IBM

Series/1

SY34-0041-3

IBM Series/1
4955 Processor and Processor Features
Theory Diagrams

Preface

This manual describes the IBM Series/1 4955 Processor and Processor Features.

The information in this manual is intended as an aid for IBM customer engineers, customer maintenance personnel, and other maintenance personnel to understand the operational and functional characteristics of the IBM 4955. This manual may also be used for reference purposes and for training purposes.

Pin numbers, voltage levels, and timing conditions referred to in this manual should not be used for troubleshooting purposes. Refer to the engineering Machine Logic Diagram (MLD) pages (shipped with the unit) for detailed and exact troubleshooting information.

Related Publications

I/O devices are described in separate publications. These publications are shipped with the device from the plant of origin.

Refer to the *IBM Series/1 Common Features Theory Diagrams,* SY34-0091, for a description of the common features used with the IBM Series/1 processors.

Refer to the *IBM Series/1 Graphic Bibliography,* GA34-0055, for a complete reference to the IBM Series/1 publications.

Fourth Edition (April 1979)

This is a major revision of, and obsoletes, SY34-0041-2.

This revision removes all information pertaining to common features. Descriptions of the common features are contained in the publication *IBM Series/1 Common Features Theory Diagrams,* SY34-0091.

A technical change to the text or to an illustration is indicated by a vertical line to the left of of the change.

Use this publication only for the purpose stated in the Preface.

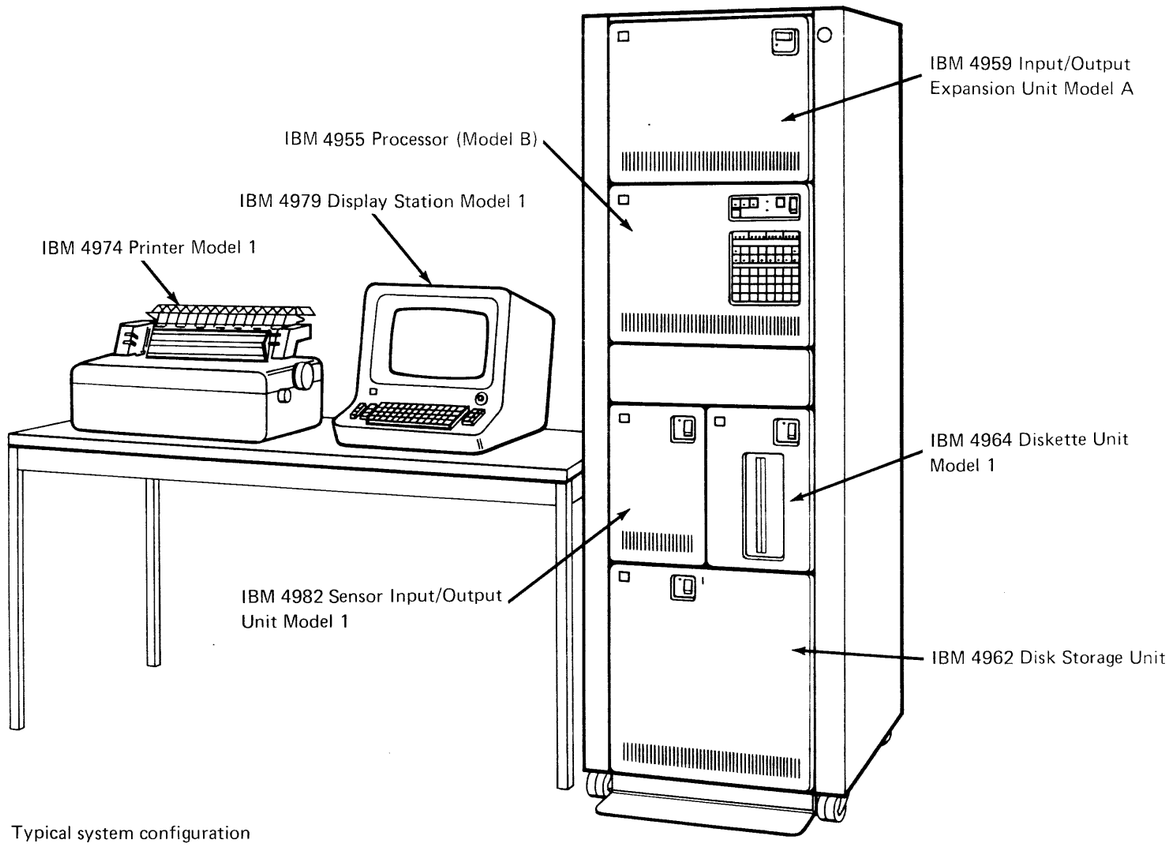
Changes are periodically made to the information herein; any such changes will be reported in subsequent revisions or Technical Newsletters.

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 services in your country.

Publications are not stocked at the address given below. Requests for copies of IBM publications should be made to your IBM representative 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 27T, P.O. Box 1328, Boca Raton, Florida 33432. IBM may use and 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.

