70
    Read Duplicated Data 71
    Write and Check Data 70
    Write Data 70
  track formatting 66
fixed processor storage locations
  BC mode 26
  EC mode 27
functional adapter 200
fixed logout area 265

general purpose instructions 29

HALT DEVICE instruction 46

IBM Systems 1401/1440/1460 Emulator Program 250
  advantages 25
  components 250
  diagnostic commands 253
  generation and execution 251
  storage requirements 257
  supported features 256
  supported I/O devices 258
  tape and disk emulation 254
  unsupported features and devices 258
  user-written routines 253
IML 42, 77
indirect data address list 126

indirect data address word 126
INSERT CHARACTERS UNDER MASK instruction 29
INSERT PAGE BITS 135
INSERT PSW KEY instruction 30
instruction buffer 21
instruction nullification 124
instruction processing damage machine check 267
instruction processing function 21
instructions
  changes to for EC mode 27
  description of ECPS:VSE mode only 130
  description of new 28
  list of new 73
integrated channel 46
internal mapping function 128
internal performance 2
interruptions
  machine check 23, 264
  Models 30 and 40 23, 269
  other than machine check 23
  page translation exception 118
  segment translation exception 118
  SET SYSTEM MASK instruction 23
interval timer 32
interval timer damage machine check 267
I/O adapters
  Communications Adapter 60
  DASD Adapter 57
  Display/Printer Adapter 50
  Diskette Drive and Adapter 60
  5424 Adapter 60
  8809 Magnetic Tape Unit Adapter 59
  attachable to the 4331 Processor 165
  operation on block multiplexer channels 54-55
I/O devices
  attachable to the 4331 Processor 165
  operation on block multiplexer channels 54-55
IPL, processor state after 269

limited channel logout 264
LOAD CONTROL instruction 22
LOAD FRAME INDEX instruction 131
LOAD REAL ADDRESS instruction 117
logouts 265
long-term fixing 109

machine check code 266
machine check interruptions 264
  analysis and logging 269
  exigent 267
  MCAR support 273
  MCH support 273
  Models 30 and 40 269
  recording modes 267
  repressible 267
  types 266
machine save function 81
main storage (See processor storage.)
MAKE ADDRESSBLE instruction 131
MAKE UNADDRESSABLE instruction 131
MCAR routine, DOS/VSE 273
MCH routine, OS/VS1 273
Model 20 comparison with the 4331 Processor 28
monitoring feature 31
MOVE INVERSE instruction 30

MOVE LONG instruction 29
multiple requesting 65

nonpaged mode of program operation 110
nonshared subchannels 48

OLTEP 275
OLTs 275
operator console 75
optional features 74
OS/VS1
  performance 2
  RMS routines 273
  support of the 4331 Processor 245

page 106, 112
page access exception 108
page fault 108
page frame 107
page in 107
page out 107
page replacement algorithm 109
page states 130
page state exception 130
page table 113, 116
page translation exception 118
paged mode of program operation 110
paging
  definition 107
  effect on system performance 141, 144-147
paging device 107
performance in a virtual storage environment 139
  factors affecting 141
  increasing 147
  relationship to virtual storage size 144
permanent fixing 109
power system 271
processor
  cycle times 21
  logout area 26, 27
processor storage 34
  calculating amount available for program use 40
  cycle and fetch times 34
  ECC on 262
  fixed locations 26, 27
  minimums available for program use 34
  processor and program sections 34-36
  sizes 34
processor storage directory
  general description 35
  use in ECPS:VSE mode 134
  use in System/370 mode 115
program event recording 28
programming systems support
  DOS Release 243
  DOS/VS Release 242
  DOS/VSE 231
  general discussion 1
  OS/VS1 245
  VM/370 247
PSW
  BC and EC mode formats 24
  change to 24
PURGE TLB instruction 122, 137

