IBM

Series/1

GA34-0230-1

File No. S1-16

IBM Series/1
4956 Processor Models C and C10
Description

IBM

Series/1

GA34-0230-1

File No. S1-16

IBM Series/1
4956 Processor Models C and C10
Description

Second Edition (January 1986)

Use this publication for the purpose stated in the preface.

Changes are periodically made to the information herein; any such changes will be reported in subsequent revisions or Technical Newsletters.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or service in your country.

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

This publication could contain technical inaccuracies or typographical errors. A form for readers' comments is provided at the back of this publication. If the form has been removed, address your comments to IBM Corporation, Information Development, Department 28E, Internal Zip 1803, P. O. Box 1328, Boca Raton, Florida 33429-1328. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Preface

This publication describes the unique functional characteristics of the IBM Series/1 4956 Processor Models C and C10, and the processor optional features. Refer to the *IBM Series/1 Principles of Operation*, GA34-0152, for the common Series/1 processor functional characteristics and instructions. This publication also provides reference information about the following:

- Processor and processor feature configurations
- Processor and processor feature operations.

The reader should understand data processing terminology and be familiar with binary and hexadecimal numbering systems.

The publication is intended primarily as a reference manual for experienced programmers who require machine code information to plan, correct, and modify programs written in the assembler language.

Chapter 1. Introduction contains a general description of the processor, processor storage, and processor features.

Chapter 2. Main Storage Addressing Using the Relocation Translator describes the relocation translator, including:

- Relocation addressing
- Storage protection mechanism
- Error-recovery considerations

Chapter 3. Console describes the keys, switches, and indicators for the basic console and the optional programmer console. Typical manual operations, such as storing into and displaying main storage, are presented.

Chapter 4. Diagnose (DIAG) Instruction describes the Diagnose instruction.

Chapter 5. Diskette Data Format describes how data and control information are formatted on the diskette surfaces.

Chapter 6. Input/Output Operations describes the I/O commands and control words that are used to operate the diskette unit. Condition codes and status information relative to the I/O operation are also explained.

Appendix A. Instruction Execution Time contains information for determining instruction execution time and instruction throughput.

Appendix B. Software Notes lists some software notes for the processor and the diskette drive unit.

Appendix C. Error Log describes the error log and explains its use as an aid in isolating errors.

Prerequisite Publication

For a description of the common processor architecture and a detailed description of the instruction set for the IBM Series/1 processors, refer to the *IBM Series/1 Principles of Operation*, GA34-0152.

Contents

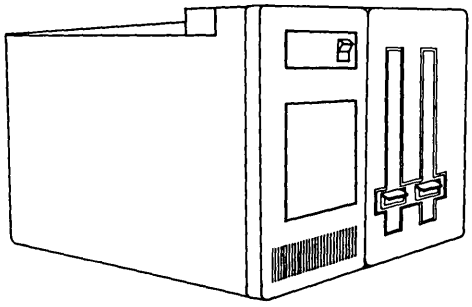
Chapter 1. Introduction

The IBM Series/1 4956 Processor Models C and C10 are compact general-purpose computers. The models are the same, except for base storage. Model C has 256 or 512 kilobytes of base storage; model C10 has 1024 kilobytes of base storage.

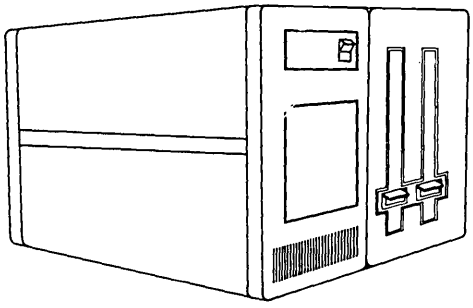
The processor is microcode-controlled for both automatic functions and program instruction functions. It includes one self-contained diskette drive as standard, and a second self-contained diskette drive may be added as an option.

The processor is a full-rack unit that can be mounted in a standard 483-millimeter (19-inch) rack. An optional stand-alone enclosure feature is also available, making this computer suitable for placement outside a rack (see Figure 1-1).

The processor contains four card sockets for data channel features and a channel repower card. One of the four card sockets must be used for the internal diskette drive attachment card, and two card sockets may be used for cards with additional processor storage.



Rack-mount enclosure (front view)



Stand-alone enclosure (front view)

Figure 1-1. IBM Series/1 4956 Processor Models C and C10

The processor has the following characteristics:

- Four priority interrupt levels, with independent registers and status indicators for each level.
- Automatic and program-controlled level switching.
- An instruction set that includes stacking and linking facilities, multiply and divide, variable-field-length byte operations, and a variety of arithmetic and branching instructions.
- Supervisor and problem states.
- A basic console that is a standard feature; a programmer console that is an optional feature.
- A storage address relocation translator that allows addressing of main storage larger than 64 kilobytes.
- An error correction code (ECC) that is implemented on the storage card to provide the capability for single-bit error correction and double-bit error detection.
- An error log, which provides a history of errors that have occurred since power-on.
- A clock/comparator. Four instructions are provided to set or copy the clock and comparator.
- Channel capability:
  - Asynchronous, multidropped channel
  - 256 input/output (I/O) devices can be addressed
  - Direct program control and cycle-steal operations
  - Maximum burst output data rate of 1.11 million 16-bit words per second (see Note)
  - Maximum burst input data rate of 1.54 million 16-bit words per second (see Note)

Note: The burst output and burst input data rates are reduced from the values shown by data channel attachment characteristics, channel loading during instruction processing, channel repowering, and processor storage refresh requirements.

Card Plugging Assignments

The processor unit contains power and space for additional features. The IBM 4959 Input/Output Expansion Unit and the IBM 4965 Diskette Drive and I/O Expansion Unit are available for adding additional features, if desired.

Figure 1-2 shows the card plugging assignments for the processor.

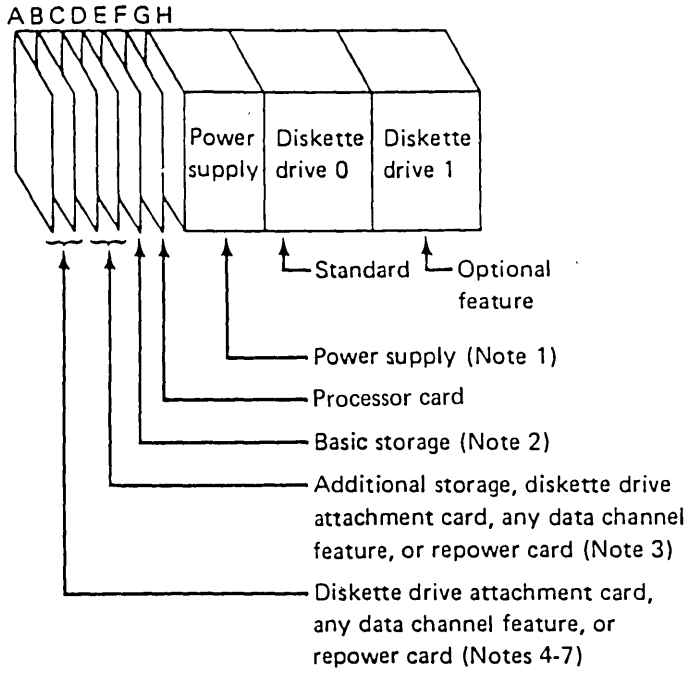


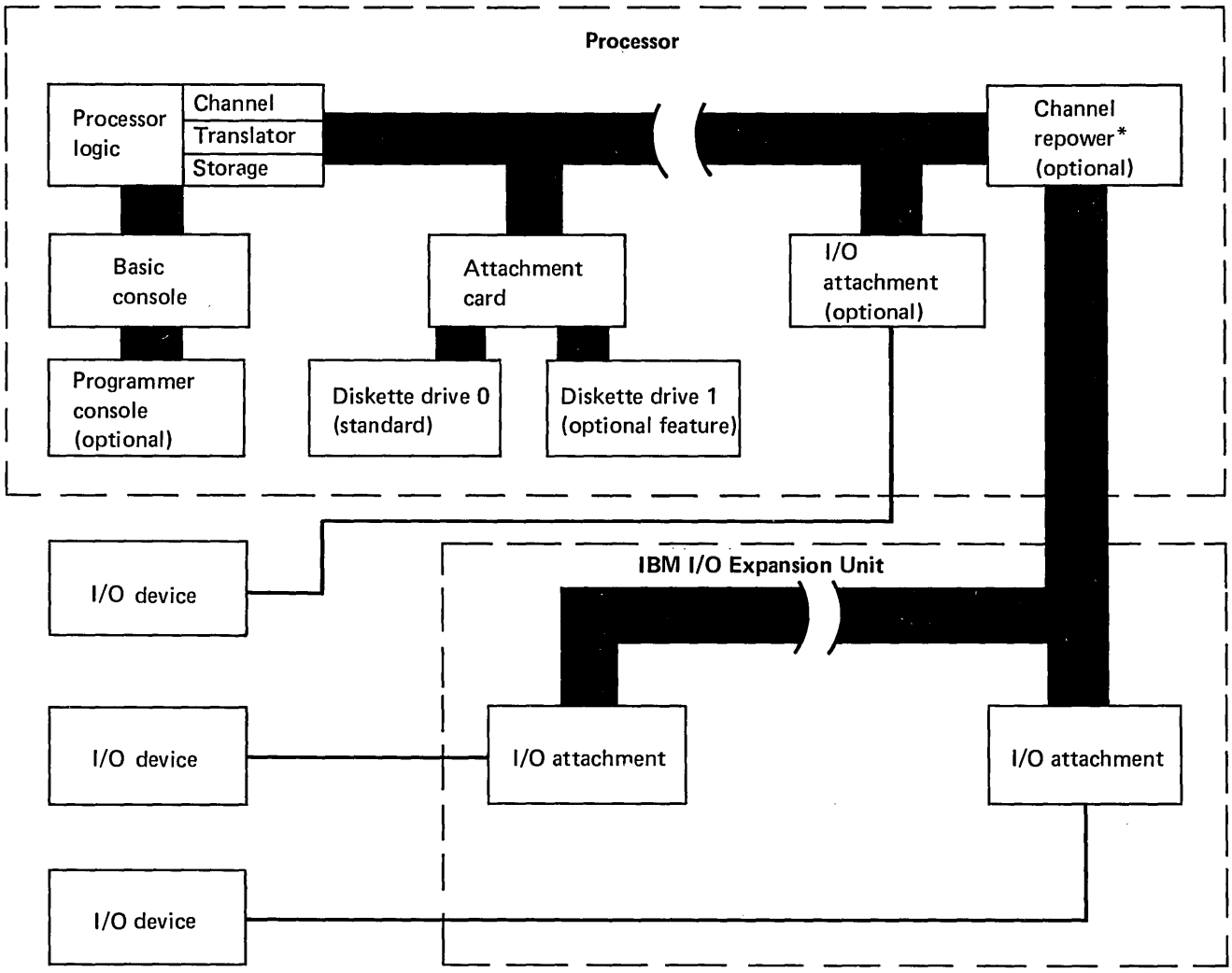
Figure 1-2. Card Plugging Assignments

Notes:

1. The pluggable high-frequency power supply plugs into card socket H.
2. The processor contains 256 KB (kilobytes), 512 KB, or 1024 KB of basic storage in socket E.
3. The processor supports three different size storage cards: 256 KB, 512 KB, and 1024 KB. Sockets C and D are available for additional storage. Any combination of storage cards may be used to obtain the desired system storage size, up to the maximum of 1024 KB. Sockets not used for additional storage can be used for the diskette drive attachment card, any channel feature, or a repower card.
4. A single diskette drive attachment card is used with either one or two diskette drives.
5. The diskette drive attachment card can be plugged into any I/O card socket position to establish the desired priority.
6. If a repower card is used, it must be plugged to the left of and adjacent to the leftmost I/O card installed.
7. A maximum of five serially connected channel repower features can be driven by each processor. Any processor system that includes an IBM I/O expansion unit with the two-channel switch feature is limited to three channel repower features.

Processor Description

The basic processor includes the processor card, a 256K-byte, 512K-byte, or 1024K-byte storage card, a basic console, one diskette drive (drive 0), and a diskette drive attachment card. Figure 1-3 shows a block diagram of the processor and an IBM 4959 I/O Expansion Unit.



*Required with an expansion unit.

Figure 1-3. Block Diagram of the Processor and an IBM I/O Expansion Unit

Four priority interrupt levels (0–3) are implemented in the processor. Each level has an independent set of machine registers. Level switching can occur in two ways: (1) by program control, or (2) automatically upon acceptance of an I/O interrupt request. The interrupt mechanism provides 256 unique entry points for I/O devices.

> **Note:** A Prepare command to levels 4–15 is executed so that condition code reporting occurs; however, the Prepare command is not executed at the addressed device and effectively results in a no-operation.

The processor instruction set contains a variety of instruction types. These include: shift, register to register, register immediate, register to (or from) storage, bit manipulation, multiple register to storage, variable byte field, and storage to storage. Supervisor and problem states are implemented, with appropriate privileged instructions for the supervisor.

The basic console is intended for dedicated systems that are used in a primarily unattended environment. Only minimal controls are provided. A programmer console, which can be added as a feature, provides a variety of indicators and controls for operator-oriented systems.

The processor supports three different size storage cards: 256 KB, 512 KB, and 1024 KB. The processor has a maximum of 1024 KB of storage. Up to two additional storage positions are available. Any combination of storage cards may be used to obtain the desired system storage size, up to the maximum of 1024 KB. (The relocation translator must be enabled to select addresses above 64K bytes.)

An error correction code (ECC) is implemented on the storage card. ECC gives the storage card the capability of single-bit error correction and double-bit error detection. ECC provides the user a higher system availability.

> **Note:** When a double-bit error in storage is detected during a processor read, a machine check interrupt occurs with PSW bit 8 set to 1 (storage parity error).

There is no storage-protect feature in the 4956 processor. However, there is a read-only protect capability provided by the address translator when it is enabled.

> **Note:** Execution of the Set Storage Key (SESK) and Copy Storage Key (CPSK) instructions results in a no-operation.

I/O devices are attached to the processor through the processor data channel. The data channel directs the flow of information between the I/O devices, the processor, and main storage. The data channel supports a maximum of 256 addressable devices.

The data channel supports:

- **Direct program control operations.** Each Operate I/O instruction transfers a byte or word of data between main storage and the device. The operation may or may not terminate in an interrupt.

- **Cycle-steal operations.** Each Operate I/O instruction initiates multiple data transfers between main storage and the device. The maximum cycle-steal transfer per device control block (DCB) is 65,535 bytes. Cycle-steal operations are overlapped with processor operations and always terminate in an interrupt.

- **Interrupt servicing.** Interrupt requests from the devices, along with cycle-steal requests, are presented and polled concurrently with data transfers.

Input/Output Units, I/O Features, and Processor Options

The floating-point feature is one of the available options. If the floating-point feature is installed, refer to Appendix A for instruction execution times. For a detailed description of this feature, refer to the *IBM Series/1 Principles of Operation*, GA34-0152.

A variety of I/O units and features, plus several processor options, are available for use with the Series/1 processor. For a list and description of system units and features, refer to the *IBM Series/1 System Selection Guide*, GA34-0143, and the *IBM Series/1 Digest*, G360-0061. Detailed information about I/O units and features can be found in separate publications. The order numbers for these publications are contained in the *IBM Series/1 Graphic Bibliography*, GA34-0055.

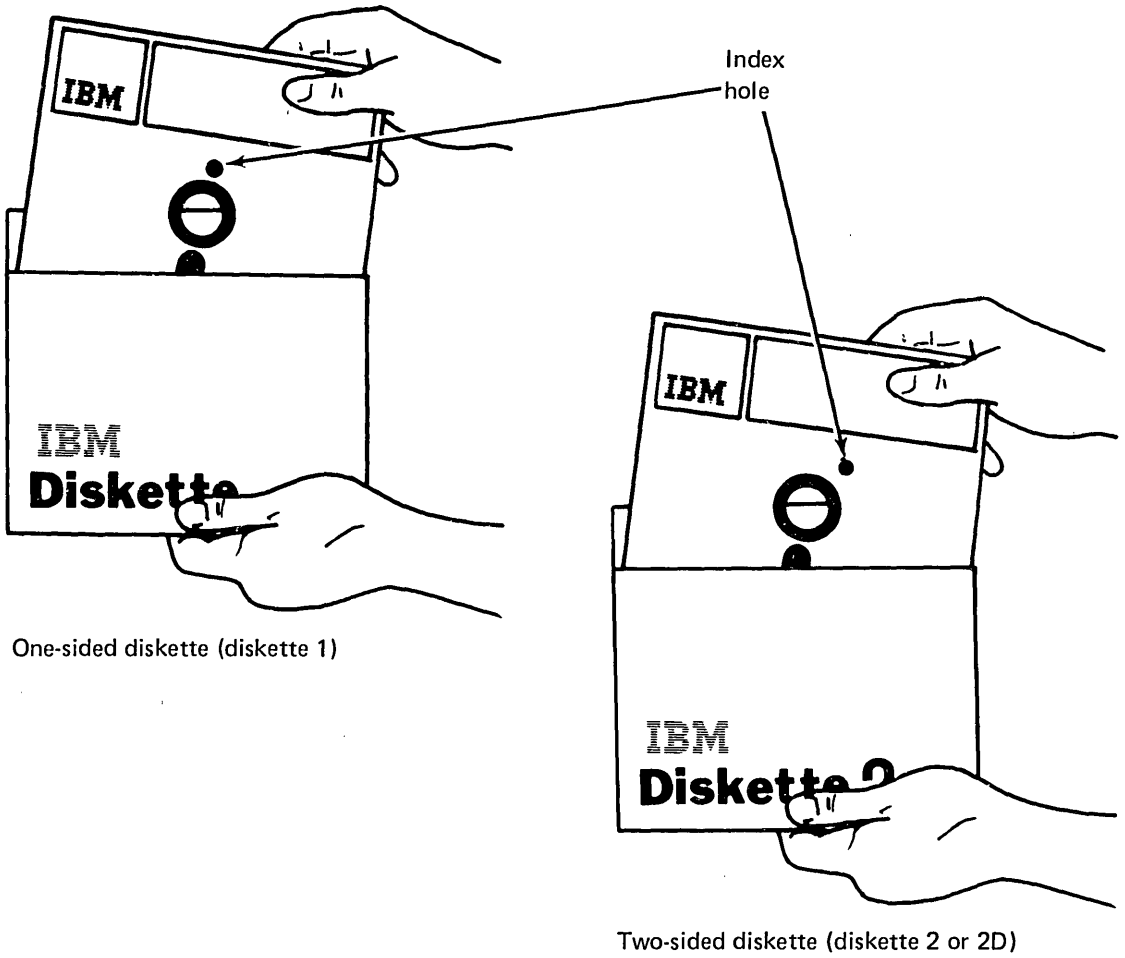
Diskette

The diskette storage medium is a flexible magnetic disk that is permanently enclosed in a jacket. The disk and jacket together are referred to as a diskette (see Figure 1-4). Small slots in the jacket allow the read/write heads to contact the surfaces as the disk rotates within the jacket (see Figure 1-5).

Three types of diskettes are used in the diskette drive:

- Diskette 1. This diskette is intended for the recording of single-density (FM) data on one diskette surface (side 0).
- Diskette 2. This diskette is intended for the recording of single-density (FM) data on both diskette surfaces (sides 0 and 1).
- Diskette 2D. This diskette is intended for the recording of double-density (MFM) data on both diskette surfaces (sides 0 and 1).

  Note: Index holes on the one-sided and two-sided diskettes occupy different locations on the diskette.



One-sided diskette (diskette 1)

Two-sided diskette (diskette 2 or 2D)

Figure 1-4. Diskette Types

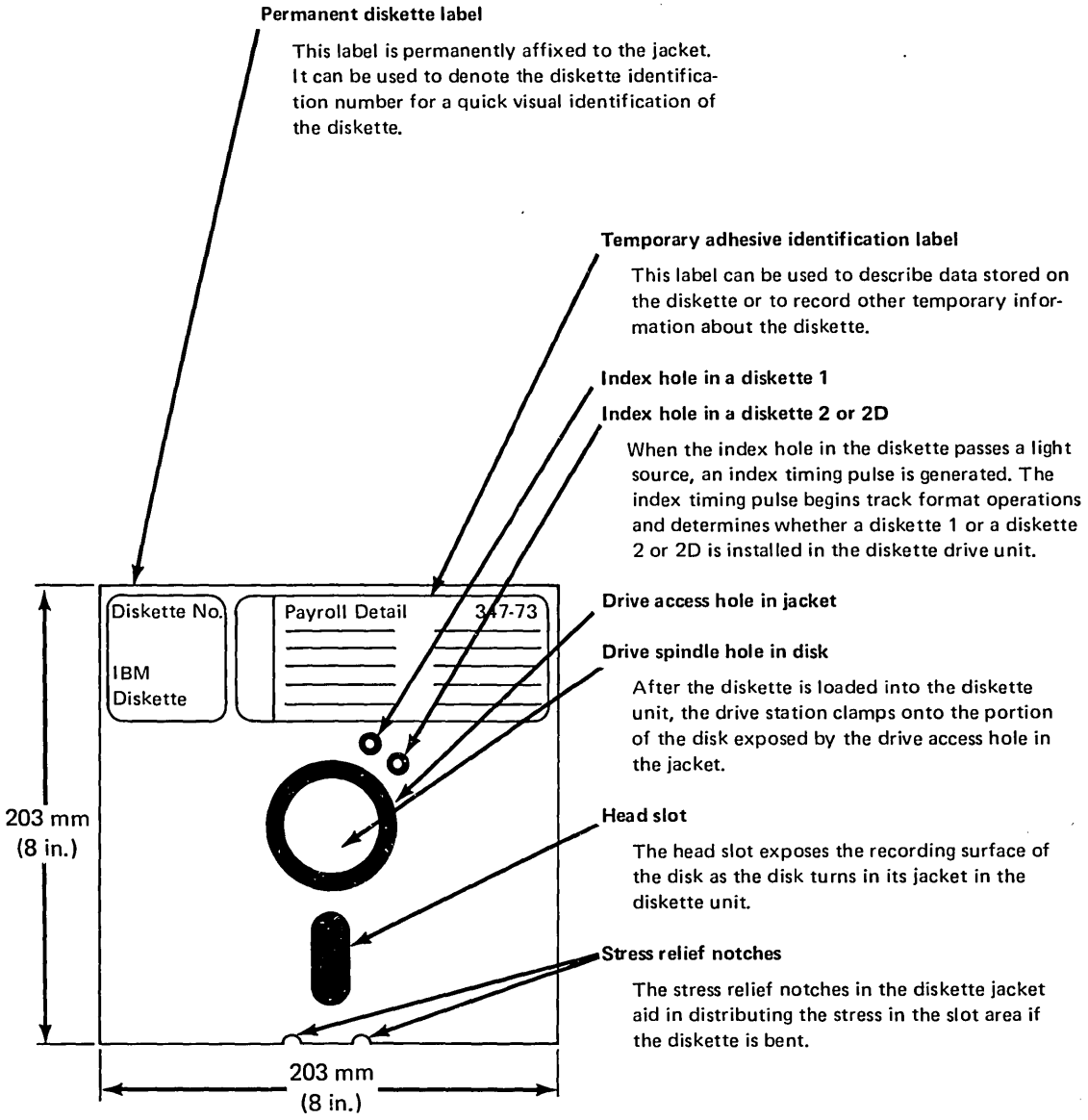
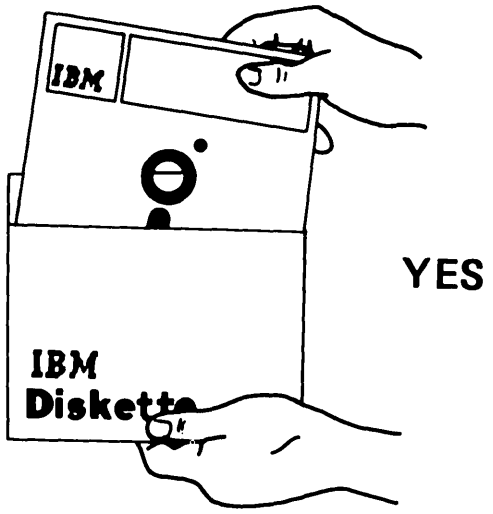


Figure 1-5. Diskette Characteristics

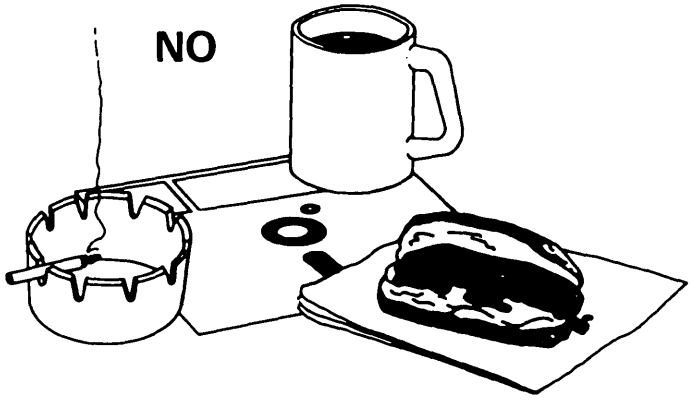
Diskette Protection

To prevent damage to diskettes or the loss of data that has been recorded on them, diskettes should be handled with care at all times. The following precautions should be observed:

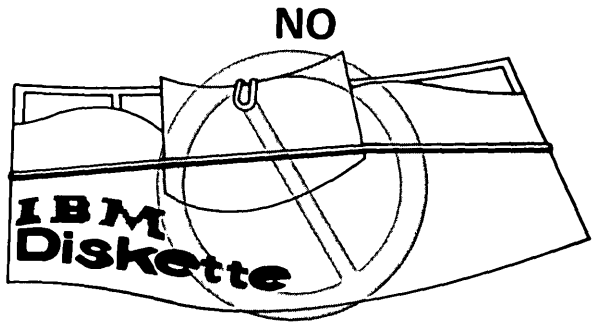
- Return diskettes to their envelopes whenever they are removed from the diskette drive.



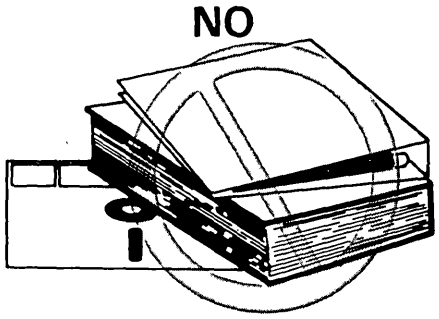
- Do not lay diskettes near food, drink, or ashtrays.



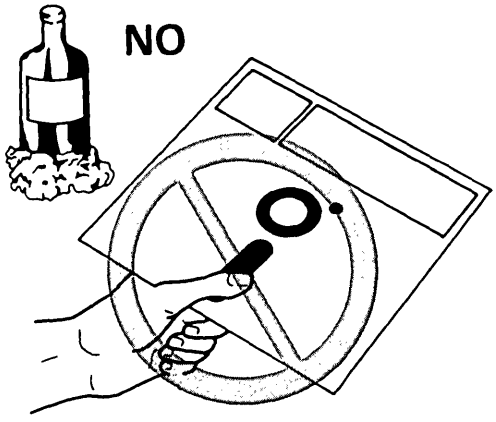
- Do not use clips or rubber bands on diskettes.



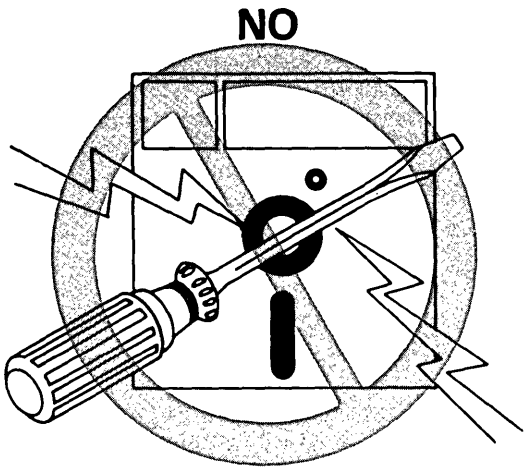
- Do not place heavy objects on diskettes.



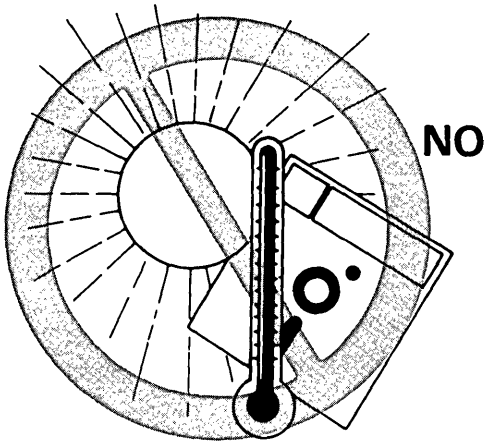
- Do not touch or attempt to clean diskette surfaces. Contaminated diskettes must be discarded.



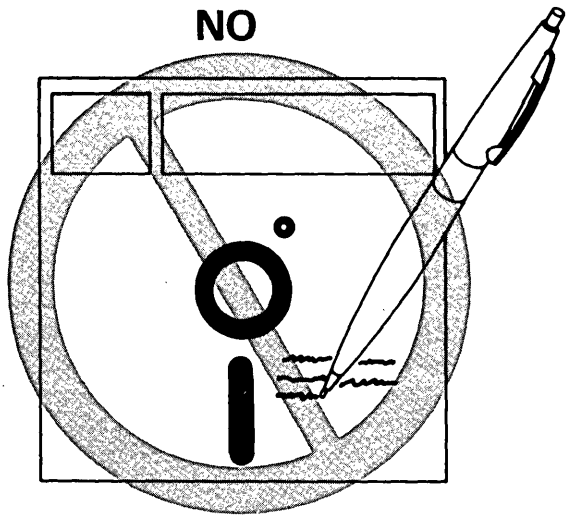
- Do not place diskettes near materials that might be magnetized. (Data can be lost from a diskette that is exposed to a magnetic field.)



- Do not expose diskettes to heat greater than 51.5°C (125°F) or to direct sunlight.



- Do not write on diskettes outside the label area.



Storage

When diskettes are not in use, they should be placed in their protective envelopes and stored in the following environment:

- Temperature: 10°C to 51.5°C (50°F to 125°F)
- Relative humidity: 8% to 80%
- Maximum wet bulb: 29.4°C (85°F)

If a diskette has been exposed to an environment outside of the given range, the diskette should be given at least five minutes to acclimate to a suitable environment before use. During that time, the diskette should be removed from its shipping container.

Shipping

Diskettes should be shipped only in special shipping cartons (available from IBM). With the diskette in place, the package weighs 280 grams (10 ounces).

Shipping cartons should be labeled:

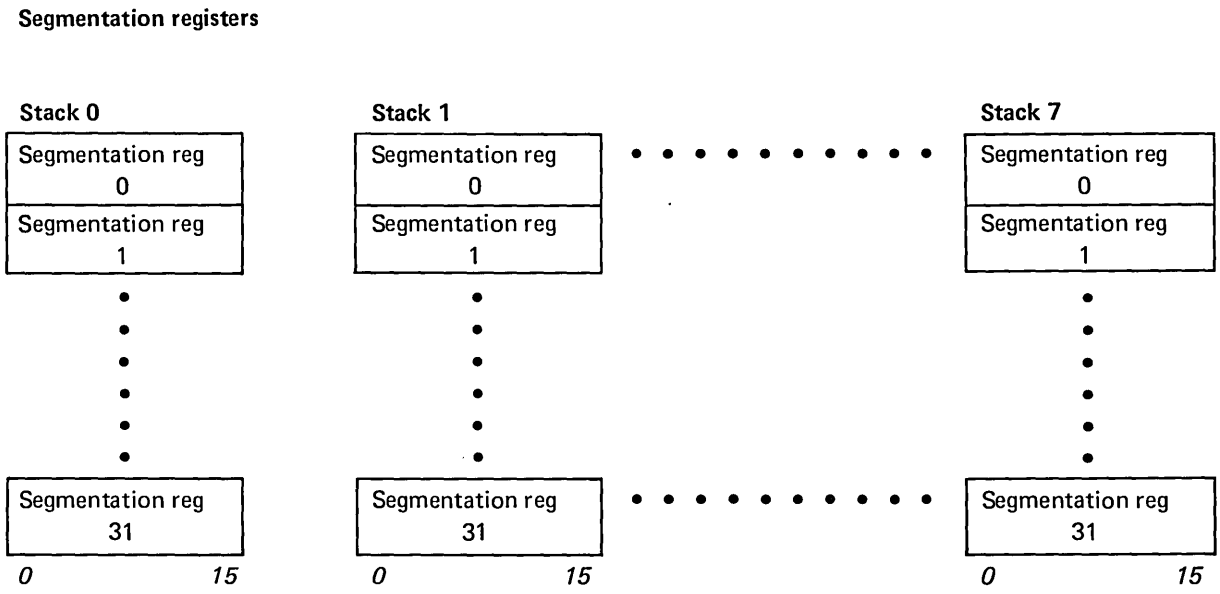
DO NOT EXPOSE TO HEAT OR SUNLIGHT

Chapter 2. Main Storage Addressing Using the Relocation Translator

The relocation translator and segmentation registers permit addressing of main storage locations beyond 64K bytes and provide a read-only type of storage protection. The first 64K bytes can be addressed directly when the translator is disabled; therefore, the translator must be enabled when main storage above 64K bytes is accessed.