Contents



Typical system configuration

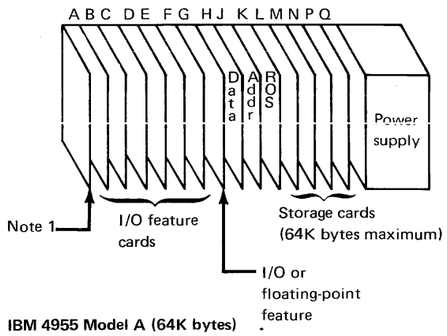
Chapter 1. Introduction

IBM 4955 Processor

The basic IBM 4955 processor, Model A, B, C, or D, includes a three-card processor, 16K bytes of storage (Models A and B) or 32K bytes of storage (Models C and D), a multilevel power supply, and a basic console. The basic IBM 4955 processor Model E includes a four-card processor, 64K bytes of storage, a multilevel power supply, and a basic console. The basic IBM 4955 processor Model F includes a four-card processor, 128K bytes of storage, a high-frequency power supply, and a basic console. The 4955 is a standard 48.3-cm (19-in) rack-mountable assembly.

4955 Model A

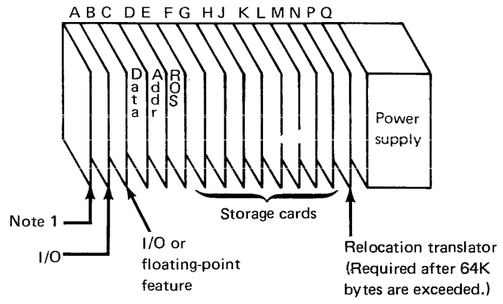
The 4955 Model A occupies the full width of the standard rack. It has the capacity for up to 64K bytes of storage and eight I/O features. The floating-point feature, if selected, must be plugged into socket H. Power is provided by a multilevel power supply.



IBM 4955 Model A (64K bytes)

4955 Model B

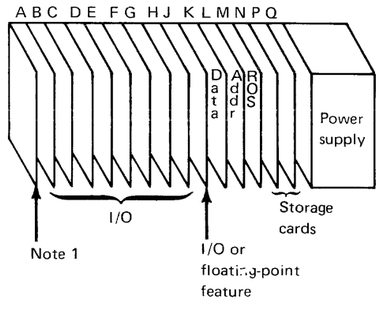
The 4955 Model B occupies the full width of the standard rack. It has the capacity for up to 128K bytes of storage (the storage address relocation translator feature is required for over 64K) and three I/O features. The floating-point feature, if selected, must be plugged into socket C. Power is provided by a multilevel power supply.



IBM 4955 Model B (128K bytes)

4955 Model C

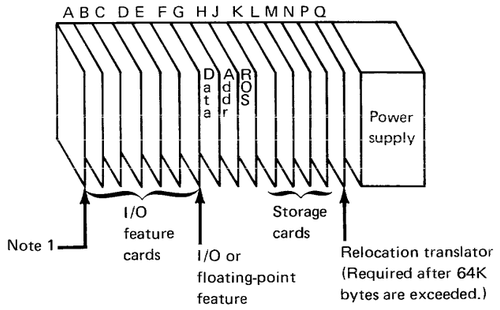
The 4955 Model C occupies the full width of the standard rack. It has the capacity for up to 64K bytes of storage and 10 I/O features. The floating-point feature, if selected, must be plugged into socket K. Power is provided by a multilevel power supply.



IBM 4955 Model C (64K bytes)

4955 Model D

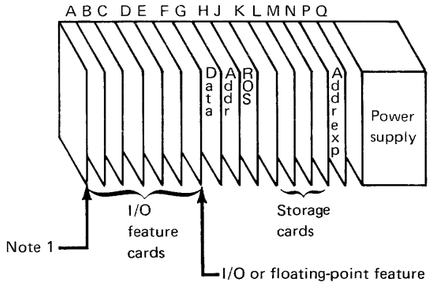
The 4955 Model D occupies the full width of the stand standard rack. It has the capacity for up to 128K bytes of storage (the storage address relocation translator feature is required for over 64K) and seven I/O features. The floating-point feature, if selected, must be plugged into socket G. Power is provided by a multilevel power supply.



IBM 4955 Model D (128K bytes)

4955 Model E

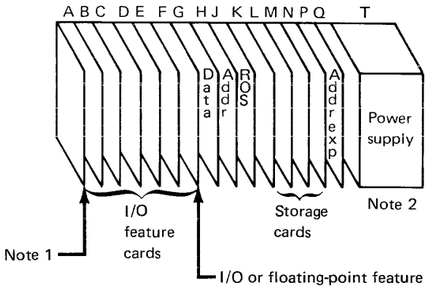
The 4955 Model E occupies the full width of the standard rack. It has the capacity for up to 256K bytes of storage and seven I/O features. The floating-point feature, if selected, must be plugged into socket G. Address translation for storage addresses greater than 64K bytes is basic and is provided by the address expansion card. Power is provided by a multilevel power supply.



IBM 4955 Model E (256K bytes)

4955 Model F

The 4955 Model F occupies the full width of the standard rack. It has the capacity for up to 512K bytes of storage and seven I/O features. The floating-point feature, if selected, must be plugged into socket G. Address relocation for storage addresses greater than 64K bytes is basic and is provided by the address expansion card. Power is provided by a high-frequency power supply.



