GA22-7066-2
File No. S370-01

Systems

IBM 3031 Processor Complex and 3031 Attached Processor Complex Functional Characteristics

IBM

Second Edition (December 1978)
This edition is a major revision of, and obsoletes, GA22-7066-0 and Technical Newsletters GN22-0556 and GN22-0564. Information about the IBM 3031 Attached Processor Complex is included. A technical change to the text or an illustration is indicated by a vertical line to the left of the change. Changes are periodically made to the information herein; before using this publication in connection with the operation of IBM equipment, refer to the latest *IBM System/370 Bibliography,* GC20-0001, for the editions that are applicable and current.

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

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Product Publications, Dept. B98, PO Box 390, Poughkeepsie, NY U.S.A. 12602. 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.

This manual describes the functional characteristics and features of the IBM 3031 Processor Complex and 3031 Attached Processor Complex. This manual provides management, programming, and operations personnel experienced in System/370 operation with a fundamental understanding of the 3031 Processor Complex and 3031 Attached Processor Complex.

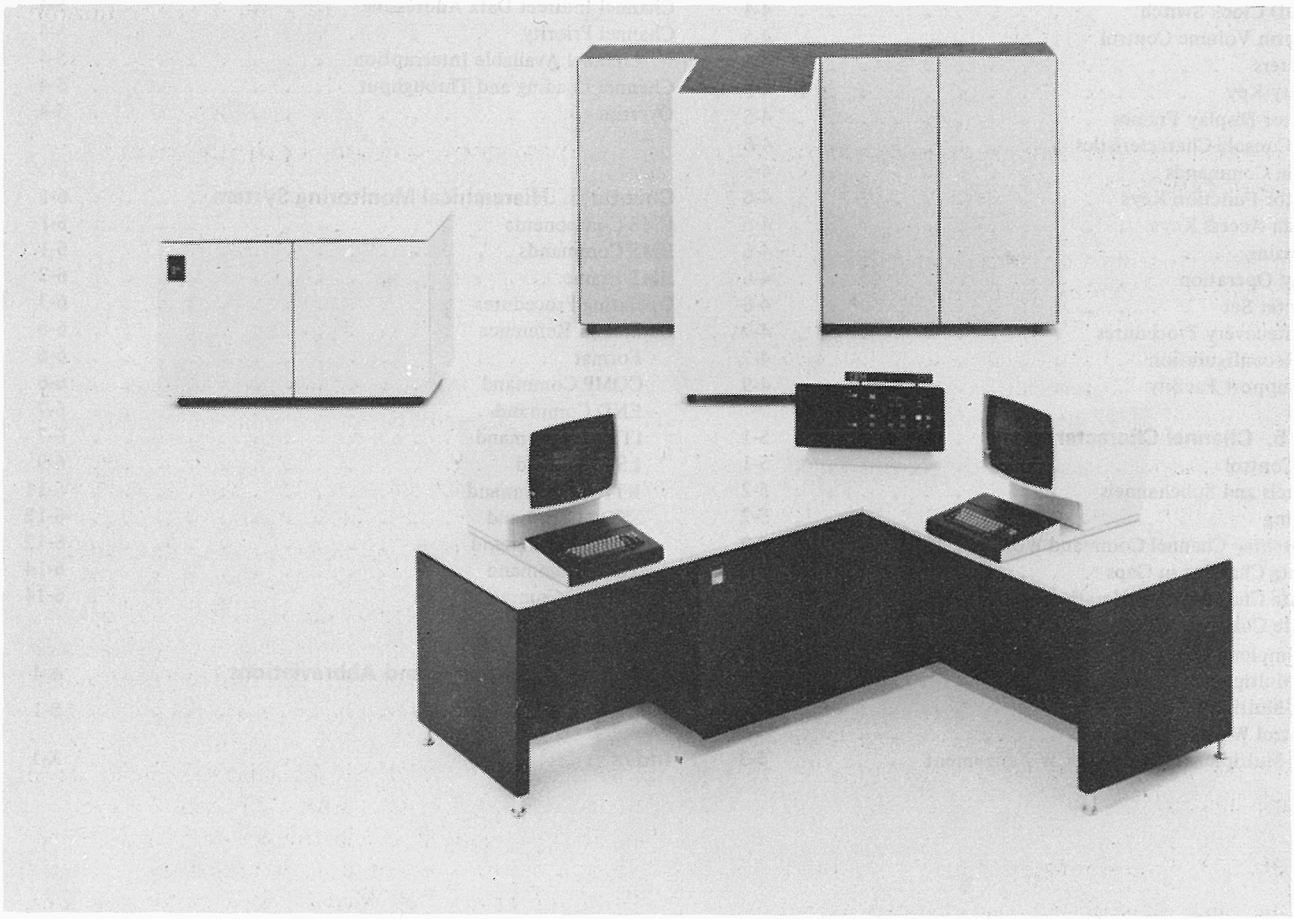
The reader should have an understanding of data processing systems, including a fundamental knowledge of the IBM System/370 as defined in *IBM System/370 Principles of Operation,* GA22-7000.

Only information that is of particular concern to the 3031 Processor Complex and 3031 Attached Processor Complex user is discussed in this manual.

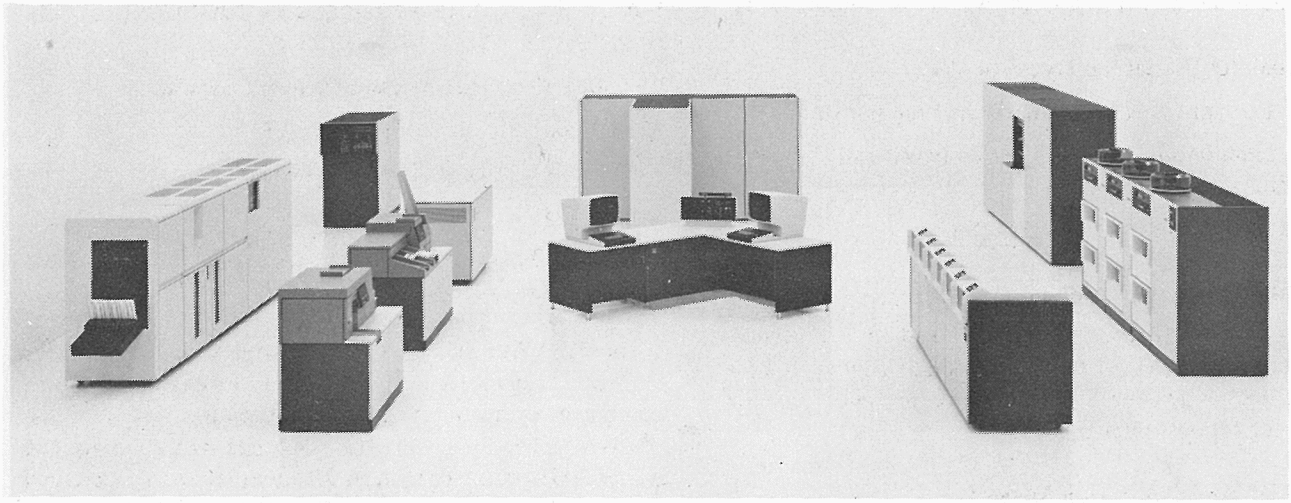
This publication contains six chapters and an appendix:

- Chapter 1 introduces the 3031 Processor Complex and 3031 Attached Processor Complex and describes their highlights, programming compatibility, and programming support.
- Chapter 2 describes the standard and optional features.
- Chapter 3 describes the logical elements of the IBM 3031 Processor and IBM 3041 Attached Processor from the viewpoint of function.
- Chapter 4 describes the controls and functions of the IBM 3036 Console Model 1.
- Chapter 5 describes the 3031 Processor's channel characteristics, expanding on the channel information presented in Chapter 3.
- Chapter 6 describes the Hierarchical Monitoring System.
- Appendix A contains a glossary and list of abbreviations.
- Appendix B describes deviations from *IBM System/370 Principles of Operation,* GA22-7000, and *IBM System/360 and System/370 I/O Interface Channel to Control Unit Original Equipment Manufacturers' Information,* GA22-6974.

Contents



IBM 3031 Processor Complex (Design Model)



IBM 3031 Processor Complex with Representative Input/Output Devices (Design Model)

IBM 3031 PROCESSOR COMPLEX AND 3031 ATTACHED PROCESSOR COMPLEX

The IBM 3031 Processor Complex (Frontispiece) and 3031 Attached Processor Complex offer high-speed performance for medium-scale scientific and business data processing applications. These complexes consist of certain combinations of the following IBM machines:

The IBM 3031 Processor Model 2, 3, 4, 5, or 6:

- Has a basic machine cycle time of 115 nanoseconds
- Uses monolithic circuit technology and efficient algorithms
- Has processor storage with capacity of 2M, 3M, 4M, 5M, or 6M bytes (M is 1,048,576 bytes of storage)
- Has a 32,768-byte high-speed buffer
- Provides four-way storage interleaving
- Has six integrated channels (one byte-multiplexer channel and five block-multiplexer channels controlled by a channel director).

The IBM 3031 Processor Model A2, A3, A4, A5, or A6:

- Has the same physical and operational characteristics, model for model, as the 3031 Processor Model 2, 3, 4, 5, or 6.
- Includes a function for communication and coordinated operation with the IBM 3041 Attached Processor Model 1.

Note: The 3031 Models 2, 3, 4, 5, and 6 can be field converted to Models A2, A3, A4, A5, and A6, respectively.

The IBM 3041 Attached Processor Model 1:

- Has a basic machine cycle time of 115 nanoseconds
- Uses monolithic circuit technology and efficient algorithms
- Has a 32,768-byte high-speed buffer.

The IBM 3036 Console Model 1:

- Has two stations (operator station and service support station) and a control panel that permit manual control of functions by operator and maintenance personnel (see Chapter 4 in this publication).

The IBM 3017 Power Unit Model 1

- Provides 208 volts at 415 Hertz to the 3031 Processor (any model) or the 3041 Attached Processor. One is required for each processor.

IBM 3031 PROCESSOR COMPLEX HIGHLIGHTS

The 3031 Processor Complex consists of:

- IBM 3031 Processor Model 2, 3, 4, 5, or 6
- IBM 3036 Console Model 1
- IBM 3017 Power Unit Model 1

The air-cooled 3031 Processor provides arithmetic, logical, storage, channel, and control functions for the processor complex. The 3031 Processor uses virtual storage, which permits users to program as if the system has as many as 16,777,216 bytes of processor storage. The high-speed buffer enables significant reduction of the effective processor storage access times. The six-channel group is physically integrated within the processor, but operates independently of other processor functions. The channels achieve high aggregate data rates by using dedicated buffers and the four-way interleaved processor storage.

The 3036 Console Model 1 enables the operator to communicate with the processor complex. Each of the two stations has a station processor, CRT display, keyboard, diskette drive, and an I/O interface to a channel. Either station may be used for operation or maintenance.

The 3017 Power Unit Model 1 is a motor generator, which, by means of a 40-foot (12.2-meter) cable, can be placed in the computer room with other elements of the 3031 Processor Complex or 3031 Attached Processor Complex.

Figure 2-1 lists the features of the 3031 Processor Complex and shows its plan view (not to scale).

IBM 3031 ATTACHED PROCESSOR COMPLEX HIGHLIGHTS

The 3031 Attached Processor Complex consists of:

- IBM 3031 Processor Model A2, A3, A4, A5, or A6
- IBM 3041 Attached Processor Model 1
- IBM 3036 Console Model 1
- IBM 3017 Power Unit Model 1 (one for the 3031 and one for the 3041)

The 3031 Attached Processor Complex (in AP mode) has all the capabilities of the 3031 Processor Complex combined with the instruction processing power of the 3041 Attached Processor. The 3041 uses virtual storage and shares 3031 processor storage. I/O instructions are executed only by the 3031 Processor. The 3031 and 3041 operate together under a single control program.

Figure 2-1 lists the features of the 3031 Attached Processor Complex and shows its plan view (not to scale).

Prefixing

Both processors in a shared-storage attached processor complex require an area of real storage for permanently assigned locations and logout areas. Because there is only one set of absolute addresses in shared processor storage, a means of assigning addresses to two different areas (one for each processor) is necessary. The technique used is called prefixing.

Prefixing allows assigning addresses 0 through 4095 to any 4,096-byte storage area, starting at any address that is a multiple of 4,096. (A 4,096-byte storage area that is assigned addresses 0 through 4095, for either processor, is called a permanent storage area.) The prefix is a 12-bit number in the prefix-value register of each processor. The contents of the register can be set by the Set Prefix instruction and can be inspected by the Store Prefix instruction. Prefixing is described in detail under "Multiprocessing" in the *IBM System/370 Principles of Operation,* GA22-7000.

Signaling and Response between Processors

Signaling and response between processors of the attached processor complex is provided by the Signal Processor instruction and the signaling and response facility (described in detail in the *IBM System/370 Principles of Operation*, GA22-7000).

3041 Concurrent Maintenance

The 3041 Attached Processor can be taken offline for most maintenance activities while the 3031 continues processing in uniprocessor mode.

Operating Mode Selection

Operating mode (UP/AP) is selected by setting a switch (on the 3041 operator panel) to either the uniprocessor (UP) or the attached processor (AP) position.

Power Control

In AP mode, the 3041 is powered up in sequence (after the 3031) from the control panel. In UP mode, the 3041 is powered up separately (after the 3031) by means of a display frame on the 3036 Console.

Time-of-Day Clock

In AP mode, the time-of-day (TOD) clock of one processor is synchronized by the TOD clock of the other processor. Once synchronized, the two clocks appear as one to the complex. (See "TOD Clock Synchronization" in the *IBM System/370 Principles of Operation,* GA22-7000, for detailed information.)

Malfunction Alert

When either processor enters the check-stop state, a malfunction alert signal is sent to the other processor. This signal generates a request for an external interruption. The interruption request remains pending until the interruption is taken or until the complex is reset.

PROGRAMMING COMPATIBILITY

Any program written for System/370 will operate on the 3031 or 3041, except when the program:

- Depends on machine facilities (storage size, I/O equipment, optional features, etc.) not included in the configuration
- Is time dependent
- Uses results defined in the *System/370 Principles of Operation* to be unpredictable or model dependent
- Uses unassigned fields in machine formats (control registers, instruction formats, etc.) not explicitly made available for program use
- Depends on interruptions caused by format errors, such as unassigned operation or command codes

Programs that include the Read Direct and Write Direct instructions must provide for the fact that, on the 3031 Processor and 3041 Attached Processor, these instructions use real rather than logical operand addresses.

A System/360 program that is to run on the 3031 Processor or 3041 Attached Processor must not violate the preceding restrictions; additionally, the program:

- Must not use program status word (PSW) bit 12 as an American National Standard Code for Information Interchange (ASCII) bit
- Must not depend on processor-storage locations assigned specifically for System/370
- Must not depend on the validity of data in processor storage after system power has been turned off, and then restored

For input/output operations, the program must take into account the effects of channel prefetching, command retry, and the operation-code assignment for the Halt Device instruction.

The System/370 functions that differ from System/360 functions are described in detail in Appendix B of the *IBM System/370 Principles of Operation,* GA22-7000.

PROGRAMMING SUPPORT

The 3031 Processor and 3041 Attached Processor run under Operating System/Virtual Storage (OS/VS) using dynamic address translation (DAT).

Programming support for the 3031 Processor Complex or the 3031 Attached Processor Complex running in uniprocessor mode includes:

- MVS (multiple virtual storage)
- MVS/System Extensions program product
- SVS (single virtual storage)
- VM/370 (IBM Virtual Machine Facility/370)
- VM/System Extensions program product
- VS1 3031 Processor Support (system control program support)
- DOS/VS (Disk Operating System/Virtual Storage)

Programming support for the 3031 Attached Processor Complex running in attached processor mode includes:

- MVS (OS/VS2)
- MVS/System Extensions program product
- VM/370 (IBM Virtual Machine Facility/370)
- VM/System Extensions program product

DATA SECURITY AND VALIDITY

Data security in the 3031 Processor Complex and 3031 Attached Processor Complex is maintained through the following standard features:

- Storage protection
- Console security keylock
- Data parity checking
- Storage error checking and correction (ECC)
- Instruction retry
- Channel retry
- CPU ID

Storage Protection

Storage protection includes both store protection and fetch protection. If store protection is violated, data is not stored into the protected area. If fetch protection is violated, data is not retrieved from the protected area of storage. Additional protection is provided for certain storage locations that are vital to operating-system availability.

Console Security Keylock

The console security keylock is located on the right-hand side of the display. When the security key is removed or is in the vertical position, system data security is placed under program control. The operator is restricted to the currently displayed frame, and any attempt to change frames causes an alarm to sound.

Data Parity Checking

All data transfers, arithmetic operations, and logical operations include parity checking. Detection of even parity causes a machine-check interruption, when this type of interruption is allowed.

Storage Error Checking and Correction

When the storage control function (SCF) operates in diagnostic parity mode, each data byte in processor storage is maintained with odd parity. Data is checked as it is read out of storage. If even parity is detected, a machine-check interruption is signaled. When the SCF operates in error checking and correction (ECC) mode, each doubleword includes an ECC check byte, which is generated during store operations to replace the parity bits. The ECC check byte detects and corrects all single-bit errors and detects all double-bit errors.

Instruction Retry

When a machine error is detected during instruction execution or interruption handling, an attempt is made to retry the instruction or interruption sequence. No invalid results are computed if retry is not successful.

Channel Retry

When a machine error is detected during channel operations, the channel director attempts to retry the channel function if possible. For situations that cannot be retried, the channel requests an I/O interruption to indicate the error.

CPU ID

Model-dependent information identifying the processor is available in storage for use in logouts and the Store CPU ID instruction.

VIRTUAL STORAGE

When the 3031 Processor or 3041 Attached Processor is operating in extended control (EC) mode with dynamic address translation (DAT) invoked, all logical addresses within the System/370 24-bit addressing structure are available, regardless of the real storage capacity. Therefore, the maximum logical (virtual) decimal address is 16777215.

The 3031 Processor and 3041 Attached Processor, which support this virtual storage environment, are not subject to the restraints normally imposed on programming applications by the amount of available real storage. Consequently, the operational flexibility of the installation is enhanced.

RELIABILITY, AVAILABILITY, AND SERVICEABILITY

System interruptions are reduced through automatic recovery facilities such as channel retry, instruction retry, interruption retry, and error checking and correction (ECC) for storage.

Availability is enhanced through reduction of system interruptions and improved serviceability features.

ATTACHABLE INPUT/OUTPUT DEVICES

Input/output (I/O) devices that can be attached to the channels of the IBM 3031 Processor in the 3031 Processor Complex or the 3031 Attached Processor Complex are listed in the *IBM System/370 Input/Output Configurator,* GA22-7002.

Many of the standard and optional features of the 3031 Processor and 3041 Attached Processor are described briefly in this section. Figure 2-1 lists the features as they apply to the 3031 and 3041. More detailed descriptions are given in the *IBM System/370 Principles of Operation,* GA22-7000.

STANDARD FEATURES

System/370 Universal Instruction Set

The System/370 universal instruction set contains 156 instructions and includes features such as byte-oriented operand, storage protection, interval timer, time-of-day clock, and monitoring.

Byte-Oriented Operand

The byte-oriented operand allows the user to ignore, in part, the restriction that all operands in processor storage be aligned on integral boundaries (for example, halfword operands on halfword boundaries). A significant amount of programming time can be saved by using this feature, but processor performance is slightly degraded when it is used excessively.

Key-Controlled Storage Protection

Key-controlled storage protection (for both stores and fetches) makes it possible to protect the contents of processor storage from undesired destruction or unauthorized use. Key-controlled storage protection includes the privileged instructions Insert Storage Key (ISK) and Set Storage Key (SSK).

Interval Timer

The interval timer is a counter that generates an external interruption request whenever it decrements to a negative value. The timer has a 15.5-hour cycle and a 3.33-millisecond resolution.

Time-of-Day (TOD) Clock

The TOD clock provides accurate elapsed-time measurements and the time of day. The clock is updated each microsecond. Clock operation is not affected by system resets or by any system activity other than turning off power or executing the Set Clock instruction. The clock runs independent of the state of the processor, when the processor is in the wait state, in the stopped state, and is executing programs.