Translator Description

The translator provides eight stacks of 16-bit segmentation registers. The stacks are numbered 0–7 to correspond to the eight possible values of the address keys. Each stack consists of 32 registers (0–31):



The stacks of segmentation registers are under supervisory program control. Four privileged instructions are used with the relocation translator and segmentation registers.

- **Set Segmentation Register (SESR).** This instruction loads one segmentation register.
- **Copy Segmentation Register (CPSR).** This instruction allows the supervisor to inspect the contents of a segmentation register.
- **Enable (EN).** This instruction enables the relocation translator. Until the translator is enabled, 16-bit addressing is in effect for the low-order 64K bytes of storage. Any storage above 64K bytes is not accessible to the program until the translator is enabled.
- **Disable (DIS).** This instruction disables the relocation translator.

Refer to the *IBM Series/1 Principles of Operation,* GA34-0152, for detailed descriptions of the preceding instructions.

Storage Mapping

Mapping of main storage is achieved through the segmentation registers. Each segmentation register controls a 2K-byte segment of storage. The SESR instruction is used to load each segmentation register with the unique physical address of a 2K-byte segment of storage.

> **Note:** More than one segmentation register can be loaded with the same segment address. For example, stack 0, register 15 (associated with the supervisor address key of 0), can be loaded with the same number as stack 1, register 6. This arrangement allows the supervisor to address control blocks within a problem program even though the address key for the supervisor is different than the key for the problem program. Once loaded, each stack of segmentation registers contains a complete map of 64K bytes divided into 2K-byte physical segments.

Relocation Addressing

The relocation translator generates a physical address that allows any byte in storage to be addressed. Figure 2-1 shows an example of address translation. The letters in the following description correspond to the letters in Figure 2-1:

Ⓐ The active address key from the address key register selects a segmentation register stack. The address key pertains to the instruction being executed on the current priority level.

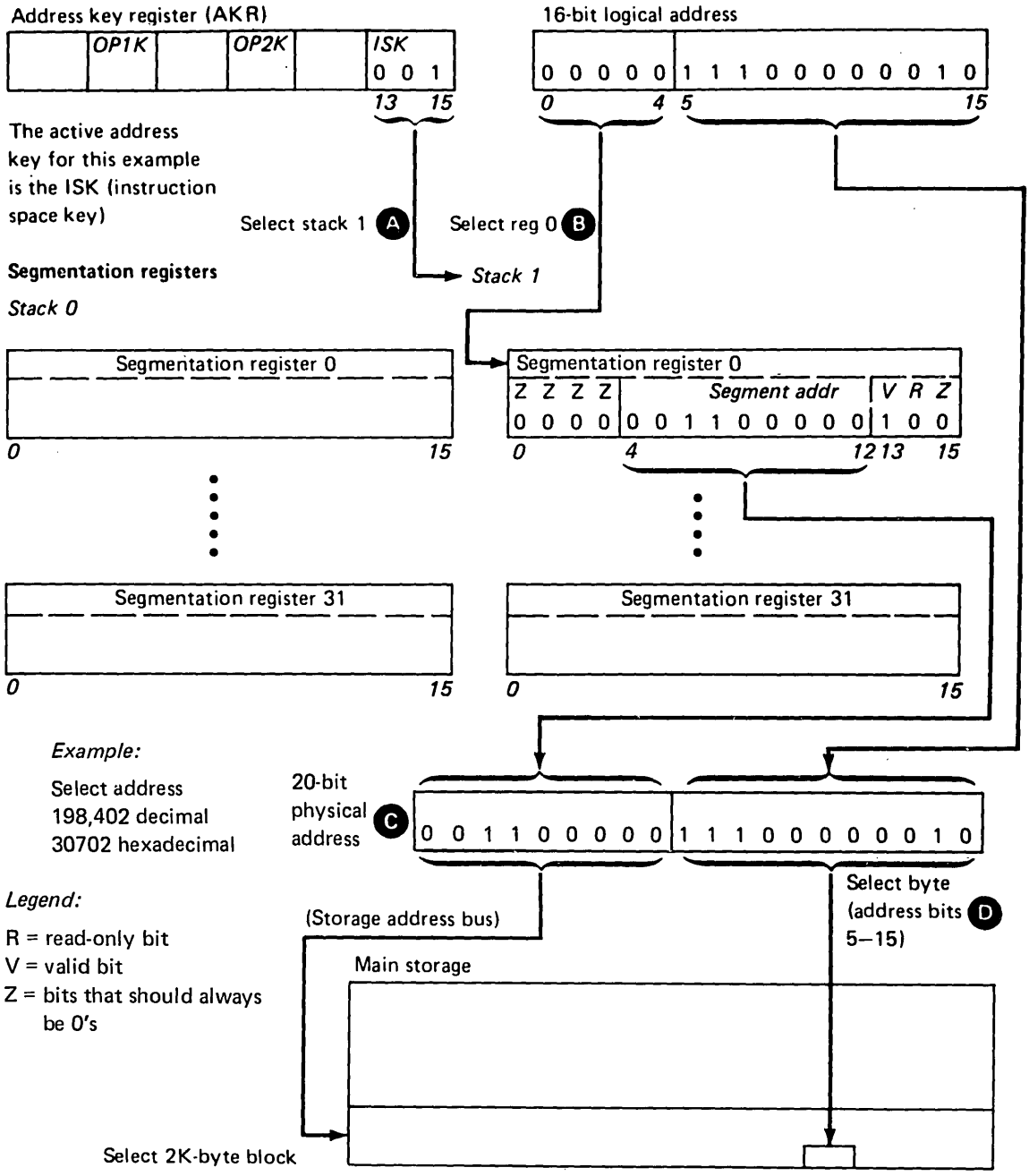
Ⓑ The five high-order bits (0—4) of the 16-bit address (generated for the instruction being executed) select a segmentation register within the stack selected in description Ⓐ. These bits define the logical segment.

Ⓒ The physical address is generated. The high-order bits are from the segmentation register; these bits specify the physical address of a 2K-byte segment of storage.

Bit 13 - Valid Bit: When set to 1, this bit specifies that the contents of the segmentation register are valid; the segmentation register can be used to perform the translation. When bit 13 is a 0, the segmentation register cannot be used for translation (no access). If translation is attempted, a program-check interrupt occurs with invalid storage address set in the processor status word (PSW). (All valid bits are set to 0's after power is switched on.)

Bit 14 - Read-Only Bit: When set to 1, this bit specifies that the block is read-only. If an attempt is made to write into storage using a segmentation register with the read-only bit set to 1, a program-check interrupt occurs with protect check set in the PSW. Storage is not changed. Bit 14 is ignored by a cycle-steal access or when the processor is in supervisor state.

Ⓓ The 11 low-order bits (5—15) of the physical address are the 11 low-order bits (5—15) of the 16-bit logical address (generated for the instruction being executed); these bits specify the byte address within the 2K-byte segment.



Note: When the translator is disabled, address bits 0–15 only are used for main storage address selection.

Figure 2-1. Address Translation Example

I/O Storage Access Using the Relocation Translator

All storage access requests from I/O devices are translated by the same hardware that handles storage requests from the processor. The device control blocks (DCBs) must reside in the supervisor's address space; therefore, all I/O devices must use address key 0 to gain access to the DCBs and to store the individual residual status blocks. The address key of the process requiring a cycle-steal operation resides in a DCB. An I/O device presents this address key, along with a 16-bit logical address, to the relocation translator. This allows an I/O device to directly address the storage space for a particular process. The address key allows I/O storage protection to be established between address spaces, assuming that the supervisor ensures the integrity of the DCBs.

Status of Translator After Power Transitions and Resets

The translator is enabled by the Enable (EN) instruction, or by the PSW key of the programmer console, if installed. The translator is disabled by any of the following:

- Disable (DIS) instruction
- Power-on reset
- Check Restart key on programmer console
- Initial program load (IPL)
- System Reset key on programmer console

All translator controls are reset when the translator is disabled.

Notes:

1. A machine-check interrupt does not disable the translator.
2. The segmentation registers are not reset when the translator is disabled.
3. The valid bits are all set to 0's when power is switched on.

Error-Recovery Considerations

Invalid Storage Address (ISA)

The invalid storage address bit (bit 1 of the PSW) is set to 1 by any one of the following:

- Storage access was attempted using a physical address greater than the physical storage size installed.
- Storage access was attempted with bit 13 (valid bit) of the segmentation register set to 0. This signifies that the contents of the segmentation register are invalid.

The specific nature of the invalid storage address can be resolved as follows:

- Store the segmentation register following the program-check interrupt.
- Test the value of bit 13 in the selected segmentation register. When set to 1, this bit specifies that the contents of the segmentation register are valid; the segmentation register can be used to perform the translation. When bit 13 is a 0, the segmentation register cannot be used for translation (no access). If translation is attempted, a program-check interrupt occurs with invalid storage address set in the processor status word (PSW).
- Ensure that the segment address does not exceed the limits of the physical processor storage installed.

Protect Check

When the translator is enabled, a program-check interrupt with protect check set in the PSW is caused by an attempt to write into storage, while in the problem state, using a segmentation register with bit 14 (read-only) set to 1.

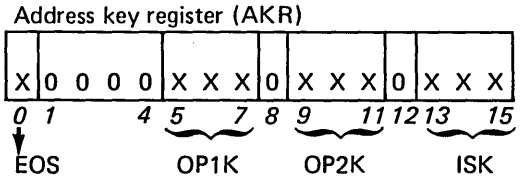
Storage is not changed. Bit 14 is ignored by a cycle-steal access, or when in supervisor state.

Active Address Key

Cycle-steal devices have a cycle-steal address key specified in their device control block.

Any one of the four address keys (ISK, OP1K, OP2K, or the cycle-steal address key) may be used during a storage access as the active address key. The address key in use (active) depends on the type of operation being performed at a specific instant in time. The active address key defines storage access through a particular block of segmentation registers.

Each priority level in the processor has an associated address key register (AKR) that contains three address keys and an equate-operand-spaces (EOS) bit.



EOS — *Equate operand spaces.* This bit, when set to 1, causes all data operands to use the OP2K address key. See "Equate Operand Spaces (EOS)" in this chapter.

OP1K — *Operand 1 key.* These bits contain the binary-coded operand 1 address key, with bit 7 as the low-order bit.

OP2K — *Operand 2 key.* These bits contain the binary-coded operand 2 address key, with bit 11 as the low-order bit.

ISK — *Instruction space key.* These bits contain the binary-coded instruction-space address key, with bit 15 as the low-order bit.

Equate Operand Spaces (EOS)

The equate operand spaces bit (bit 0) in the address key register controls the use of the OP1K address key.

When the EOS bit is set to 1 (enabled), all processor data fetches use a single address space defined by the OP2K address key. The OP1K is ignored, but not changed, and all normal OP1K operations use OP2K as an active key. When the EOS bit is set to 0 (disabled), the OP1K address key functions in a normal manner.

Equate operand spaces (EOS) may be enabled by an Enable (EN) instruction, a Set Level Block (SELB) instruction, or a Set Address Key Register (SEAKR) instruction. EOS may be disabled by a Disable (DIS) instruction, a Set Level Block (SELB) instruction, or a Set Address Key Register (SEAKR) instruction. The EOS is also disabled by a priority interrupt or a class interrupt. These instructions are described in the *IBM Series/1 Principles of Operation*, GA34-0152.

Address Space

When the relocation translator is enabled, an address key defines a specific address space where:

- The address space is a range of logically contiguous storage.
- The address space is accessible by the effective address without operating system intervention (the address space is not greater than 64K bytes).

All instruction fetches use the address space defined by the instruction space key (ISK). For storage-to-storage instructions, all reads and writes for data operand 1 use the address space defined by the OP1K, assuming that the EOS bit is a 0. All other storage data accesses, reads, and writes use the address space defined by the OP2K, excluding branch and jump instructions.

Examples:

ISK=OP1K=OP2K. For instruction processing, all storage accesses occur within the same address space.

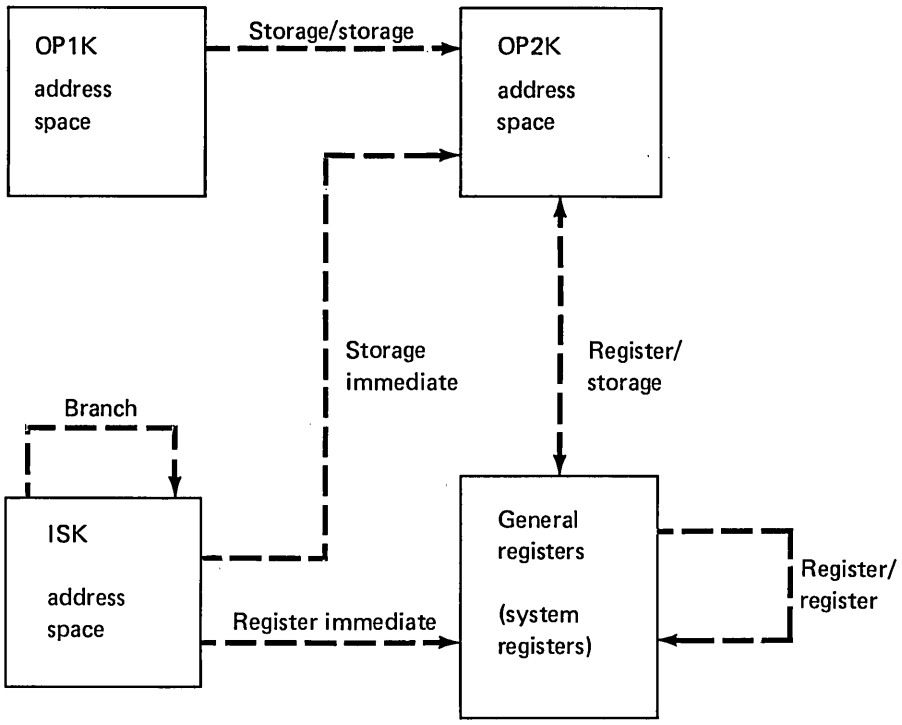
ISK≠OP1K, OP1K=OP2K. Instruction fetches occur in the ISK address space. Data access occurs in the OP2K address space.

ISK≠OP1K, OP1K≠OP2K. Refer to Figure 2-2 for this example.

I/O operations that access main storage also use an address key. Cycle-steal operations (read or write) use the cycle-steal address key specified within the device control block. An address key of 0 is used when the device fetches the device control block. Direct program control (DPC) operations that write data to storage use the OP2K address key.

Other defined uses of the address key register are as follows:

- All indirect access for branching uses the ISK.
- Effective-address generation occurs in the address space of the particular data operand. The appended words in the instruction are accessed by the ISK.
- Storage access from the console is defined by the SAR address key. Stop-on-address is based on the Stop On Address key when the translator is enabled.
- System reset and IPL set all address keys and the EOS bit to 0's.



Assembler syntax for address spaces (see Appendix A)

ISK	OP1K	OP2K	Example instructions	
	addr5	addr4	AW	addr5,addr4
	(reg)	(reg)	MVFD	(reg),(reg)
Bits 13-15 of AKR			MVBI	byte,reg
Bits 13-15 of AKR			B	longaddr*

*Indirect addressing.

Notes:

1. OP1K is only used for the source operand in storage-to-storage operations.
2. OP2K is used for storage data access in all other operations (excluding branch/jump).
3. ISK (bits 13-15 of the AKR) is used for instruction fetch and branch/jump operations.

Figure 2-2. Data Movement in Address Spaces When ISK≠ OP1K, OP1K≠OP2K

Address Key Values After Interrupts

When priority or class interrupts occur, certain values are set in the address keys of the affected AKR. These values anticipate the address spaces that the programmer might need for interrupt processing. Figure 2-3 shows the resulting AKR values for each type of interrupt:

Interrupt	EOS	OP1K	OP2K	ISK
Priority	0	0	0	0
Supervisor call	0	Note 1	0	0
Machine check	0	Note 2	0	0
Program check	0	Note 2	0	0
Soft-exception trap	0	Note 1	0	0
Trace	0	Note 3	0	0
Console	0	0	0	0
Power/thermal warning	0	0	0	0

Notes:

1. OP1K is set to the preceding key contained in OP2K.
2. OP1K is set to the last active processor address key.
3. OP1K is set to the preceding key contained in the ISK.

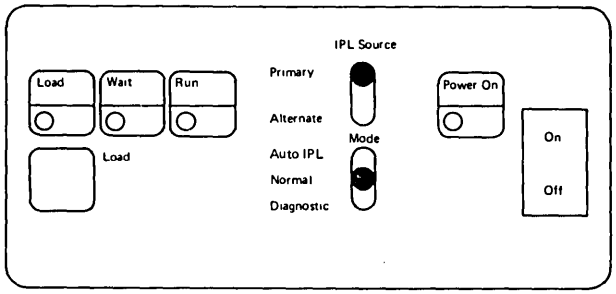
Figure 2-3. Resulting AKR Values

All interrupt service routines reside in address space 0; therefore, the ISK and OP2K are set to 0's when an interrupt occurs. Necessary information for processing a specific interrupt may reside in an address space other than 0. The address key related to the particular interrupt is placed in OP1K. The OP1K is set in anticipation of a storage-to-storage move of information from the interrupting address space to address space 0.

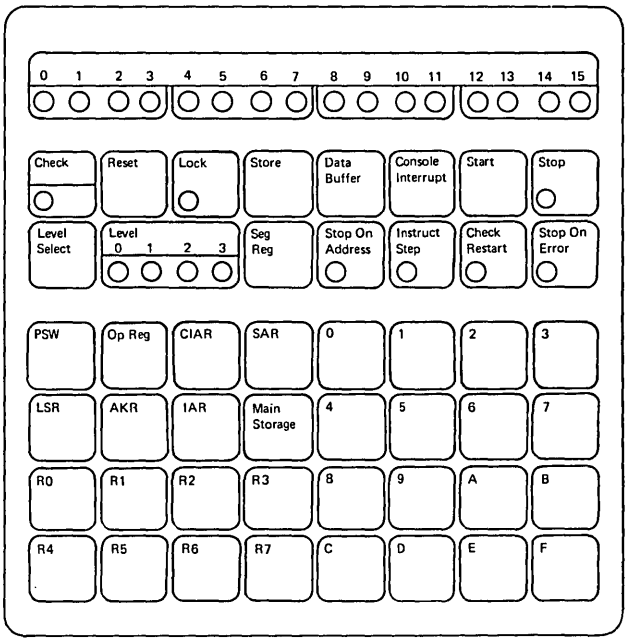
> **Note:** Class interrupts cause a hardware-controlled storing of a level status block. This operation uses address key 0.

Chapter 3. Console

The basic console is standard; the programmer console is an optional feature.



Basic console



Programmer console

The basic console is intended primarily for those systems that are totally dedicated to a particular application, where operator intervention is not needed during the execution of the application.

The programmer console is intended for operator-oriented systems where various programs are entered and executed. This type of environment requires a more versatile console arrangement for program and machine problem determination, and for manual alteration of data and programs in storage.

Basic Console

Each 4956 comes equipped with a basic console, which provides the following:

- Power On/Off switch for the processor unit
- IPL Source switch to select a primary or alternate IPL device
- Load key for initial program load (IPL)
- Mode switch to select: Auto IPL, Normal, or Diagnostic mode
- Load, Wait, Run, and Power On indicators

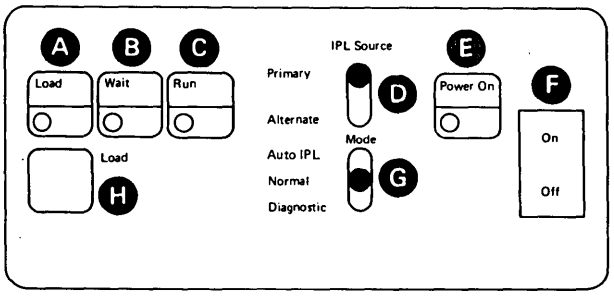
F Power On/Off: When this switch is set to the On position, power is applied to the processor unit. After all power levels are up, the Power On indicator is turned on. When this switch is set to the Off position, power is removed from the processor unit and the Power On indicator is turned off.

D IPL Source: This switch selects the I/O device to be used for program loading. In the Primary position, the device that was pre-wired as the primary IPL device is selected. In the Alternate position, the device that was pre-wired as the alternate IPL device is selected.

H Load: Pressing this key causes a system reset, and the initial program load (IPL) sequence is started. The Load indicator is turned on and remains on until the IPL sequence is completed. When the IPL sequence is completed, instruction execution begins at location 0 on priority level 0.

G Mode: This switch has the following positions:

- Auto IPL—In this position, an IPL is initiated after a successful power-on sequence. Bit 13 of the PSW is set to indicate to the software that an automatic IPL was performed. In this mode, Stop instructions are treated as no-ops.
- Normal—In this position, Stop instructions are treated as no-ops.
- Diagnostic—This position has no function without the programmer console. This position places the processor in diagnostic mode if the programmer console is attached. When the processor is in diagnostic mode, Stop instructions cause the processor to enter the stop state.



Indicators

Ⓐ **Load:** On when the machine is performing an initial program load (IPL).

Ⓑ **Wait:** On when an instruction that exits the active level has been executed and no other priority interrupts or levels are pending.

Ⓒ **Run:** On when the machine is executing instructions.

Ⓔ **Power On:** On when the proper power levels are available to the system.

Programmer Console

The programmer console is an optional feature that can be ordered with the 4956 or field-installed at a later date. The programmer console provides the following:

- Start and stop of the processor.
- Ability to display or alter any storage location.
- System reset.
- Selection of any one of the four interrupt levels for the purpose of displaying or altering data.
- Displaying or altering of the storage address register (SAR), instruction address register (IAR), SAR address key register (AKR), stop-on-address address key register (AKR), level address key register (AKR), segmentation registers, console data buffer, or any general purpose register.
- Displaying, but not altering, the level status register (LSR), current instruction address register (CIAR), op register, or processor status word (PSW). Note that the following bits of the PSW and LSR may be altered: PSW bit 14 (translator enabled), LSR bit 8 (supervisor state), and LSR bit 11 (summary mask).
- Stop on address.
- Stop on error.
- Instruction stepping.
- Check restart.
- Request for a console interrupt.
- Check indicator. The Check indicator is a light emitting diode (LED) that lights when a machine check or program check class interrupt occurs.
- Lock console.
- CE mode. The CE mode is used to display the error log.

The programmer console is touch-sensitive, with an audio-tone generator providing an audio response tone whenever a key is pressed and the information has been accepted and serviced by the processor.

Console Display

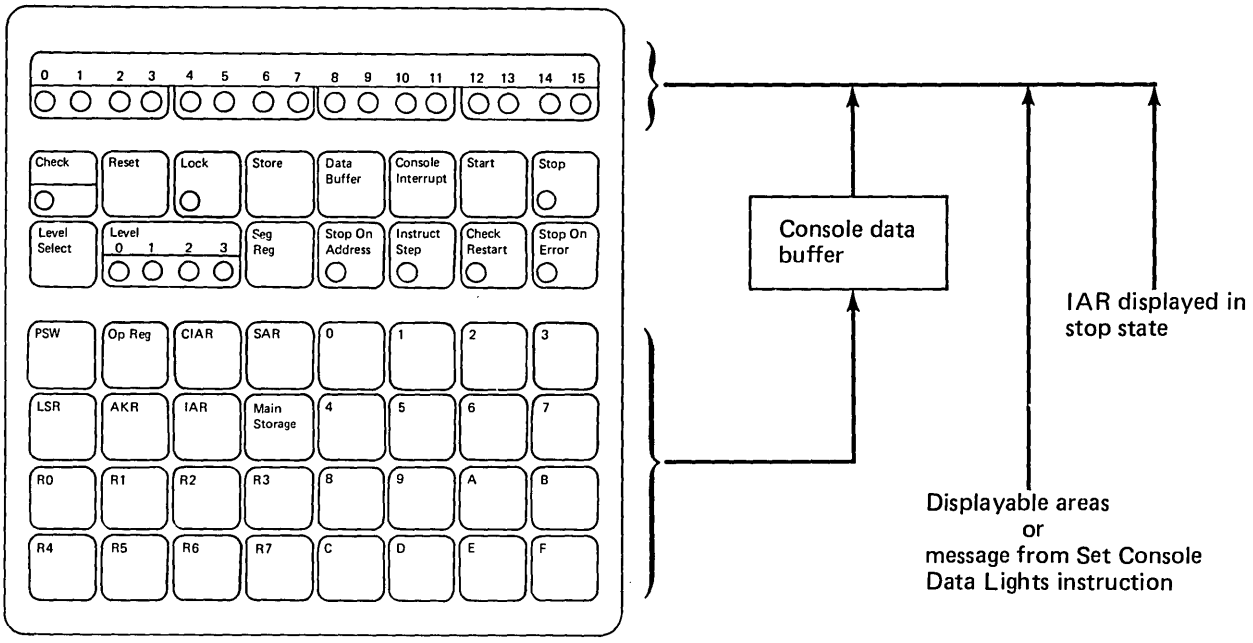
Run or Wait State

When the processor is in run or wait state, the console data buffer is displayed in the data display indicators. An exception to this is when a Set Console Data Lights (SECON) instruction writes a message to the data lights and does not change the buffer. When the Data Buffer key is pressed, the console data buffer is again displayed in the indicators.

When the console data buffer is being displayed, the console data buffer and the display are changed by entering new data with the data entry keys.

Stop State

When the processor enters stop state, the IAR is displayed in the data display indicators. Any system resource that has a corresponding select key on the console can be displayed. For example, the console data buffer can be displayed by pressing the Data Buffer key.



Power-On Reset

After a successful power-on reset, the data display indicators are set on, and the Stop indicator is set on (if the Mode switch is not positioned for Auto IPL).

Indicators

A Data Display: When the processor is in run state, the console data buffer is displayed in the data display indicators.

The Set Console Data Lights (SECON) instruction can write a message to the data display.

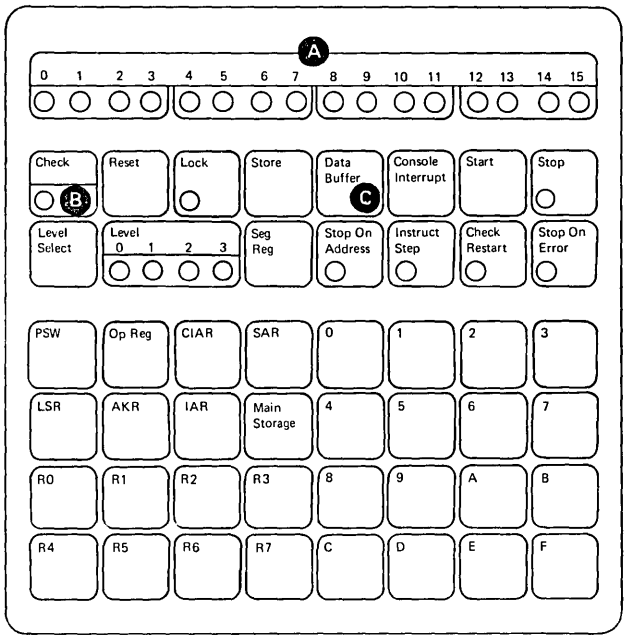
When the processor enters stop state, the IAR is displayed until another system resource is selected.

To display the contents of the console data buffer after a system resource has been displayed, press the Data Buffer key **C**

B Check: On when a machine-check or program-check has been recognized. The Check indicator is turned off by:

- Clearing the check condition.
  - Reset key.
  - Load key.
  - Executing a Copy Processor Status and Reset (CPPSR) instruction. This instruction resets bits 0-12 of the PSW.
- Pressing any console key while in the stop state. The check condition is not cleared unless the Reset key or the Load key is pressed.

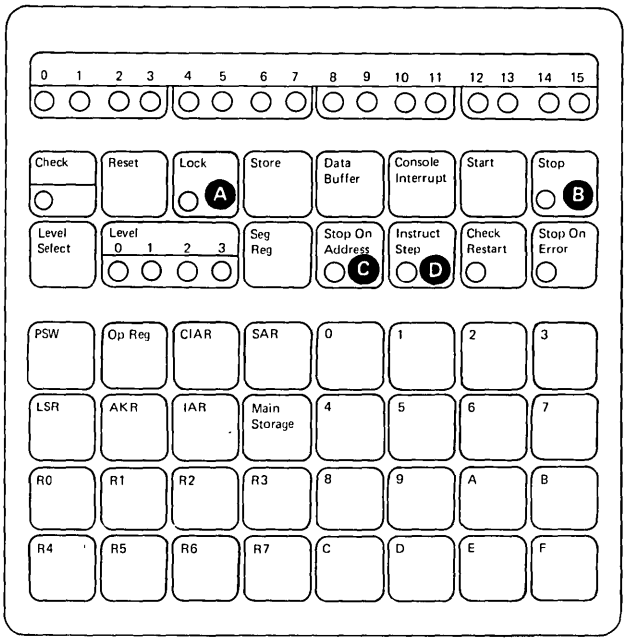
While in the stop state, the Check indicator is used to indicate main storage parity errors or invalid storage addresses during display operations. Refer to "Displaying Main Storage Locations" in this chapter.



Combination Keys/Indicators

There are six combination keys/indicators:

- Lock
- Stop
- Stop On Address
- Instruct Step
- Check Restart
- Stop On Error



A Lock: Pressing the Lock key first (Lock LED begins flashing), then pressing four hex keys and the Store key locks the console. A locked console is indicated by an illuminated indicator on the Lock key. The data LEDs are automatically set to the previous value. Displays or alterations cannot be performed with the programmer console keys while in the Lock mode. The console remains locked until the same sequence of hex keys that locked the console is repeated and then followed by pressing the Store key.

The only data displayed during the lock mode is data set by the program or data displayed during a maintenance procedure (CE) mode.

Lock mode is automatically reset during a power-on sequence. If the console is locked and an auto IPL occurs after a power failure, the console will not be locked after the power-on.

If the console is locked in the stop mode, the only active switches are Lock, Store, and the hex keys. At this time, the run mode cannot be entered; therefore, for a normal lock function, lock mode should only be set during the run mode.

CE Mode: The CE mode may be used (to allow the user to display the error log) by the following:

1. Press the Lock key; the Lock LED flashes.

2. Press the hex keys in the sequence : C, E, 0, 0.

3. Press the Store key; the most recent entry in the error log is indicated by the display LEDs.

4. Press the Lock key once; the previous entry in the error log is indicated by the display LEDs. The Lock key may now be pressed as many times as desired. Each time the Lock key is pressed, the next previous error log entry appears on the display LEDs. (Refer to Appendix C for a description of the error log.)

   Other keys may be pressed between subsequent operations of the Lock key. CE mode is exited when all 64 entries have been displayed, or when the Store key is pressed immediately after the Lock key is pressed.

   Upon entering a lock/unlock/CE mode sequence, the Lock LED flashes. The SECON instruction is disabled until the lock/unlock/CE mode sequence is terminated. The console data LEDs then assume their former value, their value upon entering stop state if in stop state, or the last value sent to them if SECON instructions have occurred.

Ⓑ **Stop:** This indicator is on when the processor is in the stop state. Stop state is entered in the following ways:

- By pressing the Stop key.
  - In run state, the current instruction is completed.
  - In wait state, stop state is entered directly.
  - In stop state, the contents of the instruction address register (IAR) prior to entering the present stop state are restored to the IAR and displayed in the data display indicators. The level that was active upon entering stop state is reselected (becomes active).
- By execution of the Stop instruction (diagnostic mode only).
- When an address compare occurs in stop-on-address mode.
- When an error occurs in stop-on-error mode.
- By pressing the Reset key.
- When a power-on reset occurs.
- By selecting instruction-step mode while in run state.

The Stop On Address key and the Instruct Step key are mutually exclusive. When one is pressed, the other is reset if it is on.

Ⓒ **Stop On Address:** Pressing this key places the processor in stop-on-address (SOA) mode and turns on the Stop On Address indicator. Pressing this key a second time resets stop-on-address mode and turns off the indicator.

Ⓓ **Instruct Step:** Pressing this key places the processor in instruction-step mode and turns on the Instruct Step indicator. Pressing this key a second time resets instruction step mode and turns off the indicator.

If the processor is in run or wait state, pressing this key causes the processor to enter stop state. Pressing the Instruct Step key a second time resets instruction-step mode; the processor remains in stop state.