IBM 4955 Model F (512K bytes)

Notes:

1. Card socket A is used for the I/O channel cables that carry the channel lines to an I/O expansion unit. When power isolation is required, the channel repower feature card is plugged into card socket A. If the cabling is not required, certain feature cards may be plugged into card socket A. These are:
   - Teletypewriter Adapter Feature, using TTL voltage levels
   - Teletypewriter Adapter Feature, using isolated current loop where customer supplies external ±12V power
   - Timers
   - 4982 Sensor Input/Output Unit Attachment
   - Integrated Digital Input/Output Non-Isolated
   - Customer Direct Program Control Adapter
2. The power supply plugs directly into card socket T.

IBM 4955 Processor Features

Standard Features

- FET (field-effect transistor) main storage.
  - Read or write time: 300 nanoseconds (660 ns required between two access cycles). Parity is generated by byte. Models A and B have 16K bytes (basic); Models C and D have 32K bytes (basic); Model E has 64K bytes (basic); Model F has 128K bytes (basic).
- Storage protection.
- Channel capability:
  - 21 I/O attachment cards supported over a distance of five feet (electrical) without repowering. (Channel repower feature is available for additional I/O capability.)
  - 256 I/O devices can be addressed.
- Four priority-interrupt levels with independent registers and status indicators for each level. Automatic and program-controlled level switching.
- Microprogram control. Microcycle time: 220 nanoseconds.
- Instruction set that includes: stacking and linking facilities, multiply and divide, variable-field-length operations, and a variety of arithmetic and branching instructions. Operates on bits, bytes, words, and doublewords.
- Automatic IPL on power up.
- Basic console.
- Cooling fans.
- 300-watt multilevel power supply for processor Models A, B, C, and D. Power-failure detect, thermal warning.
- 400-watt multilevel power supply for processor Model E. Power-failure detect, thermal warning.
- High-frequency power supply for processor Model F.

Special Features

- Storage address relocation translator, Models B and D (permits addressing of main storage beyond 64K bytes)
- Storage addition–16,384 bytes:
  - Provides storage in 16K-byte increments for processor Models A, B, C, and D.
  - Model A has a limit of four 16K cards (64K bytes total).
  - Model B has a limit of eight 16K cards (128K bytes total).
  - Model C and D have a limit of one 16K card, which which must be installed as the last storage card (any 32K cards would be installed between the 16K card and the processor cards).
- Storage addition–32,768 bytes:
  - Provides storage in 32K-byte increments for processor Models C, D, and E.
  - Model C has a limit of two 32K cards (64K bytes total).
  - Model D has a limit of four 32K cards (128K bytes total).
  - Model E has a limit of one 32K card, and it must be installed as the last storage card. The 32K card cannot be installed in card location L.
- Storage addition–65,536 bytes:
  - Provides storage in 64K-byte increments for processor Model E.
  - Model E has a limit of four 64K cards (256K bytes total).
- Storage addition–131,072 bytes: (Model F only)
  - Provides storage in 128K-byte increments for processor Model F only.
  - Model F has a limit of four 128K cards (512K bytes total).
- Programmer console
- Floating-point
- IBM 4999 Battery Backup Unit Models 1 and 2
- Channel repower feature
- Timers
- IBM 4959 I/O Expansion Unit Model A
- IBM 4997 Rack Enclosure Models 1 and 2

I/O Features

I/O features are added to the IBM Series/1 on an ongoing basis. Refer to *IBM Series/1 System Selection Guide*, GA34-0143, and the *IBM Series/1 System Summary*, GA34-0035, for a complete list of features.

Communications Feature

Refer to the *IBM Series/1 Communications Theory Diagrams* manual, SY34-0059, for a detailed description of the communication features.

IBM 4982 Sensor I/O Unit Features

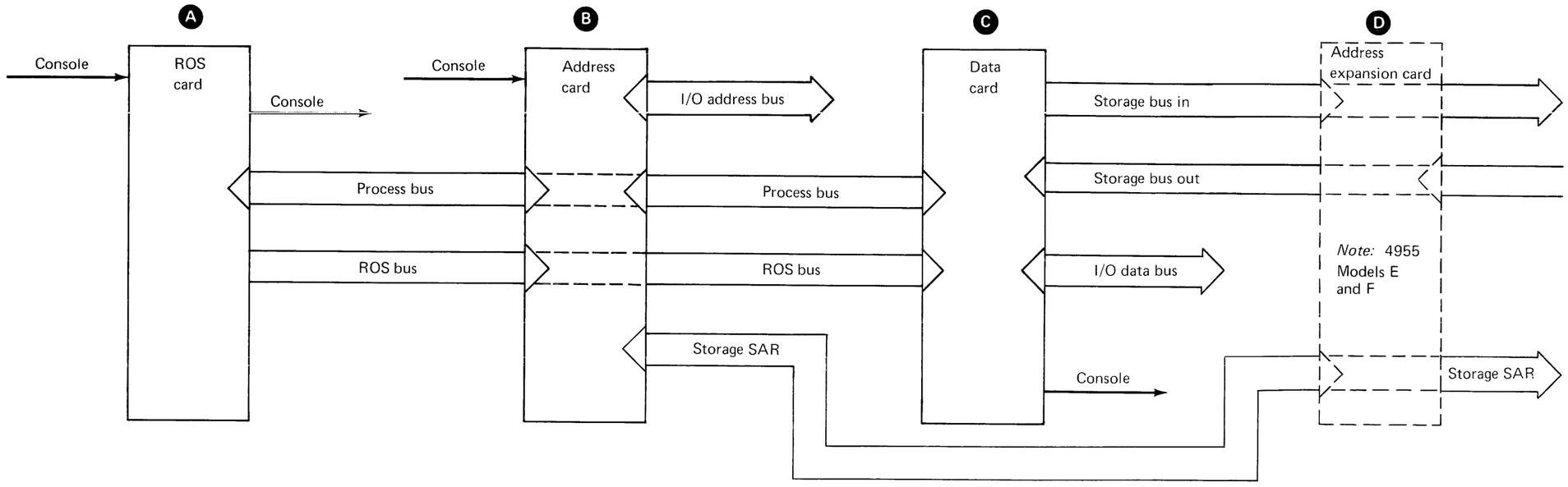
- Digital Input/Process Interrupt Non-Isolated
- Digital Input/Process Interrupt Isolated
- Digital Output Non-Isolated
- Analog Input Control
- Amplifier Multirange
- Multiplexer–Reed Relay
- Multiplexer–Solid State
- Analog Output