RAS features 260
real storage 92
recovery features, RAS 260
reference and change recording 124
reference bit 124
reference code 270
reloadable control storage 37
Remote Support Facility 277
repressible machine check conditions 267
RESET REFERENCE BIT instruction 124
RETRIEVE STATUS AND PAGE instruction 82
RMS routines 273
RMSR 274
RPS (rotational position sensing) with block multiplexing 64

segment 106, 112
segment table 113, 116
segment translation exception 118
selector channel mode 52
SET CLOCK COMPARATOR instruction 33
SET CLOCK instruction 32
SET CPU TIMER instruction 33
SET PAGE BITS instruction 135
SET PSW KEY FROM ADDRESS instruction 30
SET SYSTEM MASK instruction interruption 23
shared subchannels 48
SHIFT AND ROUND DECIMAL instruction 29
short-term fixing 109
slot 106
standard features 72
storage
  control 37
  data local 22
  external page 166
  processor (main) 34
  protect key expansion 25
  real 92
  virtual (see virtual storage)
STORE CHANNEL ID instruction 29
STORE CHARACTERS UNDER MASK instruction 29
STORE CLOCK COMPARATOR instruction 33
STORE CLOCK instruction 32
STORE CONTROL instruction 22
STORE CPU ID instruction 29
STORE CPU TIMER instruction 33
store status function 81
STORE THEN AND SYSTEM MASK instruction 30
STORE THEN OR SYSTEM MASK instruction 30
subchannels 48
  block multiplexer 52
  byte multiplexer 51
  Communications Adapter 61
  DASD Adapter 57
  Display/Printer Adapter 51
  nonshared 48
  number required by device 54
  shared 48
  type required by device 56
  231X mode 52
  5424 Adapter 60
  8809 Magnetic Tape Unit Adapter 59
support bus adapter and support bus 44, 270
support processor subsystem 41
support processor 41, 269
support processor 44

system damage machine check 267
system diskette 43, 270
system diskette drive 43
system initialization 42
System/3 Data Import 58, 202
System/370 mode
  address translaton using DAT hardware 117
  addresses translated 122
  advantages 17
  architecture 16
  channel indirect data addressing 125
  channel program translation 127
  compatibility with ECPS:VSE mode 18
  compatibility with System/360 and System/370 16, 18
  minimum program processor storage size 34
  performance 2
  processor storage directory 115
  translation lookaside buffer 120
  virtual storage organization 111

technology
  System/360 14.2
  System/370 14.3
  4331 Processor
    advantages 14.9
    control storage 14.16
    dynamic versus static storage 14.12
    engineering design system 14.11
    logic 14.5
    processor storage 14.14
temporary fixing 109
time-of-day clock 32
timing facilities damage machine check 267
translation lookaside buffer
  ECPS:VSE mode 137
  System/370 mode 120

UCWs 48

Virtual Machine Facility/370 2, 152
virtual machines
  advantages 163
  definition 152
  general operation 154
virtual storage
  advantages 99
  definition 92
  need for 88
  organization
    for ECPS:VSE mode 128
    for System/370 mode 111
  relationship between size and performance 144
  resources required to support 139
virtual storage address fields 114
virtual storage page 106

warning machine check interruption 267

1401/1440/1460 Compatibility feature 250

2311/2314/2319/3310 Direct Access Storage Compatibility feature 177
  address correspondance 178
  buffers 180
  configuration limitations 59
  functions supported 177

general operation 179
I/O instruction simulation 183
mixed mode operations 177
read command emulation 182
search command emulation 181
seek command emulation 181
volume conversion 184
volume mapping 179
write command emulation 182

3203 Model 5 Printer 228

3262 Line Printer 50

3278 Model 2A Display Console
address comapre display 85
audible alarm 75
check control display 85
Communications Adapter displays 86
display controls and indicators 79
display mode 76
display/alter display 83
general description 75
interval timer display 86
keyboard 78
machine status area 86
maintenance 87
mode selection display 80
native displays and printers display 86
operator's control panel 77
printer-keyboard mode 76
program load display 82
security keylock feature 75
user diskette displays 86

3278 Model 2 Display Station 50

3287 Printer 50

3289 Model 4 Line Printer 50

3310 Disk Storage
advantages
231X users 179
3340 users 176
alternate sectors 169
capacity 167
command operation 169
description of drives 167
error detection and correction code 169
error retry functions 172
initialization 169
powering 168
statistics 173
string configurations 167
timing characteristics 167
track formatting 168
231X compatibility feature 177