Monitoring

Monitoring, with the Monitor Call (MC) instruction, provides a means of selectively recording designated events in the execution of a program.

Translation

The translation feature provides for dynamic address translation, program event recording, and extended control mode.

Dynamic Address Translation

Dynamic address translation (DAT) provides the means for translating virtual addresses to absolute addresses. DAT is described in greater detail under "Buffer Control Function" in Chapter 3.

Program Event Recording

Program event recording (PER), a debugging aid, is controlled by bit 1 of the EC-mode program status word (PSW). PER allows the program to be alerted to:

- Successful execution of a branch instruction
- Alteration of the contents of designated general registers
- Fetching of an instruction from the contents in designated locations of processor storage or alteration to them

Extended Control Mode

Extended control (EC) mode permits the use of system facilities and functions not available with basic control (BC) mode, such as dynamic address translation and program event recording. EC mode is implemented with a modified PSW format and with permanently assigned areas of processor storage.

Extended-Precision Floating Point

Extended-precision floating point includes instructions that handle extended-precision (28 hexadecimal-digit) floating-point operands. Extended-precision operands may also be rounded off to long-precision operands, and long-precision operands may be rounded off to short-precision operands.

CPU Timer

The CPU timer provides a means for measuring elapsed processor time and for causing an external interruption when a prespecified length of time has elapsed. Unlike the

Processor: IBM 3031 Processor

Attached Processor: IBM 3041 Attached Processor

Basic Machine Cycle Time: 115 nanoseconds

Standard Features: 3031 and 3041

System 370 universal instruction set (Note 1)
- Byte oriented operand
- **Key-controlled storage protection**
- Interval timer
- Time of day clock
- Monitoring

Translation
- Dynamic address translation
- Program event recording
- Extended control mode

Multiprocessing feature instructions (3031 A-series models and 3041)
Extended precision floating point
CPU timer
Clock comparator
Conditional swapping
PSW key handling
System 370 extended facility
Error checking and correction
Instruction retry
OS VS1 ECPS (extended control program support)
Virtual machine assist
High-speed buffer storage
Reloadable control storage

Standard Features: 3031 Only

Processor Storage
Storage configuration control (via IBM 3036 Console)
Byte-multiplexer channel (Notes 2, 3, and 4)
Block-multiplexer channels (five) (Notes 2, 4, and 5)
Channel indirect data addressing
Limited channel logout
I/O extended logout
Channel retry
Command retry (block-multiplexer channels)
Start I/O fast release
Clear I O
High speed transfer
Unit emergency power off

Notes:

1. The instruction set for the 3041 does not contain I O instructions.
2. As many as eight control units can be attached; available subchannels are shown on chart (see Note 5).
3. Operates in either burst mode or byte mode; multiplexing capability on bytes, groups of bytes, or blocks.
4. IBM input output devices that can be attached to the 3031 Processor are listed in the *IBM System 370 Input Output Configurator*, GA22-7002.
5. Operates in burst mode only; multiplexing capability on blocks or multiple blocks. Available subchannels for standard channel group are shown on the following chart:

Byte-Multiplexer Subchannels		Block-Multiplexer Subchannels	
Without Sharing	With Sharing	Without Sharing	With Sharing
256	256 nonshared less 8, 16, or 32 for each subchannel configured for sharing	1,280	Up to 40 for channel group

6. The channel to channel adapter, attached to a byte- or block-multiplexer channel, permits interconnection of two channels. One control-unit position on the 3031 Processor channel can be connected to one control-unit position on any other System 360 or System 370 channel. Only one adapter is needed per connection. The adapter counts as one control unit for both channels.

Optional Features: 3031

Channel-to-channel adapter (Note 6)
Direct control

Optional Features: 3041

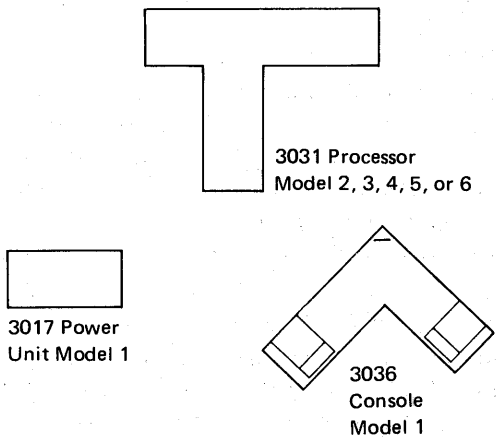
Direct control

Processor Storage: Part of IBM 3031 Processor

Processor Model	*Storage Capacity (Bytes)*
2, A2	2M (2,097,152)
3, A3	3M (3,145,728)
4, A4	4M (4,194,304)
5, A5	5M (5,242,880)
6, A6	6M (6,291,456)

Representative Plan Views

IBM 3031 Processor Complex



IBM 3031 Attached Processor Complex

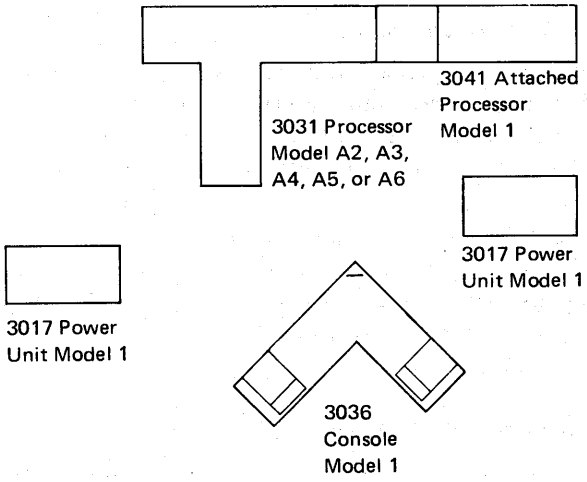


Figure 2-1. 3031 Processor Complex and 3031 Attached Processor Complex Configuration Guide

TOD clock, the CPU timer does not run when the processor is in the stopped state but does provide accurate measurement of elapsed processor time.

Clock Comparator

The clock comparator causes an external interruption when the TOD clock reaches a value specified by the program.

Conditional Swapping

Conditional swapping, by means of the Compare and Swap (CS) and the Compare Double and Swap (CDS) instructions, provides for the controlled sharing of common storage areas by programs that operate in a configuration using multiprogramming or multiprocessing.

PSW Key Handling

PSW key handling allows the four-bit protection key (for store and fetch operations), which is part of the current PSW, to be inserted into general register 2 by means of the Insert PSW Key (IPK) instruction; or the key may be replaced by part of an address in storage by means of the Set PSW Key from Address (SPKA) instruction.

Channel Indirect Data Addressing

Channels do not use dynamic address translation. Channel command words (CCWs) in virtual storage must be translated by the control program before execution. This feature allows contiguous areas of virtual storage to be mapped into noncontiguous areas of real storage.

System/370 Extended Facility

The System/370 extended facility provides the means to:

1. Help reduce the time needed to execute several frequently used supervisor functions of MVS
2. Increase the efficiency of dynamic address translation (DAT)
3. Improve system reliability, availability, and serviceability by additional protection of certain storage locations that are vital to operating-system availability.

This facility works in conjunction with the MVS/System Extensions program product.

Error Checking and Correction

Error checking and correction circuits automatically detect all single- and double-bit errors and most multiple-bit errors in data read from processor storage. Single-bit errors are automatically corrected.

Instruction Retry

Instruction retry automatically examines any failing instruction to determine whether execution has passed the retry threshold. If not, error recovery procedures can usually retry the instruction.

OS/VS1 ECPS

OS/VS1 ECPS (extended control-program support) improves the performance of OS/VS1 by emulating in the real machine certain frequently used supervisor functions.

Virtual Machine Assist

Virtual machine assist allows VS operating systems operating under VM/370 to emulate certain privileged operations.

High-Speed Buffer Storage

The 32K (32,768-byte) buffers of the 3031 Processor and 3041 Attached Processor satisfy many requests for storage, making the effective storage-access time much shorter than the actual processor-storage cycle time.

Reloadable Control Storage

Monolithic reloadable control storage (RCS), contains microprograms that control instruction execution and channel operations. Two RCS locations are in the 3031: instruction execution function (IEF) and channel director. One RCS location is in the 3041 (IEF). RCS is loaded from a diskette drive in the 3036 Console.

Storage Configuration Control

Storage configuration control permits processor storage to be configured from the console in 1M (1,048,576-byte) increments.

Byte-Multiplexer Channel

In byte mode, the 3031 byte-multiplexer channel provides control for the concurrent operation of a number of low-speed I/O devices or, in burst mode, for the operation of one high-speed I/O device at a time.

Block-Multiplexer Channels

The 3031 block-multiplexer channels provide control for the concurrent operation of a number of high-speed I/O devices.

Limited Channel Logout

Limited channel logout allows for the storing of a word of model-independent information related to equipment errors detected by a channel.

I/O Extended Logout

I/O extended logout allows the storing of extensive information related to equipment errors in a storage area designated by a pointer (address in locations 173-175).

Channel Retry

Channel retry allows the storing of channel retry information for use in the subsequent retry of the Start I/O instruction or reinitiation of the I/O operation.

Command Retry

Command retry is a procedure initiated by the control unit to retry channel commands. No I/O interruptions are required. The number of retries is device dependent.

Start I/O Fast Release

Start I/O fast release provides for early release of the processor by a channel during execution of the Start I/O Fast Release (SIOF) instruction. Start I/O fast release reduces the processor delay associated with the initiation of the I/O operation.

Clear I/O

Clear I/O allows the use of the Clear I/O (CLRIO) instruction, which causes the current operation with the addressed device to be discontinued and the state of the operation at that time to be indicated in the stored channel status word (CSW).

High-Speed Transfer

High-speed transfer, a feature for block-multiplexer channels, permits an increase of data transfer rates. This feature provides a data-in line and a data-out line between the control unit and channel. The data-in and data-out lines operate alternately with the service-in and service-out lines in communicating requests for data and indicating the transmission or receipt of data.

Unit Emergency Power Off

Activating the Unit Emergency Power Off switch removes power from the processor and all cable-connected devices under direct control of the processor.

OPTIONAL FEATURES

The following optional features can be added to provide additional facilities:

- Channel-to-channel adapter (3031 only)
- Direct control (3031 and 3041)

Channel-to-Channel Adapter

The channel-to-channel adapter provides the data path and the synchronization for data transfers between two channels. This adapter, which operates only in selector mode, can be connected to any type of channel on any System/360 or System/370. To a channel, the adapter appears as a control unit and responds to either channel the same as a control unit.

The adapter normally is connected between channels associated with different processors, thus establishing a loosely coupled multiprocessing system.

The data rate for an adapter is limited by the slower of the two communicating channels.

Direct Control

The direct-control feature provides the two privileged instructions Read Direct and Write Direct, and six external interruption lines. The read and write instructions provide for the transfer of a single byte of information between an external device and processor storage. When active, each of the external signal lines sets up the conditions for an external interruption. When used with a 3031 or 3041 these instructions use real addresses rather than virtual addresses. For more information, see *IBM System/360 and System/370 Direct Control and External Interruption Features, OEMI,* GA22-6845.

Chapter 3. IBM 3031 Processor and 3041 Attached Processor

The IBM 3031 Processor comprises the following logical elements (Figure 3-1):

- Instruction execution function (IEF)
- Buffer control function (BCF)
- Storage control function (SCF)
- Processor storage
- Director/channels (six)

The channels are described in this chapter under "Director/Channels" and also in Chapter 5 of this publication; the console is described in Chapter 4.

The IBM 3041 Attached Processor comprises the following logical elements (Figure 3-1):

- Instruction execution function (IEF)
- Buffer control function (BCF)

The descriptions of functional elements in this chapter apply to both the 3031 and 3041.

INSTRUCTION EXECUTION FUNCTION

Data Paths

The instruction execution function (IEF) contains three basic data paths: an I-fetch path, a one-byte path, and a four-byte path.

Components of the I-fetch data path allow prefetching and buffering of instructions from processor storage. Components of the other two paths allow the execution of variable field length instructions by the one-byte path, and fixed-point and floating-point instructions by the four-byte path. Instruction-fetch sequences may overlap processor operations if processor storage is not busy when a fetch sequence starts.

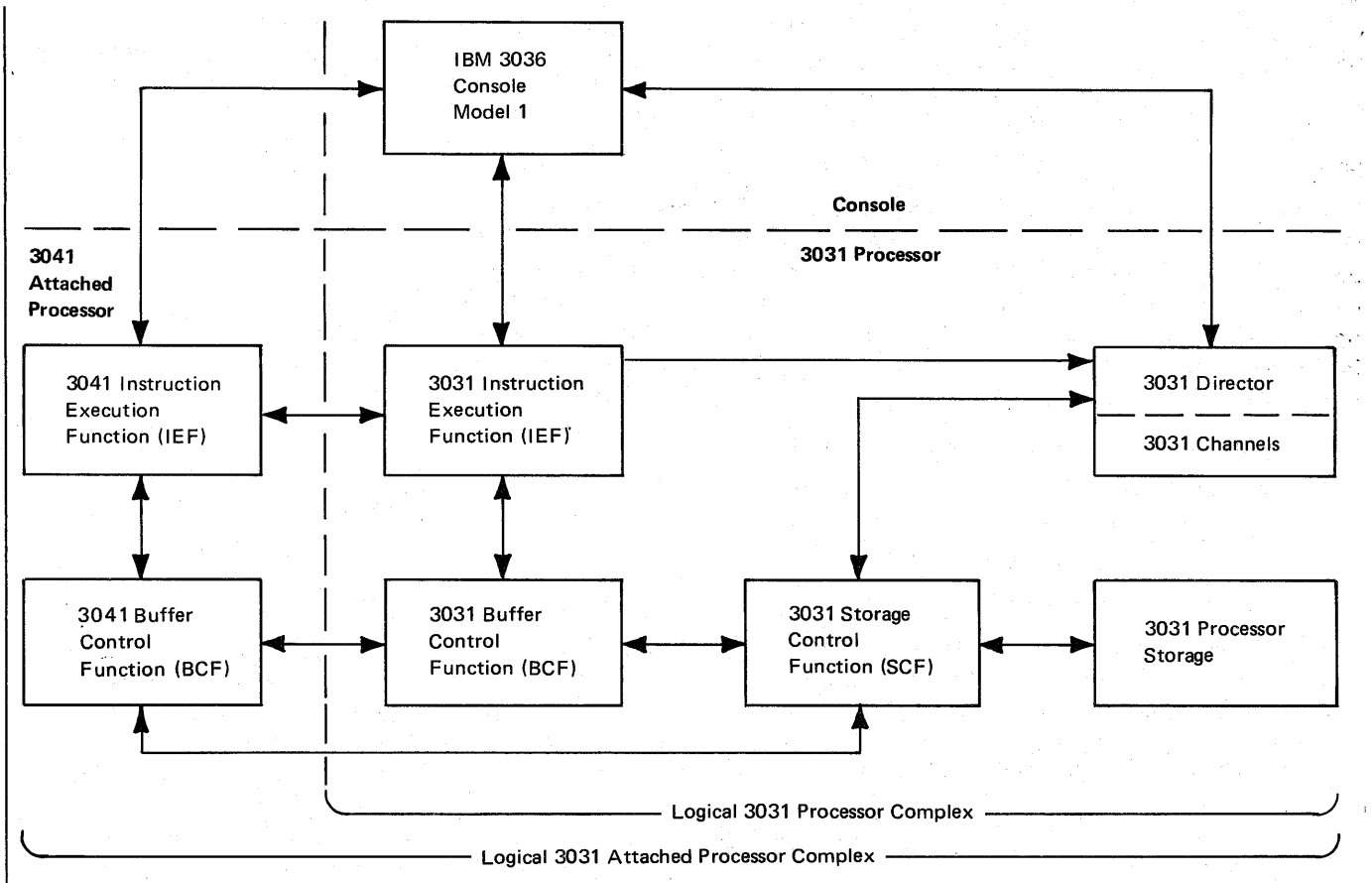


Figure 3-1. Logical Structure of the 3031 Processor, or the 3031 Processor and 3041 Attached Processor, with the 3036 Console

Status Registers

Four status registers retain status information for use in branch and control functions and in retry of instructions.

PSW Register

The PSW register retains selected status information, some of which is related to the PSW.

Execution Array

The execution array increases the processing speed of certain instructions.

IEF Local Storage

IEF local storage includes 16 general registers, four floating-point registers, working storage, and 16 control registers.

Four-Byte Logical Element

The four-byte (word-wide) logical element aligns operands, performs logical operations, and sets condition codes on certain instructions.

Reloadable Control Storage

Reloadable control storage (RCS) in the IEF contains microprograms; these may be either basic control programs or microdiagnostic programs. RCS contains 8,192 control words of 72 bits each, and is accessed once each machine cycle. Control storage consists of RCS plus associated logic.

RCS is loaded either by turning power on or by an RCS load request from the appropriate console display frame. Control storage is volatile; it must therefore be reloaded whenever power is interrupted.

IEF Functional Controls

IEF functional controls include:

- An IEF clock, which generates pulses for IEF cycles
- System reset functions
- Microcode, the 72-bit control word that specifies the functions performed during each IEF cycle
- Control registers (16) in local storage
- Program event recording (PER), which allows program interruptions under these conditions:
  1. A successful branch
  2. The alteration of a designated general register
  3. An instruction-fetch from a designated processor storage location
  4. The alteration of a designated processor storage location
- Nullification of an instruction, because of a translation exception, so that it cannot alter processor storage
- IEF identity by use of the Store CPU ID instruction
- Diagnostic capability by use of the Diagnose instruction

System Timing Facilities

The system timing facilities include the time-of-day (TOD) clock, the clock comparator, and the CPU timer, all of which are defined in *IBM System/370 Principles of Operation,* GA22-7000.

Error Handling Facilities

Error handling facilities include those for error checking, instruction retry, and machine-check interruption.

Each data path in the IEF is checked for parity on each machine cycle. Each of the 16 sectors (512 words in each) in reloadable control storage is checked for parity and sector sum during system reset. Each RCS word is checked for parity whenever it is used.

Retry techniques provide the ability to recover from intermittent IEF control and data errors. IEF retry is accomplished by microprogram routines that save the source data before it is altered during an operation and, in the event of an error, return the IEF to the beginning of the function or to a point in the operation that was correctly executed. The IEF proceeds from that point.

For recording purposes, a machine-check interruption is taken at the completion of a successful retry.

The definitions and implementations of machine-check handling are in *IBM System/370 Principles of Operation,* GA22-7000.

BUFFER CONTROL FUNCTION

The buffer control function (BCF), which provides a link between processor storage and the instruction execution function, includes:

- High-speed buffer
- Dynamic address translation (DAT)
- Storage protection
- Storage data register (SDR)

High-Speed Buffer

The 32,768-byte buffers of the 3031 and 3041 provide high-speed access to instructions and data from processor storage. A fetch operation from the buffer takes about one-fourth the time required for the same operation from processor storage (see "Storage Control Function"). The objective of the buffer is to hold that portion of processor storage currently in use by the program. Buffer action is automatic.

The 3031 processor storage is divided into 4K-byte rows (from 512 rows in the 2M-byte capacity storage to 1,536

rows in the 6M-byte capacity storage); buffer storage is divided into eight 4K-byte compartments. The rows and compartments are further divided into 32-byte blocks. Buffer space is reserved on a block basis and is loaded one block at a time.

The buffer index array provides a reference to the storage addresses of data in the buffer. The array may contain any or all of the 128 block addresses of a given row, or any combination of rows, in processor storage.

During each IEF storage reference, the index array is interrogated to determine whether the referenced data resides in the buffer. If no buffer block that corresponds to the referenced processor storage block is found, a buffer block is automatically assigned to the referenced processor storage block; the block address is placed in the buffer address array, and a buffer storage block load is initiated. While the block load is being executed, the address is made invalid until the fetch is complete.

Block assignment in buffer storage is based on usage. When a new assignment is required, the dynamic trace algorithm causes replacement of the oldest of eight entries.

For a channel store operation, only processor storage is updated. A check is made to determine whether the referenced data is in high-speed buffer storage; if the data is found there, buffer storage data is invalidated. Channel fetch requests are made only to processor storage.