To operate in instruction step mode:

1. Key the desired starting address and store into the IAR.

2. Press the Instruct Step key.

3. Press the Start key. The instruction located at the selected address is executed, and the processor returns to stop state. The IAR is updated to the next instruction address; this address is displayed in the data display indicators.

Each time the Start key is pressed, one instruction is executed and the IAR is updated to the next instruction address.

> **Note:** Priority and class interrupts are not inhibited during execution of the instruction.

Stop-On-Address Mode

The processor must be in stop state to set the compare address.

Stop On Address (Relocation Translator Disabled)

1. Press the Stop On Address key.

   Contents of the stop-on-address register are indicated by the display LEDs.

2. Enter the selected stop-on-address address by pressing the hex entry keys for a four-digit hex address.

3. Press the Store key.

   Contents of the updated stop-on-address register are indicated by the display LEDs.

4. Press the Start key.

   Execution begins at the current IAR address on the level that was active prior to entering the stop state.

   When the selected address is loaded into the SAR, the processor enters the stop state. If a stop-on-address compare occurs during the instruction fetch, the stop state is entered immediately with the compare SAR address indicated by the display LEDs. If a stop-on-address compare occurs during an operand fetch/store, the stop state is entered after completing the instruction and the next instruction address is indicated by the display LEDs. To exit stop state, press the Start key; execution begins at the next sequential address.

   If the selected address is an instruction address:

   - When the compare occurs, the stop state is entered with the compare SAR address displayed in the data display indicators.

   - Certain machine conditions occur that cause the stop state to be entered on the wrong instruction address. When this happens, continue to press the Start key until the selected instruction address is displayed.

Stop On Address (Relocation Translator Enabled)

1. Press the Stop On Address key.

   Contents of the stop-on-address (SOA) register are indicated by the display LEDs.

2. Press the AKR (address key register) key.

   Contents of the stop-on-address address key register are displayed.

3. Enter the desired address key by pressing one hex entry key for a digit value (hex 0 through 7).

4. Press the Store key.

   Contents of the updated stop-on-address key register are displayed.

5. Press the Stop On Address key.

   Contents of the stop-on-address register are indicated by the display LEDs.

6. Enter the selected compare address by pressing the hex entry keys for a four-digit hex address.

7. Press the Store key.

   Contents of the updated stop-on-address register are indicated by the display LEDs.

   The selected stop-on-address key register and stop-on-address register are used to compute a 20-bit physical address. Whenever the value in the segmentation register is changed, the physical address is recomputed.

   > **Note:** The contents of the stop-on-address key register and the stop-on-address register may be displayed on the console; however, the 20-bit physical address cannot be displayed.

8. Press the Stop On Address key.

   The processor is now in stop-on-address mode.

9. Press the Start key.

   Execution begins at the current IAR address on the level that was active prior to entering the stop state.

   When the selected physical address is computed using the SAR and the active address key, the processor enters the stop state. If the stop-on-address compare occurs during instruction fetch, the stop state is entered immediately with the compare SAR address indicated by the display LEDs. If the stop-on-address compare occurs during an operand fetch/store, the stop state is entered after completing the instruction. The next instruction address is displayed by the LEDs. To exit stop state, press the Start key; execution begins at the next sequential address.

If the selected address is an instruction address:

- When the compare occurs, the stop state is entered with the compare SAR address displayed in the data display indicators.
- Certain machine conditions occur that cause the stop state to be entered on the wrong instruction address. When this happens, continue to press the Start key until the selected instruction address is displayed.

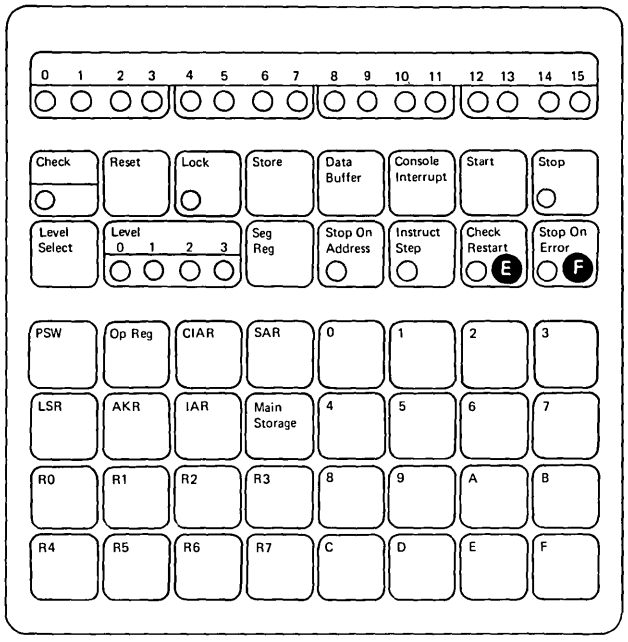
The Check Restart key and the Stop On Error key are mutually exclusive. When one is pressed, the other is reset if it is on.

(E) **Check Restart:** Pressing this key places the processor in check restart mode. While in this mode, a program-check, machine-check, or power/thermal-warning class interrupt causes the processor to be reset and execution to restart at address 0 on level 0.

Note: The power/thermal-warning stop-on-error condition is controlled by the summary mask.

(F) **Stop On Error:** Pressing this key places the processor in stop-on-error mode. Any program-check, machine-check, or power/thermal-warning class interrupt causes the processor to enter stop state. To determine the cause of the error, display the PSW. To restart the processor, press the Reset key and then the Start key. Pressing only the Start key allows the processor to proceed with the class interrupt as if stop mode had not occurred. Note that the Check indicator may have been turned off while in stop state. After the class interrupt routine is completed, control may be returned to the instruction that caused the error and an attempt to reexecute the instruction may be made. Some instructions are not reexecutable because operand registers or storage locations were changed before the instruction was terminated (because of the initial error). In these cases, the operator must be familiar with the program because manual restoration of affected locations must be made before restart is attempted.

Note: The power/thermal-warning class interrupt is controlled by the summary mask.



Keys and Switches

Ⓐ **Reset:** This key initiates a system reset that performs the following functions:

- IAR on level 0 set to 0
- AKR on level 0 set to 0
- Interrupt mask set to all levels enabled
- LSR on level 0—indicators set to 0's, summary mask enabled, supervisor state and in-process flag turned on, trace disabled
- LSRs for levels 1–3 set to 0's
- PSW bits 0–12 and 14 set to 0's (bit 14 set to 0 indicates translator disabled); bits 13 and 15 retain their state prior to system reset
- SAR set to 0
- CIAR set to 0
- Console display LEDs are turned off
- Clock class interrupts are disabled
- Error logging set to the enabled state

After the system reset is completed, the processor is placed in the stop state with the Stop indicator on.

The following resources are not affected by system reset:

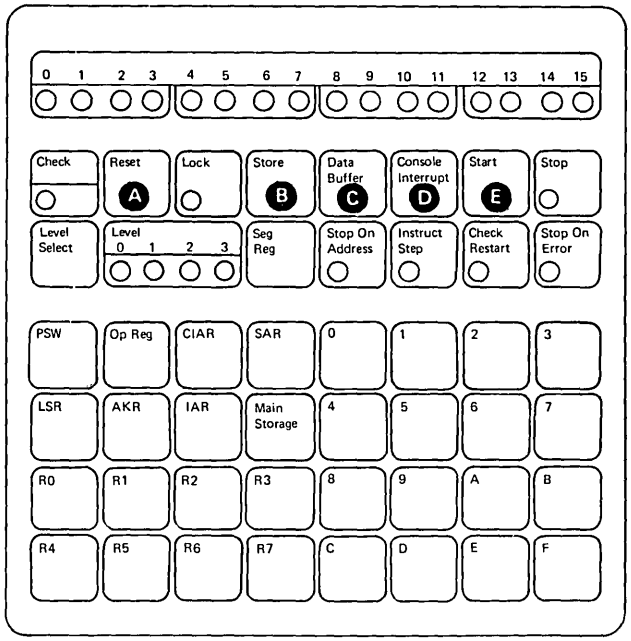
- General registers (all levels)
- IARs (levels 1–3)
- AKRs (levels 1–3)
- Main storage
- Console data buffer
- Segmentation registers
- Stop-on-address register
- Clock
- Comparator

Ⓑ **Store:** This key is effective only when the processor is in stop state. Pressing this key causes the last data entry to be stored in the last selected resource.

Ⓒ **Data Buffer:** Pressing this key causes the console data buffer to be selected. The contents of the console data buffer are displayed in the data display indicators.

D Console Interrupt: The effect of this key depends on the state of the processor. If the processor is in the stop or load state, this key has no effect. If the processor is in the run or wait state and the summary mask is enabled prior to the key action, a console-class interrupt occurs. The audio-response tone is generated when the interrupt is processed.

E Start: This key is effective in stop state only. Stop state is exited and the processor resumes execution at the address in the IAR on the current level. If stop state was entered from system reset, execution begins at address 0, level 0. If stop state was entered from wait state, the processor returns to wait state.



(H) PSW: Pressing this key selects the processor status word. The contents of the PSW are displayed in the data display indicators. Only PSW bit 14 (translator enabled) can be stored into the PSW from the programmer console.

(J) Op Reg: Pressing this key selects the op register and displays the contents in the data display indicators. Data cannot be stored into the op register from the console.

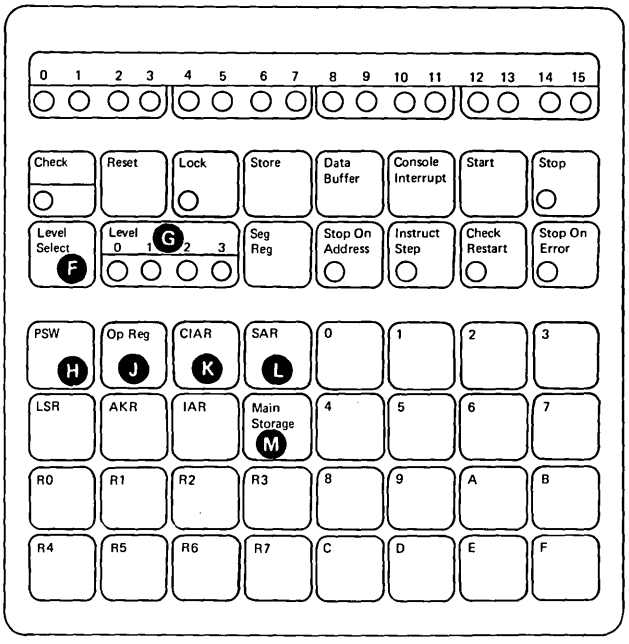
(K) CIAR: Pressing this key, after entering stop state, causes the address of the instruction just executed to be displayed. Data cannot be stored into the CIAR from the console.

(L) SAR: Pressing this key, while in stop state, displays the contents of the storage address register. An address can be stored into the SAR to address main storage or the segmentation registers for display or store operations. Bit 15 of the SAR cannot be set from the console.

(M) Main Storage: Pressing this key selects main storage as the facility to be accessed by the console. When this key is pressed, the contents of the main storage location addressed by the SAR are displayed in the data display indicators. Procedures for displaying and storing main storage are described in subsequent paragraphs in this chapter.

(F) Level Select: In the stop state, the Level-Select key should be pressed first, before selecting a new level. The desired level may then be selected by pressing either the 0, 1, 2, or 3 hex key.

The current active level (Level 0, 1, 2, or 3) is always displayed by one of the four level indicators at (G).



Level-Dependent Keys

The following keys select registers that are duplicated in hardware for each of the four interrupt levels:

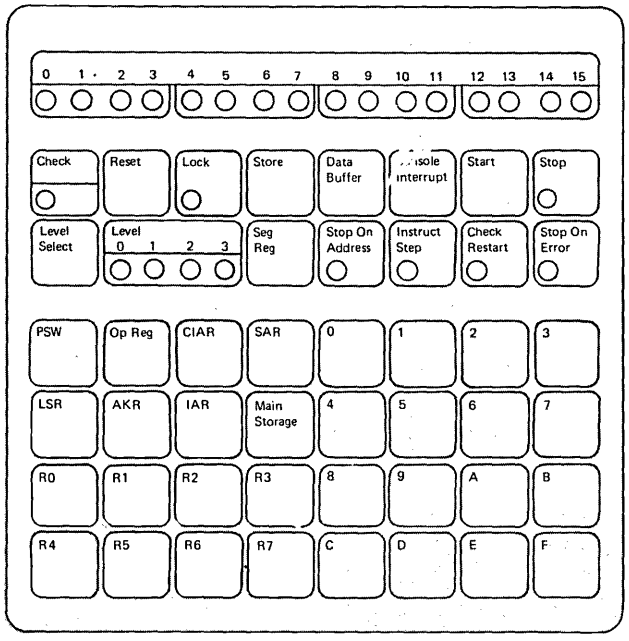
- LSR
- AKR
- IAR
- R0–R7 (General purpose registers 0–7)

Pressing any of these keys, once a level has been selected, causes the contents of that register to be displayed in the data display indicators.

The level status register (LSR) is displayable only, except bits 8 (supervisor state) and 11 (summary mask) can be stored into this register.

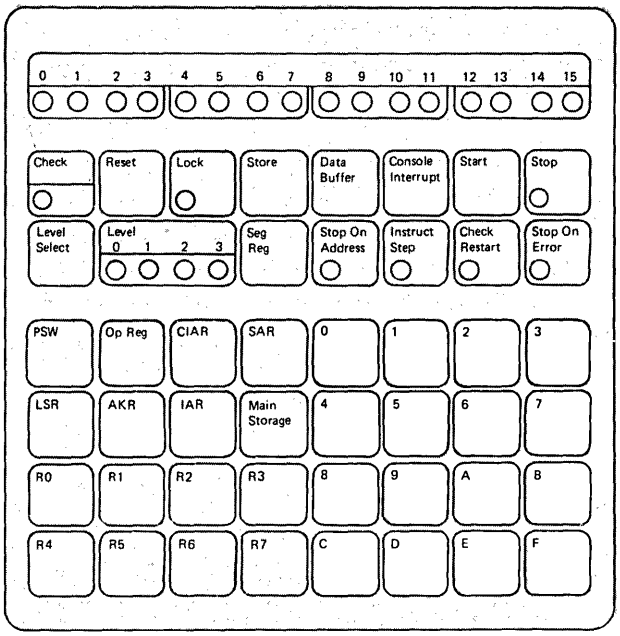
To display an AKR for a given level, enter the desired level, and then press the AKR key. The *level* AKR, bits 0, 5–7, 9–11, and 13–15 (EOS, OP1K, OP2K, and ISK) are displayed in the data display indicators.

To display SAR AKR, first press SAR, then press AKR. To display the stop on address AKR, first press the Stop On Address key, then press AKR. To display CIAR AKR, first press CIAR, then press AKR (three bits, ISK). An AKR store is accomplished by first displaying the level AKR, then entering four hexadecimal digits, followed by pressing the Store key. When the Store key is pressed, the new level AKR is displayed. After the SOA AKR, or the SAR AKR is displayed, enter one hexadecimal digit and press the Store key. The CIAR AKR is displayable only.

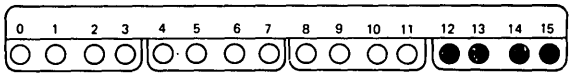
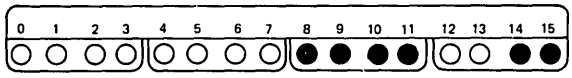
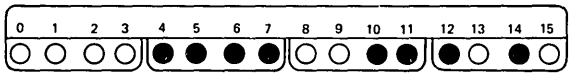
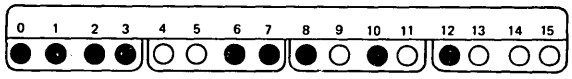


Data Entry Keys

The 16 data entry keys are used to enter data into a selected resource, such as main storage or a general register. When data is entered, it is shifted through the indicators, as shown in the following example:



Example: Data to be entered: F3A8

Action	Data display indicators
Press data entry key F	
Press data entry key 3	
Press data entry key A	
Press data entry key 8	

Legend:

● – Indicator on

○ – Indicator off

Displaying Registers

The processor must be in stop state.

1. Select the proper level by first pressing the Level Select key Ⓒ, then the appropriate 0, 1, 2, or 3 hex data key.

   The contents of any register associated with the selected level can now be displayed by pressing a register key.

2. Press the desired register key. The contents of that register are displayed in the data display indicators Ⓐ.

Storing Into Registers

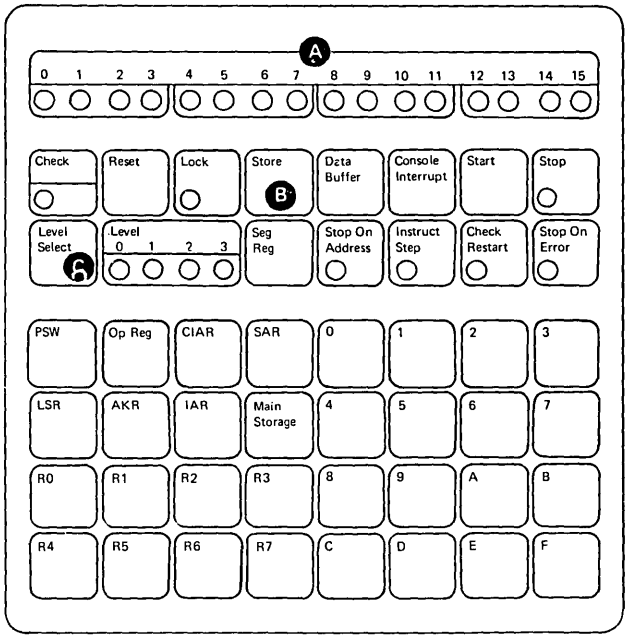
The processor must be in stop state.

1. Select the proper level by pressing the Level Select key Ⓒ, then the appropriate 0, 1, 2, or 3 hex data key.

2. Press the key for the register where data is to be stored. The contents of that register are displayed in the data display indicators Ⓐ.

3. Key in the data that is to be stored. This data is displayed in the data display indicators Ⓐ.

4. Press the Store key Ⓑ. The data that is displayed is stored into the selected register.

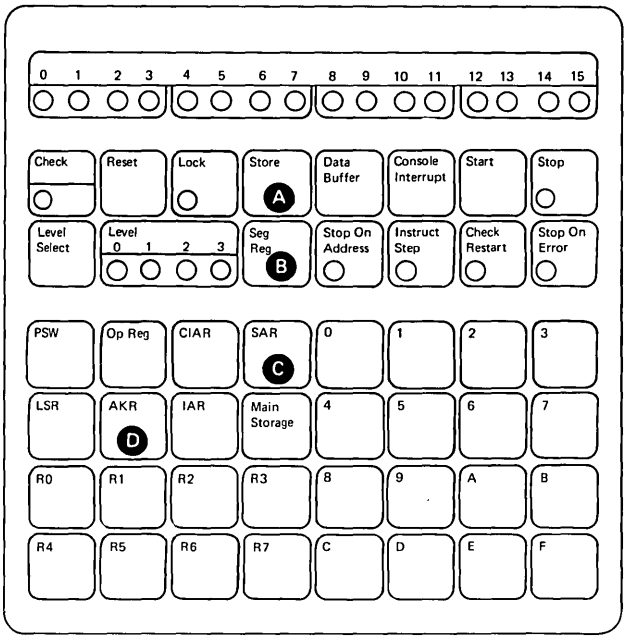


Displaying Segmentation Registers

The address relocation translator provides eight stacks (0–7) of 32 segmentation registers (0–31) in each stack, for a total of 256 segmentation registers. Refer to "Relocation Addressing" in Chapter 2.

The processor must be in the stop state.

1. Press the SAR key **C**. The contents of the SAR are displayed in the data display indicators.
2. Key in a hexadecimal four-digit number with the five high-order bits equal to the binary address (bits 0–31) of the desired segmentation register.
3. Press the Store key **A**. The address is stored in SAR.
4. Press the SAR key **C**. The selected address is displayed in the data display indicators.



5. Press the AKR key **D**. The contents of the SAR address key register (AKR) are displayed in the data display indicators.
6. Key in one hexadecimal character to select the desired segmentation stack (0–7).
7. Press the Store key **A**. The value is stored in the SAR AKR.
8. Press the Seg Reg key **B**. The contents of the selected segmentation register (defined by the five high-order bits of the SAR and the three SAR AKR bits) are displayed in the data display indicators.

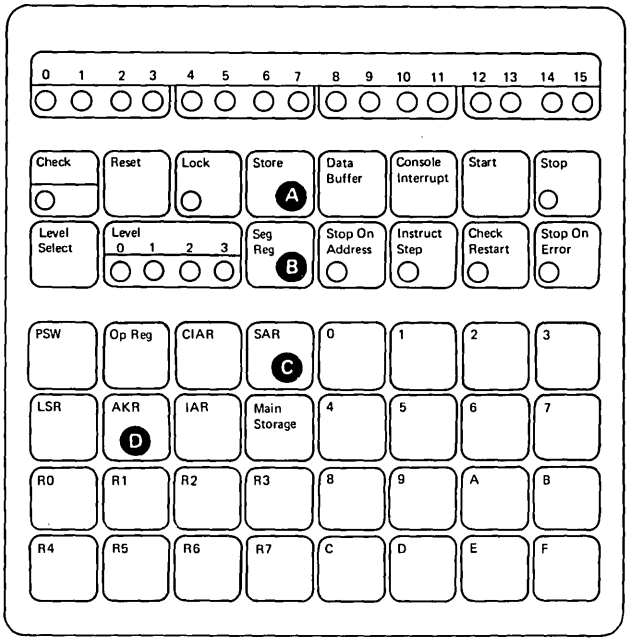
> **Note:** Each time the Seg Reg key is pressed, the segmentation-selection address is incremented by 1 until the last segmentation register in the stack is selected. Then, the segmentation-selection address wraps from 31 to 0. When the segmentation-selection address wraps from 31 to 0, the SAR AKR is incremented by 1 (a new segmentation-register stack is selected); the new segmentation-register contents are displayed in the data display indicators. When all segmentation register stacks have been selected, the SAR AKR value then wraps from 7 to 0.

Storing Into a Segmentation Register

The address relocation translator provides eight stacks (0–7) of 32 segmentation registers (0–31) for a total of 256 segmentation registers. Refer to "Relocation Addressing" in Chapter 2.

The processor must be in the stop state.

1. Press the SAR key Ⓒ. The contents of the SAR are displayed in the data display indicators.



2. Key in the value that selects the desired segmentation register within a stack (four hex characters entered with the data entry keys).

3. Press the Store key Ⓐ. The selected address is stored in the SAR.

4. Press the SAR key Ⓒ. The selected address is displayed in the data display indicators.

5. Press the AKR key Ⓓ. The contents of the SAR address key register (AKR) are displayed in the data display indicators.

6. Key in one hex character with a data entry key (any value from 0 through 7, which is the new address key that selects a segmentation-register stack). This character is displayed in bits 12–15 of the data display indicators.

7. Press the Store key Ⓐ. The contents of the SAR address key register (AKR) are updated to the value entered from the data entry keys.

8. Press the Seg Reg key (B). The contents of the selected segmentation register (defined by the five high-order bits of the SAR and the three SAR AKR bits) are displayed in the data display indicators.

9. Key in the value (four hex characters entered at the data entry keys) that provide both the desired nine high-order bits of the 20-bit physical main storage address (select a 2K-byte block of main storage) and that contain the correct value for the valid bit and the read only bit.

10. Press the Store key (A). The selected segmentation register is updated to the value in the data display indicators.

    Note: Each time the Store key is pressed, the last value keyed is entered into the selected segmentation register and the segmentation selection address is incremented by 1 until the last segmentation register in the stack is selected. Then, the segmentation selection address wraps from 31 to 0. When the segmentation selection address wraps from 31 to 0, the SAR AKR is incremented by 1 (a new segmentation-register stack is selected); the new segmentation-register contents are displayed in the data display indicators. When all segmentation register stacks have been selected, the SAR AKR value then wraps from 7 to 0.

The segmentation registers can be written into by the Set Segmentation Register (SESR) instruction and can be displayed by the Copy Segmentation Register (CPSR) instruction. Refer to *IBM Series/1 Principles of Operation*, GA34-0152, for additional information.

Displaying Main Storage Locations

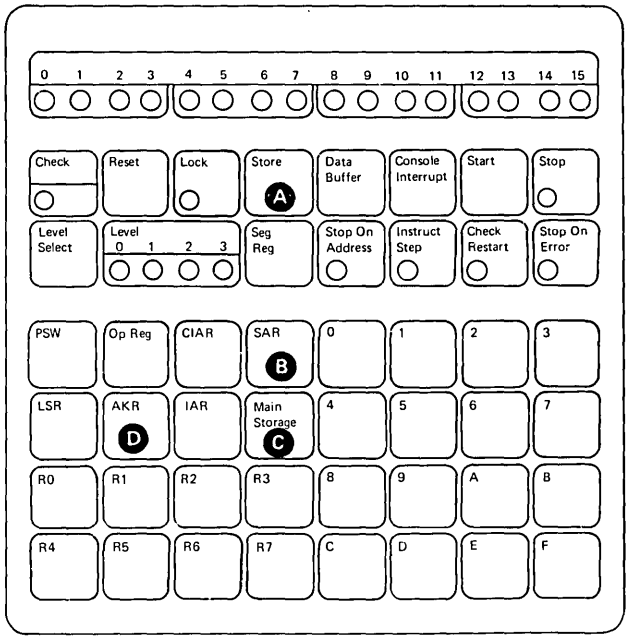
The processor must be in stop state.

If the storage address relocation translator is enabled, start at step 1; otherwise, start at step 5.

> **Note:** If steps 1 through 4 of the procedure are used, it is assumed that the operator has a thorough knowledge of the relocation translator and the storage mapping assigned by the program.

1. Press the SAR key (B). The contents of SAR are displayed in the data display indicators.

2. Press the AKR key (D). The contents of the SAR AKR are displayed in the data display indicators.



3. Key in one hex character (value of 0 through 7, which is the new address key). This character is displayed in bits 13–15 of the data display indicators.

4. Press the Store key (A). The new address key is stored into the SAR AKR.

5. Press the SAR key (B). The contents of the SAR are displayed in the data display indicators.

6. Key in the selected address (four hex characters). This address is displayed in the data display indicators.

7. Press the Store key (A). The address that is displayed is stored into the SAR.

8. Press the Main Storage key (C). The contents of the addressed storage location are displayed in the data display indicators and SAR is incremented by 2. Each time the Main Storage key is pressed, the location addressed by SAR is displayed in the data display indicators and then SAR is incremented by 2.

Notes:

1. If an invalid storage address occurs:

   a. The program check is suppressed.

   b. PSW bit 1 is set to 1.

   c. The Check indicator is turned on.

   d. PSW bit 1 set does not cause a class interrupt to occur upon entering the run state (unless the check indicator is not reset). The bit is only an indication, to the operator, of an error while displaying main storage.

2. If a storage location with bad storage parity occurs:

   a. The program check is suppressed.

   b. PSW bit 8 is set to 1.

   c. The Check indicator is turned on.

   d. PSW bit 8 set does not cause a class interrupt to occur upon entering the run state (unless the check indicator is not reset). The bit is only an indication, to the operator, of an error while displaying main storage.

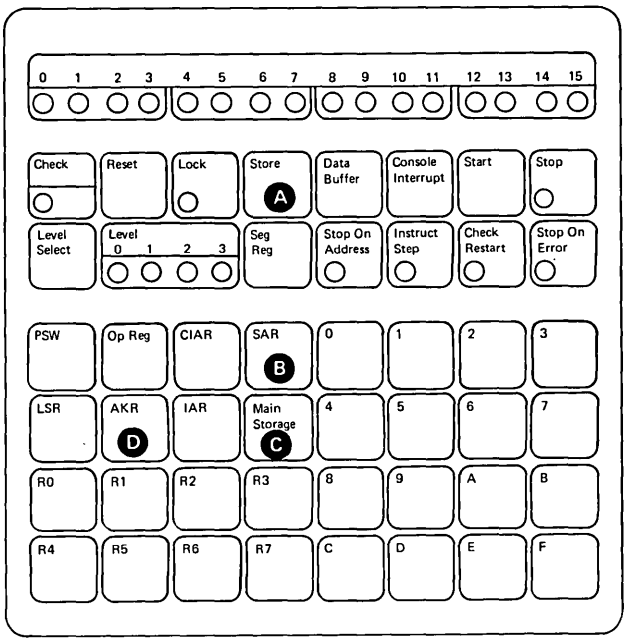
Storing Into Main Storage

The processor must be in stop state.

If the storage address relocation translator is enabled, start at step 1; otherwise, start at step 5.

> **Note:** If steps 1 through 4 of the procedure are used, it is assumed that the operator has a thorough knowledge of the relocation translator and the storage mapping assigned by the program.

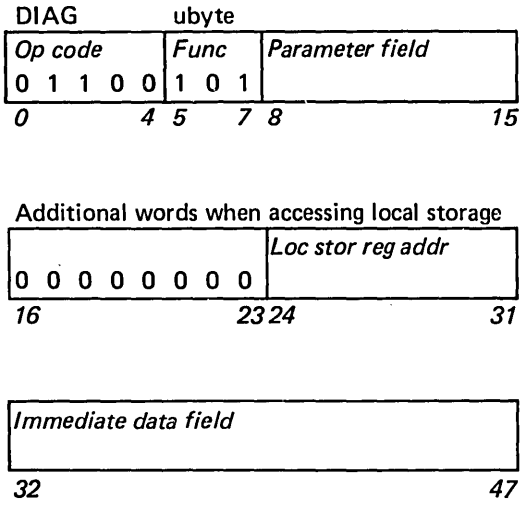
1. Press the SAR key (B). The contents of SAR are displayed in the data display indicators.
2. Press the AKR key (D). The contents of the SAR AKR are displayed in the data display indicators.



3. Key in one hex character (a value of 0–7 which is the new address key). This character is displayed in bits 13–15 of the data display indicators.
4. Press the Store key (A). The new address key is stored into the SAR AKR.
5. Press the SAR key (B). The current contents of the SAR are displayed in the data display indicators.
6. Key in the selected address (four hex characters). The address is displayed in the data display indicators.
7. Press the Store key (A). The address displayed in the data display indicators is stored into the SAR.
8. Press the Main Storage key (C). The contents of the addressed storage location are displayed in the data display indicators.
9. Key in the data that is to be stored into main storage. This data is displayed in the data display indicators.
10. Press the Store key (A). The data that is displayed is stored at the selected storage location and SAR is incremented by 2. Repeat steps 9 and 10 to store in sequential storage word addresses, or repeat steps 8, 9, and 10 if sequential storage words are to be displayed before alteration.

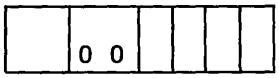
Chapter 4. Diagnose (DIAG) Instruction

The DIAG instruction is used for controlling or testing various hardware functions.



The parameter field is used to define and select the functions of the DIAG instruction. The bits in the parameter field are as follows:

Parameter field



Note: Bits 10 and 11 must always be set to 0's.

Bits	Value	Function
8–9	00	Storage select (word)
	01	Storage select (byte/error correction code bits)
	10	Local storage register select
	11	Channel select
10	0	Not used (must be set to 0)
11	0	Not used (must be set to 0)
12	0	Storage-to-register data transfer
	1	Register-to-storage data transfer
13	0	Enable all other parameter bit functions
	1	Set system ID (all other parameter bit functions disabled)
14	0	Disable (Error correction code, error log, channel-interrupt requests, and channel cycle-steal requests)
	1	Enable (Error correction code, error log, channel-interrupt requests, and channel cycle-steal requests)
15	0	Enable all other parameter bit functions
	1	Error log select (storage select, local storage register select, and channel select functions disabled)

Storage Select

The storage select function provides for testing of the error correction code (ECC) generation, single error correction, and double error detection on the storage card.

> **Note:** During a write storage cycle, ECC generates six code bits for the 16-bit data word written (for byte writes, a read must first occur to form the 16-bit data word) thus creating a 22-bit word in storage. These code bits provide for the single error correction/ double error detection capability on the read storage cycle. When a double error in storage is detected on a processor read, a machine check interrupt occurs with PSW bit 8 set to 1 (storage parity error).

Storage select can be either by word or by byte/ECC code bits. Parameter-field bits that are common to word or byte/ECC code bits selection are described in the following:

Bit 12: Specifies the direction of the data transfer.

- Bit 12=0. Transfer is from main storage to a register (read storage).
- Bit 12=1. Transfer is from a register to main storage (write storage).

Bit 13: Must be set to 0.

Bit 14: Specifies ECC generation, single error correction, and double error detection.

- Bit 14=0. ECC is not generated, single errors are not corrected, and double errors are not detected.
- Bit 14=1. ECC is generated, single errors are corrected, and double errors are detected.

Bit 15: Must be set to 0.