Refer to the *IBM Series/1 4982 Sensor I/O– Theory Diagrams* manual, SY34-0048, for a detailed description of these features.

Processor Cards

- **A** ROS card—provides the majority of controls for processor operations.
- **B** Address card—contains all the program-accessible hardware, such as data and status registers. Storage and I/O addresses are formed in this card. The majority of I/O controls are generated by this card.
- **C** Data card—performs all arithmetic and logical operations. Provides the gating for data to and from the I/O bus and the storage data buses.
- **D** Address expansion card—basic card in 4955 Models E and F. Performs address translation for inner and outer storage (Model E storage addresses 0-256K; Model F storage addresses 0-512K). Contains the segmentation registers and outer storage controls. Performs storage protection for all translated storage cycles. Controls dynamic storage refresh timings for both inner and outer storage of the Model F only.

Note: Only the major buses are shown here; many individual signal line connections exist between cards.



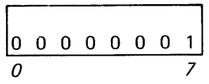
Main Storage

Main storage holds data and instructions for jobs to be processed on the system. The data and instructions are stored in units of information called bytes. Each byte consists of eight binary data bits plus an associated parity bit. Odd parity by byte is maintained throughout storage; even parity results in a machine-check condition.

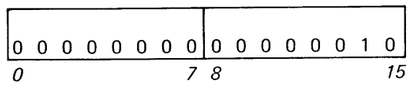
The bits within a byte are numbered consecutively, left to right, 0 through 7. When a format consists of multiple bytes, the numbering scheme is continued (for example, the bits in the second byte would be numbered 8 through 15). Leftmost bits are sometimes referred to as high-order bits and rightmost bits as low-order bits.

Bytes can be handled separately or grouped together. A *word* is a group of two consecutive bytes that begin on an even-byte boundary, and is the basic building block of instructions. A *doubleword* is a group of four consecutive bytes that begin on an even-byte boundary. Refer to Chapter 2 for a detailed description of main storage.

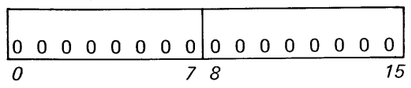
Byte

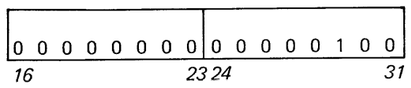


Word



Doubleword





Addressing Main Storage

Each byte location in main storage is directly addressable. Byte locations in storage are numbered consecutively, starting with location 0; each number is considered the address of the corresponding byte. Storage addresses are 16-bit unsigned binary numbers. This permits a direct addressing range of 65,536 bytes.

When the storage access is byte-type, a word is always addressed. In write operations, the processor generates a 'write byte 0' or 'write byte 1' line to select the proper byte to be written into. On read operations, a word is always read from storage, and any byte selection is done by the processor.

Address Range		
16-bit binary address	**Hexadecimal**	**Decimal**
0000 0000 0000 0000	0000	0
to	to	to
1111 1111 1111 1111	FFFF	65,535

Note: Addresses that overflow or underflow the addressing range always wrap modulo 65,536.

When the storage address relocation translator is installed and enabled in 4955 processor Models B and D, or the translator function of the address expansion card is enabled in 4955 processor Models E and F, the 16-bit address is used as a logical address to generate a 24-bit physical address.

Instruction and Operand Address Boundaries

As previously stated, all storage addressing is defined by byte location. Instructions can refer to bits, bytes, words, or byte fields as data operands. All word and doubleword operand addresses must be on even-byte boundaries. All word and doubleword operand addresses point to the most-significant (leftmost) byte in the operand. Bit addresses are specified by a byte address and a bit displacement.

All instructions must be on an even-byte boundary. This implies that the effective address for all branch-type instructions must be on an even byte boundary to be valid.

If any of the aforementioned rules are violated, a program check interrupt occurs with *specification check* set in the processor status word (PSW). The instruction is suppressed.

Storage Protection

The storage-protect mechanism protects against:

- Access (reading and writing) to a storage area not assigned to the current operation.
- Writing into a storage area designated as read-only.

The protection is accomplished by (1) comparing a storage key (associated with a storage block) against an address key (associated with the current operation), and (2) interrogation of a read-only bit in the storage key register.

When storage address relocation translation is enabled, the standard storage protection mechanism is disabled. The translator provides the storage protection while it is active. Refer to Chapter 2 for a detailed description of the storage protection mechanism and its operation.