Dynamic Address Translation

Dynamic address translation (DAT) translates virtual addresses to absolute addresses when the system is in DAT mode and during the execution of the Load Real Address (LRA) instruction. DAT is invoked by turning on bit 5 of the PSW while the processor is operating in extended control mode. (The user may select the extended control mode by turning on bit 12 of the PSW.) For complete information on dynamic address translation, see *IBM System/370 Principles of Operation,* GA22-7000.

Note: In the 3031 Processor, real and absolute addresses are the same.

Translation Lookaside Buffer

The translation lookaside buffer (TLB) retains current address translations. The TLB holds up to 128 translations in 64 addressable locations each containing one odd- and one even-page translation. Subsequent translations for these addresses are avoided so long as they remain in the TLB.

TLB Operation

Each virtual address supplied by the program causes an access to both the high-speed buffer (to examine the data at the address) and to the TLB (to determine whether the absolute address is there).

If the virtual address has been previously translated and its absolute address now resides in the TLB, and if the fetch data is in the high-speed buffer, the data is available to the IEF in two cycles.

Whether or not the virtual address has been previously translated, if it does not currently reside in the TLB, a full translation must take place. Assuming no I/O interference, six to 26 machine cycles are required, depending on the locations of the segment- and page-table entries required for the translation.

If the absolute address is available from the TLB, it is compared with addresses read out of the high-speed buffer address array to determine whether the data field required is in the high-speed buffer (Figure 3-2).

When required by the operation, the absolute address is also used to access processor storage. If the absolute address is not available from the TLB, the virtual address is translated; the TLB is updated with the newly translated address.

Issuing a Purge TLB (PTLB) instruction purges the translation lookaside buffer. The TLB may also be purged either by using SYSTEM RESET on the operator display frame (OP frame) or by a system-generated reset.

TLB Operation Example: Assume that a given virtual address is requested by the IEF (64K segment, 2K page).

Virtual address bits 13-15 and 17-20 select the entry line in the TLB. Virtual address bits 8-12 and 16 are compared with the entry from the TLB. If the TLB compare is unsuccessful, a full translation is performed. Before going to processor storage to do the full translation, a determination is made to see whether the required translation entries (or any part of them) are in the buffer. If they are, the translation is made, using the buffer entries. If only part (or none) of the entries are in the buffer, then part (or all) of the translation is made, using processor storage.

If the TLB compare is successful, no translation is required and the absolute address is transferred from the TLB to the buffer address array to determine whether the entry is in the buffer. This ultimately determines whether the fetch is made from the buffer or from processor storage.

Storage Protection

Storage protection prevents unauthorized access of information stored in processor storage. Each 2K (2,048 bytes) of storage is protected by one of 15 possible key values (key bits 0-3). For store operations, protection-key bits 0-3 from the currently active PSW are compared with storage-key bits 0-3 from the processor storage protection array. If the keys do not match, the SCF is notified of a protection violation, and the data is not stored. The same protection is active for processor storage fetch operations if bit 4 (the fetch-protect bit) is on.

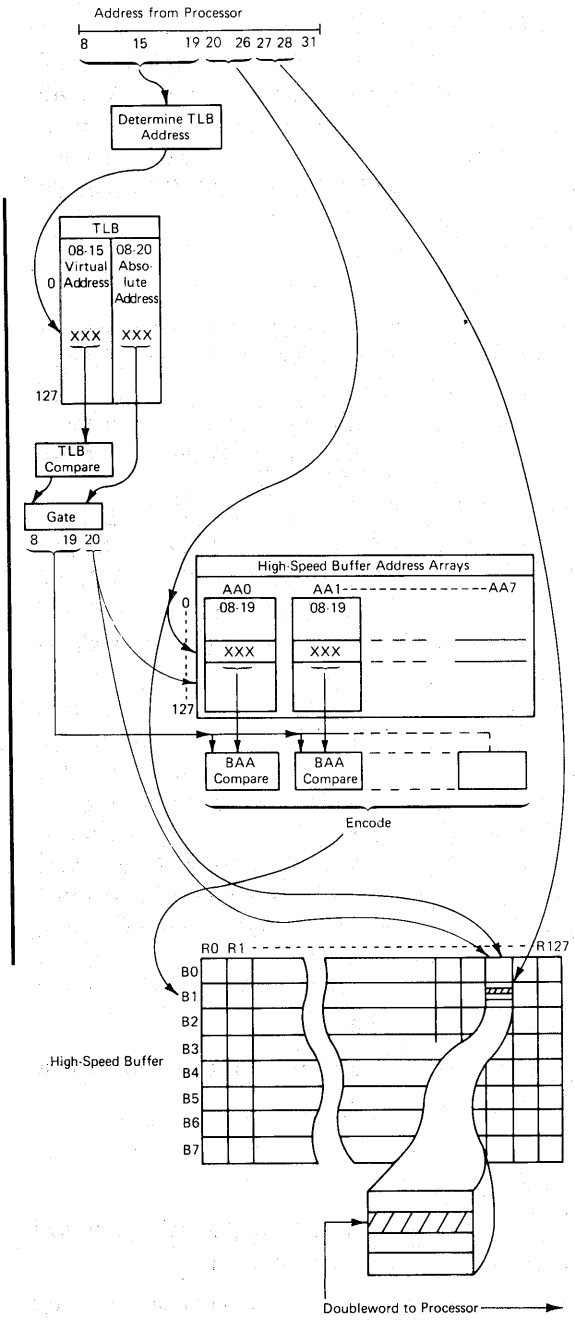


Figure 3-2. TLB and High-Speed Buffer Operation

Each protection key has a corresponding reference (R) bit and change (C) bit. The R-bit is turned on each time data is fetched from the corresponding 2K-byte block of storage. Both bits are turned on each time data is stored into the corresponding 2K-byte block of storage. Also, both bits indicate (to the storage management supervisor program) which pages of storage:

- Are not being used, and therefore can be replaced
- Have been altered and must be saved on an auxiliary storage device before being replaced in storage

Storage Data Registers

The two 16-byte storage data registers provide a means of moving data between the components of the instruction execution function and processor storage.

STORAGE CONTROL FUNCTION

By means of the channel/IEF bus controller, the storage control function (SCF) controls and processes all requests for storing data into processor storage or fetching data from it.

The channel/IEF bus controller transfers information from a channel over a bus in a series of steps. For example, during the first step, the storage address is transmitted; during the second, a word of data and byte marks are transmitted. The series is repeated until all the required information is sent. Information from the IEF is transferred over two buses: one for data; the other for addresses, marks, and commands.

The SCF performs a separate operation every half cycle. The operations include:

- Receiving a request for service
- Requesting that the channel/IEF bus controller resend the data if parity is incorrect
- Determining the storage priority of the requester
- Invalidating channel data residing in high-speed buffer storage
- Sending the address (prefixed in an attached processor complex) to storage protect (director requests only)

The 3031 Processor cycle times and storage access times are:

Operation	*Time (in Nanoseconds)*
3031 Processor machine cycle	115
Processor storage cycle	115
Control storage cycle	115
Processor storage read/write of a doubleword on a doubleword boundary	345
Processor storage read of a doubleword	575
Processor storage write of a partial doubleword (fewer than eight bytes)	690
IEF fetch of four bytes from processor storage	805
IEF store of a doubleword in processor storage	345
IEF store of a partial doubleword in processor storage	345
IEF store of a quadword in processor storage	460
IEF store of a partial quadword in processor storage	460
IEF block fetch (buffer block load)	1,150
IEF fetch of four bytes from the buffer	230
IEF fetch of eight bytes from the buffer	345

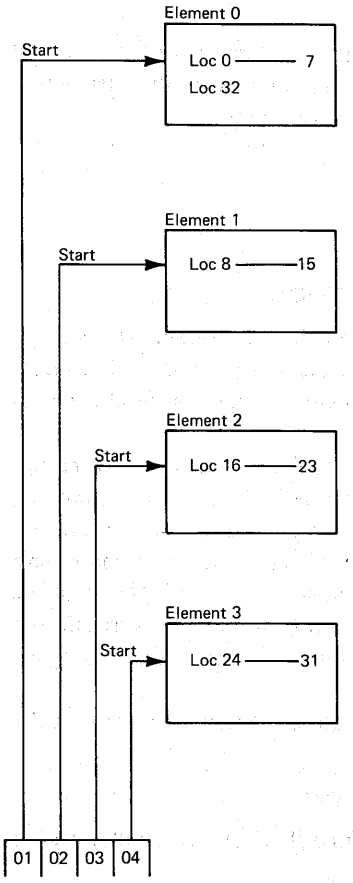


Figure 3-3. Interleaving

PROCESSOR STORAGE

Processor storage provides the system with four-way interleaved real storage, and is available in five capacities:

2M (2,097,152 bytes)
3M (3,145,728 bytes)
4M (4,194,304 bytes)
5M (5,242,880 bytes)
6M (6,291,456 bytes)

Note: The 2M-byte capacity is standard.

Processor storage has a storage distribution element (SDE) that controls fetch and store operations in the monolithic storage data array. The data array is divided into four storage elements. The SDE controls the interleaving of storage operations by selecting different elements during each machine cycle.

Error checking and correction (ECC) bits are stored in the data arrays, along with the data. In a fetch operation, the ECC bits detect and correct single-bit errors, and detect double-bit errors.

Permanent Storage Assignment

Processor storage addressing begins at location 0 and continues through the highest storage byte location. All processor storage is available for programming functions except the permanent storage assignment areas, which are described in detail in *IBM System/370 Principles of Operation,* GA22-7000.

Interleaving

Interleaving allows processor storage elements to operate independently and concurrently for effective reduction of the storage cycle time (Figure 3-3).

In four-way interleaving, four functionally independent storage elements (each providing eight bytes per storage access) make up processor storage.

Assume that the four elements are 0, 1, 2, and 3. Storage locations 0-7 are in element 0, locations 8-15 are in element 1, locations 16-23 are in element 2, and locations 24-31 are in element 3. Storage locations 32-39 are in element 0, and the address distribution sequence continues through all available storage locations.

An attempt to reference a processor storage location (element) may be made during any cycle. A storage reference is accepted on any cycle during which the functionally independent storage element containing the requested location is not busy. (A storage element is defined as busy when it has not completed a storage cycle after being selected.) Once the storage element is selected for a storage reference, it cannot again be referred to until the completion of the storage cycle.

Storage Configuration Control

The four storage elements can be configured in various ways under control of the console.

DIRECTOR/CHANNELS

The 3031 Processor has six integrated channels controlled by a channel director. The director includes:

- The director microprocessor that controls, through microcode, all functions of the director
- A 64-word local storage that provides working storage for the director and unit control word (UCW) storage for the channels
- A 32K-byte storage that provides UCW storage, subchannel (UCW) index array, and working storage for the microprogram
- Data buffer local storage of 64 words that provides communications between the director data flow and the channel data flow
- Reloadable control storage, which holds 4,096 control words of 72 bits each, and is shared by the director and the channels

Director

Within the director, the microprogram controls all communications with the IEF, the SCF, and the console. Additionally, each block-multiplexer channel includes registers and controls needed to maintain data transfer between I/O devices and storage.

The director communicates with the channels on a priority basis: first with the block-multiplexer channels in ascending order of channel number, then with the byte-multiplexer channel.

The director and each channel, operating within their own microprograms, share logic by switching control at specified points in the microprograms. This change in control is called break-in. When a break-in occurs, the current microprogram is halted temporarily while another microprogram is given control.

Byte-Multiplexer Channel

The byte-multiplexer channel is microprogram controlled for I/O-command execution, I/O interface data transfer, and interruptions; it is hardware controlled for device selection. Upon request by a device for service, the channel breaks in on the director, takes control with its own address register, services the device, sets up the conditions for subsequent break-ins for further service to the device, and eventually restores control to the director.

Any block-multiplexer channel can break in on a byte-multiplexer channel so long as the break-in is not inhibited by the executing microprogram.

Because the byte-multiplexer channel can service several devices in byte mode, unit control words (UCWs) are available on a device-address basis.

Block-Multiplexer Channels

Block-multiplexer channels are both microprogram controlled and hardware controlled. I/O-command execution, device-selection initiation, and interruptions are microprogram controlled, whereas interface data handling and command-chaining reselection are hardware controlled. The channel is under microprogram control for processor-storage data transfers.

Channel Retry

Channel retry enables each channel to retry channel functions when an error occurs. The information needed to retry channel functions is provided by affected channels and is held in a pair of microprogram-controlled registers called the threshold register and the retry-code register. The two registers record information needed by the microprogram to determine and invoke the appropriate method of recovery (reissue Start I/O or pass control to a device error-recovery program).

Limited Channel Logout

Limited channel logout provides a field (locations 176-179) for model-independent information related to equipment errors detected by the channel. This information is used to provide detailed machine status when errors have affected I/O operations and as a method of reporting errors successfully circumvented by the channel.

I/O Extended Logout

I/O extended logout provides a pointer (address in locations 173-175) that is set by the program to designate an area to be used by channels for storing information related to equipment errors (logout information) instead of the fixed logout area (locations 256-351).

Channel Performance

Byte-Multiplexer Channel

Byte-multiplexer channel performance is highly dependent on I/O-device interface transition response times. These device delays show wide variations among the devices attachable to a byte-multiplexer channel. Another factor to be considered in performance calculations is the effect of block-multiplexer channel interference, because block-multiplexer channel routines can always break in on a byte-multiplexer channel routine.

Block-Multiplexer Channels

Each block-multiplexer channel may attain a data rate of 1.5 megabytes per second.

Channels 1 through 5 can operate in a disconnected command-chaining mode (referred to as block multiplexing), which causes the channel to disconnect a device at channel-end time if command chaining occurs. During the interval between channel end and device end, another device on the channel can be started. Each CCW must complete its data transfer.

MAINTENANCE CONTROLS

The maintenance controls provides registers that enable interaction of the 3031 and 3041 with the console for manual operations and servicing features, such as:

- Causing the processor, or processor and attached processor, to enter the stopped state
- Changing the processing rate while the processor is in the stopped state
- Enabling a system reset, CPU reset, or check reset
- Enabling or disabling the interval timer or a timer interruption

The IBM 3036 Console (Figure 4-1) is a two-station physically separate device that provides the manual control functions needed to:

- Select modes
- Display console messages and system information
- Enter data manually into the system
- Monitor and control system power

The console provides two physically separate operating stations. Each station is individually addressable and has a CRT display, a keyboard, a diskette drive, a micro-programmed console processor, and an I/O interface. A control panel permits configuration for selecting station activity. Both stations have a turnable display and keyboard.

One station is selected to be used primarily by the operator, and is called the operator station. The other station then becomes the one used for service support, and is called the service station.

Either one may be used by the operator or service personnel. If one station becomes inoperative, the other may be used to continue operations, but with reduced console performance.

OPERATOR CONTROLS

The operator controls include the keyboard, control panel, and security keylock.

Keyboard

The keyboard (Figure 4-2) provides alphabetic characters A through Z (both uppercase and lowercase), numbers 0 through 9, and 26 graphic characters. In addition, eight keys are provided for cursor control, and six keys provide additional special functions.

The keyboard provides the primary means of manually interacting with the system for certain functions. When a character is entered via the keyboard, that character is recognized by the station processor and is executed. The

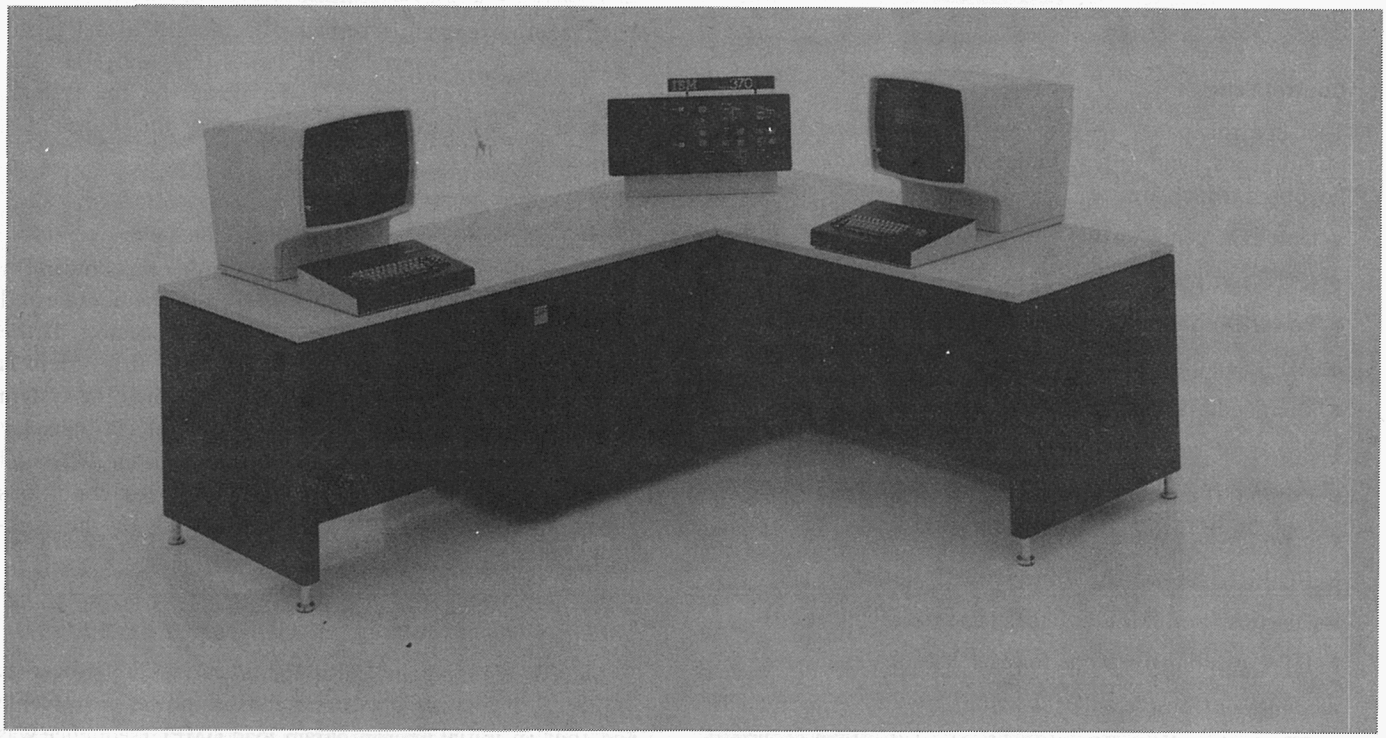


Figure 4-1. IBM 3036 Console Model 1 (Design Model)

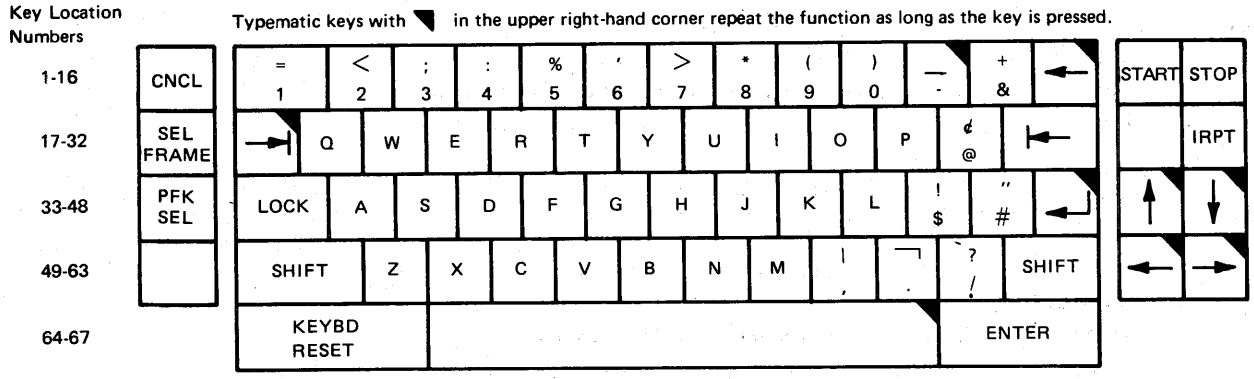


Figure 4-2. Console Station Keyboard

operator and service personnel functions for any particular display frame can be activated by entering the character for a desired function. When the character is entered, the microprogram invokes the appropriate routine to provide the action required. For example, the operator can invoke the PSW restart function by displaying the operator (OP) display frame and by entering the characters assigned to the PSW restart function. The station processor then begins a PSW restart.