Storage Select Word

Parameter-field bits 8 and 9 are set to 0's.

The storage address for this data transfer cycle is contained in register 7 of the current priority level; the data is contained in register 0 of the current priority level. Two bytes of data are transferred.

Bit 14: Disable/enable.

- Bit 14=0. Disable.

  Read storage — Single errors are not corrected, and double errors are not detected.

  Write storage — ECC is not generated. Only the 16-bit data word is updated in storage; the six code bits in the 22-bit word remain unchanged.

- Bit 14=1. Enable.

Read storage — Single errors are corrected, and double errors are detected.

Write storage — ECC is generated.

When bit 14=1 (enable), a normal read or write word (bit 12 set to 0 or 1) occurs in storage.

Storage Select Byte/ECC Code Bits

Parameter-field bit 8 is set to 0 and bit 9 is set to 1.

The storage address for this data transfer cycle is contained in register 7 of the current priority level.

Bit 14: Disable/enable.

- Bit 14=0. Disable.

  Read storage — The six code bits are transferred from the 22-bit storage word to bits 10–15 of current priority level register 0. Bits 0–9 of register 0 are set to 0's. Single errors are not corrected, and double errors are not detected.

  Write storage — Bits 10–15 of current priority level register 0 are transferred to the six code bit positions of the word in storage. The 16 data data bits in the 22-bit word remain unchanged. Bits 0–9 of register 0 are ignored.

- Bit 14=1. Enable.

  Read storage — The storage data byte is transferred from storage into bits 8–15 of current priority level register 0. Bits 0–7 of register 0 are set to 0's. Single errors are corrected, and double errors are detected.

  Write storage — The data byte in bits 8–15 of current priority level register 0 is transferred to storage. Bits 0–7 of register 0 are ignored. ECC is generated.

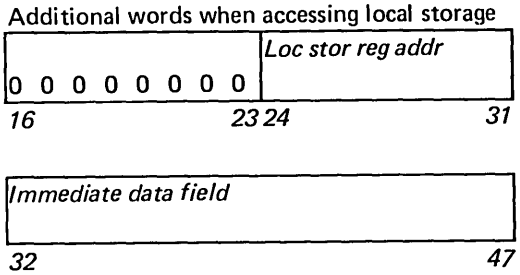
When bit 14=1 (enable), a normal read or write byte (bit 12 set to 0 or 1) occurs in storage.

Local Storage Register Select

The local storage register select function permits the transfer of data between main storage and any local storage register. The recommended use of this function is to read the error log. (See Appendix C.) To select this function, parameter-field bit 8 is set to 1 and bit 9 is set to 0.

> **Note:** The processor uses the local storage registers to store various machine parameters. Changing these parameters is not recommended because the results cannot be predicted.

The DIAG instruction has two additional words appended when this function is specified.



The bits in the two additional words are defined as follows:

Bits	Significance
16–23	Not used (must be set to 0's)
24–31	Local storage register address (00–FF hex)
32–47	Immediate data to be transferred

Parameter-field bits that are used with this function are defined as follows:

Bit 12: Specifies the direction of the data transfer.

- Bit 12=0. Transfer is from the immediate data field to the specified local storage register.
- Bit 12=1. Transfer is from the specified local storage register to the immediate data field.

Bit 13: Must be set to 0.

Bit 15: Must be set to 0.

Channel Select

The channel select function is determined by parameter field bits 8 and 9 being set to 1's.

This function, with bit 14 of the parameter field set to 0 (disable), inhibits and logically isolates I/O interrupts and cycle-steal operations. With bit 14 of the parameter field set to 1 (enable), this function allows I/O interrupts under the control of the summary mask, and cycle-steal operations are enabled.

Parameter-field bit functions are defined as follows:

Bit 13: Must be set to 0.

Bit 14: Specifies whether channel priority interrupts and cycle-steal requests are disabled or enabled.

- Bit 14=0. Channel priority interrupts and cycle-steal requests are disabled.
- Bit 14=1. Channel priority interrupts and cycle-steal requests are enabled.

Bit 15: Must be set to 0.

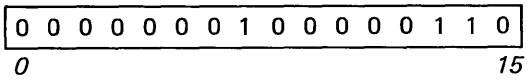
> **Note:** Pressing the Start button while in the stop state or executing any instruction that causes a level status block to be loaded (LEX instruction, SELB instruction, class interrupt, etc.) returns priority interrupt masking to program control. Also, the following operations cause cycle-stealing to be resumed:
>
> - Setting or reseting Stop on Address mode.
> - Performing the EN, DIS, SESR, or CPSR instruction.

Set System ID

The set system ID function is selected by parameter field bit 13 being set to 1.

This function sets the system ID into register 0 of the current priority level. The system ID for the 4956 is hex 0106.

Register 0 is set as follows:



> **Note:** When this function is selected, all other parameter-field functions are ignored.

Error Log Select

The error log select function provides a control to prevent logging of errors generated by diagnostic tests. To select this function, parameter-field bit 15 must be set to 1.

Parameter-field bit functions are defined as follows:

Bit 13: Must be set to 0.

Bit 14: Enables or disables error logging.

- Bit 14=0. Error logging is disabled.
- Bit 14=1. Error logging is enabled.

Bit 15: Must be set to 1.

> **Note:** If error log select is specified, storage select, local storage register select, and channel select functions are disabled.

Indicators

Indicators are not changed by the DIAG instruction; however, the data in local storage registers may be changed. Refer to "Local Storage Register Select" in this section.

Program-Check Condition

The DIAG instruction is a privileged instruction. If this instruction is encountered during the problem state, the instruction is suppressed, a program-check interrupt occurs, and the privilege violate bit (bit 2) in the PSW is set to 1.

Chapter 5. Diskette Data Format

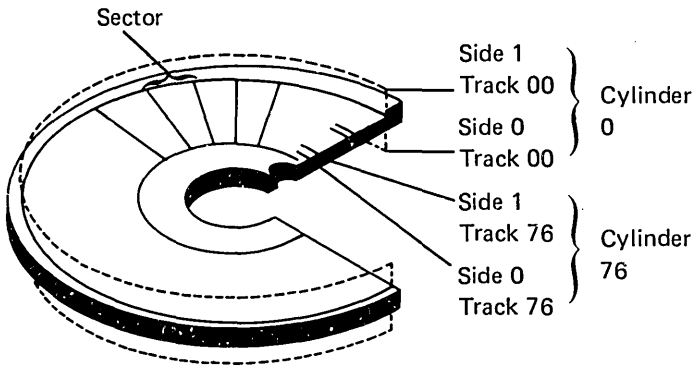
This chapter describes how data and control information are formatted on the diskette.

Depending on the type of diskette (1, 2, or 2D), one or both sides of the diskette are formatted with tracks that are divided into sectors.

Diskette 1 contains 77 tracks on the head-0 side of the diskette; the head-1 side is not used. Diskettes 2 and 2D contain 77 tracks on each side of the diskette.

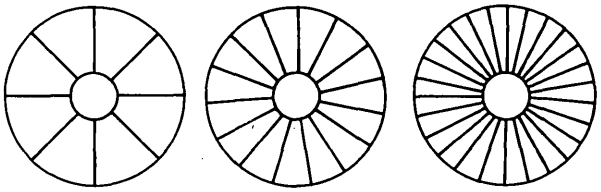
Track 00 is used for the system as a label track. Tracks 01 through 74 are used for data. Tracks 75 and 76 are used as alternate tracks. These two tracks are used as replacements for tracks that are defective.

When one-sided diskettes (diskette 1's) are used, the most data that can be read or written on a track without moving the read/write head is that track over which the read/write head is positioned. When two-sided diskettes (diskette 2's or 2D's) are used, the most data that can be read or written on two tracks (one for each side of the diskette) without moving the read/write head assembly, are the two tracks over which the read/write heads are positioned. When two tracks are used, the total amount of area that can be accessed is referred to as a cylinder.



Every track on the diskette is divided into sectors. The number of sectors depends on the recording density and the length of the sectors.

Number of sectors	Bytes per sector single density	Bytes per sector double density
8	512	1024
15	256	512
26	128	256



8 sectors 15 sectors 26 sectors

Sector numbers are assigned beginning with 1 for the first sector after the index. Each sector contains a sector identification (ID) field and a corresponding data field.

Track Format

The sectors on a track are numbered sequentially, starting with sector 1, following the index hole. The sectors are separated from each other and from the index hole pulse by various gaps that help to establish synchronization. (See Figure 5-1.)

Within each sector, the sector identification (ID) field and the sector data field are also separated by gaps for synchronization.

Index

An index pulse occurs each time the index hole in the diskette passes the light-emitting diode/phototransistor (LED/PTX) of the diskette drive. The index pulse indicates to the adapter that sector 1 of that particular track is the next sector to reach the read/write heads.

Gap 1

Gap 1 is the post-index gap required to accommodate index pulse timing variations. This gap contains 73 bytes of hex FF (all 1's) in FM format and 146 bytes of hex 4E (0100 1110) in MFM format.

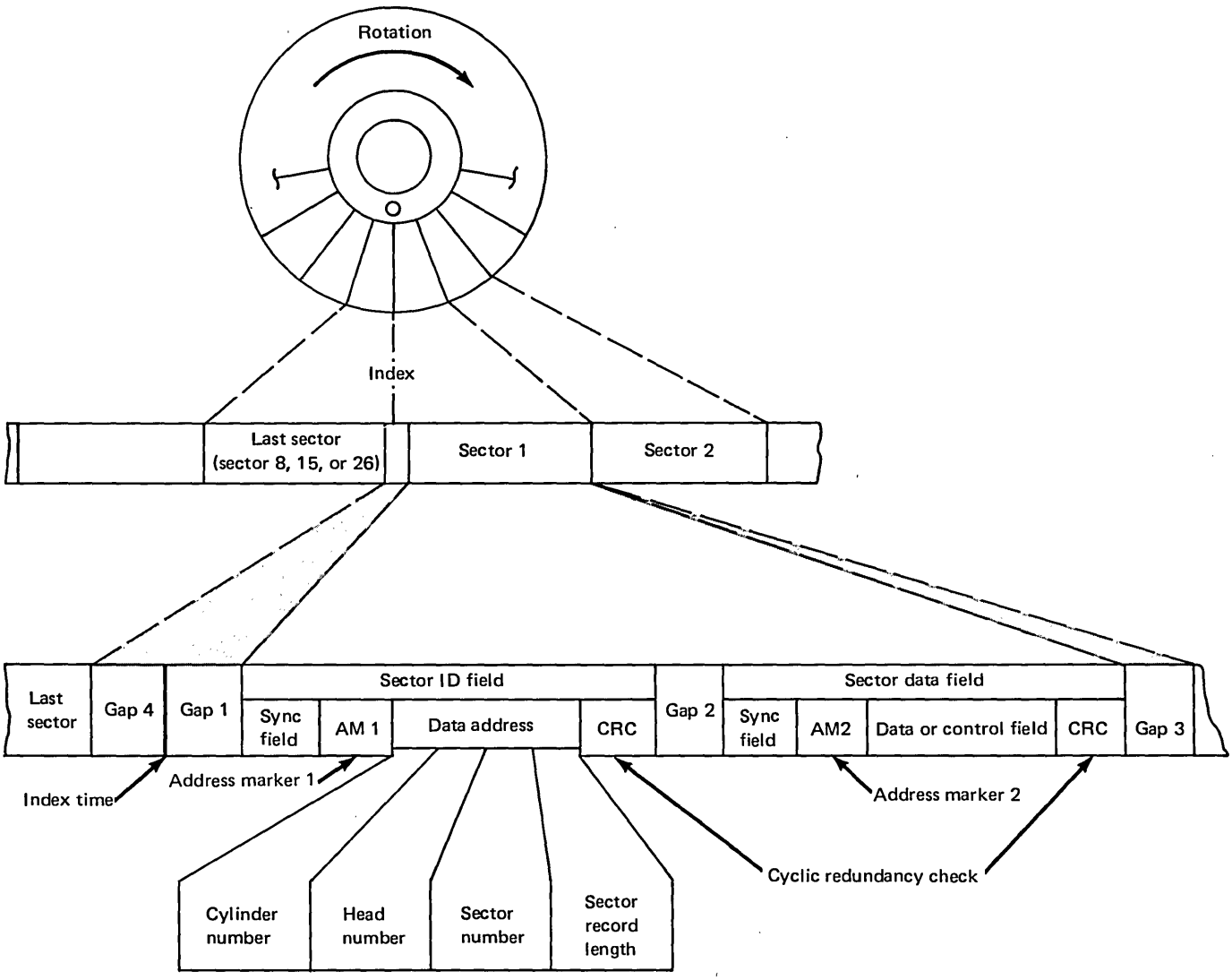
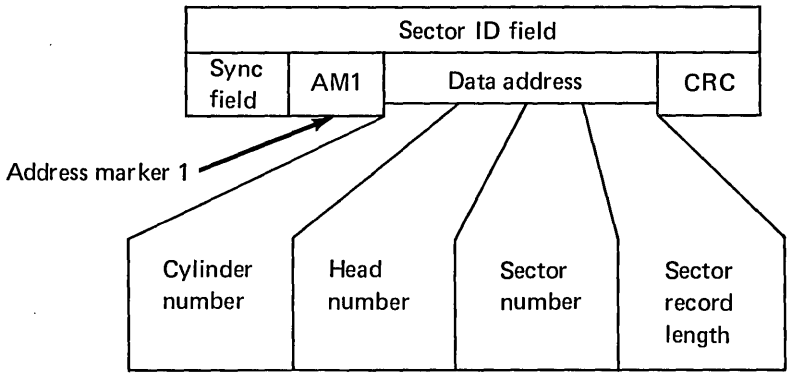


Figure 5-1. Diskette Track Format

Sector ID Field

The sector ID field contains data address information and the related synchronization and check information. This field consists of:

- A synchronization (sync) field
- Address mark 1 (AM1)
- Data address information
- Cyclic redundancy check (CRC)



Sync Field: This field is used to synchronize electronic circuitry with information being read from the diskette. In FM mode, this sync field contains six bytes of hex 00 (all 0's); in MFM mode, this field contains 12 bytes of hex 00.

Address Mark 1 (AM1): Address mark 1 (AM1) signals that the next field is a data address. In FM format, AM1 contains one byte that is always hex FE (1111 1110). In MFM format, this field contains four bytes that are always hex A1A1A1FE.

Data Address: The data address of each sector on the diskette identifies that sector's location with the following four bytes of information:

Cylinder Number: This byte contains a right-adjusted binary number equivalent to a decimal value of from 0 to 76. This number indicates which cylinder the sector is located on.

Head Number: This byte identifies which diskette surface the sector is on (side 0 or side 1).

Sector Number: This byte identifies the sector number. It contains a right-adjusted binary number that represents a decimal value of the sector number.

Sector Record Length: This byte contains a binary number equivalent to a hex value of from 00 to 03 to indicate the maximum number of data or control bytes in the sector data field.

Hex value	Bytes per sector
00	128
01	256
02	512
03	1024

Note: For a defective track, the cylinder number, the head number, the sector number, and the sector length will contain four bytes of hex FF.

Cyclic Redundancy Check: A two-byte cyclic redundancy check (CRC) is calculated as each sector ID field is written. The CRC bytes are written immediately after the sector ID field. The CRC bytes are recalculated each time information is read from the diskette and compared with the previously written CRC bytes. An unequal comparison indicates an error.

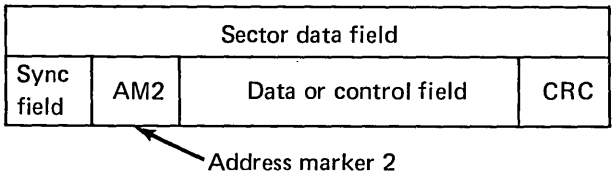
Gap 2

Gap 2 is the post-ID gap required to accommodate erase head delay and rotational tolerance. This gap contains 11 bytes of hex FF (all 1's) in FM format and 22 bytes of hex 4E (0100 1110) in MFM format.

Sector Data Field

The sector data field is the area into which data or control information, and the related synchronization and check information, is written and from which it is read. This field consists of:

- A synchronization (sync) field
- Address mark 2 (AM2)
- Data or control information
- Cyclic redundancy check (CRC)



Sync Field: This field is used to synchronize electronic circuitry with information being read from the diskette. In FM mode, this sync field contains six bytes of hex 00 (all 0's); in MFM mode, this field contains 12 bytes of hex 00.

Address Mark 2 (AM2): Address mark 2 (AM2) identifies the following field as either a data field or a control field.

Type of field indicated	FM format (1 byte)	MFM format (4 bytes)
Data	FB	A1A1A1FB
Control	F8	A1A1A1F8

Data or Control Field: If data is specified by address mark 2, this field contains either 128, 256, 512, or 1024 bytes of data, depending on the sector record length and the recording density.

If control information is specified, the control address mark flag (the first byte in this field) contains one of the following:

- A hex C4, which indicates that the data in the sector has been logically deleted. Sector access depends upon the setting of bits 0 and 1 of the control address marker mask. See "DCB Word 1 (Parameter Word 1)" under "Start" in Chapter 6.

- A hex C6, which indicates that the sector data surface is defective and the data can be found in the next physical sector. Sector access depends upon the setting of bits 0 and 1 of the control address marker mask. See "DCB Word 1 (Parameter Word 1)" under "Start" in Chapter 6.

- A hex 4B, which indicates that the sector data surface is defective and that data is relocated to the sector designated by the program.

Cyclic Redundancy Check (CRC): A two-byte cyclic redundancy check (CRC) is calculated as each sector data field is written. The CRC bytes are written immediately after the sector data field. The CRC bytes are recalculated each time information is read from the diskette and compared with previously written CRC bytes. An unequal comparison indicates an error.

Gap 3

Gap 3 separates one sector from another. The length of this gap depends on the record length and recording format, as shown in the following table:

Record format	Record length	Gap size
FM	128 bytes	27 bytes
FM	256 bytes	42 bytes
FM	512 bytes	58 bytes
MFM	256 bytes	54 bytes
MFM	512 bytes	84 bytes
MFM	1024 bytes	116 bytes

Gap 4

Gap 4 occurs after the last sector of a track and separates that sector from the index pulse. This gap contains a variable number of bytes of hex FF in FM format and a variable number of bytes of hex 4E in MFM format. The actual number of bytes depends on the speed of the diskette. The length of this field is variable to allow interchangeability of diskettes between diskette drives.

Diskette Labels

Labels may be recorded on every diskette to provide identification and recording format information. These labels are implemented by programming.

Chapter 6. Diskette Drive Input/Output Operations

This chapter describes the operations of the diskette drive(s) internal to the IBM Series/1 4956 Model Processor C. It includes descriptions of the Operate I/O instruction and its associated commands, status words, and condition codes.

The processor initiates all diskette drive operations by issuing an Operate I/O instruction, and then uses the processor data channel to transfer data to and from the diskette attachment card.

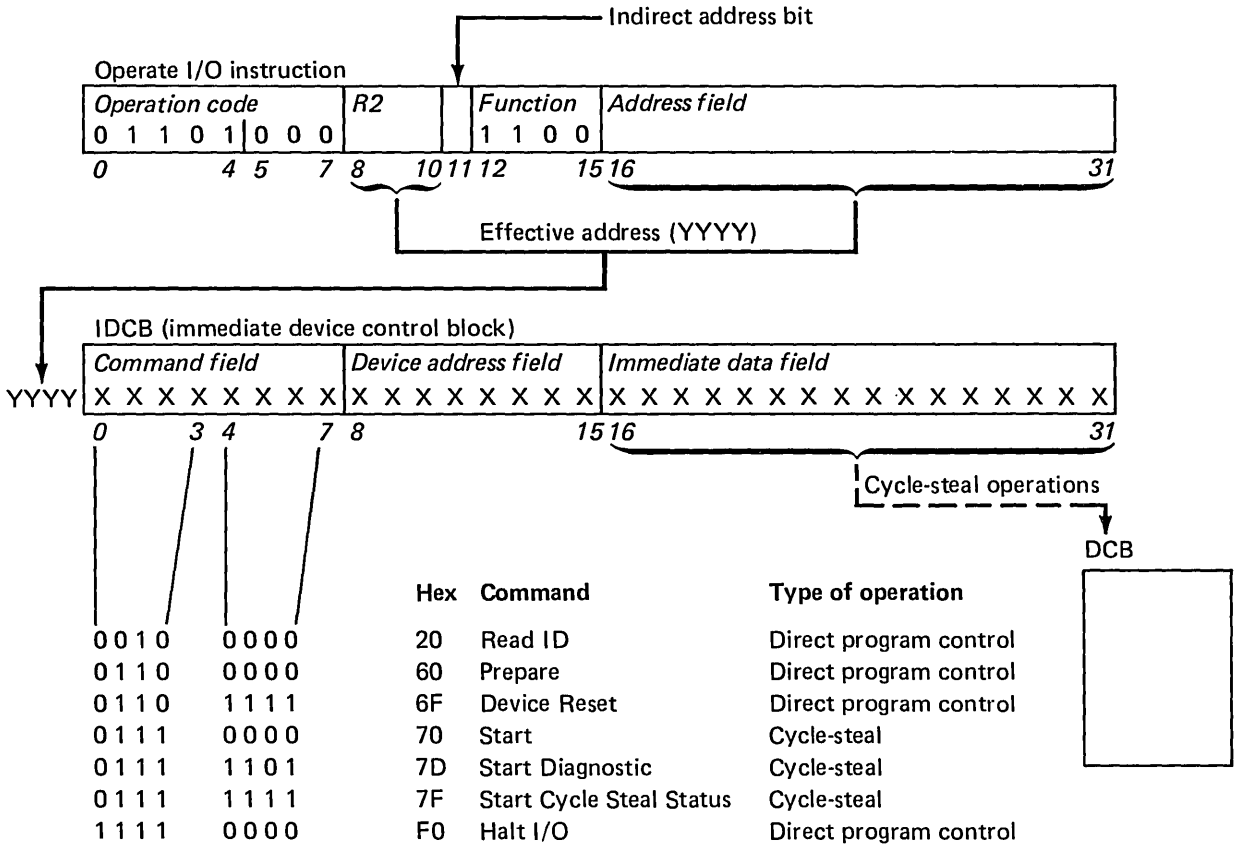


Figure 6-1. Operate I/O Instruction

The Operate I/O instruction is a privileged instruction. Its effective address (the combination of the R2 and address fields) points to an immediate device control block (IDCB) in processor storage. The IDCB contains a command, a device address, and an immediate data field (see Figure 6-1). The command defines the type of I/O operation; the device address identifies the device on which the operation is to be performed. The use of the information in the immediate data field depends on the mode of operation. For direct program control (DPC) operations, the immediate data field contains a data word; for cycle-steal operations, this field points to a device control block (DCB) that contains additional information needed to perform the operation.

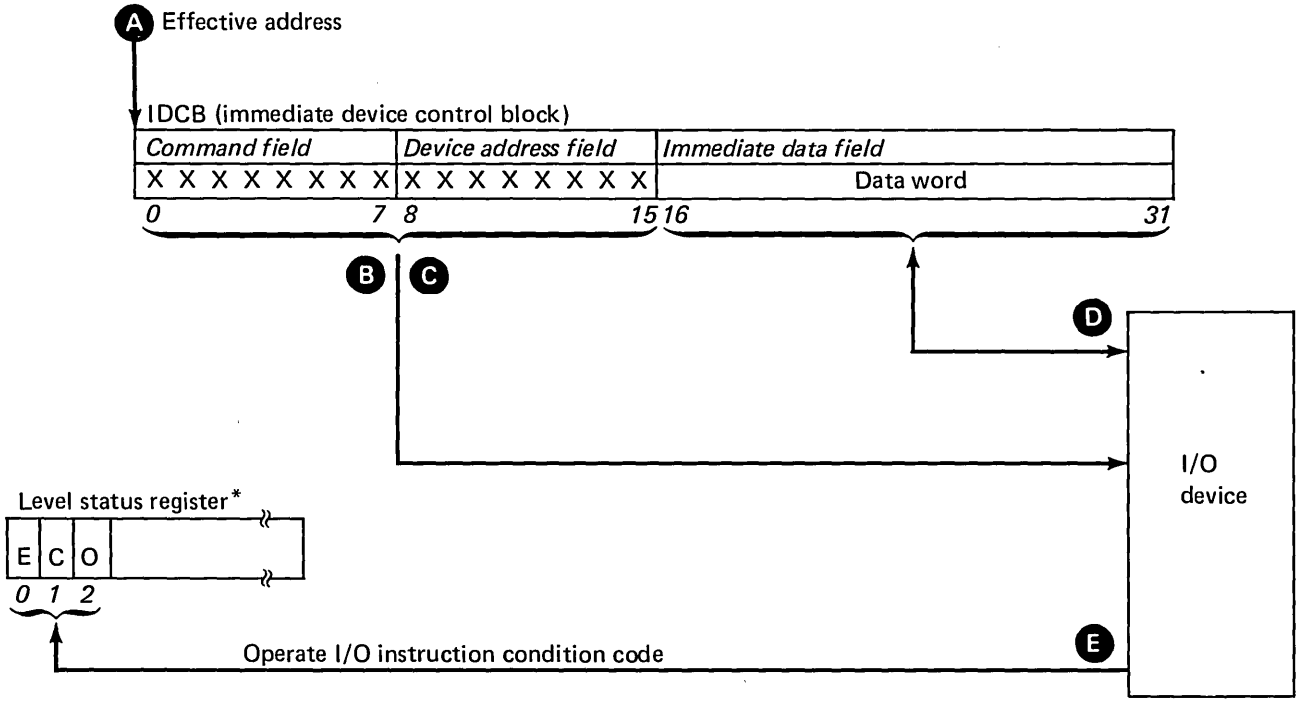
The IDCB must be on a fullword boundary. For a more detailed description, refer to *IBM Series/1 Principles of Operation*, GA34-0152.

Direct Program Control (DPC)

A DPC operation causes an immediate transfer of data or control information to or from the diskette drive.

An Operate I/O instruction must be executed for each data transfer. Each execution of this instruction causes the following events (refer to Figure 6-2):

1. The Operate I/O instruction's effective address points the program to an IDCB in processor storage (A).
2. The data channel uses the IDCB's command field (B) to determine the operation to perform and the device address field (C) to select the diskette drive.
3. The processor transfers the contents of the immediate data field to the diskette drive, or transfers information from the diskette drive to the immediate data field, depending on the command being executed (D).
4. The diskette drive sends a condition code to the level status register (LSR) in the processor (E). Condition codes are explained under "Condition Codes" in this chapter.



***Level status register**

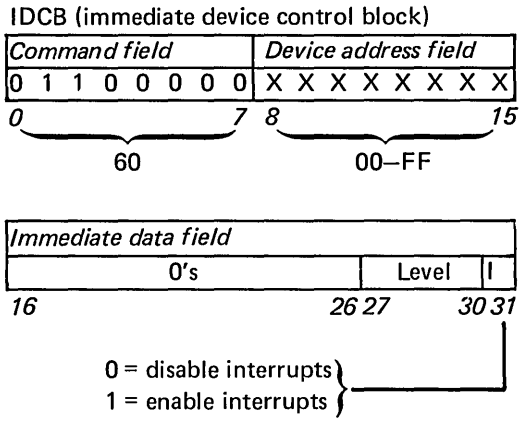
Bit 0	even indicator
Bit 1	carry indicator
Bit 2	overflow indicator

Figure 6-2. Operate I/O Instruction Execution

The following commands cause diskette drive DPC operations:

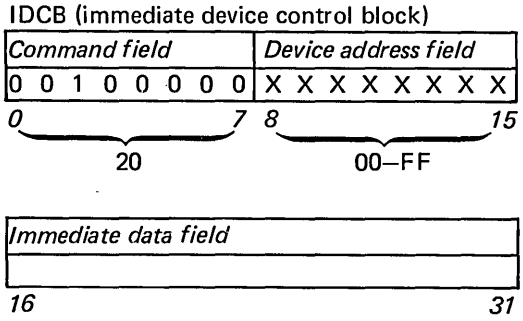
- Prepare
- Read ID
- Device Reset
- Halt I/O

Prepare

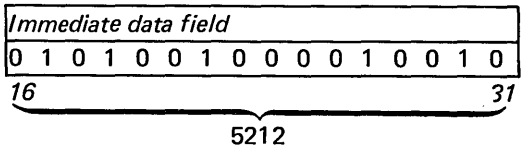


The Prepare command loads the interrupt level and I-bit into the diskette drive. The I-bit (bit 31) defines whether the diskette drive can present I/O interrupt requests to the processor. If the I-bit is a 1, requests are presented on the level defined by the level field (bits 27–30); if the I-bit is a 0, the diskette drive cannot present interrupt requests.

Read ID



The Read ID command transfers the identification (ID) word for the diskette attachment card into the immediate data field of the IDCB. After command execution, the immediate data field contains hex 5212, the device ID for the diskette drive.



Device Reset

IDCB (immediate device control block)

Command field	*Device address field*
0 1 1 0 1 1 1 1	X X X X X X X X
0 *7*	*8* *15*
6F	00–FF

Immediate data field
0's
16 *31*

The Device Reset command resets any pending interrupt requests or busy conditions in the diskette drive. The prepare level register and the residual address register are not affected. The immediate data field is not used.

Halt I/O

IDCB (immediate device control block)

Command field	*Device address field*
1 1 1 1 0 0 0 0	
0 *7* *8*	*15*
F0	

Immediate data field
0's
16 *31*

The Halt I/O command is a channel-directed command that halts all I/O activity on the data channel and resets all devices. The IDCB's immediate data field is not used. Any pending interrupt request or busy condition is reset. The prepare level register and the residual address register are not affected.

Cycle-Steal

Cycle-steal mode permits overlapping an I/O operation with processor operations and other I/O operations (see Figure 6-3). The processor transfers the IDCB, under direct program control, from processor storage to the diskette drive Ⓐ. After the diskette drive accepts the IDCB, it sends a condition code back to the processor Ⓑ. The processor is now free to continue with other operations while the diskette drive uses the information in the IDCB to execute the command. The IDCB's immediate data field contains the address of a DCB. This eight-word DCB contains parameters that define and control the I/O operation. The diskette drive cycle-steals the DCB words Ⓒ and data Ⓓ it needs to perform the operation. When the number of bytes specified in DCB word 6 has been transferred, an interrupt request is sent to the processor. The processor then accepts the interrupt condition code and an interrupt ID word from the diskette drive.

The following commands cause cycle-steal operations:

- Start
- Start Cycle Steal Status
- Start Cycle Steal Diagnostic

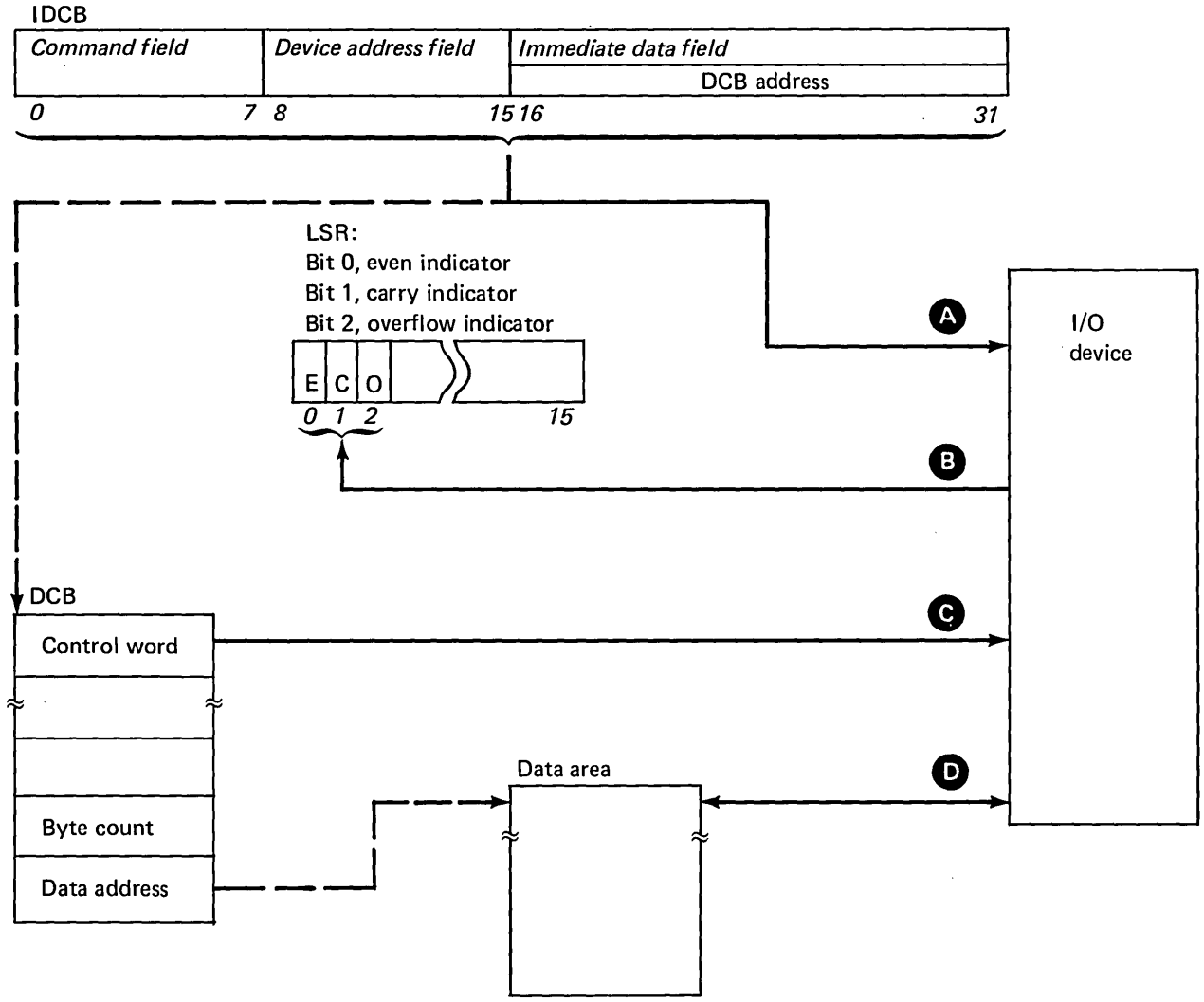
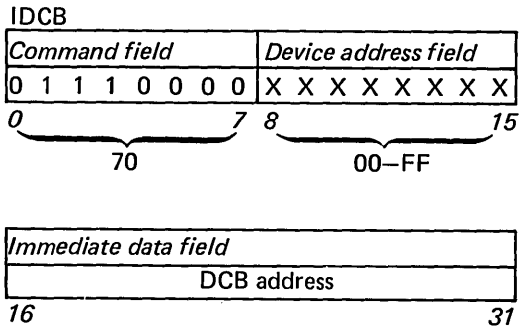


Figure 6-3. Cycle-Steal Operation

Start



The Start command initiates diskette drive I/O operations that transfer data to or from processor storage in cycle-steal mode. An interrupt request is sent to the processor when the I/O operation ends. The control information and parameters required for a particular diskette drive operation must be stored in the DCB associated with that operation.