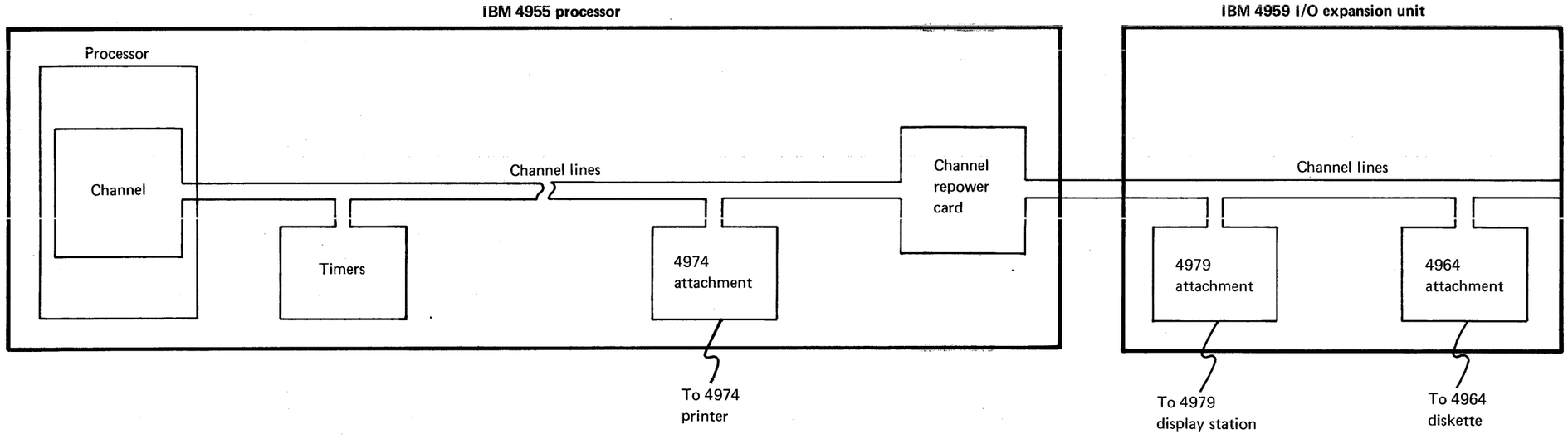
Channel

The 4955 channel logic which is contained within the processor cards, provides the timing, control, and the responses for the channel signal lines.

The channel is asynchronous, and allows the attachment of various I/O devices and processor feature cards. Asynchronous means that the response from a given device triggers the next sequential action rather than a specific timing condition. The channel provides comprehensive error-checking, including timeouts, sequence checking, and parity checking.

The channel consists of 68 lines, which include two bi-directional buses, a 16-bit address bus, and a 16-bit plus two parity bits data bus.

Refer to Chapter 2 for a detailed description and corresponding timing charts for the channel.

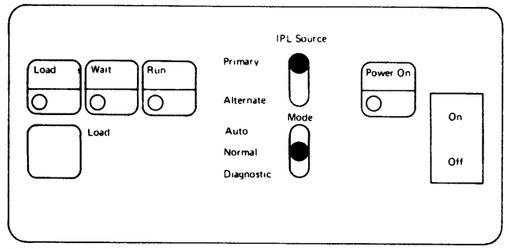


Console

Two configurations of consoles are available for the IBM 4955 Processor. The basic console is standard, and remains with the processor. The programmer console is an optional feature that is added to the processor when the option is selected. The programmer console is also available as a branch office tool.

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

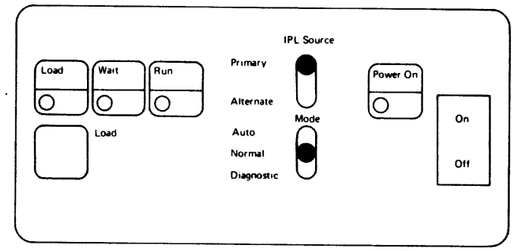
Refer to Chapter 2 for a detailed description of the console functions and operating features.

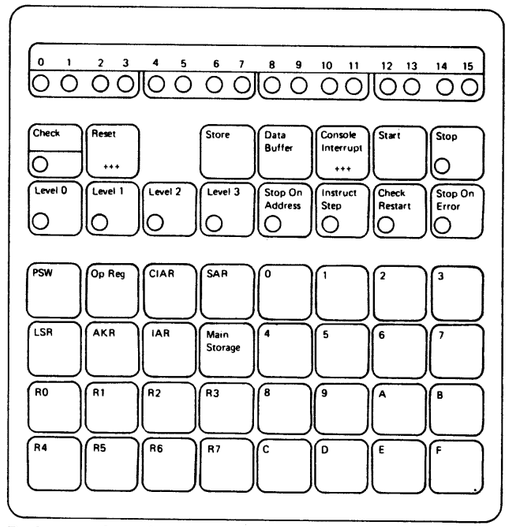


Basic console

The basic console and programmer console are 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.

Refer to Chapter 2 for a detailed description of the console functions and operating features.