Note that the key location numbers (1-67) run from far left to far right, and from top to bottom.

The function of each of the keys for the program (PR) frame is shown in Figure 4-3. Some of the functions vary according to which frame is active.

Control Panel

The control panel (Figure 4-4) contains the switches, pushbuttons, and other functions not included on the keyboard, which are:

- Unit Emergency switch
- IPC Reset pushbutton
- Power On pushbutton (backlighted)
- IMPL Pending indicator
- Microcode Power Control indicator
- Power Off Pending indicator
- Power Off pushbutton
- Power Select switch
- I/O Interface switches (one for each station)
- Diagnostic on IMPL switches (one for each station)
- IMPL pushbuttons (one for each station)
- Operator Console on IMPL switch
- TP Active–Key Reset pushbutton (backlighted) with Activate TP keylock
- TOD Clock switch
- Alarm Volume control
- Meters
- Key switch for control of the meters

Unit Emergency Switch

Setting the Unit Emergency switch to the Power Off position removes all power from the console, the 3031 Processor, and 3041 Attached Processor, but not from the motor generator. When set, this switch latches in the down position and can be restored by service personnel only.

IPC Reset Pushbutton

When console power is turned off, the IPC Reset pushbutton resets any power fault latches that are set in the initial power controller (IPC). To power up the system, press this pushbutton before pressing the Power On pushbutton.

Power On Pushbutton

After pressing the IPC Reset pushbutton, the Power On pushbutton is pressed to turn power on in the console and to load the microcode into both console processors. If the Power Select switch is set to System, then the system is powered up; if this switch is set to Console, then the system is not powered up, and the power control (PC) display frame is displayed for possible further action. When all power supplies are powered up and no check conditions exist, the console changes the backlighting of the pushbutton from red to white.

IMPL Pending Indicator

The IMPL Pending indicator lights during the power-on sequence, indicating that basic console power is available and that an initial microprogram load (IMPL) sequence was started.

Key Location Numbers	*Name*	*Description*
1	CNCL (Cancel)	Presents an attention interruption to the channel.
2-13, 19-29, 35-45, 51-60	Data Keys	Alphameric characters.
14	← (Backspace)	Moves the cursor backward one space.
15	START	Starts processor instruction execution.
16	STOP	Stops processor instruction execution.
17	SEL (Select) FRAME	Causes the selected frame to be displayed.
18	→| (Forward Tab)	Causes the cursor to space to the first character location of the next unprotected field. If the buffer is unformatted or if there are no unprotected fields, the cursor is repositioned to character location 0.
30	|← (Backward Tab)	If the cursor is located in the attribute character or the first alphameric character of an unprotected field or any character of a protected field, the cursor goes to the first alphameric character of the first preceding unprotected field. If the cursor is located in the character of an unprotected field other than the first, it is moved to the first character location of that field. If the buffer is unformatted or has no unprotected fields, the cursor is moved to character location 0.
32	IRPT (Interrupt)	Presents an external interruption to the processor.
33	PFK SEL (Select)	Changes keys 1 through 0, -, and & to program function (PF) keys 1-12 and displays the letters PFK at the bottom of the screen.
34	LOCK	Causes logical lock action (thereby putting the keyboard in uppercase mode) until either SHIFT key is pressed and released.
46	↵ (New Line)	If the buffer is formatted, the cursor goes to the first unprotected character location of the next line containing unprotected characters. If there are no unprotected fields, the cursor goes to character position 0. If the buffer is unformatted, the cursor goes to the first character of the next line.
47, 48, 62, 63	↑ ↓ ← → (Cursor Keys)	Control the cursor in the direction of the arrow.
50, 61	SHIFT	Puts the keyboard in uppercase mode.
64	KEYBD (Keyboard) RESET	Resets an inhibit keyboard condition and a pending interruption condition, if the keyboard is not disabled because of an I/O operation in progress.
65, 66	Spacebar	Writes a blank in the cursor location and moves the cursor forward one space. If the cursor was under a protected character, the keyboard becomes disabled.
67	ENTER	Causes an attention interruption on which device-independent display operator console support (DIDOCS) issues a Read Modified command to retrieve previously entered data.

Figure 4-3. Key Functions for the Program Frame

Microcode Power Control Indicator

The Microcode Power Control indicator lights when the Power Select switch is turned to System, indicating that system power is under microcode control.

Power Off Pending Indicator

Pressing the Power Off pushbutton turns on the Power Off Pending indicator and causes a power-off interruption to the station processor. When the station processor takes the interruption, system power is turned off.

Power Off Pushbutton

Pressing the Power Off pushbutton starts the power-off sequence in the console. The console sequences the power off in the reverse order from power on.

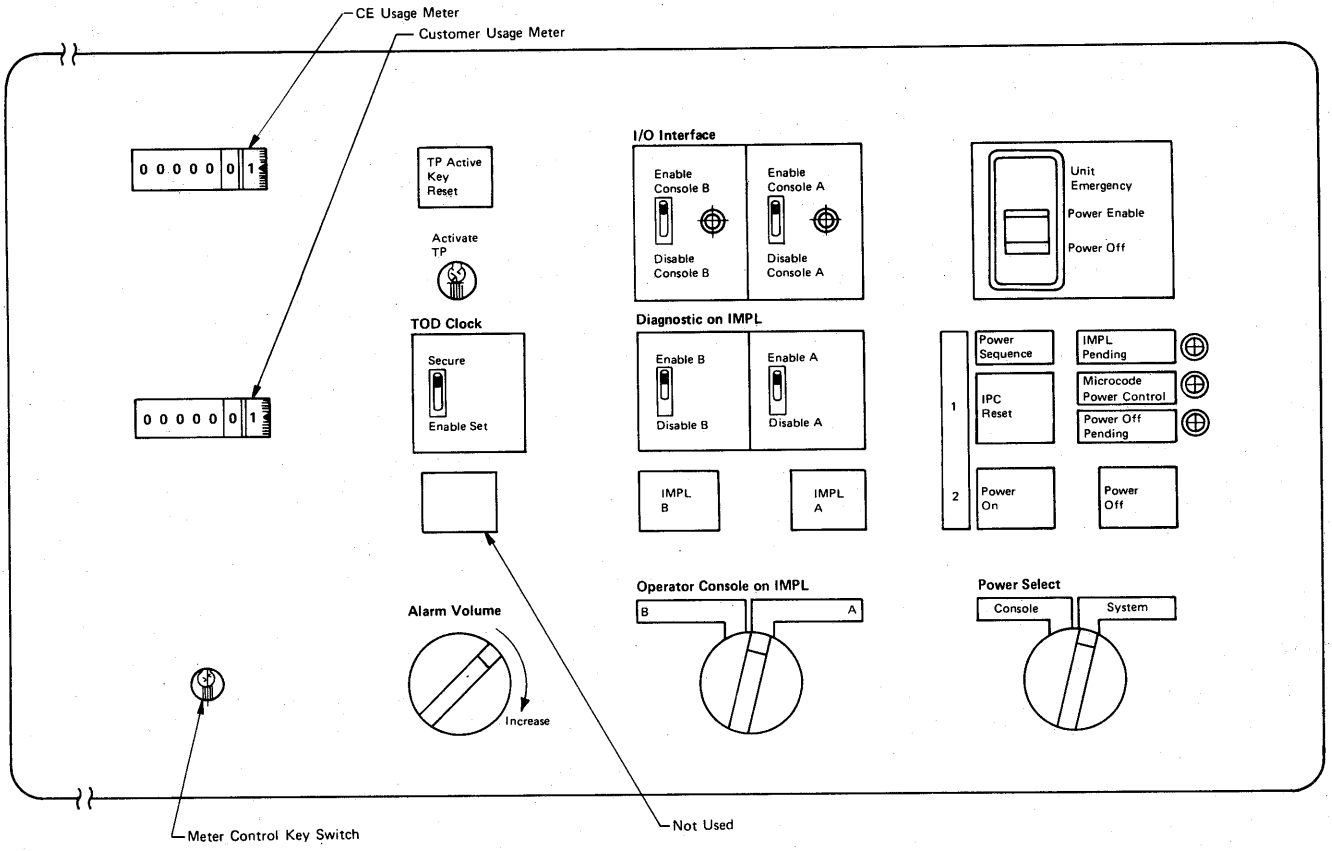


Figure 4-4. Control Panel of the IBM 3036 Console

Power Select Switch

The Power Select switch determines whether power is turned on for the entire system or just for the console.

I/O Interface Switches

Each of the two I/O Interface switches can either permit (enable) or inhibit (disable) the interface between the associated station and the corresponding channel. Both switches may be enabled at the same time.

Diagnostic on IMPL Switches

If the IMPL pushbutton for either station is pressed while the corresponding Diagnostic on IMPL switch is enabled, the station processor of the corresponding station executes the basic console microdiagnostic program before loading the control microprogram.

IMPL Pushbuttons

Pressing either the IMPL A or IMPL B pushbutton causes initial microprogram loading (IMPL) of the corresponding station, if that station's security key is turned on (horizontal). If the corresponding Diagnostic on IMPL switch is in the Enable position, console diagnostic programs can be run before the IMPL.

Operator Console on IMPL Switch

The Operator Console on IMPL switch is used in conjunction with the configuration display frame (C1 frame) and the IMPL pushbuttons to reverse the configuration of the operator station and the service station when required.

TP Active–Key Reset Pushbutton

The TP Active–Key Reset pushbutton is used in conjunction with the TP-link (TP) display frame for data communication (teleprocessing) operations. Enabling this pushbutton requires that the Activate TP key switch (located below the pushbutton) be turned on with the CE key. When the key switch is turned on, the pushbutton is backlighted yellow, and the CE key may be removed. Pressing the pushbutton unlatches TP Active and causes all data communication operations to cease.

TOD Clock Switch

The TOD Clock switch provides an interlock, with the Set Clock (SCK) instruction, as a means of guarding against an improper change to the TOD clock. When the switch is held down, the instruction can change the clock. When this

switch is in the normal (Secure) position, then the clock cannot be set and is referred to as being secure. The switch has a spring return; when released, it returns to the normal position.

Alarm Volume Control

The Alarm Volume control controls the volume of the console alarm.

Meters

Two meters, which are not needed to operate the console, indicate time usage. The key switch, located below the meters, determines which meter is active.

Security Key

The security key fits the security keylock located on the right-hand side of each station display. No frame changes are allowed if the security key is not turned on (horizontal). To be removed, the key must be turned off (vertical).

With the security key turned off or removed from either station, the IMPL pushbutton for that station is disabled, and system data security is under console microprogram control. The SEL FRAME key (on the keyboard) becomes inoperative, thereby preventing any changes in the frame and precluding access to any other frames. By turning the security key on, the SEL FRAME key becomes operable and permits the operator to change the frame.

Operator Display Frames

The operator display frames are requested by the operator as needed, using the index (IN) display frame (Figure 4-5). The frames listed on the IN frame depend on the station (operator or service) and its mode of operation (see "Console Reconfiguration" in this chapter). In addition to the IN frame, the operator display frames include:

Operator (OP) Frame, which controls basic operations of the 3031 Processor Complex and 3031 Attached Processor Complex such as IPL, reset, restart, address-compare stop, and store status.

Program (PR) Frame, which allows operator/system communication of system messages, operator commands, and status displays. This frame serves as the output device for the control program that operates the system.

Alter/Display (AD) Frame, which allows the operator to alter or display storage and specific registers.

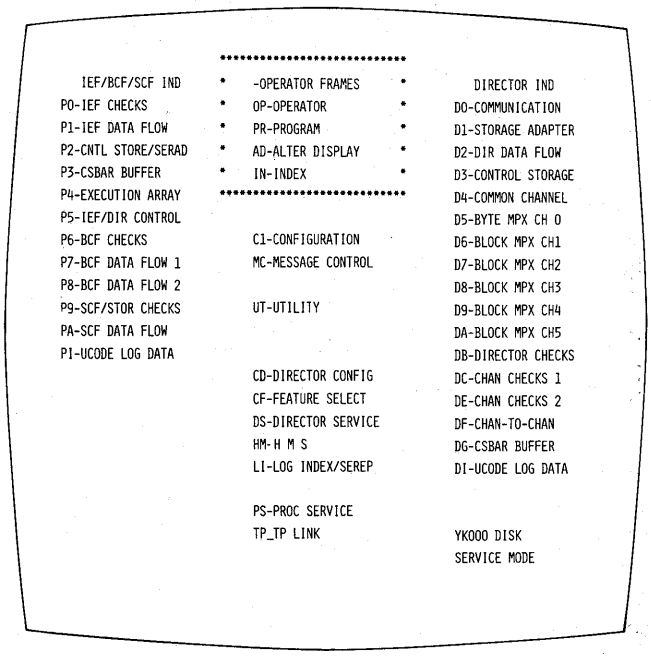


Figure 4-5. Index (IN) Frame at the Service Station

Configuration(C1) Frame, which displays the station/port configuration and the logical/physical element configuration of processor storage and allows either to be changed.

Director Configuration (CD) Frame, which allows the operator to enable or disable the channel-to-channel adapter, displays shared control unit assignments that can be altered, and allows saving the shared control unit assignment data on a new diskette.

Feature Selection (CF) Frame, which allows the operator to configure available features to the IEF.

Message Control (MC) Frame, which displays messages that indicate two types of error conditions: those that may respond to operator action and those that call for the services of a customer engineer.

Power Control (PC) Frame, which allows the operator to control the power for the entire system or for individual units, and to control the display of associated monitoring activity.

Initial microprogram load (IMPL) causes the configuration (C1) frame to be displayed at the operator station. The security key switch must be unlocked for any other frame to be displayed. In normal operating mode, the index, configuration, power control, and program frames are displayed at the operator station; all others are displayed at the service station. In maintenance mode, any frame can be displayed at the operator station.

OPERATOR CONSOLE CHARACTERISTICS

The following information applies when a station is used as the operator station.

The station appears to the operating system as an IBM 3277 Display Station Model 2 with a keyboard attached to an IBM 3272 Control Unit, and obeys a subset of the 3277 commands and orders.

Console Commands

The console accepts the following commands:

Command	*Hexadecimal Code*
Test I/O (TIO)	00
Write	01
Read Buffer	02
No Operation (No-Op) *	03
Sense	04
Erase/Write	05
Read Modified	06
Select *	0B
Erase Unprotected *	0F
Sound Alarm	E3
Sense I/O	E4

The console rejects the following commands:

Copy	07
Diagnostic Write	09
Diagnostic Read	0A

* Immediate operation commands, which cause the channel-end condition to be signaled during the initiation sequence.

Operator Function Keys

The console provides and supports the RESET key, the cursor control keys, and the forward tab and backward tab keys.

Program Access Keys

The console provides and supports the ENTER key, the cancel (CNCL) key, and the program function (PF) keys.

Addressing

The console has one unit address. The unit address consists of an eight-bit byte (plus a parity bit) and can be set to any of the 256 possible addresses at installation time.

Display Operation

The program (PR) frame display has two parts. The upper part is used for communications between the operator and the system. Positioning and deletion of messages from the upper part is under program control. The lower part is the bottom line of the screen which is used for the presentation of system status indicators.

Character Set

Ninety-six characters are specified for transfer to the console and for storage in the display buffer:

- 26 uppercase alphabetic characters (A-Z)
- 26 lowercase alphabetic characters (a-z)
- 10 numeric characters (0-9)
- Space
- 33 graphic characters (shown with their hexadecimal character codes)

Graphic Character	*Hexadecimal Code*	*Graphic Character*	*Hexadecimal Code*
¢	4A	↑	64
.	4B	↓	65
<	4C	→	66
(	4D	←	67
+	4E	←	68
|	4F	,	6B
&	50	%	6C
!	5A	_	6D
$	5B	>	6E
*	5C	?	6F
)	5D	:	7A
;	5E	#	7B
¬	5F	@	7C
-	60	'	7D
/	61	=	7E
→	62	"	7F
←	63		

The console permits hexadecimal codes for four control-character representations and another graphic character. If the hexadecimal codes are transmitted to the display, they are stored in the buffer as data, but are displayed as blanks.

	Hexadecimal Code
NL (New Line)	15
EM (End of Medium)	19
DUP (Duplicate)	1C
FM (Field Mark)	1E
¦	6A

The display buffer does not provide for storage of hexadecimal character codes for any other characters or character representations. If other character codes are transferred to the display, the data characters stored and displayed are not defined. The character codes returned, as the result of a buffer read operation, may or may not be the same as the character codes transferred to the display.

Error Recovery Procedures

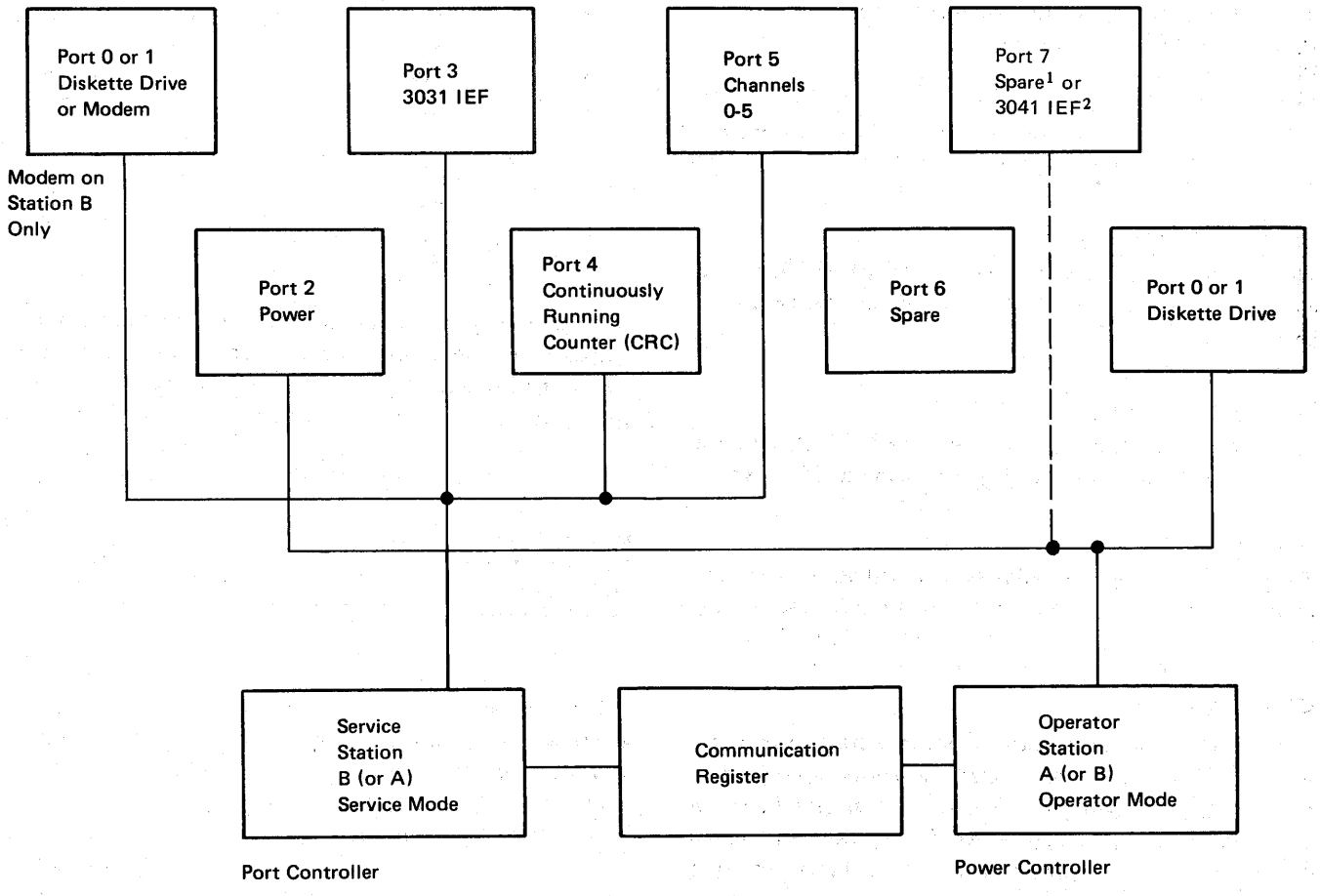
The 3277-2 error recovery procedures apply to the console. They are described in the *IBM 3270 Information Display System Component Description,* GA27-2749.