3340 Direct Access Storage Facility
advantages summary 202
alternate tracks 191, 194, 195
attachment via the DASD Adapter 200
buffers 200
capacity 185

defect skipping 199
description of 3340 drives 185
error detection and correction code 196
features table 203
fixed head feature 187, 194, 196
intermixing 3340 and 3310 strings on the DASD Adapter 57
physical address bytes 199
read-only feature 189
rotational position sensing 200
seek verification 199
servo system 190
string configurations 185
timing characteristics 198

3340 Direct Attach feature 58

3348 Data Module, for the 3340 Direct Access Storage Facility
advantages 187
capacities 185
cylinder and read/write head layout
Model 35 192
Model 70 193
Model 70F 195
general description 185
initialization 190
layout of physical and logical tracks 189
loading and unloading 188
sealed cartridge design 187
track formatting 196
track layout on the recording surface 190

3370 Direct Access Storage
attachment via the DASD Adapter 57
physical characteristics 10

3410/3411 Magnetic Tape Subsystem
general description 221.1
table of characteristics 221.5

3803/3402 Magnetic Tape Subsystem
advantages 216
automatic threading and cartridge loading 207
data rates 205
data security erase command 211
densities 208, 213
Dual Density feature 208
engineering design 207, 213
general description 205
group-coded recording 213
MODE SET command 209
phase-encoded recording 206
reliability and serviceability features 211, 215
Seven Track feature 208
Single Density feature 208
table of characteristics 217
tape switching features 209, 214
Two-Channel Switch 211, 215
3420 Magnetic Tape Unit
Models 3, 5, 7 205
Model 4 212
3803 Storage Control
Model 1 205
Model 2 212
6250 Density feature 214
6250/1600 Density feature 214

7-Track NRZI feature 208

5424 Adapter 60

8809 Magnetic Tape Unit Adapter 59, 226

8809 Magnetic Tape Unit
attachment to the 4331 Processor 222
commands 224
general description 222
models 222
modes of operation 223
serviceability features 225
transport design 224

A Guide to the IBM 4331 Processor

GC20-1878-0

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:

_____

_____

_____

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.)

Note: Staples can cause problems with automated mail sorting equipment. Please use pressure sensitive or other gummed tape to seal this form.

GC20-1878-0

Reader's Comment Form

Cut or Fold Along Line

Fold and tape | **Please Do Not Staple** | Fold and tape

First Class
Permit 40
Armonk
New York

Business Reply Mail
No postage stamp necessary if mailed in the U.S.A.

Postage will be paid by:

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

A Guide to the IBM 4331 Processor Systems Printed in U.S.A. GC20-1878-0

IBM Technical Newsletter

This Newsletter No.	GN20-3935
Date	March 19, 1979
Base Publication No.	GC20-1878-0
File No.	S/370-01
Prerequisite Newsletters	None

A Guide to the
IBM 4331 Processor

This Technical Newsletter provides replacement pages for the subject publication. Pages to be inserted and/or removed are:

Contents
14.1 - 14.18 (added)
167, 168
221 - 222 (includes added pages 221.1 - 221.8)
305 - 308

Summary of Amendments

Two subsections have been added – one discussing the logic and storage technology implemented in the 4331 Processor and another describing the 3410/3411 Magnetic Tape Subsystem.

A vertical rule in the left margin indicates a change. Absence of a vertical rule on a page bearing a 'revised' notice means only that existing copy has been moved or that a minor typographical error has been corrected.

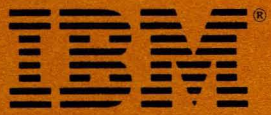
Please file this cover letter at the back of the manual to provide a record of changes.

IBM Corporation, Technical Publications, Dept. 824, 1133 Westchester Ave., White Plains, N.Y. 10604

Printed in U.S.A.

GC20-1878-0

A Guide to the IBM 4331 Processor Systems Printed in U.S.A. GC20-1878-0



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