The eight words in the DCB and their bit configurations are illustrated in Figure 6-4, and explained in the following descriptions. (The Start command operations are described after the DCB word descriptions.)

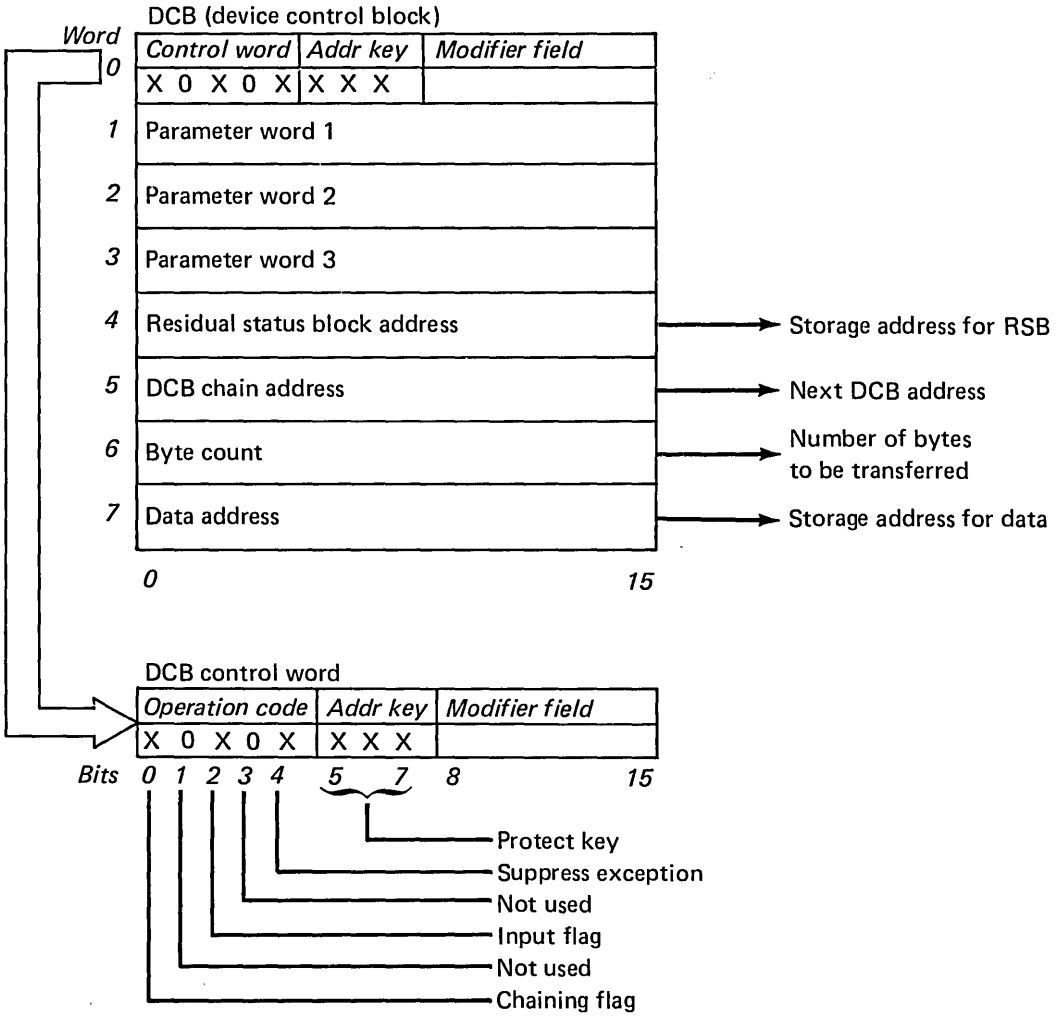


Figure 6-4. Device Control Block

DCB Word 0 (Control Word)

DCB word 0 is a 16-bit word that defines the cycle-stealing operation. This word contains two bytes of control parameters to be used with the particular Start command to be performed.

Bit 0 **Chaining flag.** When this bit is a 1 and the operation can be chained, the diskette drive performs a chaining operation. Chaining means that the diskette drive completes the current operation but does not present an interrupt request to the processor. Instead, the diskette drive fetches the next DCB in the chain and performs the next operation. DCB word 5 indicates the location of the next DCB. Chaining continues until a DCB that has the chaining flag in the control word (DCB word 0) set to 0 is fetched, thus indicating the last operation in the chain. If an error occurs, chaining to succeeding DCBs is automatically suspended, and an interrupt request is sent to the processor. Normally, an interrupt is not sent until the diskette drive has completed the last operation in the chain. DCB chaining for the diskette drive is valid only for a Start command.

If the suppress exception bit (bit 4) is a 1, each DCB in the chain stores an RSB to report any soft error retries.

Bit 1 This bit is not used and must be 0.

Bit 2 **Input flag.** This bit indicates to the diskette drive which direction the data is to be transferred. When this bit is a 1, the diskette drive transfers data to processor storage. When this bit is a 0, the data transfer is from processor storage to the diskette drive. This bit is also 0 for a command such as seek or format where no data is transferred.

The input flag value for each operation is given in the Input Flag/Modifier Field Table in this chapter.

Input Flag/Modifier Field Table

Operation	Input (bit 2)	Modifier field (bits 8–15)
Seek	0	0000 0000
Recalibrate head	0	0000 0010
Format track	0	0000 X100
Format track defective	0	0000 X101
Set FM/MFM bit	0	0000 0111
Verify format track/compare data	0	0000 X110
Read data	1	S001 X000
Read verify/cyclic redundancy check	0	S001 X001
Read verify/compare data	0	S001 X010
Read sector ID	1	0001 X100
Read diagnostic record	1	0001 X110
Write data/data address marker	0	S010 X000
Write data/control address marker	0	S010 X001
Write data with read verify	0	S010 X010
Read attachment storage	1	0101 0000
Write attachment storage	0	0110 0001

Notes:

1. The S in bit 8 denotes spiral read/write as an option and is explained in this chapter with the individual operations that support this option. The automatic seek option should be used with all spiral operations; otherwise, a no-record-found (NRF) error might be returned. Refer to "Automatic Seek Option" later in this chapter. (S=0 indicates no spiral; S=1 indicates spiral.)
2. The X in bit 12 denotes automatic seek as an option and is explained in this chapter with the individual operations that support this option. Refer to "Automatic Seek Option" later in this chapter. (X=0 indicates automatic seek; X=1 indicates suppress automatic seek.)
3. The seek operation should be used during the formatting of a new or unformatted diskette. For seek operations after formatting is complete, the automatic seek option should be specified. Refer to "Automatic Seek Option" later in this chapter.

Bit 3 This bit is not used and must be 0.

Bit 4 **Suppress exception.** When this bit is a 1:

- It suppresses the reporting of some exception conditions that otherwise would cause an exception interrupt.
- It allows certain diskette drive operations to be retried (see individual operations in this chapter).
- It stores the diskette drive status at the address specified by the residual status block address field of DCB word 4.

If a permanent error does not occur, the residual status block is available at the end of the operation that uses suppress exception. The format of the residual status block (RSB) is shown in the Residual Status Block Format Table.

If the suppress exception bit (bit 4) is a 1, each DCB in the chain stores an RSB to report any soft error retries.

Residual Status Block Format Table

Word 0—Residual count

This word is always 0's.

Word 1—Residual status block flags

This word contains the residual status block (RSB) flags. The RSB flags and their respective bit fields are:

Bit	*Meaning*
0	End of chain
1	Hardware retry
2–14	Reserved
15	No exception

Word 2—Retry counts word 1

This word contains temporary error retry counts. The error conditions and their respective bit fields are:

Bit	*Meaning*
0	Storage data check count
1	Invalid storage address count
2	Protect check error count
3	Interface data check
4–6	Seek errors retry count
7	Reserved
8–15	Control address markers encountered count

Word 3—Retry counts word 2

This word is an extension of the temporary error retry counts. The error conditions and their respective bit fields are:

Bit	*Meaning*
0	Reserved
1	Parity error count
2–7	Cyclic redundancy check error retry
8–9	Reserved
10–15	No record found error count

Word 4—Retry counts word 3

This word is an extension of the temporary error retry counts. The error conditions and their respective bit fields are:

Bit	*Meaning*
0,1	Reserved
2–7	No-data-found error count
8–9	Underrun/overrun error count
10–15	Equipment check error count

Word 5—Error status word

This word contains the temporary error retry bit:

Bit	*Meaning*
0–14	Reserved
15	Temporary error retry

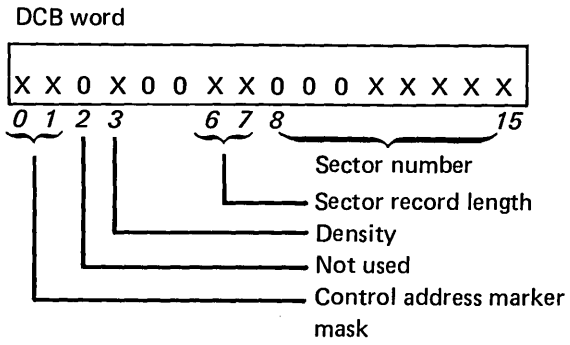
Bits 5–7 **Address key.** This is a three-bit key that the diskette drive presents, during data transfers, to verify that the program has authorization to access processor storage. An incorrect address key causes an exception interrupt request (condition code 2).

Bits 8–15 **Modifier field.** These bits are modifiers of the Start command. A bit configuration that represents the operation to be performed must be selected. The selected operation must be compatible with the setting of the input flag (bit 2).

The modifier field for each operation is given in the Input Flag/Modifier Field Table in this chapter.

DCB Word 1 (Parameter Word 1)

DCB word 1 (along with DCB word 2) provides information for most diskette operations. The information is used when seeking an ID on the diskette surface prior to the transfer of data to or from processor storage. The format of this word is as follows:



Bits 0,1 **Control address marker mask**. The combinations of these bits are as follows:

00 Complete the read of the control AM sector and terminate.

01 Continue the operation only if the control address marker flag indicates that the sector has been logically deleted; otherwise, complete the read of the control AM sector and terminate.

10 Continue the operation only if the control address marker flag indicates that the sector has been either logically deleted or relocated to the next sector; otherwise, complete the read of the the control AM sector and terminate.

11 Reserved.

Bit 2 Reserved; must be 0.

Bit 3 **Density values**. The meaning of this bit is:

0 Single density (FM)
1 Double density (MFM)

Bits 4, 5 Reserved; must be 0's.

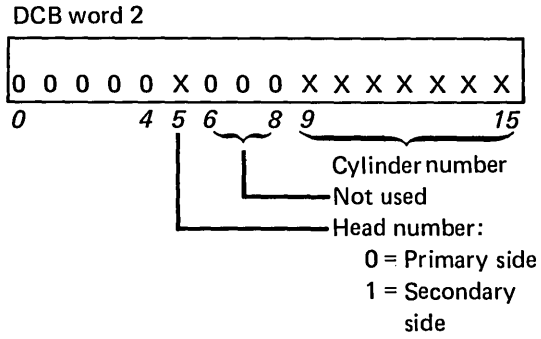
Bits 6, 7 **Sector record length values**. The combinations of these bits are as follows:

00 128 bytes per sector
01 256 bytes per sector
10 512 bytes per sector
11 1024 bytes per sector

Bits 8–15 **Sector number**. These bits specify the starting sector for the operation.

DCB Word 2 (Parameter Word 2)

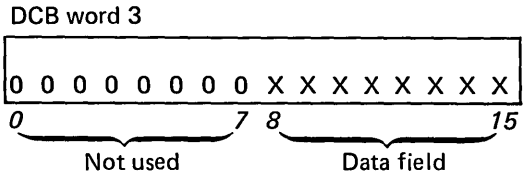
DCB word 2 provides information for all operations except read attachment storage, write attachment storage, and set FM/MFM. The information is used when searching for an ID on the diskette surface prior to the transfer of data to or from processor storage. The format of this word is as follows:



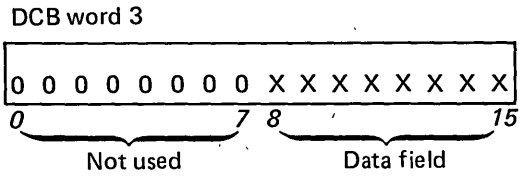
DCB Word 3 (Parameter Word 3)

DCB word 3 has different meanings depending upon the diskette drive operation to be performed.

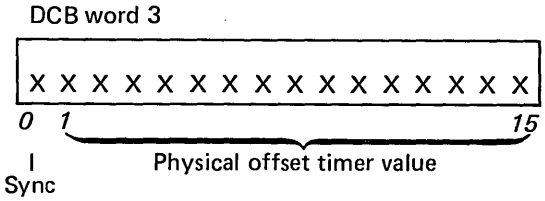
When the format track or format track defective operations are performed, bits 0–7 are not used. Bits 8–15 contain data fill characters for every sector to be written. A further explanation of these bits is under "Format Track" and "Format Track Defective" in this chapter.



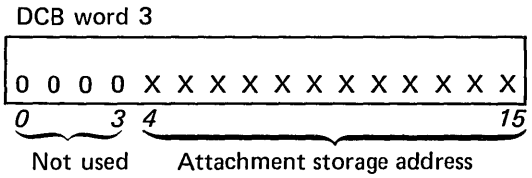
When the verify format track/compare data operation is performed, bits 0–7 are not used. Bits 8–15 contain data that is compared to the data being read. A further explanation of these bits is under "Verify Format Track/Data Compare" in this chapter.



When the read diagnostic record operation is performed, bit 0 is the sync field bit and bits 1–15 are the physical offset timer bits. A further explanation of these bits is under "Read Diagnostic Record" in this chapter.



When performing the read attachment storage or write attachment storage operations, bits 0–3 are not used. Bits 4–15 contain the diskette's attachment storage address. This address is used when tailoring diagnostic programs for the diskette drive. A further explanation of these bits is under "Read Attachment Storage" and "Write Attachment Storage" in this chapter.



DCB Word 4 (Residual Status Block Address)

The address contained in DCB word 4 points to the beginning of a processor storage area where the residual status block is stored. The residual status block is stored only when the suppress exception bit is a 1 and a permanent error did not occur. The address must be even (bit 15 is a 0), or a DCB specification check occurs.

DCB Word 5 (Chaining Address)

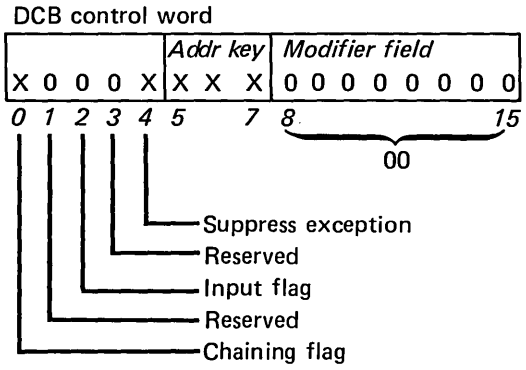
DCB word 5 is the location of the next DCB to be executed if chaining flag bit 0 of DCB word 0 is a 1. If the chaining address is odd, an interrupt request is posted and the DCB specification check bit (bit 3) in the ISB is set to 1. If a permanent error occurs, condition code 2 (exception) is reported and chaining stops.

DCB Word 6 (Byte Count)

DCB word 6 contains a 16-bit unsigned integer representing the number of data bytes to be transferred for the current DCB. If the byte count equals 0, no data is transferred. If the byte count is greater than the maximum allowed for a particular operation, or if the byte count is odd, the DCB specification check bit (bit 3) in the ISB is set to 1. When the interrupt request is accepted, condition code 2 (exception) is reported. For spiral read/write the maximum byte count is 64K bytes. No check is made to determine if the byte count is greater than the remaining bytes on the diskette. If all the data cannot be processed, end of track (bit 11 of cycle-steal status word 7) is reported.

DCB Word 7 (Data Address)

DCB word 7 contains the starting storage address for the data associated with the operation to be performed. If the starting address is odd, an interrupt request is posted and DCB specification check bit 3 in the ISB is set to 1. When the request is accepted, condition code 2 (exception) is reported.



The seek operation moves the data heads to the specified cylinder and selects the head designated by DCB word 2. The error-recovery procedure for this operation is determined by the suppress exception bit (bit 4) of DCB word 0. When the operation is complete, the status is stored at the location indicated by the residual status block address in DCB word 4. If an exception interrupt request occurs, status is available to a Start Cycle Steal Status command. If the suppress exception bit is a 0, retries are not attempted and any errors result in the end of the operation with status available to a Start Cycle Steal Status command.

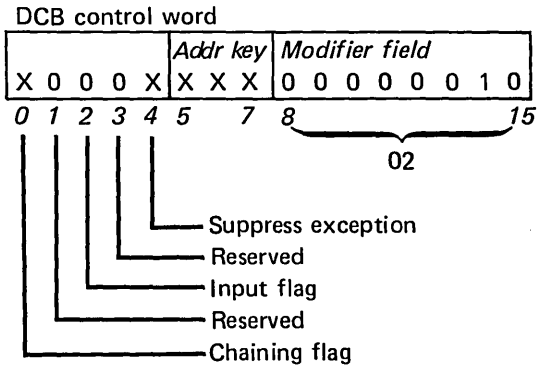
This operation should be used only during the formatting of a new or unformatted diskette. For seek operations after formatting is complete, the automatic seek option should be specified.

> **Note:** There is no automatic read sector ID operation associated with the seek operation.

Programming Considerations: After the seek operation, the processor program should perform the necessary checks to ensure that the correct track has been selected prior to altering its data. All tracks (usable and defective) are counted by the cylinder number circuits when a stand-alone seek operation is performed.

The following DCB fields must be specified: head number, cylinder number, residual status block (RSB) address (if the suppress exception bit is a 1), and chain address (when chaining).

Recalibrate Head



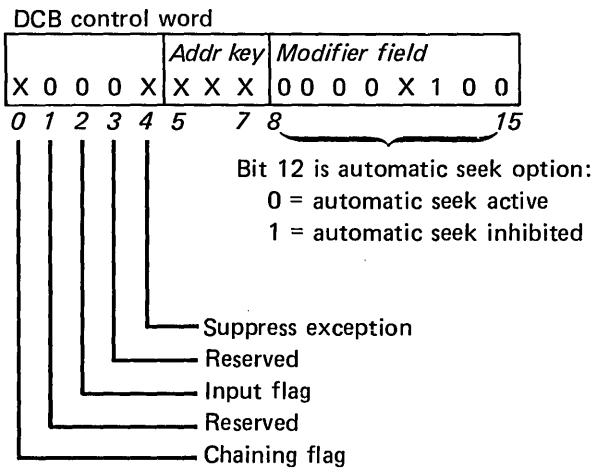
The recalibrate head operation causes the data heads to seek to track 0, head 0. When the operation is complete and the suppress exception bit (bit 4 of DCB word 0) is a 1, the residual status is stored at the location indicated by the residual status block address in DCB word 4. Any errors result in the end of the operation with an exception interrupt request. In that case, status is available to a Start Cycle Steal Status command.

There is no automatic Read Sector ID associated with this operation.

Programming Considerations: This operation is time consuming and not recommended for use other than in the error-recovery procedure or diskette initialization routines.

The following DCB fields must be specified: the RSB address (if the suppress exception bit is a 1) and the chain address (when chaining).

Format Track



The format track operation is used to initialize the track format on the diskette.

For a diskette 1 or 2, the tracks are initialized into either twenty-six 128-byte sectors, fifteen 256-byte sectors, or eight 512-byte sectors. On a diskette 2D, the tracks are initialized into either twenty-six 256-byte sectors, fifteen 512-byte sectors, or eight 1024-byte sectors.

The DCB for this operation contains the density, sector record length, head number, cylinder number, and residual status block address.

Bit 3 of DCB word 1 defines the recording density to be used. When this bit is a 0, single density is used. When this bit is a 1, double density is used.

Bits 6 and 7 of DCB word 1 specify the sector record length.

Bits 9–15 of DCB word 2 contain the cylinder number. The cylinder number must be a binary number from 0 to 76 (decimal). This number is written in the sector ID field of each sector formatted on the track.

Bits 8–15 of DCB word 3 contain the data fill character that is propagated through all data fields. Each sector data field is written with a data address marker.

When performing this operation, tracks are verified by reading the information and performing a cyclic redundancy check of each sector.

The error-recovery procedure for this operation is determined by the suppress exception bit (bit 4) of control word 0. When this bit is a 1 and an error occurs, the diskette drive retries the operation, depending on the type of error, before it posts a permanent error (see the **retries** for a write-type operation under "Error-Recovery Procedures" in this chapter). When the operation is complete, the status is stored at the location indicated by the residual status block address in DCB word 4. If an exception interrupt request occurs, status is available to a Start Cycle Steal Status command. If the suppress exception bit is a 0, no retries are attempted and any errors result in the end of the operation with status available to a Start Cycle Steal Status command.

This operation supports the automatic seek option (bit 12) in the DCB control word. When bit 12 is a 0, the cylinder number is checked against the current location and a seek operation is initiated, as required.

Programming Considerations: If a new or unformatted diskette is being formatted, the program must use a seek operation (no automatic Read ID occurs) to position the heads. A format track operation with DCB word 0, bit 12, set to 1, can then be issued. If the diskette has been previously formatted and has sector ID fields, then an automatic seek option can be specified to position the heads because the seek is checked using the sector ID fields. A Read Sector ID operation can be used to check the cylinder location of the heads before each format track operation is performed. If automatic seek is indicated and the diskette has been previously formatted, a Read Sector ID operation is performed automatically. The cylinder byte value (bits 9–15 of DCB word 2) must specify the address of the track to be formatted.

Byte 72 of the data information for cylinder 00, head 0, sector 7 of a diskette is the density **byte**. The density byte has the following values:

- Hex 40 for a diskette 1
- Hex F2 for a diskette 2
- Hex D4 for a diskette 2D

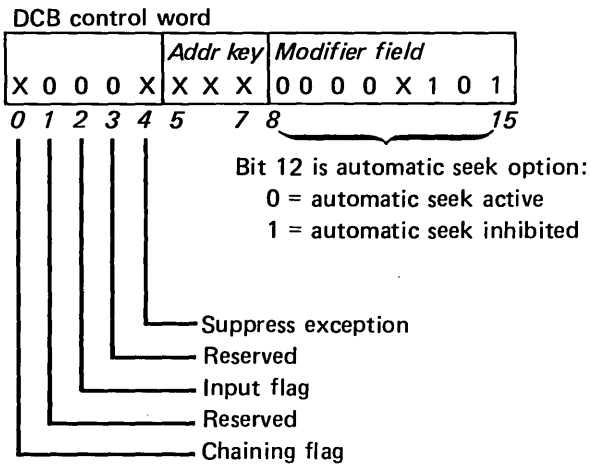
This byte should be set by the processor program to the proper value during diskette initialization. If this byte is not initialized and the diskette is removed from the diskette drive, or if there is a power failure, the density of a formatted diskette cannot be recognized.

Also, track 00 (cylinder 00, head 0) is always recorded in single density, twenty-six 128-byte sectors. When a new diskette is inserted, the density byte is read to determine the format of the diskette.

> **Note:** The above programming considerations also apply to the **format track defective** and the **verify format track/compare data** operations.

The following DCB fields must be specified: sector record length, density, head number, cylinder number, data fill character, RSB address (if the suppress exception bit equals 1), and chain address (when chaining).

Format Track Defective

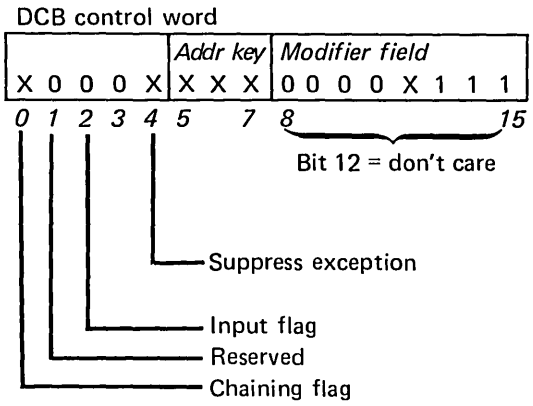


The format track defective operation is similar to the format track operation (see "Format Track," earlier in this chapter). Unlike the format track operation, no cyclic redundancy checks are performed on the sectors. Instead, the only check performed is to ensure that at least one sector ID on the track is readable to identify the track as defective. All bits of the ID for a defective track are set to 1's. If there are no readable sector IDs, bit 7 of cycle-steal status word 7 is set to 1. If a readable ID cannot be found, the diskette must be replaced.

This operation supports the automatic seek option (bit 12) in the DCB control word. When bit 12 is a 0, the cylinder number is checked against the current location and a seek operation is initiated, as required.

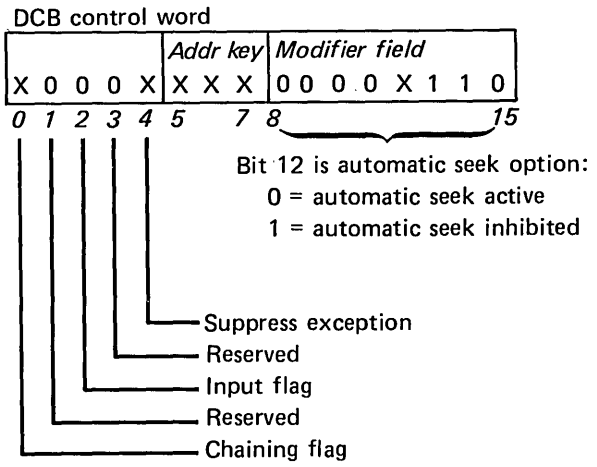
The following DCB fields must be specified: sector record length, density, head number, cylinder number, data fill character, RSB address (if the suppress exception bit is a 1), and chain address (when chaining).

Set FM/MFM Bit



The set FM/MFM bit operation can be used to set the internal FM/MFM bit as specified by bit 3 of DCB word 1 (FM if bit 3 is a 1, and MFM if bit 3 is a 0). Under normal operating conditions, this operation is not used.

Verify Format Track/Compare Data



The verify format track/compare data operation is used to validate the format and data written on diskette. It is similar to the read verify/compare data operation except that no data is transferred from processor storage while the operation is being performed. This operation:

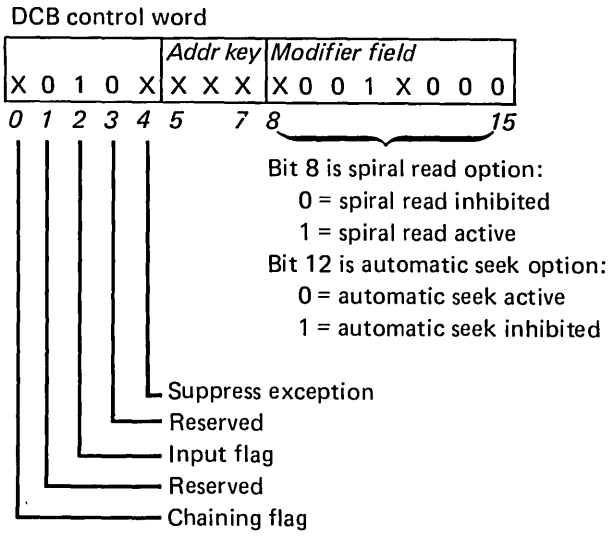
- Verifies the data and ID fields written on a track by the format track operation
- Compares the bytes of data in the data field of a selected track to the byte in bits 8–15 of DCB word 3

The error-recovery procedure for this operation is determined by the suppress exception bit (bit 4), DCB word 0. When this bit is a 1 and an error occurs, the diskette drive retries the operation, depending on the type of error, before it posts a permanent error (see the **retries** table for a stand-alone verify type operation under "Error-Recovery Procedures" in this chapter). When the operation is complete, the status is stored at the location indicated by the residual status block address in DCB word 4. If an exception interrupt request occurs, status is available to a Start Cycle Steal Status command. If the suppress exception bit is a 0, no retries are attempted and any errors result in the end of the operation with status available to a Start Cycle Steal Status command.

This operation supports the automatic seek option (bit 12) in the DCB control word. When bit 12 is a 0, the cylinder number is checked against the current location and a seek operation is initiated, as required. When this bit is a 1, the automatic seek option is inhibited.

The following DCB fields must be specified: sector record length, density, head number, data fill character, RSB address (if the suppress exception bit is a 1), and the chain address (when chaining).

Read Data



The read data operation causes data to be read from the diskette drive into processor storage in cycle-steal mode. The DCB contains the density, sector record length, sector number, head number, cylinder number, residual status block address, byte count, and data address needed by the operation. The records are transferred to processor storage as indicated by the data address field of the DCB. The byte count specifies the number of bytes to be transferred. The data address and byte count must be an even number.

Data transfer ends when:

- The specified number of bytes are read.

  Note: If the specified number of bytes has been read before the last sector has been completely read, a cyclic redundancy check is performed on the full sector.

- The diskette drive detects the end of the selected track and the spiral bit is set to 0.

- A control address marker is detected and the control address marker mask (bits 0–1 in DCB word 1) indicates that reading should stop. For a read data operation, the control address marker mask bits have the following effects:

  Bits 0, 1

  00 The diskette drive reads data until the end of the sector that contains the control address marker, ends the operation, and requests an interrupt. Bit 6 (control address marker found) of cycle-steal status word 7 is set to 1; an exception condition code is reported to the processor.

01 Reading continues with the next data sector only if the first byte of the control information (the control address marker flag) in the present sector indicates that the record has been logically deleted. The present sector is bypassed; reading continues when the next data sector is found and continues until the specified number of bytes has been read. A device end condition code is reported at this time. If the control address marker flag in the present sector is other than hex C4 (record logically deleted), the entire sector is read, the operation terminates, an interrupt is posted, and an exception condition code is reported when the processor accepts the interrupt. Bit 6 (control address marker found) of cycle-steal status word 7 is set to 1.

10 Reading continues with the next data sector only if the first byte of the control information (the control address marker flag) in the present sector indicates that the record has been either logically deleted or relocated to the next sector. The present sector is bypassed; reading continues when the next data sector is found until the specified number of bytes has been read. A device end condition code is reported at this time, with the permissive device end bit 0 of the IIB set to 1. If the control address marker flag in the present sector is other than hex C4 (record logically deleted) or hex C8 (record relocated to the next sector), the entire sector is read, the operation terminates, an interrupt is posted, and an exception condition code is reported when the processor accepts the interrupt. Bit 6 (control address marker found) of cycle-steal status word 7 is set to 1.

11 This value causes a DCB specification check; an exception condition code is reported to the processor.

Sectors read are transferred to processor storage, one after another, in the order in which they are read from the diskette. If a control address marker is detected and the control address marker mask is set to allow reading to continue, the control address marker sector is bypassed. The number of control address markers encountered is contained in bits 8–15 of RSB word 2 and cycle-steal status word 2. Reading and data transfer is resumed at the beginning of the next data sector.