CONSOLE RECONFIGURATION

The two console processors can service as many as eight different access points (called *ports*) in different configurations, under control of the configuration (C1) display frame at the service station. Figure 4-6 shows the normal (default) configuration of the console after a power on is executed, in which the operator and service stations operate in their normal modes (operator and service, respectively).

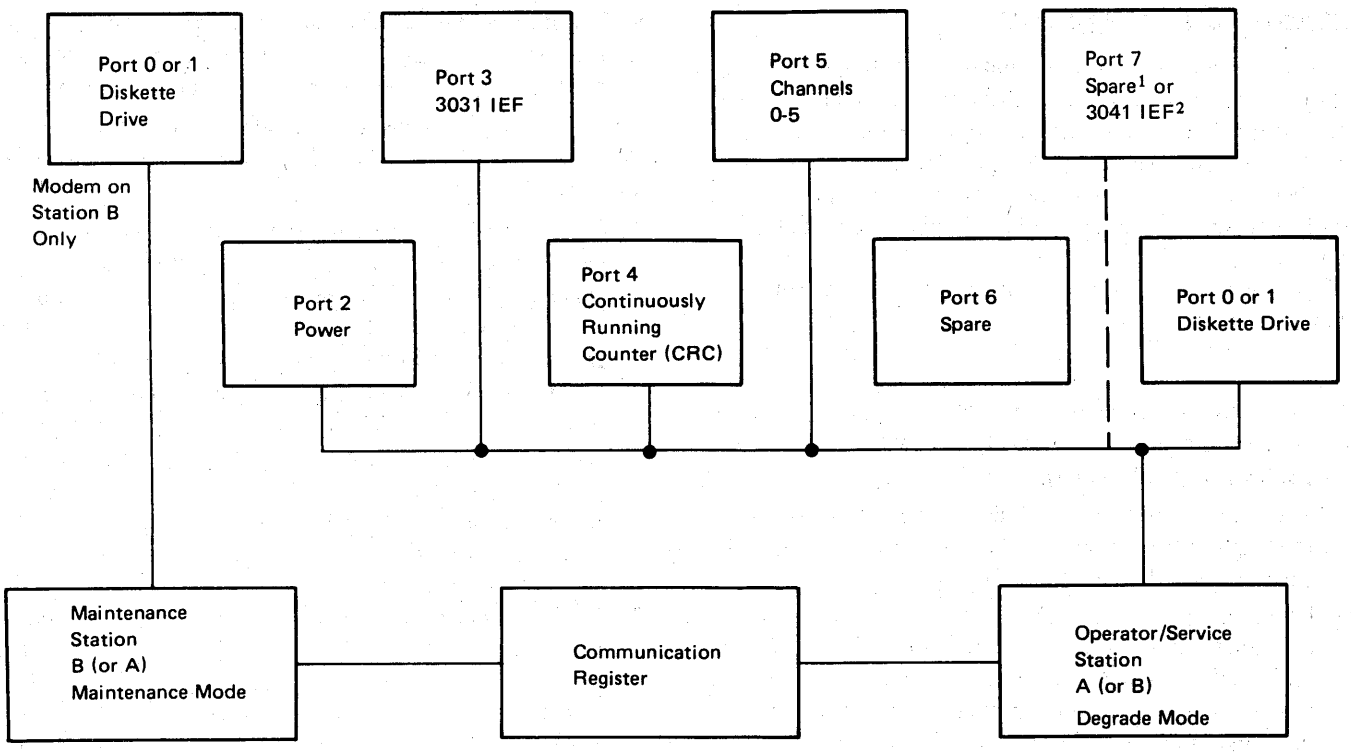
If a failure occurs in one of the console processors, displays, keyboards, I/O interfaces, diskette drives, or the 3041 Attached processor, the ports are reconfigured. Reconfiguration allows system operation through the combined operator/service station concurrent with maintenance on the console itself, or on the 3041. After reconfiguration there may be a slight slowing of response to the operator when an execption condition (such as a channel error) occurs simultaneously with operator responses or requests. Because of this change in response, the mode of the operator/service station is referred to as degrade mode. The mode of the other station is referred to as maintenance mode until the problem is corrected and the console is reconfigured to the normal configuration. The port configuration for system operation concurrent with console maintenance is shown in Figure 4-7, and for system operation concurrent with 3041 maintenance in Figure 4-8.

The dedicated diskette drive on each console processor is the primary attached drive. The drive is also port-controlled and can be accessed by either console processor. The primary drive is attached to its dedicated console processor on port 0. When the drive is accessed by the other console processor, the drive is attached to that console processor's port 1.



[1] In a 3031 Processor Complex
[2] In a 3031 Attached Processor Complex

Figure 4-6. Normal Configuration



[1] In a 3031 Processor Complex
[2] In a 3031 Attached Processor Complex

Figure 4-7. Configuration for 3036 Console Maintenance Concurrent with System Operation

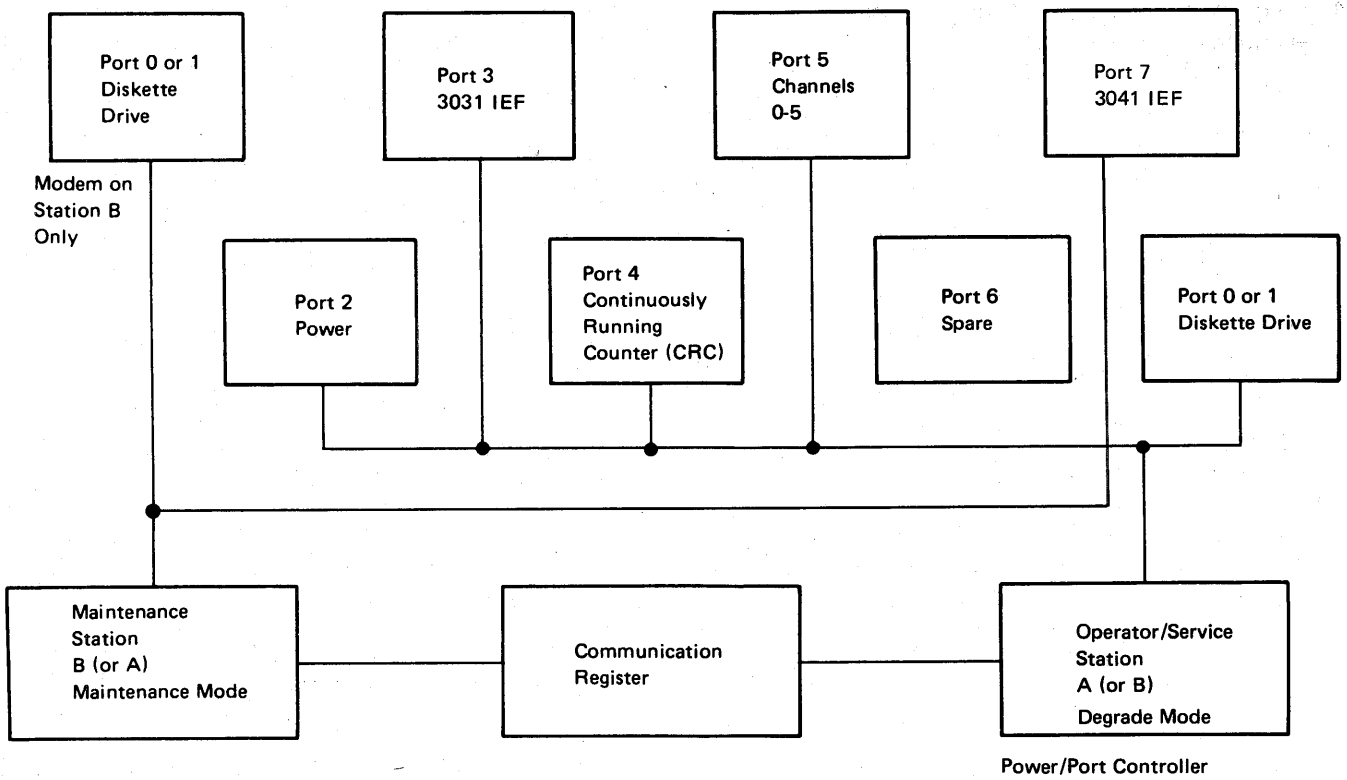


Figure 4-8. Configuration for 3041 Attached Processor Maintenance Concurrent with System Operation

REMOTE SUPPORT FACILITY

If the system malfunctions, the remote support facility allows remote support specialists to access maintenance data through a data communication (teleprocessing) network. The advantages of this support technique are evident; however, the customer's data security and privacy are subject to some exposure while the remote support facility is in use.

The security measures enlisted to ensure minimal exposure are as follows:

1. The customer's console security key is required to display the data-communication (teleprocessing) link display frame (TP frame).
2. The data-communication link is established only after thorough verification of the identity of both the customer and the remote support specialists.
3. Each mode of remote support (remote program, transmits logs, and remote console) is initiated and identified at the customer's installation. Before establishing a data-communication link, service personnel must establish the customer's security level by selecting the remote options allowed by the customer. Any options selected are indicated on the TP frame by an asterisk to the left of the options selected. Any attempt to select unallowable options results in an error message on the TP frame.
4. Every operation initiated at the remote center can be monitored by the customer.
5. The data-communication link can be disconnected at any time at the customer's installation by pressing the TP Active–Key Reset pushbutton on the control panel.

One of the following three modes of operation can be selected at the discretion of the service personnel and with the customer's approval:

1. *Remote Program*: This mode provides all of the online test executive program (OLTEP) security facilities. For example, to protect against accidental modification of customer data, OLTEP and OLTSEP diagnostic programs restrict writing to noncustomer volumes or to designated areas of customer volumes. Also, to protect against disclosure, OLTSEP diagnostic programs read and transmit the smallest amount of data that permits satisfactory diagnosis.
2. *Transmit Logs*: This mode allows the remote specialist to copy selectively the logout data retained by the service support station for offline analysis.
3. *Remote Console*: In this mode, most console functions are available to the remote center. The keyboard of the service support station is inoperative; however, the customer can terminate the operation by pressing the TP Active–Key Reset pushbutton on the control panel. Note that all data sets on the system could be accessed by the remote center; however, every operation initiated at the remote center can be monitored by the customer.

All modes of operation are independent of operating system release levels. This facility does not depend on the processor, I/O devices, or channels being operational.

Chapter 5. Channel Characteristics

The channels transfer data between processor storage and I/O devices under control of a channel program executed independently of other operations being performed by the processor. The processor is free to resume its other operations after initiating an I/O operation.

At the end of an I/O operation, the channel signals an I/O interruption request to the processor. If not disallowed, an I/O interruption occurs that places the processor under control of the I/O new PSW. When I/O interruptions are disallowed, interruption requests are queued. Until honored, an I/O interruption condition is called a pending I/O interruption.

At the end of an I/O operation, a channel has information concerning the success of the operation, or has detailed information about any lack of success. This information is available to the control program.

The channel group has facilities for performing the following functions:

- Accepting an I/O instruction from the processor
- Addressing the device specified by an I/O instruction
- Fetching the channel program from processor storage
- Decoding the channel command words (CCWs) that make up the channel program
- Testing each CCW for validity
- Executing CCW functions
- Placing control signals on the I/O interface
- Accepting control-response signals from the I/O interface
- Transferring data between an I/O device and processor storage
- Checking parity of bytes transferred
- Counting the number of bytes transferred
- Accepting status information from I/O devices
- Maintaining channel-status information
- Signaling interruption requests to the processor
- Sequencing interruption requests from I/O devices
- Sending status information to location 64 (decimal) when an interruption occurs
- Sending status information to location 64 (decimal) on processor request

CHANNEL CONTROL

A major feature of the channels is their common I/O interface connection to all System/370 input/output control units. The I/O interface provides for attachment of a variety of I/O devices to a channel. The interface is governed by six basic channel commands and a common set of input/output instructions, which include:

- Start I/O
- Start I/O Fast Release
- Test Channel
- Test I/O
- Halt I/O
- Clear I/O
- Halt Device
- Store Channel ID
- Clear Channel

All I/O instructions set the PSW condition code, and (under certain conditions) all instructions except Test Channel may cause a channel status word (CSW) to be stored. Start I/O causes an operation to begin after a device is selected; Start I/O Fast Release causes an operation to begin independently of device selection. A Test Channel instruction elicits information about the addressed channel; a Test I/O instruction elicits information about a channel and a particular I/O device. Halt I/O terminates any operation on the addressed channel, subchannel, and I/O device. Halt Device terminates only operations associated with the addressed I/O device. Store Channel ID places information identifying the designated channel in a specified location. Only Start I/O uses channel command words (CCWs).

A Start I/O instruction initiates execution of one or more I/O operations, and specifies a channel, a subchannel, a control unit, and an I/O device. Start I/O causes the channel to fetch the channel address word (CAW) from location 72 (decimal). The CAW contains the protection key and the address of the first channel command word (CCW) for the operation. The channel fetches and executes one or more CCWs, beginning with the first CCW specified by the CAW.

Six channel commands are used:

- Read
- Write
- Read Backward
- Control
- Sense
- Transfer in Channel

The first three commands are self-explanatory. Control commands specify such operations as set tape density, rewind tape, advance paper in a printer, or sound an audible alarm.

A Sense command brings information from a control unit into processor storage concerning unusual conditions detected during the last I/O operation and detailed status about the device.

A Transfer in Channel (TIC) command specifies the location in processor storage from which the next CCW in the channel program is to be fetched. A TIC may not specify another TIC. Also, the CAW may not address a TIC.

Each CCW specifies the channel operation to be performed, and (for data transfer operations) specifies contiguous locations in processor storage to be used. One or more CCWs make up a channel program that directs a channel operation.

Command retry is a channel-to-control unit procedure that can cause a command to be retried without requiring

an I/O interruption. Retry is initiated by the control unit. When the command being executed encounters an error that can be retried, the control unit presents retry status to the channel. If conditions permit, a normal device reselection occurs to reissue the previous command; if retry is not possible, any chaining is terminated and an I/O interruption follows.

Channels and Subchannels

The channel facilities required to sustain a single I/O operation are called a subchannel. Subchannels may be either nonshared or shared. A nonshared subchannel has the facilities to operate only one I/O device; a shared subchannel provides facilities to operate one of an attached set of I/O devices.

The channels maintain the following channel control information for each I/O device selected:

- Protection key
- Data address
- Identity of operation specified by command code
- CCW flags
- Byte count
- Channel status
- Address of next CCW

Chaining

A single CCW may specify contiguous locations in processor storage for a data transfer operation, or successive CCWs may be chained to specify a set of noncontiguous storage areas. Chaining to the next CCW is caused by the presence of a flag bit in a CCW.

In data chaining, the address and count information in a new CCW is used; the command code field is ignored unless a TIC is specified.

Entire CCWs, including their command code fields, may also be chained for use in a sequence of channel operations. Such coupling is called command chaining, and it is specified by a different flag bit in a CCW.

Data chaining has no effect on a device, as long as the channel has sufficient time to perform both data chaining and data transfer for the device.

In this manual, when a device is said to chain data, it means that the channel program for the device specifies data chaining.

Fetching Channel Command Words

The channel must fetch a new CCW when a CCW specifies data chaining, command chaining, or transfer in channel (TIC). The extra control activity caused by these operations takes time, and diminishes the capability of the channel to do other work.

A data-chaining fetch operation usually occurs while a channel also has a data transfer load from the same device. The time required to fetch the new CCW necessarily limits the interval of time available for successive data transfers through the channel. An absence of data chaining ordinarily permits a channel to operate with a faster I/O device.

Data Chaining in Gaps

For direct access storage devices, such as an IBM 3330 Disk Storage or an IBM 2305 Fixed Head Storage, formatting write commands causes the control unit to create gaps between count, key, and data fields on the recording track. Read and write commands that address more than one of the fields may specify data chaining to define separate areas in processor storage for the fields.

The gaps on a track have significance to channel programming considerations for direct access storage devices. The channel does not transfer data during the time a gap is created or passes under the read/write head, and this time is sufficient for a 3031 Processor to perform a command-chaining or data-chaining operation.

Command chaining ordinarily occurs only during gap time, but data chaining may occur during gap time or while data is being transferred. A data-chaining operation occurring during gap time has a lesser impact on channel facilities than when data transfers also occur. If a channel program for a direct access storage device calls for data chaining only during gap time, the overall load of the device on channel facilities is significantly less.

When a direct access storage device is said to chain data in a gap, the reference is to a gap other than a gap following a data field. The latter gap causes a device-end indication and command chaining is used in such a gap if the transfer of more information is desired. A device-end condition occurring in the absence of a CCW specifying command chaining results in termination of the operation. When command chaining continues the operation, the status information available at the end of the operation relates to the last operation in the chain.

During a read operation, an attempt at chaining data in a gap following a data field causes an incorrect-length indication in the channel status byte.

Data Chaining Considerations

Chaining checks on read data chaining operations, or overruns on write data chaining operations can occur because of discrepancies in the following:

- Data addresses that are not on fullword boundaries.
- Short byte counts (see the following guidelines).
- Device speed (on the channel that is data chaining).
- Other channel activity.

The following guidelines are recommended to reduce possible chaining checks and overruns during data chaining operations:

- Ensure that data addresses are on fullword boundaries.

- Ensure that the CCW byte count is greater than or equal to (64)S/1.2, where S is the device speed in megabytes per second.

Late Command Chaining

Operation of direct access storage devices, such as disk storage, requires the use of command chaining. Between certain operations, such as searching for a record identification key and reading a data field on a direct access storage device, the control unit has a fixed time interval during which it must receive and execute a new command. Certain I/O devices (such as the IBM 3330 Disk Storage and the IBM 3350 Direct Access Storage) can cause a command retry operation without requiring an I/O interruption. If activity on other channels causes too much delay in initiation of the operation specified by the new command, the channel program is terminated and an I/O interruption condition occurs.

CHANNEL IMPLEMENTATION

Each channel may attach as many as eight control units and can address as many as 256 I/O devices. Control units are connected to all channels through the I/O interface.

Multiplexing refers to the ability of the channels and devices to disconnect and reconnect during an operation over the I/O interface. The block-multiplexer channel operates in burst mode and can multiplex between blocks of data; the byte-multiplexer channel operates either in burst mode or in byte mode, and can multiplex between bytes, groups of bytes, or blocks.

Burst Mode is defined as operation over the I/O interface in which the device and the channel remain connected for a relatively long period of time in terms of system operation.

Byte Mode is defined as byte-interleaved operation over the I/O interface in which the channel and any one device remain connected for a relatively short period of time, typically long enough to transfer one byte or a small number of bytes.

Byte-Multiplexer Channel

A byte-multiplexer channel has a single data path that may be monopolized by one I/O device (burst mode) or shared by many I/O devices (byte mode). The design of a control unit predetermines whether its operation on the byte-multiplexer channel is in burst or byte mode. In either case, data transfer between storage and an I/O device is controlled one byte at a time. Byte-multiplexer channel operation may overlap block-multiplexer channel operation.

When multiple I/O devices concurrently share byte-multiplexer channel facilities, the operations are in byte mode. Each device in operation is selected, one at a time, for transfer of a byte or a group of bytes to or from processor storage. Bytes from multiple devices are interleaved and routed to or from the desired locations in processor storage. Therefore, the byte-multiplexer channel data path is used by one device for transfer of one byte or a group of bytes, and then another device uses the same data path. The sharing of the data path makes each device appear (to the programmer) as if it has a data path of its own. This leads to calling a device's share of the data path a subchannel.

Block-Multiplexer Channel

Each block-multiplexer channel provides a path for moving data between storage and a selected I/O device. It has storage for control information and data buffering. Data moves to or from an I/O device one byte at a time, but it is buffered to a width of 16 bytes for communications with storage.