Basic console and programmer console

Processor States

The Processor is always in one of the following machine states when power is on:

- Stop
- Wait
- Load
- Run

The current processor state may be determined by the indicators on the basic and programmer consoles. The Wait, Load, and Run indicators are located on the basic console. The Stop indicator is located on the programmer console. When the programmer console is not installed, stop state is indicated by the following:

- Load indicator off
- Wait indicator off
- Run indicator off
- Power on

Stop State

Stop state may be entered by any of the following methods:

- Pressing the Stop key on the programmer console.
- Executing the Stop instruction when the mode switch on the basic console is set to Diagnostic and the programmer console is installed.
- An address match occurs, when the processor is in stop-on-address mode. (programmer console)
- An error occurs when the processor is in stop-on-error mode. (programmer console)
- Pressing the Reset key on the programmer console.
- Power-on reset occurs when not in Auto IPL mode.

While the processor is in stop state, no interrupt requests are accepted. Stop state may be exited by the following:

- Pressing the Start key on the programmer console. After pressing the Start key, one instruction is executed before the processor accepts any interrupt requests.
- Pressing the Load key on the basic console.

Wait State

Wait state is entered when either a Level Exit (LEX) or Set Level Block (SELB) instruction, which turns off the current in-process bit, is executed. This is true provided that no other level of interrupt is pending. While in Wait state, the processor may accept interrupts.

Wait state is exited by pressing either the Stop, Reset, Load, or Console Interrupt key, or by the processor receiving and accepting an interrupt.

Load State

Load state is entered manually by pressing the Load key on the basic console, or automatically by powering up with the Mode switch set to the Auto IPL, or by remote IPL from a host system via a TP feature.

Load state is exited by successful completion of the initial program load function. Program execution begins at location 0 on level 0.

Run State

Run state is entered whenever the processor is not in the stop, load, or wait state. Run state indicates that the processor is executing instructions.

Program States

While in run state, the processor may be in one of two program states: supervisor state or problem state.

Supervisor State

In supervisor state, all instructions are valid. The storage protection mechanism is suppressed, and the program running in supervisor state has access to all of main storage. When storage address relocation translation is enabled, changes in address keys or segmentation registers may be necessary to allow access to all of main storage.

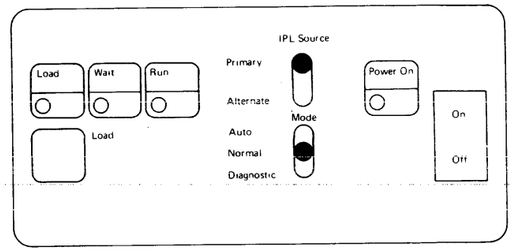
Supervisor state is entered:

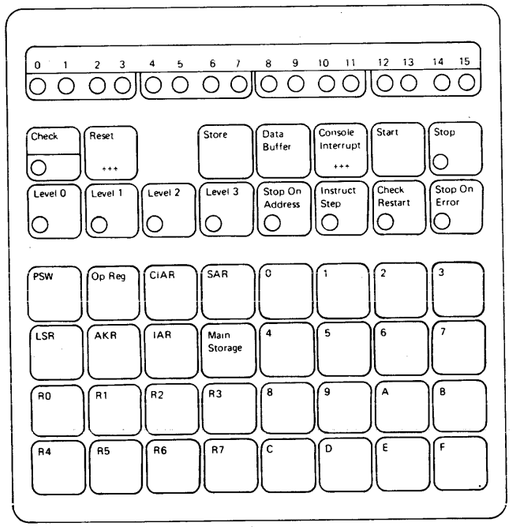
- When reset occurs
- When a Supervisor Call instruction is executed
- When a class interrupt occurs
- When IPL is successfully completed
- When an I/O interrupt is accepted

Supervisor state is exited by executing a SELB instruction with bit 8 of the LSR off.

Problem State

When the processor is running and not in supervisor state, problem state is in effect. While the processor is in problem state, the privileged instructions are invalid and any attempt to execute a privileged instruction results in a program-check interrupt, with bit 2 (privilege violate) of the PSW being set.





Interrupts

There are two types of interrupts implemented for the 4955 processor: I/O interrupts and class interrupts.

I/O Interrupts

I/O devices present interrupt requests on an interrupt priority level. Any one of four levels (0–3) can be assigned to the device. Level 0 has the highest priority, and any I/O device requesting an interrupt on this level interrupts any operation on levels 1, 2, or 3. Level 1 has the second highest priority and may interrupt operations on level 2 or 3. Level 2 interrupt requests may interrupt level 3 operations only. Level 3 has the lowest priority and cannot interrupt operations on any level.

I/O devices are assigned their priority level by a Prepare command, that is issued with an Operate I/O instruction. Refer to Chapter 2 for a detailed description of the Operate I/O instruction and the Prepare command.

Class Interrupts

Class interrupts do not cause a change in priority level and are processed on the currently active level. However, supervisor state is entered, Trace is reset, and all priority-interrupt requests are disabled.

Although class interrupts are serviced on the current active level, they are serviced in a priority order by exception condition. The seven types of class interrupts are:

- Machine Check
- Program Check
- Power/Thermal Warning
- Supervisor Call
- Soft Exception Trap
- Trace
- Console

Refer to "Class Interrupts" in Chapter 2 for a detailed description of class interrupts.

Input/Output (I/O) Operations

Communications and data transfers between the processor and input/output devices are referred to as I/O operations. The I/O operations are conducted over the channel signal lines between the channel logic and the I/O attachment cards.