The error-recovery procedure for this operation is determined by the suppress exception bit (bit 4), DCB word 0. When this bit is a 1 and an error occurs, the diskette drive retries the operation, depending on the type of error, before it posts a permanent error (see the **retries** table for a read-type operation under "Error-Recovery Procedures" in this chapter). When the operation is complete, the status is stored at the location indicated by the processor storage address in DCB word 4. If an exception interrupt request occurs, status is available to a Start Cycle Steal Status command. When the suppress exception bit is a 0, no retries are attempted and any errors result in the end of the operation with status available to a Start Cycle Steal Status command.

This operation supports the spiral read/write option (bit 8) in DCB word 0. If bit 8 is set to 1, the diskette drive automatically reads data consecutively on all tracks and sectors, beginning at the address specified in word 7 of the DCB and continues until one of the following occurs:

- The byte count (maximum 64K) in word 6 is decremented to 0.
- The operation is completed on cylinder 74 (side 0 for one-sided diskettes and side 1 for two-sided diskettes).
- An error occurs.

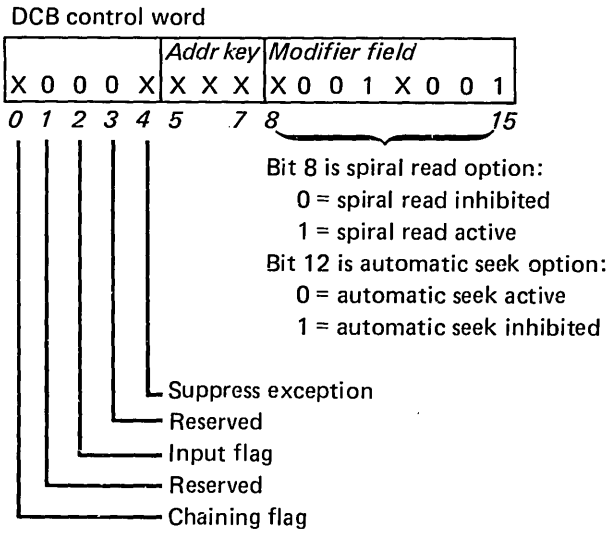
  Note: Do not attempt to read cylinder 0 using the spiral read/write option, because a no-record-found error might be reported. This occurs because cylinder 0, head 0, is always recorded in FM mode, while the remainder of the diskette may be recorded differently.

This operation supports the automatic seek option (bit 12) in DCB word 0. When bit 12 is a 0, the cylinder number is checked against the current location, and a seek operation is initiated, as required.

Programming Considerations: If a sector ID matching the sector record length, sector number, head number, and cylinder number is not located after one revolution of the diskette, or if a sector ID error is detected, the operation ends immediately (if the suppress exception bit is a 0) and an interrupt request is sent to the processor. An exception condition code and interrupt ID word containing status information are transferred to the processor when the interrupt is serviced.

The following DCB fields must be specified: input flag (must be a 1), sector number, starting sector, sector record length, density, control AM mask, head number, cylinder number, RSB address (if the suppress exception bit is a 1), chain address (when chaining), and spiral read/write bit (when performing a spiral read).

Read Verify/Cyclic Redundancy Check



The read verify/cyclic redundancy check operation is similar to the read data operation in that each sector specified is read completely (see "Read Data" in this chapter); however, data read from the diskette during this operation is not transferred to processor storage.

The cyclic redundancy check field is checked to verify the data. If the cyclic redundancy check results in an "unequal" compare, an exception interrupt request is posted and status is available to a Start Cycle Steal Status command.

If a control address marker is encountered, it is handled the same as in a read data operation.

If an error occurs, the operation terminates and a permanent error is posted. When the operation is complete, the status is stored at the location indicated by the processor storage address in DCB word 4. Retries are performed only during the seek portion of the operation, not during the verify portion. If an exception interrupt request occurs, status is available to a Start Cycle Steal Status command.

This operation supports the spiral read/write option (bit 8) in DCB word 0. If bit 8 is set to 1, the diskette drive automatically reads data consecutively on all tracks and sectors, beginning at the address specified in Word 7 of the DCB and continues until one of the following occurs:

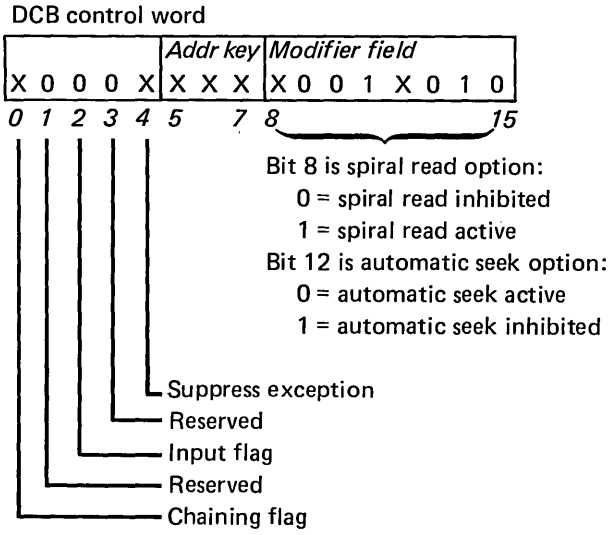
- The byte count (maximum 64K) in word 6 is decremented to 0.
- The operation is completed on cylinder 74 (side 0 for one-sided diskettes and side 1 for two-sided diskettes).
- An error occurs.

  Note: Do not attempt to read cylinder 0 using the spiral read/write option, because a no-record-found error might be reported. This occurs because cylinder 0, head 0, is always recorded in FM mode while the remainder of the diskette may be recorded differently.

This operation supports the automatic seek option (bit 12) in the DCB control word. When bit 12 is a 0, the cylinder number is checked against the current location and a seek operation is initiated, as required.

The following DCB fields must be specified: sector number, sector record length, density, control AM mask, head number, cylinder number, RSB address (if the suppress exception bit is a 1), chain address (when chaining), spiral read/write bit (when performing a spiral read), and byte count.

Read Verify/Compare Data



The read verify/compare data operation verifies and compares previously written data. The data is transferred from processor storage to the diskette drive in cycle-steal mode. The data being read from the currently loaded diskette is compared bit for bit with the data from processor storage. If an error is found during this operation, the read verify error bit (bit 12) in cycle-steal status word 6 is set to 1. This operation ends when one of the following occurs:

- The specified number of bytes has been read.
- The diskette drive detects the end of the selected track and the spiral bit is set to 0.
- The processor detects a hardware error.
- A control marker is found. Control address markers are handled for this operation the same as they are for the read data operation. (Refer to "Read Data" in this chapter.)

If an error occurs, the operation terminates and a permanent error is posted. If the suppress exception bit (bit 4 of DCB word 0) is a 1, retries are performed only during the seek portion of the operation, not during the verify portion. When the operation is complete, the status is stored at the location indicated by the residual status block address in DCB word 4. If an exception interrupt request occurs, status is available to a Start Cycle Steal Status command. No retries are attempted during the verify portion and any errors result in the end of the operation with status available to a Start Cycle Steal Status command.

This operation supports the spiral read/write option (bit 8) in DCB word 0. If bit 8 is set to 1, the diskette drive automatically reads data consecutively on all tracks and sectors, beginning at the address specified in word 7 of the DCB and continues until one of the following occurs:

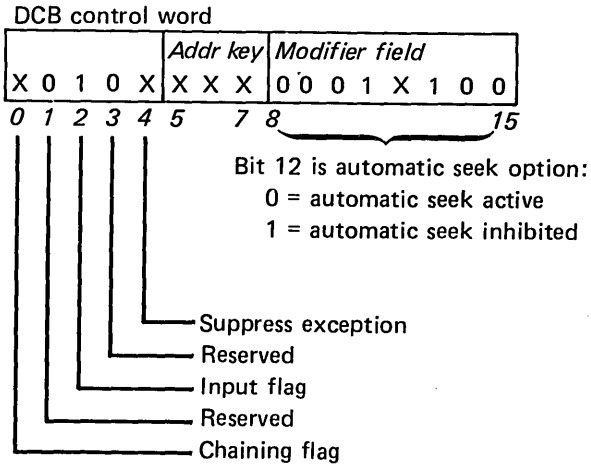
- The byte count (maximum 64K) in word 6 is decremented to 0.
- The operation is completed on cylinder 74 (side 0 for one-sided diskettes and side 1 for two-sided diskettes).
- An error occurs.

  Note: Do not attempt to read cylinder 0 using the spiral read/write option, because a no-record-found error might be reported. This occurs because cylinder 0, head 0, is always recorded in FM mode while the remainder of the diskette may be recorded differently.

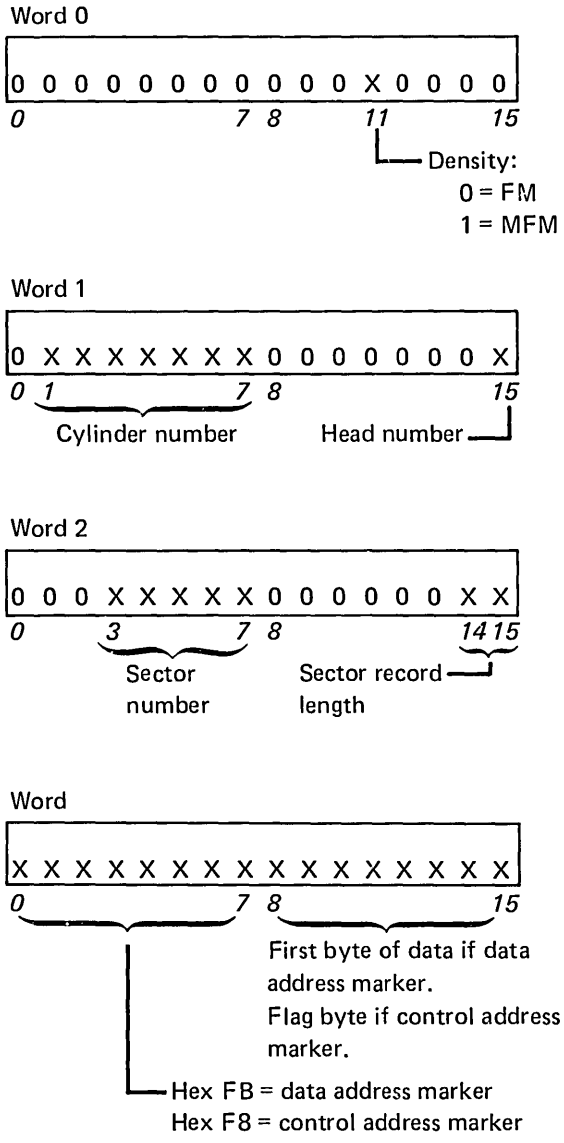
This operation supports the automatic seek option (bit 12) in the DCB control word. When bit 12 is a 0, the cylinder number is checked against the current location and a seek operation is initiated, as required.

The following DCB fields must be specified: sector number, sector record length, density, control address marker mask, head number, cylinder number, RSB address (if the suppress exception bit is a 1), chain address (when chaining), spiral read/write bit (when performing a spiral read), byte count, and data address.

Read Sector ID



Beginning with the first sector encountered after the index, the read sector ID operation reads as many ID blocks as the byte count calls for into processor storage. Each ID block consists of the sector ID and the first byte of the sector's data or control information within the data field. The data read is in the following format:



Words 1 through 3 are repeated for other sectors on the track. The byte count for this operation should be 8 with additional multiples of 6 for each additional sector ID that is to be read.

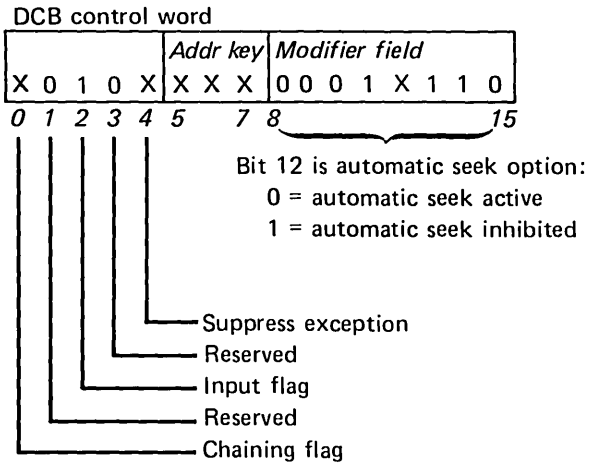
The error-recovery procedure for this operation is determined by the suppress exception bit (bit 4), DCB word 0. When this bit is a 1 and an error occurs, the diskette drive retries the operation, depending on the type of error, before it posts a permanent error (see the **retries** table for a read-type operation under "Error-Recovery Procedures" in this chapter). When the operation is complete, the status is stored at the location indicated by the residual status block address in DCB word 4. If an exception interrupt request occurs, status is available to a Start Cycle Steal Status command.

When the suppress exception bit is a 0, no retries are attempted and any errors result in the end of the operation with status available to a Start Cycle Steal Status command.

This operation supports the automatic seek option (bit 12) in the DCB control word. When bit 12 is a 0, the cylinder number is checked against the current location and a seek operation is initiated, as required.

The following DCB fields must be specified: input flag is a 1, head number, cylinder number, RSB address, chain address (when chaining), byte count, and data address.

Read Diagnostic Record



The read diagnostic record operation is used by special error-recovery programs to retrieve data on a diskette when read data operations are unable to do so. For this operation, DCB word 3 (bits 1–15) contains the physical offset timer value. See "DCB Word 3 (Parameter Word 3)" under "Start" in this chapter. The FM/MFM bit must be specified.

When searching for a sector on a given track, the diskette drive senses the index and delays the operation until the count in the timer is exhausted (decremented to 0). The amount of time that the operation is delayed is found by multiplying the physical offset timer value by 7.5 microseconds. The diskette drive then searches the data being read from the track for a sync field followed by an ID, data, or a control address marker. When this sequence is found, the information and the address marker are placed in processor storage in cycle-steal mode, beginning with the address specified in DCB word 7. The data transfer continues until the number of bytes specified in DCB word 6 has been read.

The read diagnostic record operation is not affected by defective diskette surfaces; however, termination of the operation occurs following the failure of the diskette drive to detect a sync field after passing the index twice.

If the preceding method for data retrieval fails, an additional step can be taken. By setting bit 0 in DCB word 3 to 1, the need for a data, control, or ID address marker is eliminated. Therefore, when a sync field is found, any character that is not a 0 acts as an address marker, and data transfer begins with that character.

The error-recovery procedure for this operation is determined by the suppress exception bit (bit 4), DCB word 0. When this bit is a 1 and an error occurs, the diskette drive retries the operation, depending on the type of error, before it posts a permanent error (see the **retries** table for a read-type operation under "Error-Recovery Procedures" in this chapter). When the operation is completed status is stored at the location indicated by the residual status block address in DCB word 4. If an exception interrupt request occurs, status is available to a Start Cycle Steal Status command.

If the suppress exception bit is a 0, no retries are attempted and any errors result in the end of the operation with status available to a Start Cycle Steal Status command.

If a permanent cyclic redundancy error is encountered, the data record read from the diskette on the last retry is transferred in cycle-steal mode to processor storage before the device-end interrupt is posted.

This operation supports the automatic seek option (bit 12) in DCB word 0. When bit 12 is a 0, the cylinder number is checked against the current location and a seek operation is initiated, as required.

Programming Considerations: This operation is for use during error-recovery; it is not intended for normal use. For this command, the first byte transferred to processor storage is the data address marker. But, if the byte count is equal to or more than the designated maximum size of the sector data field, the data record will not be read in its entirety, because the hardware will stop transfer of data for this sector when the full count for the sector is reached. (Refer to "Sector Record Length" under "Track Format" in Chapter 5.) Thus, the byte count (DCB word 6) must be incremented so that it is 2 more than the length of the record, and the hex value that indicates the maximum number of bytes in the sector data field must be set for the next larger value.

> **Note:** The byte count is incremented by 2 (rather than by 1) because it must always be an even number.

In this case, the data address marker, all data, and the first byte of the CRC are transferred to processor storage. If an interface data check occurs for the last byte transferred, it has no meaning. However, if a CRC check occurs, it indicates bad data.

> **Note:** The hex value indicating the sector data field size cannot exceed 3. Thus, when the actual record length is 1024 bytes (and the hex value is already set at 3), the byte count remains at 1024, and the hex value must still be set to 3. Then, the last byte of data cannot be read.

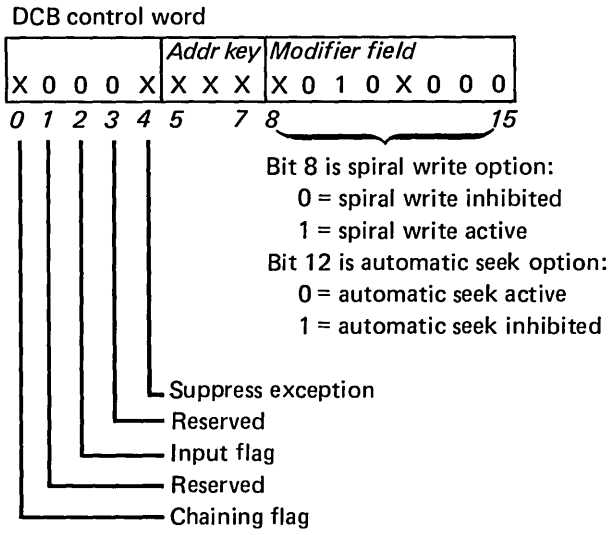
The following DCB fields must be specified: input flag set to 1, density, physical offset and bypass bit, head number, cylinder number (automatic seek only), RSB address (if the suppress exception bit is a 1), chain address (when chaining), byte count, and data address.

Timer Values

The following table shows nominal timer values for reference purposes only. Adjustments to compensate for diskette speed variation and other tolerances may be required to either increase or decrease the timer values from those listed. If the timer value is too low, a CRC and/or an interface data check are posted and the ID field for that data field is returned. If the timer value is too high, a CRC and/or an interface data check are posted and the ID field for the next data field is returned.

Data field number (decimal)	Single density			Double density		
	Sector record length			Sector record length		
	128	256	512	256	512	1025
1	018F	01AE	018C	0194	0183	0174
2	04B7	0736	0BB8	4B4	070D	0B90
3	07E6	0CBD	15E4	7D4	0C97	15A8
4	0B10	1245	2010	AE4	1222	1FCC
5	0E3C	17CD	2A3A	E00	17AE	29D4
6	1168	1D55	3462	1120	1D38	33F0
7	1491	22D5	3E8E	1440	22C2	3E00
8	17BE	2859	48B4	1754	2858	4B20
9	1AF1	2DDE		1A74	2DE2	
10	1E1B	3364		1D90	336C	
11	2145	38E3		20AC	38F6	
12	246C	3E6A		23C8	3E80	
13	2796	43F7		26E4	43CE	
14	2AC8	497E		29F8	4958	
15	2DDA	4EFD		2D0F	4EE2	
16	311C			3026		
17	344D			333D		
18	3774			3654		
19	3AA1			396B		
20	3DCB			3C82		
21	40F6			3F99		
22	4421			4260		
23	4740			45C7		
24	4A5C			48DE		
25	4D87			4BF5		
26	50A6			4F0C		

Write Data/Data Address Marker



The write data/data address marker operation writes hex FB in the address marker 2 byte when the data field contains data recorded in single density (FM). A hex A1A1A1FB is written in address marker 2 when the data field contains data recorded in double density (MFM).

The DCB contains the density, sector record length, sector number, head number, cylinder number, RSB address (if the suppress exception bit is a 1), chaining address (when chaining), byte count, and data address needed by the operation. The records to be written are specified by the data address field of the DCB. The byte count specifies the number of bytes to be transferred. The byte count and the data address must be even numbers.

If the record being written is not a full sector, the diskette drive writes the data bytes and then fills the remainder of the sector with 0's. Records longer than one sector are written over as many sectors as are necessary to satisfy the byte count. This continues onto the next track if the spiral write bit is set to 1.

The error-recovery procedure for this operation is determined by the suppress exception bit (bit 4) of DCB word 0. When this bit is a 1, the diskette drive retries the operation, depending on the type of error, before it posts a permanent error (see the **retries** table for a write-type operation, under "Error-Recovery Procedures," later in this chapter). When the operation is complete, the status is stored at the location indicated by the residual status block address in DCB word 4. If an exception interrupt request occurs, status is available to a Start Cycle Steal Status command. When the suppress exception bit is a 0, no retries are attempted and any errors result in the end of the operation with status available to a Start Cycle Steal Status command.

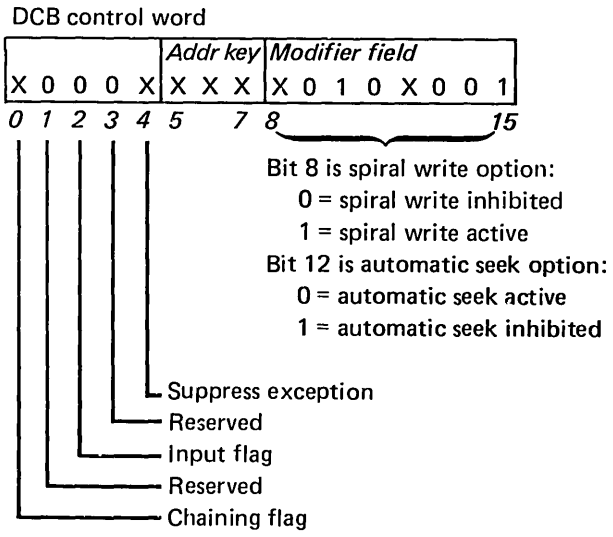
This operation supports the spiral read/write option (bit 8) in DCB word 0. If bit 8 is set to 1, the diskette drive automatically writes data consecutively on all tracks and sectors, beginning at the address specified in word 7 of the DCB and continues until:

- The byte count (maximum 64K) in word 6 is decremented to 0.
- The operation is completed on cylinder 74 (side 0 for one-sided diskettes and side 1 for two-sided diskettes).
- An error occurs.

  Note: Do not attempt to write cylinder 0 using the spiral read/write option, because a no-record-found error might be reported. This occurs because cylinder 0, head 0, is always recorded in FM mode while the remainder of the diskette may be recorded differently.

This operation supports the automatic seek option (bit 12) of DCB word 0. If this bit is a 0, the cylinder number is checked against the current location and a seek operation is initiated, as required. A read sector ID operation is initiated to assure the diskette drive that the seek operation was correct.

Write Data/Control Address Marker



The write data/control address marker operation writes hex F8 in the address marker 2 byte when the data field contains control information recorded in single density. A hex A1A1A1F8 is written when the data field contains control information recorded in double density.

The first byte of data to be transferred should be one of the following:

- A hex C4, which indicates that the physical sector data has been logically deleted.
- A hex C6, which indicates that the sector data surface is defective and the data can be found on the next physical sector.
- A hex 4B, which indicates that the sector data surface is defective and the data is relocated to the sector designated by the program.

  Note: Only one byte of meaningful data should be written in each sector with a control address marker.

The DCB contains the density, sector record length, sector number, head number, cylinder number, residual status block address (if the suppress exception bit is a 1), chaining address (when chaining), byte count, and data address needed by the operation. The records to be written are specified by the data address field of the DCB. The byte count specifies the number of bytes to be transferred and must be an even number.

If the record being written is not a full sector, the diskette drive writes the control bytes and then fills the remainder of the sector with 0's. Records longer than one sector are written over as many sectors as are necessary to satisfy the byte count.

The error-recovery procedure for this operation is the same as for a write data/data address marker operation. When the operation is complete, the status is stored at the location indicated by the residual status block address in DCB word 4. If an exception interrupt request occurs, status is available to a Start Cycle Steal Status command. When the suppress exception bit is a 0, no retries are attempted and any errors result in the end of the operation with status available to a Start Cycle Steal Status command.

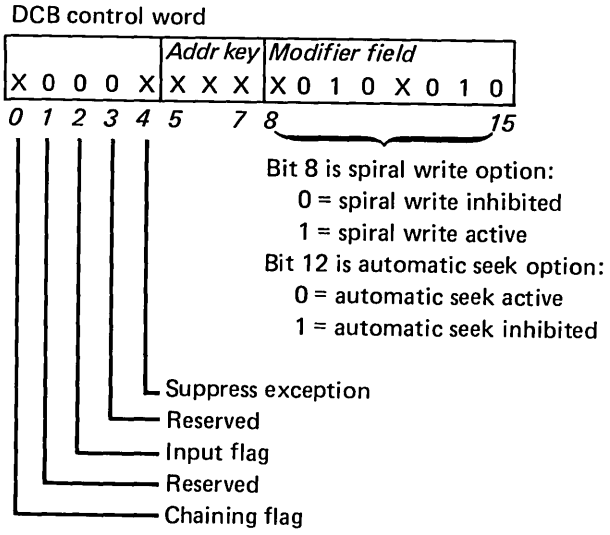
This operation supports the spiral read/write option (bit 8) in DCB word 0. If bit 8 is set to 1, the diskette drive automatically writes control information consecutively on all tracks and sectors, beginning at the address specified in word 7 of the DCB and continues until:

- The byte count (maximum 64K) in word 6 is decremented to 0.
- The operation in completed on cylinder 74 (side 0 for one-sided diskettes and side 1 for two-sided diskettes).
- An error occurs.

  Note: Do not attempt to write cylinder 0 using the spiral read/write option, because a no-record-found error might be reported. This occurs because cylinder 0, head 0, is always recorded in FM mode while the remainder of the diskette may be recorded differently.

This operation supports the automatic seek option (bit 12) of DCB word 0. If this bit is a 0, the cylinder number is checked against the current location and a seek operation is initiated, as required. A read sector ID operation is initiated to assure the diskette drive that the seek operation was correct.

Write Data With Read Verify



The write data with read verify operation writes data and automatically verifies the data written with the cyclic redundancy check before the operation is complete.

It is recommended that the write data with read verify be used as one operation instead of separate operations so that automatic error-recovery procedures can retry both the write and verify portions of the operation if the verify portion fails.

The error-recovery procedure for this operation is determined by the suppress exception bit (bit 4), control word 0. When this bit is a 1 and an error occurs, the diskette drive retries the operation, depending on the type of error, before it posts a permanent error (see the **retries** table for a write-type operation, under "Error-Recovery Procedures," later in this chapter). When the operation is complete, the status is stored at the location indicated by the residual status block address in DCB word 4. If an exception interrupt request occurs, status is available to a Start Cycle Steal Status command. When the suppress exception bit is a 0, retries are not attempted and any errors result in the end of the operation with status available to a Start Cycle Steal Status command.

This operation supports the spiral read/write option (bit 8) in DCB word 0. If bit 8 is set to 1, the diskette drive automatically writes and then verifies (all writes are completed before the verifies begin) data consecutively on all tracks and sectors, beginning at the address specified in word 7 of the DCB and continues until:

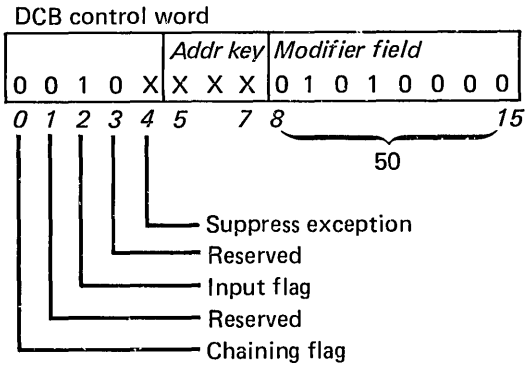
- The byte count (maximum 64K) in word 6 is decremented to 0.
- The operation is completed on cylinder 74 (side 0 for one-sided diskettes and side 1 for two-sided diskettes).
- An error occurs.

  Note: Do not attempt to write cylinder 0 using the spiral read/write option, because a no-record-found error might be reported. This occurs because cylinder 0, head 0, is always recorded in FM mode while the remainder of the diskette may be recorded differently.

This operation supports the automatic seek option (bit 12) of DCB word 0. If this bit is a 0, the cylinder number is checked against the current location and a seek operation is initiated, as required.

The DCB contains the density, sector record length, sector number, head number, cylinder number, residual status block address (if the suppress exception bit is a 1), chaining address (when chaining), byte count, spiral read/write bit (when performing a spiral write), and data address needed by the operation. The records to be written are specified by the data address field of the DCB. The byte count specifies the number of bytes to be transferred and must be an even number.

Read Attachment Storage



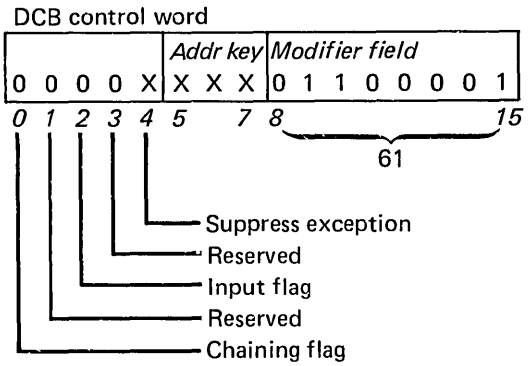
The read attachment storage operation reads from attachment storage into processor storage. This operation is intended to be used only for diagnostic purposes. The byte count specifies the number of bytes to be transferred and must be an even number.

A DCB specification check is posted and data is not transferred if any of the following conditions exist:

- Chaining bit 0, DCB word 0, is a 1.
- The byte count is too large.
- The value of DCB word 3 (attachment storage address) is not in the range from hex 0C00 to hex 0FFF or the address is odd.

DCB chaining is not supported. The following DCB fields must be specified: input flag (must be a 1), attachment storage address, RSB address (if the suppress exception bit is a 1), the byte count, and the data address in processor storage.

Write Attachment Storage



The write attachment storage operation writes into attachment storage from processor storage. Except for the direction in which data flows, this operation is identical to the read attachment storage operation. Refer to "Read Attachment Storage" for a further explanation.

Start Cycle Steal Status

IDCB (immediate device control block)

Command field	*Device address field*
0 1 1 1 1 1 1 1	X X X X X X X X
0 ... *7*	*8* ... *15*
7F	*00–FF*

Immediate data field
DCB address
16 ... *31*

The Start Cycle Steal Status command initiates the transfer of up to 13 words of status information from the diskette drive to processor storage in cycle-steal mode. This status information is used to determine why a given operation did not execute correctly. The processor storage address is specified in word 7 of the applicable DCB.

The Start Cycle Steal Status command requires an Operate I/O instruction with the address of an IDCB, an IDCB with the address of the DCB, and a DCB. The format of the DCB is as follows:

DCB (device control block)

Word		
0	*Control word*	
	0 0 1 0 0 | *Addr key*	0 0 0 0 0 0 0 0
	0 ... *4* *5* ... *7* *8* ... *15*	
1	Not used (0's)	
2	Not used (0's)	
3	Not used (0's)	
4	Not used (0's)	
5	Not used (0's)	
6	Byte count (must be even and ⩽ hex 1A)	
7	Data address of status word 0 (must be even)	
	0 ... *15*	

The 13 words of status information have the following format and meaning:

Status Word	
0	Residual address
1	Residual count
2	Retry counts word 1
3	Retry counts word 2
4	Retry counts word 3
5	Reserved for diagnostics
6	Error status word 1
7	Error status word 2
8	Not used (0's)
9	Movable head status word
10	Last DCB address
11	Current head/cylinder
12	Previous head/cylinder

Word 0 (Residual Address)

Word 0 contains the storage location of the last attempted cycle-steal transfer associated with a Start command. This address could be a DCB address, data address, or residual status block address. During a Start Cycle Steal Status operation, this address is not altered. Only a power-on reset resets the residual address to hex 0001.

Word 1 (Residual Count)

Word 1 contains the residual byte count for data requested for the last DCB operation. The count reflects the number of bytes of data not transferred.

Word 2 (Retry Counts Word 1)

Word 2 contains temporary error retry counts. The error conditions and their respective bit fields are:

Bit	Meaning
0	Storage data check count
1	Invalid storage address count
2	Protect check error count
3	Interface data check
4–6	Seek errors retry count
7	Reserved and not used
8–15	Control address markers encountered count

Word 3 (Retry Counts Word 2)

Word 3 is an extension of the temporary error retry counts. The error conditions and their respective bit fields are:

Bit	Meaning
0	Reserved and not used
1	Parity error count
2–7	Cyclic redundancy check error retry count or data verify error count
8–9	Reserved and not used
10–15	No record found error count

Word 4 (Retry Counts Word 3)

Word 4 is an extension of the temporary error retry counts. The error conditions and their respective bit fields are:

Bit	Meaning
0–1	Reserved and not used
2–7	No data found error count
8–9	Underrun/overrun error count
10–15	Equipment check error count

Word 5

Word 5 is reserved for diagnostics.