A block-multiplexer channel can operate in three different ways:

1. When operating in block-multiplex mode with a nonshared subchannel that has a UCW assigned, the channel follows all block-multiplexing rules. (These rules are in *IBM System/370 Principles of Operation*, GA22-7000.)
2. When operating in block-multiplex mode with a shared subchannel, and the subchannel block-multiplexing bit is on, the channel follows the block-multiplexing rules. If the block-multiplexing bit is off, the channel operates as a selector channel.
3. A block-multiplexer channel operates as a selector channel under control of the operating system, as determined by bit 0 in control register 0.

UNIT CONTROL WORDS

The initiation of multiple I/O operations with logic-controlled channel multiprogramming requires that the subchannels be provided channel storage to record the addresses, count, and status and control information associated with the I/O operation. In the 3031 Processor, the storage for a single set of such information is called a unit control word (UCW). UCWs are stored in a 32K-byte buffer that is referred to as UCW storage.

On both byte- and block-multiplexer channels, the channel-control information (for each I/O device selected) is maintained for each subchannel in operation. When a particular subchannel is selected by a Start I/O instruction and a channel program is initiated, the UCW locations for the subchannel are loaded with the information necessary for subchannel operation.

Block-Multiplexer Channel UCW Assignment

Block-multiplexer channels assign devices to nonshared UCWs as needed. Shared UCWs are assigned during initial microprogram load (IMPL). As many as 40 shared UCWs

are assigned for the basic channel group. Fewer may be specified, depending on customer requirements.

Assignment of the configuration is made by service personnel during system installation. This assignment can be easily changed if the system is reconfigured.

During installation or reconfiguration, the following parameters must be specified for each shared subchannel:

- Number of devices installed (8, 16, or 32) for each control unit
- Mode of operation (selector or block multiplex, for each control unit)

Each shared subchannel refers to a block of 16 of 32 contiguous device addresses of the following form:

X0 through X7	(for 8 addresses)
X8 through XF	(for 8 addresses)
X0 through XF	(for 16 addresses)
X0 through (X+1) F	(for 32 addresses)

Only one control unit can be attached to each shared subchannel.

CHANNEL INDIRECT DATA ADDRESSING

Channels do not implement dynamic address translation. CCWs in virtual storage must be translated by the control program before execution. To allow the designation of contiguous areas of virtual storage to be mapped into noncontiguous areas of real storage, channel indirect data addressing (CIDA) is provided. For further information concerning CIDA, see *IBM System/370 Principles of Operation,* GA22-7000.

CHANNEL PRIORITY

For normal operation, priority for allocation within the channel group is in the following order:

- Block-multiplexer channel data transfer
- Block-multiplexer channel data chaining
- Block-multiplexer channel command chaining
- Byte-multiplexer channel operations
- Director operations

Block-multiplexer channels receive data handling priority in numeric order, within the channel group.

I/O interruption priority goes across the channel group, and is in order of channel number, with the highest priority given to the lowest-numbered channels.

Channel Available Interruption

The 3031 Processor implements the channel available interruption on the block-multiplexer channels. The channel available interruption is not implemented on the byte-multiplexer channel.

CHANNEL LOADING AND THROUGHPUT

Each I/O device in operation places a load on its channel facilities. The magnitude of the load (and consequently the throughput rate) depends on many factors, such as the device's channel programming, its data transfer rate, and the use of command chaining.

OVERRUN

Overrun occurs when a channel does not accept or transfer data within required time limits. This data overrun may occur when the total channel activity initiated by the program exceeds channel capabilities. Depending on the device, the channel may halt operation or may continue transferring data until the end of the block is reached.

An overrun may cause a unit-check indication to be presented to the channel and stored into the CSW. Chaining, if any, is suppressed and an I/O interruption condition is generated at the end of the operation. Certain control units, however, may initiate a command retry sequence without storing a CSW or requiring an I/O interruption.

Overrun occurs only on unbuffered I/O devices. Buffered devices are not subject to overrun. Instead, when buffer service is not provided within required time limits, the device merely waits for channel service.

Devices such as the IBM 2501 Card Reader, the IBM 1419 Magnetic Character Reader, or the IBM 3705 Communications Controller (in emulation mode) may require operator intervention in overrun situations.

Chapter 6. Hierarchical Monitoring System

The Hierarchical Monitoring System (HMS) is a service aid that gathers trace data by monitoring various system operations. This data is used by service personnel in diagnosing software/microcode and hardware/microcode problems. The monitoring and the gathering of trace data are controlled by an HMS program that is generated on the HMS frame (Figure 6-1) by the user. The HMS program traps data and uses it to compare, calculate, store, and display the results.

Data is gathered at the software level through program monitoring, and at the microcode level through microcode monitoring. Program monitoring and microprogram monitoring can be mixed in any HMS program.

In program monitoring, HMS monitors instruction streams or processor storage alterations for specified conditions that, simultaneously met, halt instruction processing (trapping). A number of system facilities (for example, the program status word and the contents of general registers and control registers) can then be accessed by the display function of alter/display to compare, test, calculate, store, and display the data. When these operations are completed, instruction processing continues until conditions for the next trap command are met, or until program execution is interrupted or ended.

In microprogram monitoring, HMS monitors reloadable control storage for execution of the word at a specified address. When the specified word is encountered, the RCS address stop function stops the processor clocks to allow the HMS program to access system (status) indicators. The processor clocks remain stopped while the HMS program continues until either another trap command (ITRAP, STRAP, or RTRAP) is executed or the HMS program ends.

The data obtained from HMS monitoring can be stored in the HMS save areas from where it is displayable and is also available as input (if desired) to the HMS program. Trace data is stored in the trace buffer, which holds 32 words that are not available as input to the program, but can be displayed when the HMS program ends.

HMS COMPONENTS

The HMS facility consists of the HMS frame, which contains the HMS controls and displays HMS results, and the following microcode routines:

- The controller, which maintains the frame image, communicates with the operator, initializes and controls the HMS executor routine, and displays the status and results of the HMS operations.
- The assembler, which converts alphanumeric input into HMS commands, checks command syntax, and assembles commands into executable format.
- The executor, which decodes and executes HMS commands; starts and stops the system; initializes logical address stops or RCS address stops; communicates with the display microcode; maintains HMS trace and save areas; and performs tests, comparisons, and arithmetic calculations on data in HMS save areas or on data obtained from the system.

HMS COMMANDS

The following commands are used in the HMS program:

- ITRAP and STRAP, which set instruction and storage traps, respectively, in the processor to gain access to system facilities (such as the PSW and the contents of general registers and control registers).
- RTRAP, which stops the processor clocks to gain access to system indicators.
- LS, which loads an HMS save area with data and uses that data in calculations (with a constant or data from another save area).
- COMP, which compares save areas with each other or with a constant and provides a branch within the HMS program.
- TEST, which tests a save area by using another save area or a constant as a mask and provides a branch within the HMS program.
- TRACE, which stores data from system facilities in the trace buffer.
- STOP, which deactivates HMS and activates an alarm to alert the operator that operator action is required.
- END, which ends execution of the HMS program.

HMS can be used in a real-storage or a virtual-storage system while the system continues its normal operation. However, system performance is decreased when trap conditions are met. In addition, execution of the STOP command leaves the processor in the manual state (if the STOP command executes after an ITRAP or STRAP command) or in a clock-stopped state (if the STOP command executes after an RTRAP command). Operator intervention is then required to resume processing. The possibility of an overrun condition exists in I/O operations that occur concurrently with the execution of the RTRAP command in the director.

The Diagnose instruction allows software initiation of an HMS program. The format for this instruction is given in "Operating Procedures."

HMS FRAME

The HMS frame permits operator interaction with the HMS program through the keyboard. (The operator can type control characters, or position a cursor and type input characters as the operation requires. The operation is completed when the operator presses the ENTER key.)

The HMS frame consists of two parts: the working frame (Figure 6-1), which is initially displayed, and the input mode frame (Figure 6-2), which is selected from the working frame. The frame elements in Figures 6-1 and 6-2 are keyed to descriptions that follow the figures.

The working frame consists of:

- Controls **A**, which enable the operator to:
  Select input mode.
  Activate, terminate, and save an HMS program.
  Load a saved HMS program.
  Display trace data.
- Activity indicators **B**, which display the status of HMS (active or inactive), the number of logical address or RCS address stops that have occurred (count wraps to 0 after 1F hexadecimal), and the end code of the previous program (why the program ended). The following are the end codes and their meanings:

Code	*Reason HMS Ended*
0	END command
1	TEST or COMP command
2	END on STOP command
3	STOP on TEST or COMP command
4	Accumulator overflow
5	Accumulator underflow (result of subtraction is negative)
6	Invalid address from processor storage
7	Microcode timeout (HMS program loop)
8	Manual intervention (END HMS control character)
9	Assembler or HMS program error (error condition)

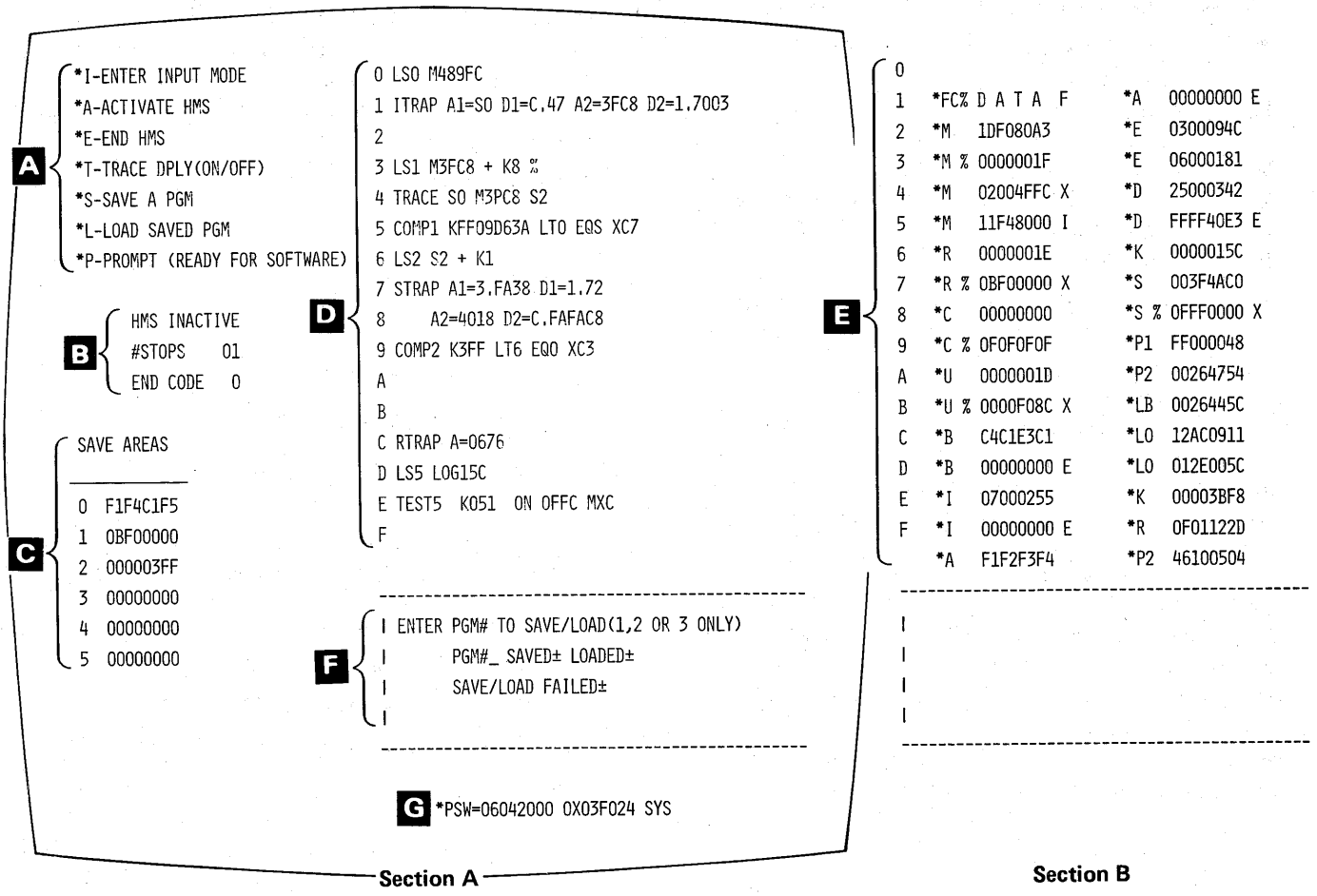


Figure 6-1. HMS Working Frame (Section A showing HMS active, Section B showing HMS inactive and trace area displayed)

Code	*Reason HMS Ended*
A	Processor or director microcode hang (error condition)
B	PSW restart, load, load clear, or system reset clear
C	HMS not being successfully activated
D	Processor or console interface parity error
E	Processor or director not available to service support station of the console
F	Console IMPL error occurred

Note: PSW restart deactivates HMS. However, the PSW restart operation can be performed without resetting HMS by performing a system reset followed by a program restart.

- Save areas **C**, which display the contents of the eight save areas (an X displayed at the right of the data indicates a translation exception during the execution of the LS command).
- A program display area **D**, which displays the HMS program when HMS is active (see Figure 6-1, Section A). A trace display area **E**, which displays the contents of the trace buffer when HMS is inactive (see Figure 6-1, Section B).

Symbols that appear with the trace data are:

'*'	HMS command number that created the trace entry
FC	Facility code is the one- or two-character code of the traced facility
DATA	Traced data follows
%	Indicates data obtained directly from processor storage
F	Flags (X, I, or E) follow:
X	Indicates that a translation exception occurred
I	Indicates that this is an invalid processor storage address
E	Indicates that an error occurred during a data fetch from the processor or director

Note: These symbols appear when a trace area is loaded and when a previously flagged save area is traced.

If less than 32 words are traced, the unfilled trace areas display all 0's, but no facility code. If more than 32 words of data are traced, the trace area is treated as a wraparound storage, with each new word of data entering at the position indicated by >. The > points to the oldest trace entry. (The trace entry immediately above the oldest is the last entry traced before HMS was stopped by the STOP command or ended by the END command.)

- Program save/load area **F**, which indicates whether the program is saved or loaded in response to entering the S- or L-control character and program number (occurs when HMS is not active and not in trace display mode). Up to three compiled HMS programs can be saved in program save areas (1-3). Each program can be loaded using the save area number as the program number (entered after PGM#).
- Current PSW **G**. When HMS is active, a blinking asterisk appears in the position immediately to the left of the PSW on all 3031 Processor Complex frames that display the current PSW.

If the working frame is displayed, HMS is inactive and trace data is not displayed. The input mode frame can be displayed by pressing the I key.

The input mode frame consists of:

- An input mode frame identifier and exit input mode character indicator **H**, which displays the control character for exit to the working frame.
- Facility codes and operators **J**, which are displayed as a reference for entering HMS subcommands (operands).
- Commands **K**, which are displayed as a reference for entering HMS commands (OP codes).
- A program area **L**, which displays the HMS program.
- A command input area **M**, which displays the command as typed and, if applicable, displays a syntax error code when the ENTER key is pressed. The error can then be corrected.
- The current PSW.

OPERATING PROCEDURES

This section explains how to:

- Select the HMS frame.
- Generate a new HMS program.
- Save an HMS program.
- Load a saved HMS program.
- Edit an existing HMS program.
- Activate an HMS program.
- Enable the software initiation of an HMS program.
- Restart processing after a stop.
- End the HMS program.

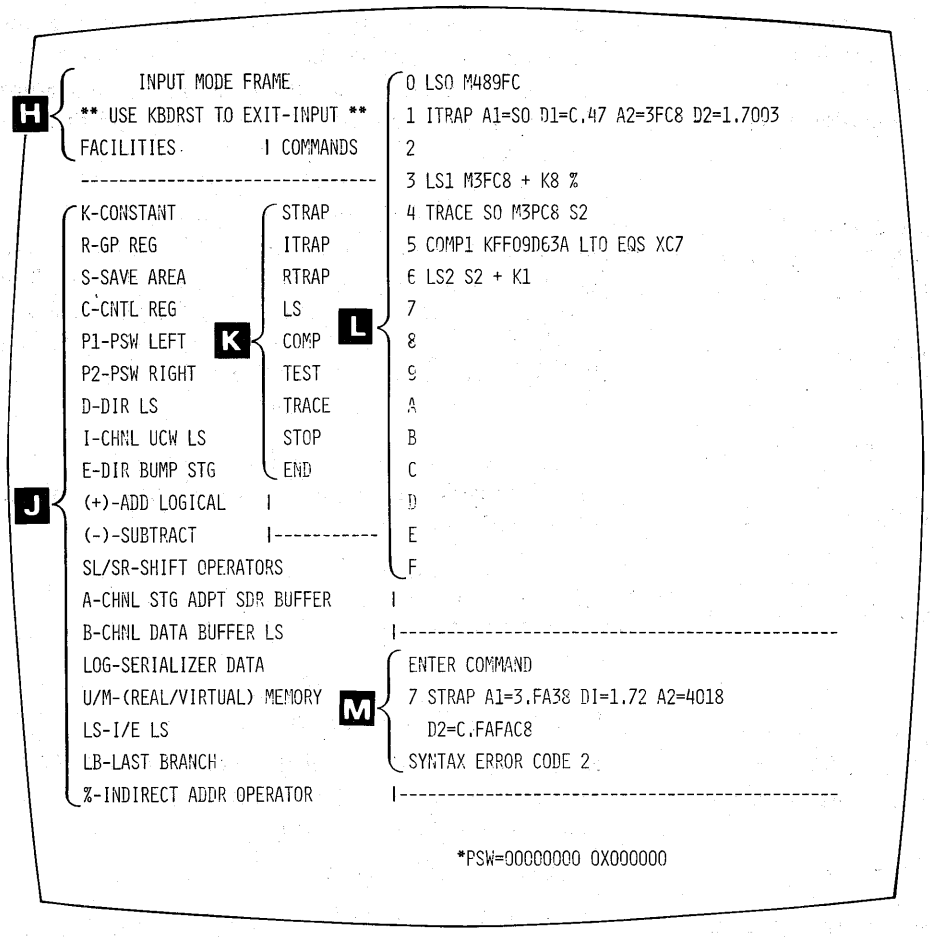


Figure 6-2. HMS Input Mode Frame

- Display the contents of the trace buffer.
- Leave the HMS frame.

To select the HMS frame:

- Press the SEL FRAME key and type 'HM'. The HMS working frame is then displayed.

To generate a new HMS program:

1. With the working frame displayed and HMS inactive and not in trace display mode, select ENTER INPUT MODE by pressing the I key. The input mode frame is displayed and the cursor appears under the first character position in the command entry area after command number 0.
2. Type the first command (see "Command Reference" in this chapter for a complete description of commands). The command appears as typed in the command input area.
3. Press the ENTER key. The command moves to position 0 in the program display area indicating it is compiled, or the command remains and a syntax error code appears. If possible the cursor moves under the incorrect character. The command syntax error codes are:

   0 Invalid command
   1 Invalid facility
   2 Illegal facility value
   3 Two assignments to an entry or invalid pairs (ITRAP,STRAP)
   4 Invalid field length
   5 Invalid combination of save areas and constants in TEST or COMP command
   6 Exhausted allotted assembly area
   7 Invalid facility/operator sequence
   8 Attempt to branch to same instruction or nonexistent instruction in TEST or COMP command
   9 Entered short format command on two lines (TEST or COMP)
   A Attempted to enter a long command over a short command
   B Attempted to enter the second half of a long command

   Correct the error and press the ENTER key to move the command to the program display area.
4. When all commands are entered, press the KEYBD RESET key. This causes a syntax error check of the program for a TEST or COMP command, which may attempt to branch to itself or to the second half of a

long command. If a TEST or COMP command is found that has an invalid branch, that command is moved to the command input area and syntax error code 8 is displayed. Correct the illegal branch condition and press the ENTER key.

To save an HMS program:

1. With the HMS working frame displayed, enter SAVE A PGM by pressing the S key. ENTER PGM# appears as a prompt and the cursor appears under the first character position after PGM#.
2. Type the number of the program to be saved (1, 2, or 3).
3. Press the ENTER key.

If the save is successful, PGM# X SAVED is displayed (where X is the number of the program).

If the save is unsuccessful, PGM SAVE FAILED is displayed.

Note: If the save operation is unsuccessful, try again; ensure that the program number is 1, 2, or 3.