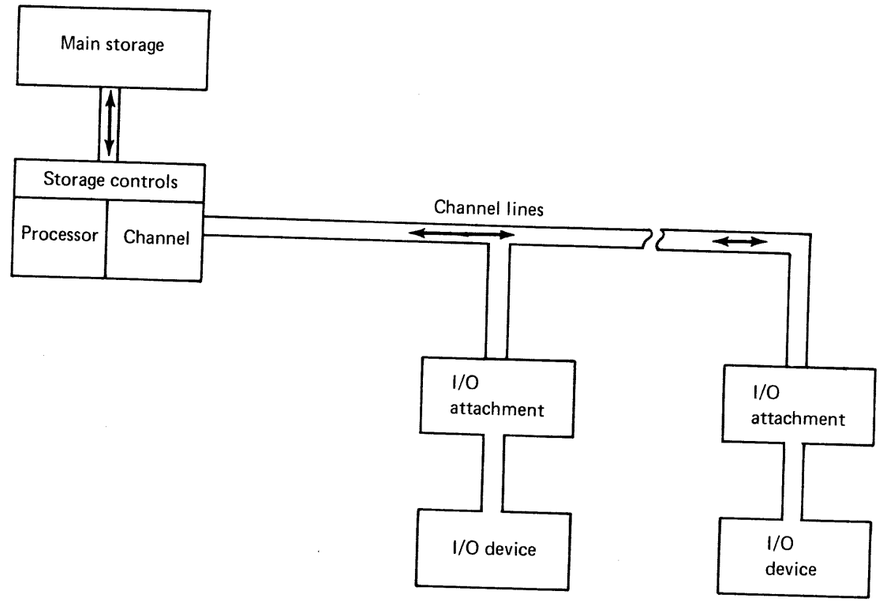
The I/O attachment cards provide special decode logic and controls that take the I/O channel information and reformat this information into data that an individual device can use. The control lines between the attachment card and the device are device-dependent.

The channel can access up to 256 unique device addresses. Four interrupt-priority levels are used to facilitate device service. Communications over the channel is either a direct program control (DPC) operation or a cycle-steal (CS) operation.

DPC operation — An immediate data transfer is made to or from the device for each Operate I/O instruction. The data may consist of one byte or one word. The operation may or may not terminate with an interrupt.

CS operation — An Operate I/O instruction can initiate cycle-stealing data transfers of up to 65,535 bytes between main storage and the device. Cycle-steal operations are overlapped with processing operations. Word or byte transfers, command chaining, burst mode, and program-controlled interrupt can be supported. All cycle-stealing operations terminate with an interrupt.

The Operate I/O instruction which initiates all I/O operations from the processor, is a privileged instruction and must be executed in supervisor state.

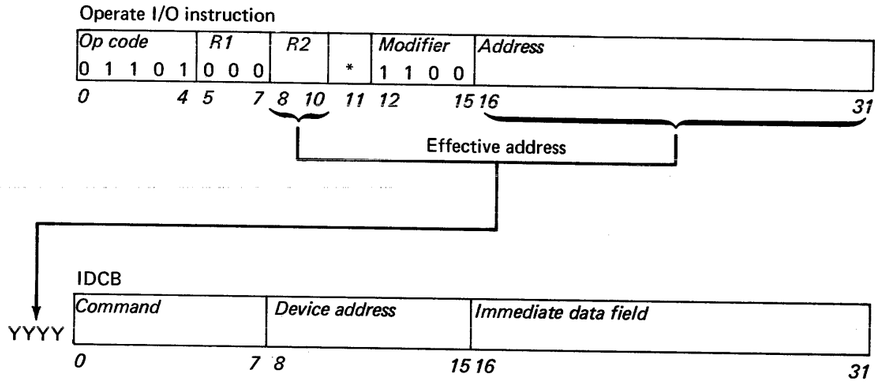


Operate I/O Instruction

The Operate I/O instruction initiates all I/O operations from the processor. This is a privileged instruction and may be fetched in supervisor state only. If this instruction is fetched in problem state, a privileged violate program check is set and a class interrupt is taken.

The effective address generated by this instruction points to an immediate device control block (IDCB). The IDCB contains the command field, device address field, and the immediate data field.

Command field	The first hex digit identifies the type of command. The second hex digit is the modifier.
Device address field	The device address field contains the device address. This address is hardware-plugged into the I/O attachment card.
Immediate data field	For DPC operations, this field contains the word to be written to the I/O device, or the word just read from the device. For cycle-steal operations, this field contains the address of the device control block (DCB).



*Indirect address bit

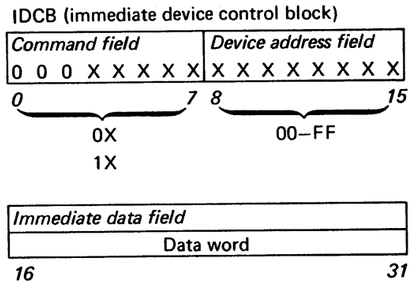
Command field	Command
0X (See Note.)	Read
1X	Read
20	Read ID
2X	Read status
3X	Reserved
4X	Write
5X	Write
60	Prepare
6X	Control
6F	Device reset
7X	Start
7F	Start Cycle-Steal Status
F0	Halt I/O

Note: X means that the modifier is device-dependent.