Word 6 (Error Status Word 1)

The following bit format and respective error conditions are described with the bit set to 1, unless otherwise specified:

Bit 0 — **Permanent error.** A permanent error condition exists. This causes the operation to terminate, sets the temporary error counter to 0, and posts an exception interrupt. When the suppress exception bit is 0, any error detected causes bit 0 to be set to 1.

Bit 1 — **Attachment detected parity check.** A parity error is detected, and an exception interrupt is posted if the suppress exception bit is a 0. If the suppress exception bit is a 1 and retries are performed unsuccessfully, an exception interrupt is posted.

Bit 2 — **Attachment time-out.** An operation has taken too long to perform, causing an exception interrupt to be posted.

Bits 3,4 — Reserved and not used.

Bit 5 — **Wrong type of diskette selected.** There are two conditions that can set this bit to 1 and cause an exception interrupt to be posted:

- A command is issued to select head 1 of a diskette and a diskette 1 (side 0 only) is loaded.
- The value of density bit 3 in DCB word 1 does not match the density of the loaded diskette.

Bit 6 — **Head seek error.** The cylinder number in the ID did not compare either after an automatic seek or an automatic seek retry. An exception interrupt is posted if the suppress exception bit is a 0. If the suppress exception bit is a 1 and retries are performed unsuccessfully, an exception interrupt is posted.

Bit 7 — Reserved and not used.

Bit 8 — **Write gate/Erase gate.** For diagnostic use only.

Bit 9 — Reserved and not used.

Bit 10 — **Attachment equipment check.** The attachment detects a hardware error, causing an exception interrupt to be posted if the suppress exception bit is a 0. If the suppress exception bit is a 1 and retries are performed unsuccessfully, an exception interrupt is posted.

Bit 11 **Overrun/underrun.** The processor I/O channel cannot accept the cycle-steal requests from the diskette drive due to a mismatch between the data rate of cycle-steal data and the data rate of the channel. An exception interrupt is posted if the suppress exception bit is a 0. If the suppress exception bit is a 1 and retries are performed unsuccessfully, an exception interrupt is posted.

Bit 12 **Read verify error.** One of the following operations fails to compare:

- Read verify/cyclic redundancy check
- Read verify/compare data

An exception interrupt is posted if the suppress exception bit is a 0. If the suppress exception bit is a 1 and retries are performed unsuccessfully, an exception interrupt is posted.

Bit 13 **Related error.** This bit is set to 1 when an error that is not directly associated with the current command occurs (error occurred during a read of the FM/MFM bit on cylinder 0, or error occurred during the seek portion of a command with automatic seek). An exception interrupt is posted if the suppress exception bit is a 0. If the suppress exception bit is a 1 and retries are performed unsuccessfully, an exception interrupt is posted.

Bit 14 Reserved and not used.

Bit 15 **Temporary error retry.** A temporary error has been retried. This bit is set to 1 if the suppress exception bit is set to 1 and temporary errors (errors that are recoverable before the retry count goes to 0) have occurred and have been stored in the RSB. (The RSB is always stored when the suppress exception bit is set to 1 and an exception interrupt is not posted.)

Word 7 (Error Status Word 2)

The following bit format and respective error conditions are described with the bit set to 1, unless otherwise specified:

Bit 0	**Cyclic redundancy check.** A cyclic redundancy check has occurred during a sector ID or data record read function. An exception interrupt is posted if the suppress exception bit is a 0. If the suppress exception bit is a 1 and retries are performed unsuccessfully, an exception interrupt is posted.
Bit 1	Reserved and not used.
Bit 2	**Exceeded control AM count.** This bit is set to 1 if more than 255 control AMs are detected during a spiral read or verify operation. An exception interrupt is posted.
Bit 3	Reserved and not used.
Bit 4	**No record found.** The cylinder, head, record number, and record length must match the corresponding fields of the ID that was read. Any field that fails to compare within two diskette revolutions can cause a no record found error. An exception interrupt is posted if the suppress exception bit is a 0. If the suppress exception bit is a 1 and retries are performed unsuccessfully, an exception interrupt is posted.
Bit 5	**No data found.** A sector is addressed, but the sector's data field cannot be found. An exception interrupt is posted if the suppress exception bit is a 0. If the suppress exception bit is a 1 and retries are performed unsuccessfully, an exception interrupt is posted.
Bit 6	**Control address marker found.** This bit is set to 1 whenever a control AM is read.
Bit 7	**ID check failed after format track defective operation.** A format track defective command was issued and the automatic read ID that follows failed to read a valid defective track ID. An exception interrupt is posted.
Bit 8	Reserved and not used.
Bit 9	**Diskette not up to speed.** The diskette is not up to its proper rotational speed after two revolutions. This can be caused by either a defective diskette or defective hardware. An exception interrupt is posted.
Bit 10	Reserved and not used.
Bit 11	**End of track.** The logical end of a track is encountered before the specified number of bytes (DCB word 6) has been read using a non-spiral command. An exception interrupt is posted.
Bits 12–14	Reserved and not used.
Bit 15	**Diskette unit not ready.** The diskette drive has a not-ready condition (no diskette installed, diskette not rotating, or diskette drive has not come up to or stayed up to its rotational speed). An exception interrupt is posted.

Word 8—Reserved and not used

Word 9 (Movable Head Status Word)

The following bits are described with the bit set to 1, unless otherwise specified:

Bits 0–3	Reserved and not used
Bit 4	Head access latch 0
Bit 5	Head access latch 1
Bits 6–13	Reserved and not used
Bit 14	The data heads are positioned at cylinder 0
Bit 15	Reserved and not used

Word 10 (Last DCB Address)

Word 10 contains the starting address of the last DCB other than Start Cycle Steal Status used by the attachment.

Word 11 (Current Head/Cylinder)

Word 11 is in the format of the current DCB word 2 and is stored for every command that involves a seek. When operations that do not require a seek are performed, this word contains the previous position (see cycle-steal status word 12 below).

Word 12 (Previous Head/Cylinder)

Word 12 is in the format of the previous DCB word 2 and is stored during the operation-ending sequence of the last command. This word contains the current head number and cylinder number if the current operation is other than a seek or automatic seek command (see cycle-steal status word 11 above).

Start Cycle Steal Diagnostic

IDCB (immediate device control block)

Command field	*Device address field*
0 1 1 1 1 1 0 1	X X X X X X X X

0 *7* *8* *15*

7D *00–FF*

Immediate data field
DCB address

16 *31*

The Start Cycle Steal Diagnostic command is used to diagnose the attachment feature card. The results are cycle-stolen back to the Series/1 processor.

DCB (device control block)

Word		
0	*Control word*	
	0 0 1 0 0 *Addr key*	0 0 0 0 0 0 0 0
	0 *4* *5* *7* *8* *15*	
1	Not used (0's)	
2	Not used (0's)	
3	Not used (0's)	
4	Not used (0's)	
5	Not used (0's)	
6	Byte count (must be even and ⩽ hex 18)	
7	Data address (must be even)	

0 *15*

The format of the data returned is as follows:

Word 0: ROS module page 0 checksum
Word 1: Expected ROS module page 0 checksum complemented
Word 2: ROS module page 1 checksum
Word 3: Expected ROS module page 1 checksum complemented
Word 4: ROS module page 2 checksum
Word 5: Expected ROS module page 2 checksum complemented
Word 6: Expected I/O wrap test result complemented
Word 7: I/O wrap test result
Word 8: Reserved
Word 9: Reserved
Word 10, byte 0: Data address register and register test result
Word 10, byte 1: I/O wrap test result
Word 11: Reserved

Automatic Seek Option

The diskette drive attachment retains only the *physical* cylinder number for individual seek operations. However, the attachment retains a *physical* and a *logical* cylinder number when performing operations that employ the automatic seek option. Thus, using the automatic seek option ensures that the logical cylinder specified by DCB word 2 is found in the event that a defective track has been replaced by an alternate track. Refer to the description of the modifier field in "DCB Word 0 (Control Word)," under "Start," earlier in this chapter.

The automatic seek option should be used, whenever supported, during either of the following situations:

- When performing any supported individual operation after a diskette has been formatted.
- When performing any spiral read/write operation.

If the specified track (cylinder) is not found and the suppress exception bit is set to 1, the automatic seek option attempts to find the specified track by executing the following retries:

1. Move to the next track in the original direction.
2. Move one more track in the same direction as retry 1.
3. Recalibrate to track 0 and re-seek to the track specified by DCB word 2.
4. Move to the next track in the inward direction.
5. Move one more track in the same direction as retry 4.

If the specified cylinder is not found within the five retries, bit 6 of cycle-steal status word 6 is set to 1 and an exception interrupt is reported.

> **Note:** Bit 6 of cycle-steal status word 6 is set to 1 if any seek error requiring a seek retry is found. An exception interrupt is posted after all retries are unsuccessful.

Spiral Operation

Spiral operations step from cylinder to cylinder in ascending order, and from head 0 to head 1 (on two-sided diskettes) within each cylinder.

Spiral operation is possible on six Start command operations:

- Read Data
- Read Verify/CRC Check
- Read Verify/Compare Data
- Write Data/Data Address Marker
- Write Data/Control Address Marker
- Write Data with Read Verify/CRC Check

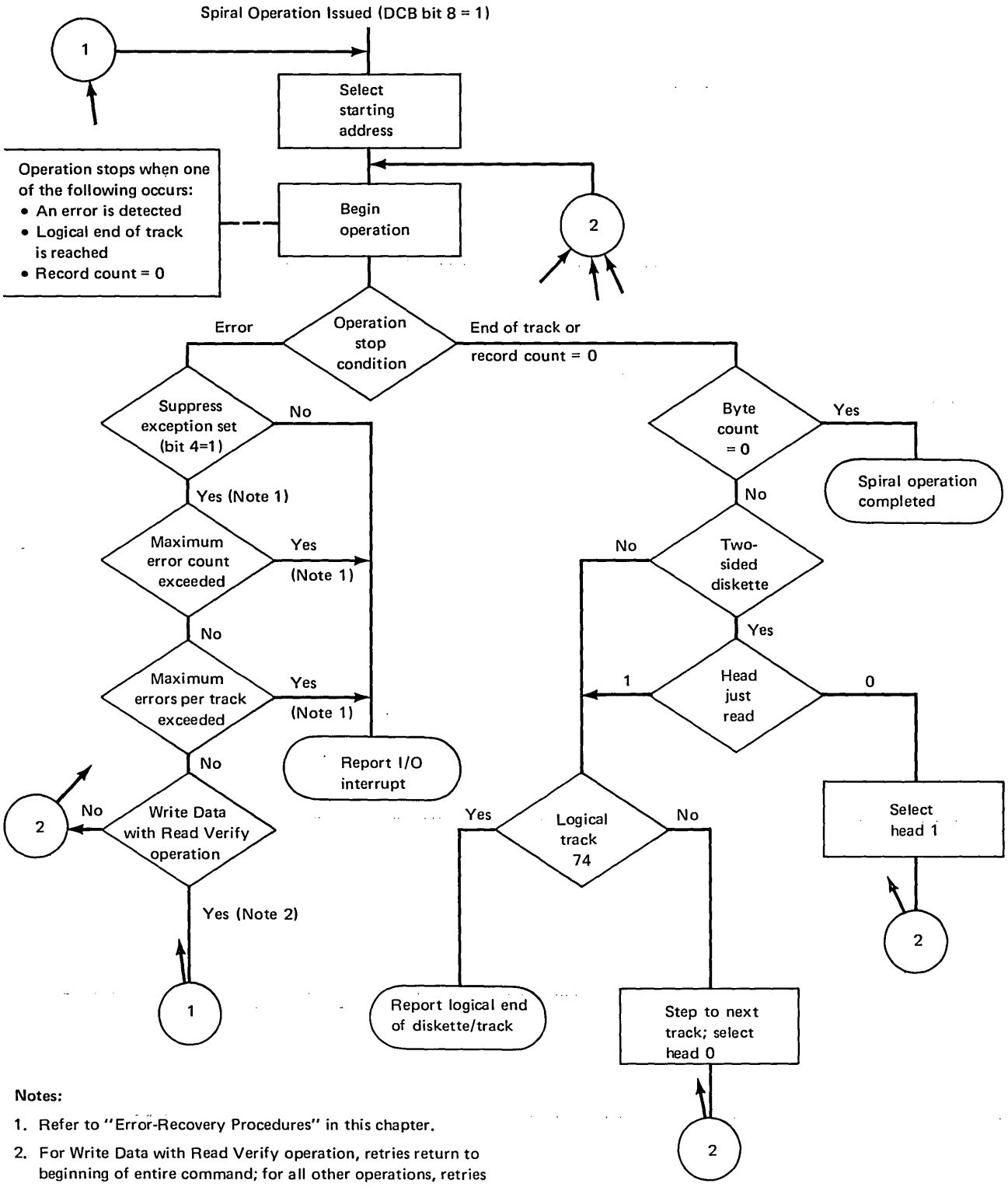
Spiral operation is activated by a 1 in bit 8 of DCB word 0. The maximum byte count in word 6 for spiral operation is 64K bytes. This mode of operation causes the diskette drive to automatically perform the assigned operation consecutively on all tracks and sectors, beginning at the designated diskette address and proceeding until one of the following conditions occurs:

- The byte count in word 6 is decremented to 0.
- The operation is completed on cylinder 74 (side 0 for one-sided diskettes and side 1 for two-sided diskettes).
- An error occurs.

For information about recovery from error conditions during spiral operation, refer to "Error-Recovery Procedures," later in this chapter.

Notes:

1. The automatic seek option should be used with all spiral operations; otherwise, a no-record-found (NRF) error might be returned. Refer to "Automatic Seek Option," in this chapter.
2. Do not attempt to read or write cylinder 0 using the spiral read/write option, because a no-record-found error might be reported. This occurs because cylinder 0, head 0, is always recorded in FM mode, while the remainder of the diskette may be recorded differently.



Notes:

1. Refer to "Error-Recovery Procedures" in this chapter.
2. For Write Data with Read Verify operation, retries return to beginning of entire command; for all other operations, retries return to beginning of current track only.

Condition Codes

A condition code is reported to the processor (1) after execution of every Operate I/O instruction and (2) upon presentation of a priority interrupt request. The condition code is available in the even, carry, and overflow bit positions of the level status register (LSR) in the processor. For information on the LSR, refer to an appropriate processor description manual listed in the *IBM Series/1 Graphic Bibliography*, GA34-0055. For operations that do not cause interrupt requests, the condition code reported after the instruction is executed is the only status information required or available.

Operate I/O Instruction

Condition Code 0 (Device Not Attached): This code is reported by the data channel when the addressed attachment is not attached to the processor data channel.

Condition Code 1 (Busy): This code is reported by the data channel when it is unable to execute a command because it is in the busy state. The diskette drive enters the busy state when it is performing an operation that generates an interrupt request. The diskette drive exits the busy state when the processor accepts the interrupt request.

Condition Code 2 (Busy After Reset): This code is reported by the attachment when it is unable to execute a command because of a reset and the attachment has not had sufficient time to return to the normal (active) state. There is no interrupt request to indicate termination of this condition.

Condition Code 3 (Command Reject): This code is reported by the attachment or the data channel when an issued command is not part of the diskette drive command set. When a cycle-steal device reports command reject, the DCB is not fetched.

Condition Code 4: Reserved and not used.

Condition Code 5 (Interface Data Check): This code is reported by the attachment or the channel when a parity error is detected on the I/O data bus during a data transfer.

Condition Code 6 (Controller Busy): This code is reported by the channel when the attachment to which two diskette drives are attached is busy.

Condition Code 7 (Satisfactory): This code is reported by the attachment when it accepts a command.

Condition Codes 0 and 1: Reserved and not used.

Condition Code 2 (Exception): This code is reported when an error or exception condition is associated with the priority interrupt request. This condition is described in the interrupt status byte (ISB). The additional device-dependent status words are obtained via the Start Cycle Steal Status command.

Condition Code 3 (Device End): This code is reported when no error exception or attention conditions occur during the I/O operation and a normal termination of the operation has occurred. If the permissive device end bit in the IIB is set to 1, some part of the command is successfully retried before a valid device end can be presented.

Condition Code 4 (Attention): This code is reported when an interrupt is caused by the diskette drive going to the ready condition.

Condition Code 5: Reserved and not used.

Condition Code 6 (Attention and Exception): This code is reported when both an attention and an exception are present.

Conditon Code 7 (Attention and Device End): This code is reported when both attention and device end are present.

Status Information

Status information is transferred from the diskette drive to the processor as the result of:

- A start cycle-steal status operation. See "Start Cycle Steal Status" in this chapter.
- Storing a residual status block. See "DCB Word 0 (Control Word)" under "Start" in this chapter.
- A priority interrupt request that stores an IIB or an ISB.

Interrupt Identification Word

Acceptance of an I/O interrupt request causes the diskette drive to present an interrupt ID word to the processor. The interrupt ID word consists of an interrupt information byte (IIB) and the diskette drive device address; it is stored in processor register 7. The format is as follows:

Interrupt ID word

IIB (ISB)								*Device address*							
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0							*7*	*8*							*15*

For attention interrupts, the IIB is always 0's. For device end and attention/device end interrupts, the IIB can have the following meanings:

Bit 0 **Permissive device end.** When this bit is set to 1, it indicates that information about temporary errors or control address markers encountered is available in the residual status block. In the case of chained DCBs, this bit indicates that at least one of the residual status blocks stored contains information about temporary errors or control address markers encountered. When chaining DCBs, this bit is set to 1 only if at least one DCB in the chain uses the suppress exception bit option and retries were performed.

Bits 1–7 **Reserved.** These bits are all 0's.

Interrupt Status Byte

The ISB stores accumulated status information.

The format of the ISB is:

Bit 0 **Device-dependent status available.** This bit is a 1 when additional status information (residual address and status bits) is available from the diskette drive in the cycle-steal status.

Bit 1 **Delayed command reject.** This bit is a 1 when the diskette drive cannot execute a command because of an invalid function or modifier in the IDCB or an odd byte address pointing to the first DCB.

Bit 2 This bit is not used by the diskette drive and is set to 0.

Bit 3 **DCB specification check.** This bit indicates that an invalid parameter, which prevented execution of the command, was found in the DCB (issued by the diskette drive). This can reside in any of the eight DCB words. The residual address received as a result of a Start Cycle Steal Status command points to the rightmost byte of the DCB word that contains the invalid parameter.

Bit 4 **Storage data check.** This bit is set only for cycle-steal output operations when incorrect parity is detected. The parity in main storage is not corrected, no machine check condition occurs, and, if the suppress exception bit is set to 1, retry and status storing are performed.

Bit 5 **Invalid storage address.** This bit is set to 1 as a result of a cycle-steal input/output operation when the main storage address provided by the attachment (for a data or DCB access) exceeds the storage size fitted on the system. If the suppress exception bit is set to 1, retry and status storing are performed.

Bit 6 **Protect check.** The attachment has attempted to access storage without the correct storage protection key. If the suppress exception bit is set to 1, retry and status storing are performed.

Bit 7 **Interface data check.** This bit is set to 1 when a parity error is detected on an interface cycle-steal data transfer. This condition can be detected by the channel. If the suppress exception bit is set to 1, retry and status storing are performed.

Error-Recovery Procedures

If an error occurs with suppress exception bit 4 of DCB word 0 set to 0, the following program events are recommended prior to posting a permanent error:

- Retry read-type errors nine times.
- Retry write-type errors three times.
- Retry seek errors five times. (Refer to "Automatic Seek Option" in this chapter.)
- For spiral read/write operations, retries are performed on a track basis with a maximum number of retries, as follows:

	Retries per track	Max. retries per diskette
Write/verify	3	9
Read	9	63
Seek	5	5

> **Note:** No retries are performed on any verify command. For write commands with automatic verify, the write portion is retried if the error is detected during the verify.

- For permanent seek errors, retry the operation one time with the suppress exception bit set to 1.
- Retry read-type or write-type operations three times if they cause an overrun or underrun.
- Retry the operation causing a parity check one time after first issuing a recalibrate head and re-seeking.

If an error occurs with suppress exception bit 4 of DCB word 0 set to 1, the following retries are automatic prior to posting a permanent error:

	Number of retries							
	Read		Write		Verify		Write/Verify	
Error	Normal	Spiral	Normal	Spiral	Normal	Spiral	Normal	Spiral
Cyclic redundancy check	9	9/63 (Note 1)	0	0	0	0	3	9
Equipment check	9	9 (Note 2)	3	3	0	0	3	3 (Notes 1 & 2)
Interface data check	1	1	1	1	1	1	1	1
Invalid storage address	1	1	1	1	1	1	1	1
Logical end of track	0	0	0	0	0	0	0	0
No data found	9	9/63 (Note 1)	0	0	0	0	3	3/9 (Notes 1 & 2)
No record found	9	9/63 (Note 1)	3	3/9 (Note 2)	0	0	3	3/9 (Notes 1 & 2)
Not up to speed (Note 3)	0	0	0	0	0	0	0	0
Overrun/underrun	3	3 (Note 2)	3	3 (Note 2)	0	0	3	3 (Note 2)
Parity check	1	1 (Note 4)	1	1 (Note 4)	0	0	1	1 (Note 4)
Seek (ID ≠ Cylinder no.) (Note 5)	5	5	5	5	5	5	5	5
Storage data check	1	1	1	1	1	1	1	1

Notes:

1. Maximum retries per track/maximum retries per diskette on a spiral operation.
2. Retry operation from beginning.
3. Not-up-to-speed error, delayed 175 milliseconds before the error is reported.
4. Retry on a track-by-track basis.
5. For seek retries, refer to "Automatic Seek Option" in this chapter.

The diskette drive initiates the required retries if all DCBs have the suppress exception bit set to 1, with the exception of those listed in the following chart. The chart describes the conditions to be tested and the actions to be taken. Perform the action number indicated and continue sequentially until the problem is found. The conditions can be tested in cycle-steal status words 6 and 7.

Cycle-steal status word	Bit	Error condition	Action numbers
7	15	Diskette unit not ready	1, 2, 3
6	2	Attachment time-out	2, 3, 4
6	1	Attachment detected parity check	2, 3, 4
6	10	Attachment equipment check	2, 3, 4
7	0	CRC check	2, 3, 4
7	4	No record found	2, 3, 4
7	5	No data found	2, 3, 4

Actions:

1. Ensure that the diskette drive is powered on and is prepared to allow interrupts.
2. Initiate a recalibrate head operation.
3. Re-seek to the original track.
4. Retry the original operation. If error persists, issue an operator message and exit the error-recovery procedures. Wait for an attention interrupt. An operator message can be used here.

Resets

Several methods of resetting controls and registers are available.

Power-On Reset: This resets all controls and registers (including the prepare register), sets the residual address register to hex 0001, sets cycle-steal status word 1 to 0, and recalibrates the heads.

System Reset: This resets all controls and registers (including the prepare register). However, it recalibrates the heads and does not reset the residual address register.

Initial Program Load (IPL): This resets all controls and registers (including the prepare register), sets the I-bit to 1, and sets priority level 0.

Halt I/O Command: This resets all controls and registers except the prepare register and the residual address register.

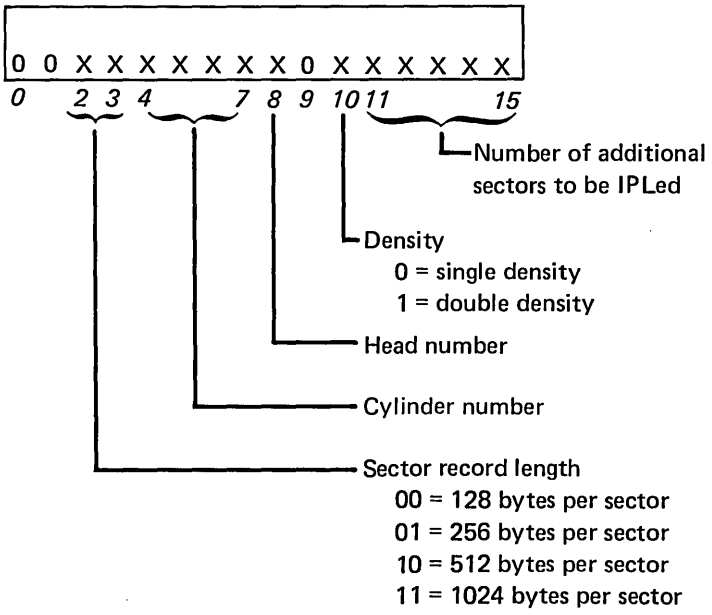
Device Reset Command: This resets all controls and registers except the prepare register and the residual address register.

Initial Program Load (IPL)

Following a successful IPL sequence, the diskette drive presents a device end interrupt on priority level 0. At interrupt time, the interrupt ID contains the diskette drive device address. Upon receiving the IPL select, the diskette drive performs the equivalent of a recalibrate head operation. Upon the completion of the recalibrate head operation, the diskette drive begins a read data operation with a byte count of hex 0100 (implying two sectors with data fields of 128 bytes each) and a storage data address of 0.

If the special code, hex 83C4, is detected in the 255th and 256th bytes, the byte count is increased to hex 0C00 (24 more sectors to be read) for a total of 3,328 bytes. This option of loading 26 sectors at IPL time is used only as a tool for loading diagnostics into the processor from a diagnostic diskette.

Another form of IPL is extended IPL, which allows up to one full track of additional information to be read into processor storage at the cylinder and head specified. To invoke this type of IPL, bytes 255 and 256 must have the following format:



For all types of IPLs that are successful, the 256th byte placed into storage contains the device address; the 255th byte contains the IIB.

Byte 72 of the data information for cylinder 00, head 0, sector 7 of a diskette is the density **byte**. The density byte has the following values:

- Hex 40 for a diskette 1
- Hex F2 for a diskette 2
- Hex D4 for a diskette 2D

This byte should be set by the processor program to the proper value during diskette initialization. If this byte is not initialized and the diskette is removed from the diskette drive, or if there is a power failure, the density of a formatted diskette cannot be recognized. Also, track 00 (cylinder 00, head 0) is always recorded in single density, twenty-six 128-byte sectors. When a new diskette is inserted, the density byte is read to determine the format of the diskette.

Appendix A. Instruction Execution Times

The processor executes instructions at a rate of approximately 434,000 instructions per second, based on an instruction mix that has been selected as being representative of Series/1 programming (refer to the Representative Instruction Mix Table in this appendix).

Several factors other than instruction mix have an effect on instruction throughput. Some factors that reduce instruction throughput are:

- Storage refresh
- Channel load interference
- Wait state
- Timer housekeeping
- Synchronization of the channel interface to the storage interface
- Correction of a single bit error in storage

The instruction mix in the following table has been selected as being a representative sample of Series/1 programming. The instruction rate is obtained as follows:

1. Multiply the weight of each instruction by its execution time. This results in a weighted time for each instruction.
2. Add the weighted times together, and divide their total into 1. This results in the number of instructions executed per second.

Representative Instruction Mix Table

Instruction		Note	Weight	Execution time (usec)	Weighted time (usec)
MVBI	byte,reg		0.0988	0.900	0.089
JC	cond,jdisp	1	0.0931	1.500	0.140
JC	cond,jdisp	2	0.0806	1.350	0.109
MVFN	(reg),(reg)	3	0.0018	18.900	0.034
MVA	addr4,reg	4	0.0196	1.850	0.036
STM	reg,addr4&lbr.,abcnt&rbr.		5,6	0.0092	10.500 0.097
LMB	addr4	7,6	0.0112	9.550	0.107
TWI	word,addr4	7,8	0.0151	3.900	0.059
CWI	word,addr4	9	0.0199	2.900	0.058
TBT	(reg,bitdisp)	10	0.0302	2.250	0.068
TBTR	(reg,bitdisp)	10	0.0170	3.600	0.061
J	jdisp		0.0841	1.500	0.126
DIS	ubyte	11	0.0411	2.700	0.111
BAL	longaddr,reg	12	0.0391	1.850	0.072
MVW	longaddr,reg	12	0.0500	2.300	0.115
MVW	reg,longaddr	12	0.0154	2.450	0.038
MVW	reg,reg		0.0976	1.350	0.132
AWI	word,reg&lbr.,reg&rbr.			0.0206	1.800 0.037
MVW	addr5,addr4	9,9	0.0237	4.300	0.102
MVD	addr5,addr4	9,9	0.0110	5.700	0.063
MVWS	reg,shortaddr		0.0372	2.300	0.086
AB	addr4,reg	13	0.0258	4.200	0.108
CW	addr4,reg	7	0.0158	2.600	0.041
MVW	reg,addr4	9	0.0307	2.600	0.080
MVWS	shortaddr,reg		0.0780	2.250	0.175
PSW	reg,addr4	9	0.0167	4.750	0.079
PW	addr4,reg	9	0.0167	4.850	0.081

Notes:

1. Taken
2. Jump not taken
3. N=17
4. AM=10, RB=0
5. AM=01
6. Specified general registers 0–4
7. AM=00
8. All selected bits=1's
9. AM=10, RB≠0
10. Test bit number 4
11. OP bit 15 (summary mask)
12. R2=0
13. AM=11, RB≠0

Execution times for individual instructions can be calculated from the following information.

Figure A-1 shows the additional time required when executing register/storage instructions or storage/storage instructions and assembler syntax for address mode.

- RS–the additional time for register/storage instructions

AM	RB	Time (microseconds)
00	Note 1	0.0
01	Note 1	0.45
10	Note 1	0.0
11	Note 1	1.05

- SS–the additional time for storage/storage instructions (all combinations of AM1 and AM2 are shown below)

AM1	RB	AM2	RB	Time (microseconds)
00	Note 1	00	Note 1	0.0
00	Note 1	01	Note 1	0.0
00	Note 1	10	Note 1	0.0
00	Note 1	11	Note 1	0.9
01	Note 1	00	Note 1	0.0
01	Note 1	01	Note 1	0.0
01	Note 1	10	Note 1	0.45
01	Note 1	11	Note 1	0.95
10	Note 1	00	Note 1	0.45
10	Note 1	01	Note 1	0.45
10	Note 1	10	Note 1	0.45

AM1	RB	AM2	RB	Time (microseconds)
10	Note 1	11	Note 1	1.35
11	=0	00	Note 1	1.25
11	=0	01	Note 1	1.25
11	=0	10	Note 1	1.55
11	=0	11	Note 1	2.15
11	≠0	00	Note 1	1.35
11	≠0	01	Note 1	1.35
11	≠0	10	Note 1	1.80
11	≠0	11	Note 1	2.40

- Assembler syntax for address modes (Note 2)

Assembler syntax		Address modes	
addr4	*addr5*	*AM, AM1, or AM2*	
(reg^{0-3})	(reg)	00	Note 1
(reg^{0-3})+	(reg)+	01	Note 1
addr	addr	10	Note 1
(reg^{1-3},waddr)	(reg^{1-7},waddr)	10	Note 1
addr*	addr*	11	RB=0
disp1(reg^{1-3},disp2)*	disp1(reg^{1-7},disp2)*	11	RB≠0
disp(reg^{1-3})*	disp(reg^{1-7})	11	RB≠0
(reg^{1-3})*	(reg^{1-7})*	11	RB≠0
(reg^{1-3},disp)*	(reg^{1-7},disp)*	11	RB≠0

Notes:

1. The value of RB does not affect the timings.
2. Register/storage instructions use assembler syntax *addr4* for address mode (AM).
   Storage/storage instructions use assembler syntax:
   - *addr5* for address mode for operand 1 (AM1)
   - *addr4* for address mode for operand 2 (AM2)

Figure A-1. Additional Instruction Times and Assembler Syntax for Address Mode

The symbols used in Figure A-1 and in the remainder of this appendix are defined as follows:

Symbol	Meaning
N	Number of bytes moved, filled, scanned, or compared
NS	Number of shifts
RS	Additional addressing-mode time for register/storage instructions
SS	Additional addressing-mode time for storage/storage instructions
*	Indirect address

The following notes and tables are used to determine instruction execution times:

Notes:

1. Subtract 1.2 microseconds if N=0, or subtract 0.45 microsecond if the operation terminates before the contents of general register 7=0.
2. Subtract 1.8 microseconds if N=0.
3. If the initial count contents of the selected general register are 0 or all 1's, the time is 2.75 microseconds.
   - If the selected general register count goes to 0, the time is 1.80 microseconds.
   - If the selected general register count does not go to 0, the time is 1.85 microseconds.
4. For each specific general register from 0 through 6, add an additional 0.65 microsecond per register.
5. MVFD or MVFN :
   - When N>3 and the addresses in the general registers (specified by the R1 and R2 fields) are both odd or both even and:
     - N is even, the MVFN instruction time is 6.75+0.675N microseconds. An additional 0.25 microsecond is added to the MVFN instruction time for the MVFD instruction.
     - N is odd, add 1 to N to make it even. The MVFN instruction time is 6.75+0.675N microseconds. An additional 0.25 microsecond is added to the MVFN instruction time for the MVFD instruction.
   - When N<4 and >0, the MVFN instruction time is 4.9+1.35N microseconds. The MVFD instruction time is the same.
   - When one address value in the general register (specified by the R1 and R2 fields) is even, the other address value is odd, and N>0, the MVFN instruction time is 4.9+1.35N microseconds. The MVFD instruction time is the same.
   - When N=0, the MVFN instruction time is 2.75 microseconds. The MVFD instruction time is the same.
6. Subtract 2.25 microseconds if N=0, subtract 0.90 microsecond if the operation terminates with the contents of general register 7=1, or subtract 0.45 microsecond if the operation terminates with the contents of general register 7>1.
7. For an SLT or SLTD instruction, if the first shift operation causes a carry, the instruction is terminated; add 0.9 microsecond.
   - If the instruction is SLT and it is terminated by a count of 0, add 0.9 microsecond.
   - If the instruction is SLTD and it is terminated by a count of 0, add 0.45 microsecond.
8. For each specified general register from 1 through 6, add an additional 0.65 microsecond per register. When only general register 0 is specified, add an additional 0.50 microsecond.
9. For a non-0 result, add 0.45 microsecond.