To load a saved HMS program:

1. With the HMS working frame displayed, enter LOAD SAVED PGM by pressing the L key. ENTER PGM# appears as a prompt and the cursor appears under the first character position after PGM#.
2. Type the number of the program to be loaded.
3. Press the ENTER key.

   If the load is successful, PGM# X LOADED is displayed (where X is the program number).

   If the load is unsuccessful, PGM LOAD FAILED is displayed.

   Note: If the load operation is unsuccessful, try again; ensure that the program number is 1, 2, or 3.

To edit an existing HMS program:

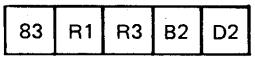
1. Perform the procedure for loading a saved HMS program if the program to be edited is not the current program displayed.
2. Select the input mode frame by pressing the I key. The input mode frame is displayed, and the cursor appears in the command input area under the first character position after the command number (0, followed by the command, is displayed).
3. Move the desired command to the command input area for editing by either of these methods:
   a. Move the cursor under the 0, type the desired command number, and press the ENTER key. (This replaces command 0 with the desired command in the input area.)
   b. Press ENTER to move the next command of the program into the command input area; repeat until the desired command appears. (This action causes a wraparound to command 0 after the last command of the program.)
4. Make the desired changes either by (1) editing the command and pressing the ENTER key to move the command to the program display areas, or (2) deleting the command by pressing the CNCL key and then the ENTER key.

To activate an HMS program:

- With the HMS working frame displayed and not in trace display mode, activate HMS by pressing the A key. The program begins execution with the command at line 0 in the program display area, or, if HMS is being reactivated after the execution of a STOP command, at the command presently indicated by the pointer.

To enable the software initiation of an HMS program:

1. Insert a Diagnose instruction with the following format into the software program:



   Where:
   83 is the DIAGNOSE op code

   R1 is any general register that contains the DIAGNOSE control word when B2, D2 = '0000'X. The control word must have the following format:

   Bits (0-31) ='0400D100'X

   R3 is any general register

   B2, D2 must be 0's if R1 contains the DIAGNOSE control word. The DIAGNOSE control word may be in processor storage instead of R1, but it must be on a doubleword boundary; the address specified by B2, D2 is assumed to be real.
2. Prior to running the software program that issues the Diagnose instruction for activating HMS, display the HMS working frame.
3. Enter an HMS program via the input mode frame by pressing the I key or load a previously saved HMS program by pressing the L key and then the desired program number key (1, 2, or 3).
4. Issue a PROMPT request by pressing the P key. When HMS is ready to be activated by the software Diagnose instruction. HMS S-ACTIVE is displayed instead of HMS INACTIVE.

To restart processing:

1. If the processor is in the manual state because of the execution of the STOP command or subcommand, press the START key. Normal processing continues with HMS inactive.
2. If the processor is in the stopped state because a STOP command or subcommand was executed after an RTRAP command without an intermediate ITRAP or STRAP command, display the Processor Service (PS) frame and press the spacebar. If the director is in the

stopped state because of the same circumstances in the director, display the Director Service (DS) frame and press the spacebar.

To end the HMS program:

- Enter END HMS by pressing the E key.

To display the contents of the trace buffer:

- With the working frame displayed and HMS inactive, enter TRACE DISPLAY by pressing the T key. Data accumulated by the TRACE command is displayed in the program area instead of HMS commands. To end the trace display, press the T key again.

To leave the HMS frame:

- With the working frame displayed, press the SEL FRAME key. Key in the ID characters of the desired frame and press the ENTER key. The desired frame is displayed.

COMMAND REFERENCE

This section contains a detailed description of the HMS command language. Included is how each of the commands and subcommands is used, its syntax, and examples of each command.

Format

HMS commands STRAP, ITRAP, LS, and TRACE can be of the long or short command format, depending on the number and type of subcommands entered with the command. Short commands use one of the 16 possible formats, while long commands always use two of the 16 possible command format numbers. Commands are classified as long if they:

1. Extend onto both of the command entry lines.
2. Are sufficiently complex, with subcommands.

The HMS frame determines and controls the long and short command formats. If commands are entered on both the command entry lines, their subcommands cannot run from one line to the next.

COMP Command

The COMP command is used to compare two save areas, or a save area and a constant, and to branch within the HMS program based on the result of the comparison.

Command	Operands
{COMPs Kconstant / COMPab}	[LTx] [GTx] [EQx] [XCx]

COMPs

The COMPs command compares a save area with a constant: where s is a number from 0 to 7 that specifies the save area to be compared. In this format, the save area is the first element.

Kconstant

The Kconstant command specifies the constant that is compared as the second element: where constant is a hexadecimal number from 1 to 8 that is right-justified in a four-byte field before the comparison is made.

COMPab

The COMPab command compares two save areas: where a is a number from 0 to 7 that specifies the save area to be used as the first element, and b is a number from 0 to 7 that specifies the save area to be used as the second element.

LTx

The LTx operand specifies the action taken if the first element is less than the second: where x is a hexadecimal number from 0 to F, or is the letter N, the letter S, the letter L, or the letter R. A hexadecimal number specifies the number of the HMS command that is branched to. The letter N means that an END command is executed. The letter S means that a STOP command is executed. The letter L indicates that serializer data is logged out. The letter R means that a program restart is forced in the processor, and HMS program execution is continued with the next HMS command. If this operand is not specified and the first facility is less than the second, control is passed to the following HMS command.

GTx

The GTx operand specifies the action taken if the first element is greater than the second: where x is used in the same manner as the x in the LT operand. If this operand is not specified and the first element is greater than the second, control is passed to the following HMS command.

EQx

The EQx operand specifies the action taken if the first facility is equal to the second: where x is used in the same manner as the x in the LT operand. If this operand is not specified and the first element is equal to the second, control is passed to the following HMS command.

XCx

The XCx operand specifies the action taken if either of the save areas on which the comparison is made is flagged as having an exception: where x is used in the same manner as the x in the LT operand. If this operand is not specified and either save area had an exception, control is passed to the following HMS command.

Example 1

Operation: Compare save area 2 to save area 6. If save area 2 is less than or equal to save area 6, branch to the beginning of the HMS program; otherwise continue with the next command.

```
COMP26 LT0 EQ0
```

Example 2

Operation: Compare save area 0 to a constant of X'6AC1'. If they are not equal, branch to command 3. If they are equal, stop the processor. If save area 0 had an exception and the HMS program ends, leave the system running.

```
COMP0 K6AC1 LT3 GT3 EQS XCN
```

END Command

The END command is used to end execution of an HMS program and to deactivate any trap that may be set up in the IEF or the director. If the system was in the system state before execution of the HMS program, it is returned to the system state. If the HMS program is subsequently activated, HMS command execution begins with the first command in the HMS program, and the save and trace areas are initialized to 0.

Command	Operands
END	

ITRAP Command

The ITRAP command is used to set up an instruction address trap in the processor. After an ITRAP command, execution of the HMS program is suspended until the trap conditions are met in the processor. If the processor was in the system state before execution of the HMS program, it is returned to the system state. As many as three address or data conditions can be specified. The trap conditions are not met until all specified conditions are met simultaneously. All specified addresses are treated as virtual addresses if the system is in translation mode. If the system is not in translation mode, all addresses are treated as real addresses. When the trap conditions are met in the processor, the processor is placed in the manual state and HMS program execution resumes with the command following the ITRAP command.

If a STOP command is executed before the next ITRAP, STRAP, or RTRAP command, the system is left in the manual state.

Command	Operands
ITRAP [/xx] [/Sa]	[HP] [AP] [BP] A1 = {hex address | n.displacement | Sb | n.Sc} [D1 = {0.hex data | 1.hex data | C.hex data}] [A2 = {hex address | n.displacement | Sb | n.Sc} D2 = {0.hex data | 1.hex data | C.hex data}] [A3 = {hex address | n.displacement | Sb | n.Sc} D3 = {0.hex data | 1.hex data | C.hex data}]

ITRAP/xx

The ITRAP/xx command indicates that the processor is to be stopped when the trap conditions have been met xx times: where xx is a hexadecimal number from 1 to FF. The default value is xx=1.

ITRAP/Sa

The ITRAP/Sa command indicates that the processor is to be stopped when the trap conditions have been met a number of times equal to the contents of byte 3 of a save area: where a is a number from 0 to 7 that specifies the save area containing the count. If byte 3 of the specified save area contains 00, the processor is not stopped until the trap conditions have been met 256 times.

HP

The HP operand specifies that the logical stop is to be set in the host processor.

AP

The AP operand specifies that the logical stop is to be set in the attached processor.

BP

The BP operand specifies that the logical stop is to be set in both processors.

A1

The A1 operand specifies the instruction address that the trap is to monitor. When the trap conditions are met, the processor is stopped after executing the instruction located at A1.

hex address

The address is specified as an absolute hexadecimal address: where hex address is from one to six digits of hexadecimal data that are right-justified in a three-byte field when the trap is set up.

n.displacement
The address is specified as a base register and displacement: where n is a hexadecimal number from 0 to F that specifies the GR within the processor to be used as the base register. If the contents of the specified base register change after the trap has been set up, the effective address of the trap is changed dynamically to agree with the new contents of the base register. If n=0, no base register is used in calculating the effective address. The displacement is one to six digits of hexadecimal data that are right-justified in a three-byte field when the effective address is calculated.

Sb
The address is specified as an absolute address contained in bytes 1 to 3 of the save area: where b is a number from 0 to 7 that specifies the save area containing the address.

n.Sc
The address is specified as a base register and displacement: where n is a hexadecimal number from 0 to F that specifies the GR within the processor used as the base register. If the contents of the specified base register change after the trap has been set up, the effective address of the trap is changed dynamically to agree with the new contents of the base register. If n=0, no base register is used in calculating the effective address. The displacement is contained in bytes 1 of 3 of a save area: where c is a number from 0 to 7 that specifies the save area containing the displacement.

D1
The D1 operand specifies one byte of data or a one-byte mask that is to be compared with or tested against the op code at A1.

0.hex data
The hexadecimal data is used as a mask that is tested against the addressed data in processor storage. The trap condition is met if all bits selected by the mask are off. If no bits are on in the mask, the condition is always met.

1.hex data
The hexadecimal data is used as a mask that is tested against the addressed data in processor storage. The trap condition is met if all bits selected by the mask are on. If no bits are on in the mask, the condition is always met.

C.hex data
The hexadecimal data is compared with the addressed data in processor storage. The trap condition is met if the two fields are equal.

A2
The A2 operand specifies a second processor storage location that is to be tested or compared before the processor is stopped.

D2
The D2 operand specifies a one- to four-byte data field that is to be compared with or used as a mask to test the data in processor storage addressed by A2. The data is left-justified in a four-byte field, and the test or comparison is done on a byte-by-byte basis from left to right, beginning with the byte of processor storage addressed by A2. Any bytes of D2 that are not specified do not enter into the test or comparison.

A3
The A3 operand specifies a third processor storage location that is to be tested or compared before the processor is stopped.

D3
The D3 operand specifies a one- to four-byte data field that is to be compared with or used as a mask to test the data in processor storage addressed by A3. The data is left-justified in a four-byte field, and the test or comparison is done on a byte-by-byte basis from left to right, beginning with the byte of processor storage addressed by A3. Any bytes of D3 that are not specified do not enter into the test or comparison.

Example 1

Operation: Trap when the instruction located at processor storage address X'37F44' is executed by the host processor.

```
ITRAP HP A1=37F44 (or)
ITRAP HP A1=0.37F44
```

Example 2

Operation: Trap when the instruction located at the effective address formed using base register D and a displacement of X'228' is executed by the attached processor, if that instruction is a branch.

```
ITRAP AP A1=D.228 D1=C.47
```

Example 3

Operation: Trap when the instruction located at the processor storage address X'239EE0' is executed by either processor, if that instruction has an op code of Bx.

```
ITRAP BP A1=239EE0 D1=1.B0 A2=239EE0 D2=0.40
```

Example 4

Operation: Trap after the fourth time that the instruction located at the effective address formed using base register B and a displacement contained in save area 5 is executed, if the last I/O interruption had unit status of channel end.

```
ITRAP/4 A1=B.S5 A2=44 D2=1.08
```

LS Command

Use the LS command to load a save area with system facilities, a constant, another save area, or the result of calculations performed on any of the preceding. If a translation exception occurs while accessing processor storage during the execution of the command, the save area that was to be loaded is flagged as having an exception. If an invalid address is used to access processor storage during execution of this command, an END command is executed (see "END Command"). If an error occurs while HMS is trying to fetch a facility from the processor IEF or the director (for example, trying to load a save area from GR after the execution of an RTRAP command), then HMS is ended with a code 9 (program error). Algebraic expressions are evaluated from left to right, and no parentheses are allowed. Intermediate results obtained during the evaluation of an expression are referred to as the current value. If an overflow on an addition or an underflow on a subtraction occurs during the evaluation of an expression, an END command is executed.

Command	Operands
LSa	facility [{ (unary operator) / (binary operator) facility } ...]

LSa
The LSa command loads a save area with a facility or the result of an algebraic expression: where a is a number from 0 to 7 that specifies the save area to be loaded.

facility
The following facilities may be used in an LS command:

Cb
The Cb operand represents control registers within the processor: where b is a hexadecimal number from 0 to F that specifies the control register to be used.

Kdata
The Kdata operand represents constants: where data is from one to eight hexadecimal digits that are right-justified in a four-byte field.

LB
The LB operand is the last successful branch address: the address of the last branch instruction in the processor in which the branch conditions were met successfully while HMS was active. If the processor was in translation mode, the address is treated as a virtual address. If the processor was not in translation mode, the address is treated as a real address. This data is accurate only if an ITRAP or STRAP command was executed previously in the HMS program.

Maddr
The Maddr is processor storage: where addr is from one to six hexadecimal digits that are right-justified in a three-byte field and are used to address processor storage. If the processor is in translation mode, the address is treated as a virtual address. If the processor is not in translation mode, the address is treated as a real address.

Uaddr
The Uaddr is an address of from one to six hexadecimal digits that are right-justified in a three-byte field and are used to address processor storage. The address is treated as a real address with the processor in basic or extended control mode.

P1
The P1 operand is the left half of the PSW.

P2
The P2 operand is the right half of the PSW.

Rc
The Rc operand is the general register in the processor, where c is a hexadecimal number from 0 to F that specifies the GR to be used.

Sd
The Sd operand is the save area: where d is a number from 0 to 7 that specifies the save area to be used.

Log*aaa(p)
The Log*aaa(p) operand is log (serializer) data of 18 or 32 bits, where * is the flag for the host processor (0), director (1), or attached processor (4), and aaa is the log (serializer) address. If p is specified, two bytes (each with a parity bit that pads out to one byte) are fetched from the serializer and saved. If p is not specified, four bytes (without parity bits) are fetched and saved. The bytes are fetched and saved. The bytes are fetched from the serializer, beginning at address aaa, and are left-justified in the save area.

E*aaaa
The E*aaaa operand is director UCW storage (one word), where * is the director flag (1) and aaaa is the address (0000-1FFF) of UCW storage.

E%*cuaw
The E%*cuaw operand is director UCW storage (one word), where % is the indirect flag, * is the director (1), cua is the channel and unit address (000-5FF), which is the address of the UCW index word that points to the first word of the UCW, and w is the UCW word number (0-3) which is added to the word fetched from the UCW index.

B*cw
The B*cw operand is director and data buffer local storage (one word), where * is the director flag (1), c is the channel number (0-5) for the buffer storage desired, and w is the word number in buffer storage for the channel selected, ranging from 0-7.

I*cw
The I*cw operand is director UCW local storage (one word), where * is the director flag (1), c is the channel number for the local storage desired (0-5), and w is the word (0-7) in the local storage for the channel selected.

LS#w
The LS#w operand is one word in IEF local storage, where # is the local storage sector number, and w is the word number (0-F).

F0
The F0 operand is the prefix register for the host processor.

F4
The F4 operand is the prefix register for the attached processor.

D*w
The D*w operand is director local storage (one word), where * is the director flag (1) and w is the word number, ranging from 0-F.

A*cw
The A*cw operand is the channel storage adapter storage data register (SDR) buffer (one word), where * is the director flag (1), c is the channel storage adapter buffer number, ranging from 0-7 (storage adapters 0-5 are the normal channel SDR buffers, 6 is a spare, and 7 is the director SDR buffer), and w is the SDR word number (0-7).

unary operators
The following unary operators may be used in an LS command:

%
The % is the indirect operator. Use bytes 1 to 3 of the current value to address processor storage. The contents of processor storage become the new current value.

If the processor is in translation mode, the address is treated as a virtual address. If the processor is not in translation mode, the address is treated as a real address.

SLe
The SLe operator shifts the current value left: where e is a hexadecimal number from 1 to 1F that specifies the number of bits that the current value is to be shifted. Zeros are shifted in from the right.

SRf
The SRf operator shifts the current value right: where f is a hexadecimal number from 1 to 1F that specifies the number of bits that the current value is to be shifted. Zeros are shifted in from the left.

binary operators
The following binary operators may be used in an LS command:

+
The symbol + represents a logical addition. The following facility is added to the current value. The sum must not exceed a four-byte field.

–
The symbol – represents a logical subtraction. The following facility is subtracted from the current value. The difference must be positive.

Example 1

Operation: Load save area 3 with the contents of processor storage location X'159BB'.

```
LS3 M159BB
```

Example 2

Operation: Increment the contents of save area 5 by 1.

```
LS5 S5 + K1
```

Example 3

Operation: Load save area 0 with the halfword immediately following the halfword addressed by GR1.

```
LS0 R1 + K2 % SR10
LS0 R1 % SL10 SR10
```

Example 4

Operation: Load save area 4 with the contents of processor storage addressed using GR6 as a base register and a displacement contained in save area 5 through three levels of indexing.

```
LS4 R6 + S5 % % % %
```

Example 5

Operation: Load save area 7 with four bytes of data and parity bits from the IEF system indicators at log address X'129'.

```
LS7 LOG0129
```

Example 6

Operation: Load save area 6 with two bytes of data and parity bits from the director indicators at log address X'021'.

LS6 LOG1021P

RTRAP Command

The RTRAP command stops the IEF or director clocks at the specified RCS address to obtain system indicator information. The processor clocks remain stopped until another RTRAP, STRAP, or ITRAP command is executed, or the HMS program ends. The processor clocks remain stopped indefinitely if a STOP command or subcommand is executed.

Command	Operands
RTRAP [/xx] [/Sa]	[U*] [A=aaaa] [A=bbbb[A=cccc[A=dddd]]] [CH#] [err] [SFT] [SEQ] [STP] [SST]

Optional RTRAP operands are position dependent and must be specified according to the following format:

```
RTRAP[/xx][U*]A=aaaa[A=bbbb[A=cccc[A=dddd]]][CH#][err]
      [/Sa]                                    [SFT]
                                               [SEQ]
                                               [STP]
                                               [SST]
```

RTRAP/xx
The RTRAP/xx command stops the IEF or director clocks and analyzes the system indicators when the RCS control word specified is executed xx times: where xx is a hexadecimal number from 0 to FF. The default value is xx=1.

RTRAP/Sa
The RTRAP/Sa command stops the IEF or director clocks and analyzes system indicators when the RCS control word specified is executed the number of times equal to the contents of byte 3 of save area a: where a is a number from 0 to 7 that specifies the save area containing the count. If byte 3 of the save area contains 00, the processor clocks are stopped when the RCS control word is executed 256 times.

U*
The U* operand specifies the element in which the trap is to be set, where 0 is the processor, 1 is the director, and 4 is the attached processor. If the U* operand is not specified, the default is the host processor (0). If the operand is specified, U* must precede the A=aaaa operand and any other options.

aaaa
The aaaa operand specifies the address of the RCS word that causes a CS stop in the processor if no other options are specified.

bbbb/cccc/dddd
The bbbb/cccc/dddd operands specify an additional one to three RCS addresses that cause a CS stop in the processor.

CH#
The CH# operand specifies that a CS stop is to occur in the director when the RCS word at aaaa is executed and channel # is in control. where # is 0-5. If CH# is not specified, the CS stop occurs whenever the word at aaaa is executed, regardless of the channel in control.

err
The err operand specifies that a simulated machine check is to occur when the RCS word at address aaaa has executed the number of times specified by /xx.

SFT
The SFT operand specifies that a simulated machine check is to occur when the RCS word at address aaaa has executed the number of times specified by /xx, but the processor goes into manual state instead of CS state. When a subsequent ITRAP, STRAP, RTRAP, or END command is executed the processor returns to the machine check microprogram.

SEQ
The SEQ operand specifies that the trap is to take effect in the processor only when the words at all four RCS addresses specified have been executed in the sequence aaaa-cccc, without executing the word at dddd.

STP
The STP operand specifies that a CS stop is to occur in the processor when the trap takes effect in the director.

SST
The SST operand specifies that the clocks are to stop (CS stop) in the host processor when the trap occurs in the attached processor (or vice versa). If the SFT option is also specified then the other processor will also be set to the manual state. Both processors will start if a subsequent RTRAP, ITRAP, STRAP, or END command is executed.

STOP Command

The STOP command ends execution of an HMS program and deactivates any trap that is set up in the IEF or director. The system alarm on the processor is sounded. Subsequently, HMS command execution resumes with the HMS command following the STOP command, and the save and trace areas are unchanged.

Command	Operands
STOP	

STRAP Command

The STRAP command is used to set up a storage trap in the processor. After an STRAP command, execution of the HMS program is suspended until the trap conditions are met in the processor. If the processor was in the system state prior to execution of the HMS program, it is returned to the system state. Up to three address or data conditions can be specified. The trap is not taken until all specified conditions are met simultaneously. All specified addresses are treated as virtual addresses if the processor is in translation mode. If the processor is not in translation mode, all addresses are treated as real addresses. When the trap conditions are met in the processor, the processor is placed in the manual state and HMS program execution resumes with the command following the STRAP command.

If a STOP command is executed before the next ITRAP, STRAP, or RTRAP command, the system is left in the manual state.

Note: Because the IEF and director have separate data paths to and from storage, STRAP commands using addresses in director storage will not execute.

Command	Operands
STRAP [/xx] [/Sa]	[HP] [AP] [SP] A1 = {hex address, n.displacement, Sb, n.Sc} [D1 = {0.hex data, 1.hex data, C.hex data}] [A2 = {hex address, n.displacement, Sb, n.Sc} D2 = {0.hex data, 1.hex data, C.hex data}] [A3 = {hex address, n.displacement, Sb, n.Sc} D3 = {0.hex data, 1.hex data, C.hex data}]

STRAP/xx
The STRAP/xx command indicates that the processor is to be stopped when the trap conditions have been met xx times: where xx is a hexadecimal number from 1 to FF. If this operand is not specified, the default value is xx=1.

STRAP/Sa
The STRAP/Sa command indicates that the processor is to be stopped when the trap conditions have been met a number of times equal to the contents of byte 3 of the save area: where a is a number from 0 to 7 that specifies the save area containing the count. If byte 3 of the specified save area contains 00, the processor is not stopped until the trap conditions have been met 256 times.

HP
The HP operand specifies that the logical stop is to be set in the host processor.

AP
The AP operand specifies that the logical stop is to be set in the attached processor.

BP
The BP operand specifies that the logical stop is to be set in both processors.

A1
The A1 operand specifies the storage address that the trap is to monitor. When the trap conditions are met, the processor is stopped after storing into the location specified by A1.

hex address
The hex address is specified as an absolute hexadecimal address: where hex address is from one to six digits of hexadecimal data that are right-justified in a three-byte field.

n.displacement
The address is specified as a base register and displacement: where n is a hexadecimal number from 0 to F that specifies the GR within the processor to be used as the base register. If the contents of the specified base register change after the trap has been set up, the effective address of the trap is changed dynamically to agree with the new contents of the base register. If n=0, no base register is used in calculating the effective address. The displacement is from one to six digits of hexadecimal data that are right-justified in a three-byte field when the effective address is calculated.

Sb
The address is specified as an absolute address contained in bytes 1 to 3 of a save area: where b is a number from 0 to 7 that specifies the save area containing the address.

n.Sc
The address is specified as a base register and displacement: where n is a hexadecimal number from 0 to F that specifies the GR within the processor to be used as the base register. If the contents of the specified base register change after the trap has been set up, the effective address of the trap is changed dynamically to agree with the new contents of the base register. If n=0, no base register is used in calculating the effective address. The displacement is contained in bytes 1 to 3 of a save area, and c is a number from 0 to 7 that specifies the save area containing the displacement.

D1
The D1 operand specifies one byte of data or a one-byte mask that is compared with or tested against the data stored in the location specified by A1.

0.hex data
The hexadecimal data is used as a mask that is tested against the addressed data in processor storage. The trap condition is met if all bits selected by the mask are off. If no bits are on in the mask, the condition is always met.

1.hex data
The hexadecimal data is used as a mask that is tested against the addressed data in processor storage. The trap condition is met if all bits selected by the mask are on. If no bits are on in the mask, the condition is always met.

C.hex data
The hexadecimal data is compared with the addressed data in processor storage. The trap condition is met if the two fields are equal.

A2
The A2 operand specifies a second processor storage location that is to be tested or compared before the processor is stopped.

D2
The D2 operand specifies a one- to four-byte data field that is compared with or used as a mask to test the data in processor storage addressed by A2. The data is left-justified in a four-byte field, and the test or comparison is done on a byte-by-byte basis from left to right, beginning with the byte of processor storage addressed by A2. Any bytes of D2 that are not specified are not tested or compared.

A3
The A3 operand specifies a third processor storage location that is tested or compared before the processor is stopped.

D3
The D3 operand specifies a one- to four-byte data field that is to be compared with or used as a mask to test the data in processor storage addressed by A3. The data is left-justified in a four-byte field, and the test or comparison is done on a byte-by-byte basis from left to right, beginning with the byte of processor storage addressed by A3. Any bytes of D3 that are not specified are not tested or compared.

Example 1

Operation: Trap when processor storage X'567FF' is altered by the host processor.

```
STRAP HP A1=567FF (or)
STRAP HP A1=0.567FF
```

Example 2

Operation: Trap when bit 3 of the processor storage byte addressed using base register B and a displacement of X'0A3' is turned on by the attached processor.

```
STRAP AP A1=B.A3 D1=1.10
```

Example 3

Operation: Trap when data of X'1A' is stored at the processor storage location addressed by save area 4, if the data located at processor storage addressed by base register E and a displacement of X'04' is equal to C'IEHPROGM'.

```
STRAP A1=S4 D1=C.1A A2=E.4 D2=C9C5C8D7
A3=E.8 D3=C.D9D6C7D4
```

Example 4

Operation: Trap after the contents of processor storage, located at the effective address formed using base register B and a displacement contained in save area 7, are altered by either processor a number of times equal to the contents of byte 3 in save area 1.

```
STRAP/S1 BP A1=B.S7
```

TEST Command

The TEST command is used to perform a test under mask operation on a save area and to branch within the HMS program based on the result of the test under mask. The mask may be a constant or it may be the contents of a save area.

Command	Operands
{TESTs Kconstant / TESTab}	[ONx] [OFFx] [MXx] [XCx]

TESTs
This command tests a save area with a constant: where s is a number from 0 to 7 that specifies the save area to be tested.

Kconstant
The Kconstant command specifies the constant to be used as a mask: where constant is specified as a hexadecimal number from one to eight digits that is right-justified in a four-byte field before the test is made.

TESTab
This command tests a save area with the contents of a save area: where a is a number from 0 to 7 that specifies the save area to be tested; where b is a number from 0 to 7 that specifies the save area containing the mask.

ONx
The ONx operand specifies the action taken if the bits selected by the mask are on: where x is a hexadecimal number from 0 to F, the letter N, the letter S, the letter L, or the letter R. A hexadecimal number specifies the number of the HMS command to be branched to. The letter N means that an END command is to be executed (see "END Command"). The letter S means that a STOP command is to be executed (see "STOP Command"). The letter L indicates that serializer data is logged out. The letter R means that a program restart is to be forced in the processor, and HMS program execution is to be continued with the next HMS command. If this operand is not specified and the selected bits are on, control is passed to the following HMS command.

OFFx
The OFFx operand specifies the action to be taken if the bits selected by the mask are off: where x is used in the same manner as the x in the ON operand. If this operand is not specified and the selected bits are off, control is passed to the following HMS command.

MXx
The MXx operand specifies the action to be taken if the bits selected by the mask are mixed 1's and 0's: where x is used in the same manner as the x in the ON operand. If this operand is not specified and the selected bits are mixed, control is passed to the following HMS command.

XCx
The XCx operand specifies the action to be taken if either of the save areas referred to by the TEST command is flagged as having an exception. The x is used in the same manner as the x in the ON operand. If this operand is not specified and either save area has an exception, control is passed to the following HMS command.

Example 1

Operation: Test the contents of save area 6 using the contents of save area 3 as a mask. If either save area is flagged as having an exception, end the HMS program. If the selected bits are on, perform a program restart. If the selected bits are off or mixed, continue with the next HMS command.

```
TEST63 ONR XCN
```

Example 2

Operation: Test bits 0, 2, 3, and 7 of save area 5. If all bits are on, branch to command 9. If all bits are off, branch to the beginning of the HMS program; otherwise, continue with the next HMS command.

```
TEST5 KB1000000 ON9 OFF0
```

TRACE Command

The TRACE command is used to store system facilities, save areas, or constants in a trace buffer. The trace buffer is 32 words long and is treated as a pushup storage with the most recent entry at the bottom. If a translation exception occurs while accessing processor storage during execution of this command or if a save area that is flagged as having an exception is traced, the trace entry is flagged as having an exception. If an invalid address is used to access processor storage during the execution of this command, the trace entry is flagged as having an IVA.

Command	Operands
TRACE	facility [{% / facility}]...

TRACE

This command traces one or more elements.

facility

The following elements may be used in a TRACE command:

Cb

The Cb element represents control registers within the processor, where b is a hexadecimal number from 0 to F that specifies the control register to be used.

Kdata

The Kdata element represents constants: where data is a hexadecimal number from one to eight digits that is right-justified in a four-byte field.

LB

The LB element is the last successful branch address: the address of the last branch instruction in the processor in which the branch conditions were met successfully while the HMS was active. If the processor was in translation mode, the address is treated as a virtual address. If the processor was not in translation mode, the address is treated as a real address. This data is accurate only if an ITRAP or STRAP command was executed previously in the HMS program.

Maddr

The Maddr element is processor storage: where addr is a hexadecimal number from one to six digits that is right-justified in a three-byte field and is used to address processor storage. If the processor is in translation mode, the address is treated as a virtual address.

Uaddr

The Uaddr is an address of from one to six hexadecimal digits that are right-justified in a three-byte field and are used to address processor storage. The address is treated as a real address with the processor in basic or extended control mode.

P1

The P1 element is the left half of the PSW.

P2

The P2 element is the right half of the PSW.

Rc

The Rc element is the general register in the processor: where c is a hexadecimal number from 0 to F that specifies the GR to be used.

Sd

The Sd element is the save area: where d is a number from 0 to 7 that specifies the save area to be used.

Log*aaa(p)

The Log*aaa(p) operand is log (serializer) data of 18 or 32 bits where * is the flag for the processor (0), director (1), or attached processor (4), and aaa is the log (serializer) address. If p is specified, two bytes (each with a parity bit that pads out to one byte) are fetched from the serializer and saved. If p is not specified, four bytes (without parity bits) are fetched and saved. The bytes are fetched from the serializer, beginning at address aaa, and are left-justified in the trace area.

E*aaaa

The E*aaaa operand is director UCW storage (one word), where * is the director flag (1) and aaaa is the address (0000-1FFF) of UCW storage.

E%*cuaw

The E%*cuaw operand is director UCW storage (one word), where % is the indirect flag, * is the director (1), cua is the channel and unit address (000-5FF), which is the address of the UCW index word that points to the first word of the UCW, and w is the UCW word number (0-3) which is added to the word fetched from the UCW index.

B*cw

The B*cw operand is director and data buffer local storage (one word), where * is the director flag (1), c is the channel number (0-5) for the buffer storage desired, and w is the word number in buffer storage for the channel selected, ranging from 0-7.

I*cw

The I*cw operand is director UCW local storage (one word), where * is the director flag (1), c is the channel number for the local storage desired (0-5), and w is the word (0-7) in the local storage for the channel selected.

LS#w

The LS#w operand is one word in IEF local storage, where # is the local storage sector number, and w is the word number (0-F).

F0

The F0 operand is the prefix register for the host processor.

F4

The F4 operand is the prefix register for the attached processor.

D*w

The D*w operand is director local storage (one word), where * is the director flag (1) and w is the word number, ranging from 0-F.

A*cw

The A*cw operand is the channel storage adapter storage data register (SDR) buffer (one word), where * is the director flag (1), c is the channel storage adapter buffer number, ranging from 0-7 (storage adapters 0-5 are the normal channel SDR buffers, 6 is a spare, and 7 is the director SDR buffer), and w is the SDR word number (0-7).

%

The % element is the indirect operator. Use bytes 1 to 3 of the previous element to address processor storage, and enter the addressed data in the trace buffer. The previous element is not entered in the trace buffer. If the processor is in translation mode, the address is treated as a virtual address. If the system is not in translation mode, the address is treated as a real address.

Example 1

Operation: Trace the right half of the PSW and the last successful branch address.

TRACE P2 LB

Example 2

Operation: Trace the channel status word (CSW) and the processor storage location address by GR7 through two levels of indexing.

TRACE M40 R7 % % %

Example 3

Operation: Trace the contents of control register E and the processor storage location addressed by save area 4.

TRACE CE S4 %

Appendix A. Glossary and Abbreviations

If the term you are seeking does not appear in this glossary, refer to *IBM Data Processing Glossary*, GC20-1699.

absolute address: An address that identifies a storage location or device without the use of any intermediate reference.

address: An identification of a storage location or an I/O device.

address compare: A technique to stop the processor at a specific address.

address modification: The process of changing the address part of a machine instruction via coded instructions.

address translation: The process of changing the address of an item of data or an instruction from its virtual storage address to its real storage address.

alphameric: Pertaining to a character set that contains letters, digits, and special characters.

attribute character: A character that describes the characteristics of the data field that follows.

basic control (BC) mode: A mode in which the features of a System/360 and additional System/370 features (such as new machine instructions) are operational on a System/370.

BC mode: See *basic control mode.*

BCF: Buffer control function.

break-in: A change in control at a specified point in a channel or director microprogram.

C-bit: Change bit.

CAW: Channel address word.

CCW: Channel command word.

CE: Customer engineer.

CIDA: Channel indirect data addressing.

CNCL: Cancel (key).

COMP: The COMPARE command of HMS.

control registers: A set of registers used for operating system control of relocation, priority interruption, program event recording, error recovery, and masking operations.

CPU: Central processing unit.

CRT: Cathode-ray tube.

CSW: Channel status word.

CTCA: Channel-to-channel adapter.

cursor: A short line (underscore) displayed on the CRT display to indicate where the next character entered will be positioned.

C1: Configuration (display frame).

DAT: See *dynamic address translation*.

DIDOCS: Device-independent display operator console support.

director: In any channel group, the element that controls the activity of the group's channels.

DOS/VS: Disk operating system/virtual storage.

dynamic address translation (DAT): (1) The change of a virtual storage address to a real storage address during execution of an instruction. (2) A hardware feature that performs the translation.

E-function: Execution function.

EBCDIC: Extended binary-coded decimal interchange code.

EC mode: See *extended control mode.*

ECC: Error checking and correction.

END: The END command of HMS.

extended control (EC) mode: A mode in which all the features of a System/370, including dynamic address translation, are operational.

FIFO: First in, first out.

hardstop: Faulty machine condition in which the processor ceased operation.

HMS: Hierarchical Monitoring System.

ID: Identifier.

IEF: Instruction execution function.

IMPL: Initial microprogram load.

IN: Index (display frame).

initialize: To set counters, switches, addresses, or storage contents to 0 or to other starting values at the beginning of, or at prescribed points in, a computer program.

IPC: Initial power controller.

ITRAP: The ITRAP command of HMS.

K byte: 1,024 bytes of storage capacity.

LCL: Limited channel logout.

logout: Recorded information about machine conditions that relate to an error.

LS: The load save area command of HMS.

M byte: 1,048,576 bytes of storage capacity.

MC: Mode control (display frame).

MFT: Multiprogramming with a fixed number of tasks.

MVS: Multiple virtual storage.

no-op: A no-operation instruction.

offline: Pertaining to resources with which the processor has no direct communications or control.

OLTEP: Online test executive program.

OLTSEP: Online test stand-alone executive program.

online: Pertaining to resources with which the processor has direct communications or control.

OS/VS: Operating system/virtual storage.

page: (1) A fixed-length block of instructions, data, or both that can be transferred between real storage and external page storage. (2) To transfer instructions, data, or both between real storage and external page storage.

page table: A table that indicates whether a page is in real storage and correlates virtual storage addresses with real storage addresses.

PC: Power control (display frame).

PER: Program event recording.

PFK or PF key: Program function key.

port: An access point for data entry or exit.

PR: Program (display frame).

processor storage: General-purpose storage that is part of a processor. Synonymous with real storage.

PSCF: Processor storage control function.

PSW: Program status word.

R-bit: Reference bit.

RAS: Reliability, availability, and serviceability.

RCS: Reloadable control storage.

real address: The address of a location in real storage.

RTRAP: The RTRAP command of HMS.

SCF: Storage control function.

SCP: System control program.

SDE: Storage distribution element.

SDR: Storage data register.

segment: A continuous area of virtual storage, which is allocated to a job or system task.

SEL FRAME: Select frame (key).

STOP: The STOP command of HMS.

STRAP: The STRAP command of HMS.

SVS: Single virtual storage.

TEST: The TEST command of HMS.

TIC: Transfer in channel.

TLB: Translation lookaside buffer.

TOD clock: Time-of-day clock.

TP: Teleprocessing. This term is synonymous with the term *data communications.*

TRACE: The TRACE command of HMS.

UCW: Unit control word.

virtual address: An address that refers to virtual storage and that must be translated into a real storage address when it is to be used.

VM/370: IBM Virtual Machine Facility/370.

VS: Virtual storage.

DEVIATIONS FROM IBM System/370 Principles of Operation

Early 'Status In' Residual Count

'Status in' received after initial status, but before any data has been transferred, results in an early 'status in' residual count in the subsequent CSW that is one higher than the count in the last-used CCW.

Program Event Recording (PER) General-Register Alteration for the Move Long (MVCL) Instruction

The MVCL instruction in the condition-code-3 case does not indicate the general-register alteration event for the odd registers (that is, R1 +1 and R2 +1). Although the registers are not changed in this case, the recognition of a PER event is required.

No-OP/Transfer-in-Channel Loops in Channel Programs

The use of no-op/transfer-in-channel loops in channel programs results in degradation of director throughput and may cause time-out errors.

DEVIATIONS FROM IBM System/360 and System/370 I/O Interface Channel to Control Unit Original Equipment Manufacturers' Information

Channel-to-Channel Adapter (CTCA) Internal Cable Resistance

The 'select out' and 'select in' internal cable resistance of the CTCA may exceed by 0.1 ohm the value specified (1.5 ohms) in the *IBM System/360 and System/370 I/O Interface Channel to Control Unit Original Equipment Manufacturers' Information,* GA22-6974.

IBM 3031 Processor Complex and 3031 Attached Processor Complex Functional Characteristics (File No. S370-01) Printed in U.S.A. GA22-7066-2

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

IBM 3031 Processor Complex and 3031 Attached Processor Complex Functional Characteristics

Order No. GA22-7066-2

READER'S COMMENT FORM

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name and mailing address:

_____

_____

_____

Note: Staples can cause problems with automated mail sorting equipment. Please use pressure-sensitive or other gummed tape to seal this form.

What is your occupation? _____

Number of latest Newsletter associated with this publication: _____

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the front cover or title page.)

Reader's Comment Form

Fold and tape | Please Do Not Staple | Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department B98
P.O. Box 390
Poughkeepsie, New York 12602

Fold and tape | Please Do Not Staple | Fold and tape

Cut or Fold Along Line

IBM 3031 Processor Complex and 3031 Attached Processor Complex Functional Characteristics (File No. S370-01) Printed in U.S.A. GA22-7066-2

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