Individual instruction timings are subject to the same factors that reduce the instruction throughput: storage refresh, channel load interference, wait state, timer housekeeping, synchronization of the channel interface to the storage interface, and correction of a single-bit error in storage.

The following tables show the instructions in alphabetical sequence based on assembler mnemonics:

Instruction execution times			
Mnemonic	**Instruction name**	**Syntax**	**Execution time (microseconds)**
AA	Add Address	raddr,reg[,reg]	1.80
		raddr,addr4	3.40+RS
AB	Add Byte	reg,addr4	3.30+RS
		addr4,reg	3.15+RS
ABI	Add Byte Immediate	byte,reg	0.90
ACY	Add Carry Register	reg	1.35
AD	Add Doubleword	reg,addr4	4.55+RS
		addr4,reg	3.65+RS
		addr5,addr4	6.80+SS
AW	Add Word	reg,reg	1.40
		reg,addr4	3.00+RS
		addr4,reg	2.45+RS
		longaddr,reg	2.45
		longaddr*,reg	3.00
		addr5,addr4	4.95+SS
AWCY	Add Word with Carry	reg,reg	1.40
AWI	Add Word Immediate	word,reg[,reg]	1.80
		word,addr4	3.40+RS
B	Branch Unconditional	longaddr	1.85
		longaddr*	2.40
BAL	Branch and Link	longaddr,reg	1.85
		longaddr*,reg	2.40
BALS	Branch and Link Short	(reg,jdisp)*	2.05
		(reg)*	2.05
		addr*	2.05
BALX	Branch and Link External	vcon,reg	See Bal
BC	Branch on Condition	cond,longaddr	Branch not taken—1.80 Taken—1.85
		cond,longaddr*	Branch not taken—1.85 Taken—2.40
BCC	Branch on Condition Code	cond,longaddr	3.20
		cond,longaddr*	3.35
BCY	Branch on Carry	longaddr	See BC
BE	Branch on Equal	longaddr	See BC
BER	Branch on Error	longaddr	See BNCC
BEV	Branch on Even	longaddr	See BC
BGE	Branch on Arithmetically Greater Than or Equal	longaddr	See BNC
BGT	Branch on Arithmetically Greater Than	longaddr	See BNC
BLE	Branch on Arithmetically Less Than or Equal	longaddr	See BC

Instruction execution times			
Mnemonic	Instruction name	Syntax	Execution time (microseconds)
BLGE	Branch on Logically Greater Than or Equal	longaddr	See BNC
BLGT	Branch on Logically Greater Than	longaddr	See BNC
BLLE	Branch on Logically Less Than or Equal	longaddr	See BC
BLLT	Branch on Logically Less Than	longaddr	See BC
BLT	Branch on Arithmetically Less Than	longaddr	See BC
BMIX	Branch if Mixed	longaddr	See BC
BN	Branch on Negative	longaddr	See BC
BNC	Branch on Not Condition	cond,longaddr	Branch not taken—1.80 Taken—1.85
		cond,longaddr*	Branch not taken—1.80 Taken—2.40
BNCC	Branch on Not Condition Code	cond,longaddr	3.20
		cond,longaddr*	3.35
BNCY	Branch on No Carry	longaddr	See BNC
BNE	Branch on Not Equal	longaddr	See BNC
BNER	Branch if Not Error	longaddr	See BCC
BNEV	Branch on Not Even	longaddr	See BNC
BNMIX	Branch if Not Mixed	longaddr	See BNC
BNN	Branch on Not Negative	longaddr	See BNC
BNOFF	Branch if Not Off	longaddr	See BNC
BNON	Branch if Not On	longaddr	See BNC
BNOV	Branch on Not Overflow	cond,longaddr	Branch not taken—1.80 Taken—1.85
		cond,longaddr*	Branch not taken—1.80 Taken—2.40
BNP	Branch on Not Positive	longaddr	See BNC
BNZ	Branch on Not Zero	longaddr	See BNC
BOFF	Branch if Off	longaddr	See BC
BON	Branch if On	longaddr	See BC
BOV	Branch on Overflow	cond,longaddr	Branch not taken—1.80 Taken—1.85
		cond,longaddr*	Branch not taken—1.80 Taken—2.40
BP	Branch on Positive	longaddr	See BC
BX	Branch External	vcon	See B
BXS	Branch Indexed Short	(reg^{1-7},jdisp)	1.65
		(reg^{1-7})	1.65
		addr	1.65
BZ	Branch on Zero	longaddr	See BC

Instruction execution times				
Mnemonic	Instruction name	Syntax	Execution time (microseconds)	
CA	Compare Address	raddr,reg	1.80	
		raddr,addr4	2.90+RS	
CB	Compare Byte	addr4,reg	2.70+RS	
		addr5,addr4	4.10+SS	
CBI	Compare Byte Immediate	byte,reg	0.90	
CD	Compare Doubleword	addr4,reg	3.80+RS	
		addr5,addr4	5.60+SS	
CFED	Compare Byte Field Equal and Decrement	(reg),(reg)	3.95+1.75N	Note 1
CFEN	Compare Byte Field Equal and Increment	(reg),(reg)	See CFED	
CFNED	Compare Byte Field Not Equal and Decrement	(reg),(reg)	See CFED	
CFNEN	Compare Byte Field Not Equal and Increment	(reg),(reg)	See CFED	
CMR	Complement Register	reg[,reg]	1.80	
CPAKR	Copy Address Key Register	addr4	4.45+RS	
		reg	3.20	
CPCL	Copy Current Level	reg	2.45	
CPCLK	Copy Clock	reg	2.75	
CPCMP	Copy Comparator	reg	3.20	
CPCON	Copy Console Data Buffer	reg	2.30	
CPIMR	Copy Interrupt Mask Register	addr4	3.35+RS	
CPIPF	Copy In-Process Flags	addr4	7.10+RS	
CPISK	Copy Instruction Space Key	addr4	See CPAKR	
		reg		
CPLB	Copy Level Block	reg,addr4	11.65+RS	
CPLSR	Copy Level Status Register	reg	1.80	
CPOOK	Copy Operand 1 Key	addr4	See CPAKR	
		reg		
CPOTK	Copy Operand 2 Key	addr4	See CPAKR	
		reg		
CPPSR	Copy Processor Status and Reset	addr4	4.10+RS	
CPSK	Copy Storage Key (no op)	reg,addr4	2.70+RS	
CPSR	Copy Segmentation Register	reg,addr4	4.45+RS	
CW	Compare Word	reg,reg	1.40	
		addr4,reg	2.60+RS	
		addr5,addr4	4.10+SS	
CWI	Compare Word Immediate	word,reg	1.80	
		word,addr4	2.90+RS	

Instruction execution times			
Mnemonic	**Instruction name**	**Syntax**	**Execution time (microseconds)**
DB	Divide Byte	addr4,reg	5.20+RS Minimum 34.45+RS Maximum
DD	Divide Doubleword	addr4,reg	5.15+RS Minimum 57.45+RS Maximum
DIAG	Diagnose	ubyte	2.70 Minimum 5.65 Maximum
DIS	Disable	ubyte	Op bits 12 13 14 15 Times 0 0 0 0 2.70 0 0 0 1 2.70 0 0 1 0 5.60 0 0 1 1 5.60 0 1 0 0 3.15 0 1 0 1 3.60 0 1 1 0 6.05 0 1 1 1 6.50 1 0 0 0 2.70 1 0 0 1 2.70 1 0 1 0 5.60 1 0 1 1 5.60 1 1 0 0 3.15 1 1 0 1 3.60 1 1 1 0 6.05 1 1 1 1 6.50
DW	Divide Word	addr4,reg	4.75+RS Minimum 34.00+RS Maximum
EN	Enable	ubyte	Op bits 12 13 14 15 Times 0 0 0 0 2.70 0 0 0 1 5.40 0 0 1 0 4.95 0 0 1 1 7.65 0 1 0 0 3.60 0 1 0 1 6.30 0 1 1 0 5.85 0 1 1 1 8.55 1 0 0 0 4.95 1 0 0 1 7.65 1 0 1 0 4.95 1 0 1 1 7.65 1 1 0 0 5.85 1 1 0 1 8.55 1 1 1 0 5.85 1 1 1 1 8.55

Instruction execution times				
Mnemonic	**Instruction name**	**Syntax**	**Execution time (microseconds)**	
FFD	Fill Byte Field and Decrement	reg,(reg)	4.55+0.9N	Note 2
FFN	Fill Byte Field and Increment	reg,(reg)	See FFD	
IO	Operate I/O	longaddr	11.30 (Halt I/O) 5.10 (minimum Read) 6.00 (minimum Write)	
		longaddr*	11.90 (Halt I/O) 5.65 (minimum Read) 6.55 (minimum Write)	
IOPK	Interchange Operand Keys Keys		4.15	
IR	Interchange Registers	reg,reg	1.35	
J	Jump Unconditional	jdisp	1.50	
		jaddr	1.50	
JAL	Jump and Link	jdisp,reg	1.50	
		jaddr,reg	1.50	
JC	Jump on Condition	cond,jdisp	Jump not taken—1.35 Taken—1.50	
		cond,jaddr	Jump not taken—1.35 Taken—1.50	
JCT	Jump on Count	jdisp,reg		Note 3
		jaddr,reg		Note 3
JCY	Jump on Carry		See JC	
JE	Jump on Equal		See JC	
JEV	Jump on Even		See JC	
JGE	Jump on Arithmetically Greater Than or Equal		See JNC	
JGT	Jump on Arithmetically Greater Than		See JNC	
JLE	Jump on Arithmetically Less Than or Equal		See JC	
JLGE	Jump on Logically Greater Than or Equal		See JNC	
JLGT	Jump on Logically Greater Than		See JNC	
JLLE	Jump on Logically Less Than or Equal		See JC	
JLLT	Jump on Logically Less Than		See JC	

Instruction execution times				
Mnemonic	Instruction name	Syntax	Execution time (microseconds)	
JLT	Jump on Arithmetically Less Than		See JC	
JMIX	Jump if Mixed		See JC	
JN	Jump on Negative		See JC	
JNC	Jump on Not Condition	cond,jdisp	Jump not taken—1.35 Taken—1.50	
		cond,jaddr	Jump not taken—1.35 Taken—1.50	
JNCY	Jump on No Carry		See JNC	
JNE	Jump on Not Equal		See JNC	
JNEV	Jump on Not Even		See JNC	
JNMIX	Jump if Not Mixed		See JNC	
JNN	Jump on Not Negative		See JNC	
JNOFF	Jump if Not Off		See JNC	
JNON	Jump if Not On		See JNC	
JNP	Jump on Not Positive		See JNC	
JNZ	Jump no Not Zero		See JNC	
JOFF	Jump if Off		See JC	
JON	Jump if On		See JC	
JP	Jump on Positive		See JC	
JZ	Jump on Zero		See JC	
LEX	Level Exit	[ubyte]	7.65 14.40	Minimum Maximum
LMB	Load Multiple and Branch	addr4	6.30 + RS	Note 4
MB	Multiply Byte	addr4,reg	5.60+RS 10.10+RS	Minimum Maximum
MD	Multiply Doubleword	addr4,reg	7.40+RS 28.05+RS	Minimum Maximum
MVA	Move Address	addr4,reg raddr,addr4	1.85+RS 2.80+RS	
MVB	Move Byte	reg,addr4 addr4,reg addr5,addr4	3.15+RS 3.10+RS 4.05+SS	
MVBI	Move Byte Immediate	byte,reg	0.90	
MVBZ	Move Byte and Zero	addr4,reg	3.35+RS	
MVD	Move Doubleword	reg,addr4 addr4,reg addr5,addr4	3.30+RS 3.50+RS 5.25+SS	
MVDZ	Move Doubleword and Zero	addr4,reg	4.60+RS	
MVFD	Move Byte Field and Decrement	(reg),(reg)		Note 5
MVFN	Move Byte Field and Increment	(reg),(reg)		Note 5

Instruction execution times				
Mnemonic	Instruction name	Syntax	Execution time (microseconds)	
MVW	Move Word	reg,reg	1.35	
		reg,addr4	2.60+RS	
		addr4,reg	2.40+RS	
		reg,longaddr	2.45	
		reg,longaddr*	2.90	
		longaddr,reg	2.30	
		longaddr*,reg	3.00	
		addr5,addr4	3.85+SS	
MVWI	Move Word Immediate	word,reg	1.85+RS	
		word,addr4	2.80+RS	
MVWS	Move Word Short	reg,shortaddr	2.30	
		reg,shortaddr*	3.10	
		shortaddr,reg	2.25	
		shortaddr*,reg	2.75	
MVWZ	Move Word and Zero	addr4,reg	3.15+RS	
MW	Multiply Word	addr4,reg	5.15+RS	Minimum
			15.00+RS	Maximum
NOP	No Operation		1.50	
NWI	AND Word Immediate	word,reg[,reg]	1.80	
OB	OR Byte	reg,addr4	3.30+RS	
		addr4,reg	3.05+RS	
		addr5,addr4	4.70+SS	
OD	OR Doubleword	reg,addr4	4.55+RS	
		addr4,reg	3.65+RS	
		addr5,addr4	6.35+SS	
OW	OR Word	reg,reg	1.40	
		reg,addr4	3.00+RS	
		addr4,reg	2.45+RS	
		longaddr,reg	2.45	
		longaddr*,reg	3.00	
		addr5,addr4	4.45+SS	
OWI	OR Word Immediate	word,reg[,reg]	1.80	
		word,addr4	3.40+RS	
PB	Pop Byte	addr4,reg	5.05+RS	
PD	Pop Doubleword	addr4,reg	5.65+RS	
PSB	Push Byte	reg,addr4	5.30+RS	
PSD	Push Doubleword	reg,addr4	5.75+RS	
PSW	Push Word	reg,addr4	4.75+RS	
PW	Pop Word	addr4,reg	4.85+RS	
RBTB	Reset Bits Byte	reg,addr4	3.30+RS	
		addr4,reg	3.05+RS	
		addr5,addr4	4.70+SS	
RBTD	Reset Bits Doubleword	reg,addr4	4.55+RS	
		addr4,reg	3.65+RS	
		addr5,addr4	6.35+SS	

Instruction execution times				
Mnemonic	Instruction name	Syntax	Execution time (microseconds)	
RBTW	Reset Bits Word	reg,reg	1.40	
		reg,addr4	3.00+RS	
		addr4,reg	2.45+RS	
		longaddr,reg	2.45	
		longaddr*,reg	3.00	
		addr5,addr4	4.45+SS	
RBTWI	Reset Bits Word Immediate	word,reg[,reg]	1.80	
		word,addr4	3.40+RS	
SA	Subtract Address	raddr,reg[,reg]	1.80	
		raddr,addr4	3.40+RS	
SB	Subtract Byte	reg,addr4	3.30+RS	
		addr4,reg	3.15+RS	
SBTB	Set Bits Byte	reg,addr4	3.30+RS	
		addr4,reg	3.05+RS	
		addr5,addr4	4.70+SS	
SBTD	Set Bits Doubleword	reg,addr4	4.45+RS	
		addr4,reg	3.55+RS	
		addr5,addr4	6.35+SS	
SBTW	Set Bits Word	reg,reg	1.4	
		reg,addr4	3.00+RS	
		addr4,reg	2.45+RS	
		longaddr,reg	2.45	
		longaddr*,reg	3.00	
		addr5,addr4	4.45+SS	
SBTWI	Set Bits Word Immediate	word,reg[,reg]	1.80	
		word,addr4	3.40+RS	
SCY	Subtract Carry Indicator	reg	1.35	
SD	Subtract Doubleword	reg,addr4	4.70+RS	
		addr4,reg	3.80+RS	
		addr5,addr4	6.80+SS	
SEAKR	Set Address Key Register	addr4	5.35+RS	
		reg	4.15	
SECLK	Set Clock	reg	3.20	
SECMP	Set Comparator	reg	3.20	
SECON	Set Console Data Lights	reg	3.90	
SEIMR	Set Interrupt Mask Register	addr4	4.10+RS	
SEIND	Set Indicators	reg	1.80	
SEISK	Set Instruction Space Key	addr4	See SEAKR	
		reg		
SELB	Set Level Block	reg,addr4	13.15+RS	Minimum
			26.65+RS	Maximum
SEOOK	Set Operand 1 Key	addr4	See SEAKR	
		reg		

Instruction execution times				
Mnemonic	Instruction name	Syntax	Execution time (microseconds)	
SEOTK	Set Operand 2 Key	addr4	See SEAKR	
		reg		
SESK	Set Storage Key	reg,addr4	2.70	
SESR	Set Segmentation Register	reg,addr4	5.95+RS	
SFED	Scan Byte Field Equal and Decrement	reg,(reg)	5.00+0.95N	Note 6
SFEN	Scan Byte Field Equal and Increment	reg,(reg)	See SFED	
SFNED	Scan Byte Field Not Equal and Decrement	reg,(reg)	See SFED	
SFNEN	Scan Byte Field Not Equal and Increment	reg,(reg)	See SFED	
SLC	Shift Left Circular	cnt16,reg	3.60	
		reg,reg	3.60	
SLCD	Shift Left Circular Double	cnt31,reg	6.85	Minimum
			7.30	Maximum
		reg,reg	6.85	Minimum
			7.30	Maximum
SLL	Shift Left Logical	cnt16,reg	4.05	Minimum
			4.95	Maximum
		reg,reg	4.05	Minimum
			4.95	Maximum
SLLD	Shift Left Logical Double	cnt31,reg	7.70	Minimum
			9.55	Maximum
		reg,reg	5.40	Minimum
			9.10	Maximum
SLT	Shift Left and Test	reg,reg	4.05+0.45NS	Note 7
SLTD	Shift Left and Test Double	reg,reg	4.05+0.9NS	Note 7
SRA	Shift Right Arithmetic	cnt16,reg	4.05	
		reg,reg	4.05	
SRAD	Shift Right Arithmetic Double	cnt31,reg	6.85	Minimum
			7.30	Maximum
		reg,reg	4.05	Minimum
			7.30	Maximum
SRL	Shift Right Logical	cnt16,reg	3.60	
		reg,reg	3.60	
SRLD	Shift Right Logical Double	cnt31,reg	6.75	Minimum
			6.80	Maximum
		reg,reg	4.05	Minimum
			6.80	Maximum
STM	Store Multiple	reg,addr4[,abcnt]	6.80+RS	Note 8
STOP	Stop	[ubyte]	2.70	
SVC	Supervisor Call	ubyte	18.35	

Instruction execution times				
Mnemonic	Instruction name	Syntax	Execution time (microseconds)	
SW	Subtract Word	reg,reg	1.40	
		reg,addr4	3.00+RS	
		addr4,reg	2.60+RS	
		longaddr,reg	2.60	
		longaddr*,reg	3.15	
		addr5,addr4	4.95+SS	
SWCY	Subtract Word with Carry	reg,reg	1.40	
SWI	Subtract Word Immediate	word,reg[,reg]	1.80	
		word,addr4	3.40+RS	
TBT	Test Bit	(reg,bitdisp)	2.25	
TBTR	Test Bit and Reset	(reg,bitdisp)	3.60	
TBTS	Test Bit and Set	(reg,bitdisp)	3.60	
TBTV	Test Bit and Invert	(reg,bitdisp)	3.60	
TWI	Test Word Immediate	word,reg	2.25	Note 9
		word,addr4	3.50+RS	Note 9
VR	Invert Register	reg[,reg]	1.35	
XB	Exclusive OR Byte	reg,addr4	3.30+RS	
		addr4,reg	3.05+RS	
XD	Exclusive OR Doubleword	reg,addr4	4.55+RS	
		addr4,reg	3.65+RS	
XW	Exclusive OR Word	reg,reg	1.40	
		reg,addr4	3.00+RS	
		addr4,reg	2.45+RS	
		longaddr,reg	2.45	
		longaddr*,reg	3.00	
XWI	Exclusive OR Word Immediate	word,reg[,reg]	1.80	

Floating-point instruction execution times				
Mnemonic	Instruction name	Syntax	Execution time (microseconds)	
CPFLB	Copy Floating Level Block	freg,addr4	18.55+RS	
FA	Floating Add	addr4,freg	7.60+RS	Minimum
			19.55+RS	Maximum
		freg,freg	7.80	Minimum
			17.80	Maximum
FAD	Floating Add Double	addr4,freg	12.05+RS	Minimum
			21.20+RS	Maximum
		freg,freg	8.80	Minimum
			23.35	Maximum
FC	Floating Compare	freg,freg	9.25	Minimum
			11.95	Maximum
FCD	Floating Compare Double	freg,freg	9.80	Minimum
			12.95	Maximum
FD	Floating Divide	addr4,freg	8.60+RS	Minimum
			75.30+RS	Maximum
		freg,freg	6.85	Minimum
			73.55	Maximum
FDD	Floating Divide Double	addr4,freg	9.90+RS	Minimum
			221.00+RS	Maximum
		freg,freg	6.95	Minimum
			218.05	Maximum
FM	Floating Multiply	addr4,freg	7.70+RS	Minimum
			60.05+RS	Maximum
		freg,freg	5.90	Minimum
			58.75	Maximum
FMD	Floating Multiply Double	addr4,freg	9.15+RS	Minimum
			105.30+RS	Maximum
		freg,freg	5.90	Minimum
			102.35	Maximum
FMV	Floating Move	addr4,freg	5.40+RS	
		freg,freg	4.20	
		freg,addr4	5.50+RS	
FMVC	Floating Move and Convert	addr4,freg	6.10+RS	Minimum
			8.35+RS	Maximum
		freg,addr4	7.70+RS	Minimum
			9.05+RS	Maximum
FMVCD	Floating Move and Convert Double	addr4,freg	6.75+RS	Minimum
			9.90+RS	Maximum
		freg,addr4	8.40+RS	Minimum
			11.25+RS	Maximum
FMVD	Floating Move Double	addr4,freg	6.25+RS	
		freg,freg	4.20	
		freg,addr4	6.70+RS	

Floating-point instruction execution times				
Mnemonic	Instruction name	Syntax	Execution (microseconds)	
FS	Floating Subtract	addr4,freg	10.05+RS	Minimum
			20.00+RS	Maximum
		freg,freg	8.25	Minimum
			18.25	Maximum
FSD	Floating Subtract Double	addr4,freg	12.05+RS	Minimum
			26.75+RS	Maximum
		freg,freg	8.80	Minimum
			23.80	Maximum
SEFLB	Set Floating Level Block	addr4,freg	22.15+RS	

Appendix B. Software Notes

Notes for Processor

Note 1

Instruction streams that are self-modifying cannot be guaranteed. An executing instruction cannot modify the next sequential instruction stream word.

Example:

```
        MVWI    hex 5000,LOC1
LOC1    J       THERE
```

This example illustrates a self-modifying instruction stream. The Move Word Immediate (MVWI) instruction moves a hex 5000 to LOC1. A hex 5000 in machine code is a no-operation instruction (NOP). The program is attempting to execute a NOP instruction, instead of the Jump (J) instruction. This type of programming is not permitted on the 4956 Processor.

Note 2

Four priority interrupt levels (0–3) are implemented in the processor. A Prepare command to levels 4–15 is executed so that condition code reporting occurs; however, the Prepare command is not executed at the addressed device and effectively results in a no-operation.

Note 3

There is no storage-protect feature in the 4956 processor. Execution of the Set Storage Key (SESK) and Copy Storage Key (CPSK) instructions results in a no-operation.

Note 4

Byte write operations to storage locations that contain two bit errors are not executed by the storage card. The storage location must first be corrected with a word write before the byte write operation can be executed correctly.

> **Note:** After a successful power-on reset, all storage locations are initialized to some value.

Notes for Diskette Drive

Note 1

Diskette recording format (FM or MFM) is specified by byte 72 of cylinder 0, head 0, sector 7. This record is read whenever a new diskette is inserted in the diskette drive. Any format type command to a track other than cylinder 0, head 0 sets FM or MFM with no check as to whether the diskette is FM or MFM. The format type commands are:

- Format Track
- Format Track Defective
- Verify Format Track
- Set FM/MFM Bit

Note 2

Do not leave the FM/MFM byte (byte 72 cylinder 0, head 0, sector 7) in an uninitialized state. If this byte is left uninitialized and an error occurs or power is lost so that the diskette has to be read back, the diskette drive will not be able to determine whether the diskette is formatted in MFM. This could result in read/write errors and/or seek errors.

Note 3

Track 0 is always recorded in FM format and 128 byte sectors. The diskette drive automatically reads the FM/MFM byte whenever a new diskette is inserted.

Note 4

Seek errors can occur if the SE bit is set to 0. If the SE bit is a 0 and a defective track is encountered or passed over as a result of an automatic seek command, an exception interrupt is reported with bit 6 of cycle-steal status word 6 (head seek error) posted.

If the SE bit is a 1 and the same conditions are present, the device does retries. This includes stepping to the next track and looking for the correct ID twice, and then recalibrating the diskette and repeating the procedure. Since the maximum number of defective tracks per diskette is two, the desired track would be found.

Note 5

A defective track must have at least one ID that is readable and that can be decoded as being defective. The format track defective command performs this test automatically after each format defective command is complete, and the microprocessor determines if one sector is readable and contains the correct defective ID data.

In the case where no sector IDs are readable or are wrong when read, the microprocessor flags a format track defective error in bit 7 of cycle-steal status word 7. In this case, the program should do a Read Sector ID Map command and examine all the IDs, looking for one that is correct. If none of the IDs are found readable and correct, the diskette must then be replaced.

Note 6

A Recalibrate Head command must follow a Start Diagnostic command.

Note 7

Controller end is not reported by the diskette drive. When an Operate I/O instruction is issued, a controller busy condition code may be issued to a second Operate I/O instruction for a maximum time of 1 millisecond. If a tight loop on controller busy or if a loop with no provisions for other interrupts is incorporated into a program, system performance may be degraded. The program should contain provisions to handle a defective attachment card that has 'controller busy' stuck in the active state.

Note 8

On the first data transfer to a diskette, either an implied seek or one of the format commands including Set FM/MFM Bit must be used.

Note 9

Spiral read/write should not be used on cylinder 0. This is because cylinder 0 is always recorded in FM (N=0) and the remainder of the diskette may be recorded differently. During a spiral read/write under this condition, a "No Record Found" could be returned.

Note 10

It is strongly recommended that all software conform to the label and data set specifications found in *IBM Diskette Reference Manual*, GA21-9182.

Note 11

Implied seeks should be used with spiral commands. Otherwise a "No Record Found" may be returned.

Appendix C. Error Log

Purpose

The error log provides a history of errors that have occurred since power-on. The log is useful in isolating the cause of a particular problem. At power-on, all 64 entries in the error log are cleared; at any other time, the error log contains the most current 64 errors of the types of errors that are logged. This information is readily available to the CSR via the programmer's console or the Diagnose instruction.

Structure

Error log entries are placed in 64 local store registers (addresses hex 40 through 7F). The following errors are included in the log:

- Processor ISA check
- Specification check
- Storage parity error (double error detected in storage)
- I/O check with sequence indicator
- CPU control check
- Timer overrun (refer to "Stall Detector/Timer Overrun Error")
- Storage protect check

When one of these errors occurs, the SDR contents (the last word read or written by the processor to main storage) are loaded into local store address hex 05. Also included in the log are Operate I/O condition codes (busy after reset, command reject, and interface data check), and priority interrupt condition codes (exception, and attention and exception). To access the log, the lock key and a special code are used to display the error entries in the data lights. The Diagnose instruction may be used to dump the error log or local store address hex 05 to main storage for viewing.

Entries are placed in the error log starting with the first at local store address hex 40. Subsequent entries are placed in the log by incrementing the last previous address by 1. Local store address hex 0F contains the address of the last entry in the error log, in the low byte bits 8–15. (The high byte bits 0–7 contain machine parameters and must not be modified.) When all 64 entries have been written, the log wraps; that is, the next entry replaces the first. During power on, the local store address hex 0F is set to hex 003F and the error log is initialized to 0's. Thus, a location that contains hex 0000 indicates the end of the log, unless it is the first entry of a two-location log entry.

All errors are grouped into two types: machine check and program check. Timer overrun, CPU control check, I/O check, and storage parity error are machine check errors; processor ISA check, specification check, and storage protect check are considered program errors. Multiple errors within machine check and program check types are recorded. However,if any machine check condition occurs, program check errors are ignored.

Machine Check

Multiple machine check errors are logged in the following order:

1. Timer overrun (refer to "Stall Detector/Timer Overrun Error")
2. CPU control check
3. I/O check with sequence indicator
4. Storage parity error

If an I/O check occurs and the sequence indicator is set, no device address is logged (bits 8–15 of the log entry are set to 0's). If the sequence indicator is not set, the device address is placed in bits 8–15 of the log entry.

Program Check

Program check (processor ISA check, specification check, and storage protect check) generates two log entries for each error. The first entry to be logged is the contents of the CIAR. The second entry identifies the type of error, the supervisor state, and the last active address key. For all errors with two log entries, the second entry begins with binary 11. When displayed from the console, the second entry precedes the first, since the order of the display is from the last entry written.

Stall Detector/Timer Overrun Error

The stall detector is a unique and independent hardware timing mechanism in the 4956 processor. Its purpose is to check data integrity of the clock registers (correct time). If these registers are not updated every millisecond, the stall detector is activated. In addition, certain erroneous microcode branches that result in excessively high instruction execution time and impaired data integrity are detected by the stall detector.

When the stall detector activates, a machine check interrupt occurs with bit 10 (CPU control check) in the PSW set to 1. A timer overrun condition is then entered in the error log.

Format of Log Entries

Machine Check

Timer overrun: 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 (hex 1000)

CPU control check: 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 (hex 0800)

I/O check:

Bits 0–3	0 1 0 0
Bit 4	Sequence indicator
Bits 5–7	0 0 0
Bits 8–15	Device address if available

Storage parity error (two log entries):

First entry: 16-bit address (contents of SAR)

Second entry:

Bits 0–3	1 1 1 0
Bit 4	0
Bit 5	Supervisor state
Bits 6–7	Level
Bit 8	0
Bits 9–11	LAAK (Note 1)
Bit 12	0
Bits 13–15	CAAK (Note 2)

Program Check

Program check contains two log entries:

First entry: 16-bit address (contents of CIAR)

Second entry:

Bits 0–3	1 1 0 0 (processor ISA check) 1 1 0 1 (specification check) 1 1 1 1 (storage protect check)
Bit 4	0
Bit 5	Supervisor state
Bits 6–7	Level
Bit 8	0
Bits 9–11	LAAK (Note 1)
Bit 12	0
Bits 13–15	CAAK (Note 2)

Notes:

1. LAAK (last active address key) contains the last address key (ISK, OP1K, or OP2K) used.
2. CAAK (current active address key) contains the current instruction space key (ISK) used.

Priority Interrupt Entries

Bits 0–3	0 1 1 0 (exception check) 0 1 1 1 (attention and exception check)
Bit 4	0
Bit 5	Supervisor state
Bits 6–7	Level
Bits 8–15	Device address

Operate I/O Entries

Bits 0–3	0 0 1 0 (busy after reset check) 0 0 1 1 (command reject check) 0 1 0 1 (interface data check)
Bit 4	0
Bit 5	Supervisor state
Bits 6–7	Level
Bits 8–15	Device address

Index

G

H

I

K

L

M

O

P

IBM Series/1 4956 Processor Models C and C10
Description

Order No. GA34-0230-1

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. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

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

Please use pressure sensitive or other gummed tape to seal this form.

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

GA34-0230-1
Printed in U.S.A.

Reader's Comment Form

Cut or Fold Along Line

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
Information Development, Department 28E
1803 (Internal Zip)
P.O. Box 1328
Boca Raton, Florida 33429-9960

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

IBM®

International Business Machines Corporation

GA34-0230-1

GA34-0230-1
Printed in U.S.A.