I/O Commands

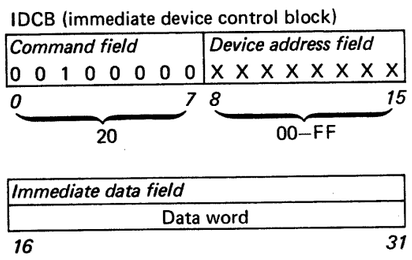
This section describes each I/O command and shows the related IDCB. The command field (bits 0–7) of the IDCB contains the hexadecimal representation of the command. An X in this field means that the value is device-dependent.

Read



The Read command transfers a word or byte from the addressed device to the data word of the IDCB. If a single byte is transferred, it is placed in bits 24–31 of the data word with bits 16–23 set to 0's. Correct parity is always maintained and checked for both bytes on the I/O channel. The individual device may use either the 0X or 1X type of read command.

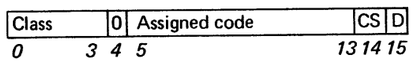
Read ID



The Read ID command transfers an identification word from the device to the data word of the IDCB. The device identification word contains physical information about the device, and is used by diagnostic programming to tabulate a system configuration.

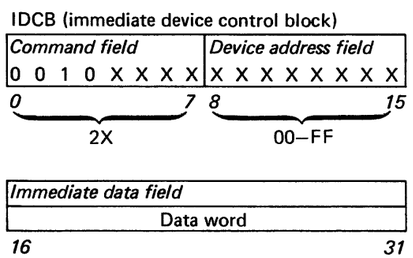
Note: This word is not related to the interrupt ID word associated with interrupt processing.

The device ID word format is:



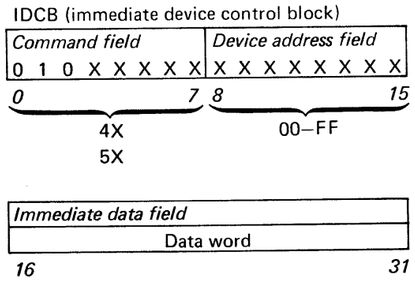
Bits 0–3	Assigned class code
Bit 4	Reserved; always 0
Bits 5–13	Assigned code
Bit 14	0 – not a cycle-steal device
	1 – cycle-steal device
Bit 15	0 – IBM device
	1 – OEM device

Read Status



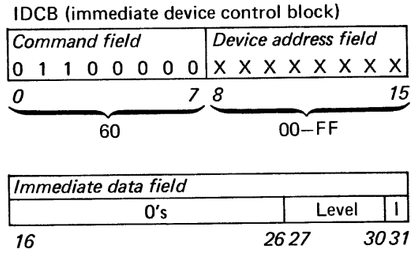
The Read Status command transfers a device status word from the device to the data word of the IDCB. Contents of the status word are device-dependent.

Write



The Write command transfers a word or byte to the addressed device from the data word of the IDCB. If a single byte is to be transferred, it must be placed in bits 24–31 of the data word, and bits 16–23 must be set to 0's.

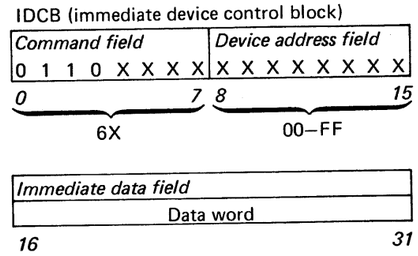
Prepare



The Prepare command transfers a word to the addressed device that controls the device's interrupt parameters. The word is transferred from the second word of the IDCB in the format shown. A priority interrupt level is assigned to the device by the level field. The I-bit (device mask) controls the device interrupt capability. If the I-bit equals 1, the device is allowed to interrupt.

Note: The 4955 does not recognize a priority level other than 0–3. Lost interrupts result if a device is prepared for a level other than 0–3.

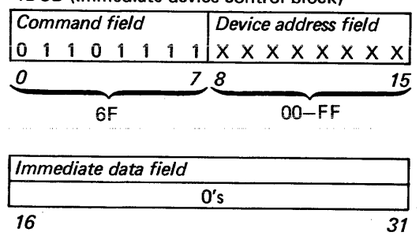
Control



The Control command initiates a control action in the addressed device. A word or byte transfer from the data word of the IDCB to the addressed device may or may not occur, depending on device requirements.

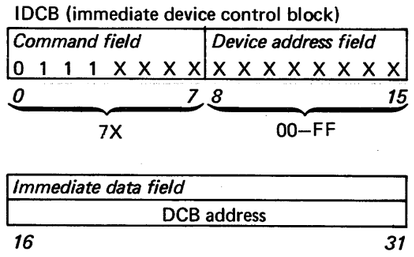
Note: The second word of the IDCB is fetched by the channel and placed on the I/O data bus (with good parity), even if the word is not required by the device.

Device Reset



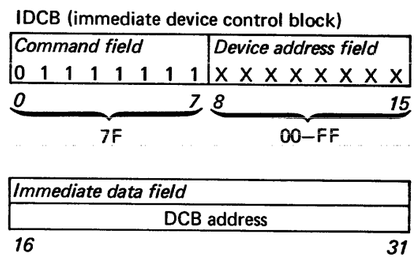
The Device Reset command resets the addressed device. A pending interrupt from this device (or a busy condition) is cleared. The device mask (I-bit) is not changed and there is no change to the assigned priority level for the device. The residual address (device status) and output sensor points are not affected. Parity checking of the IDCB data word is not performed.

Start



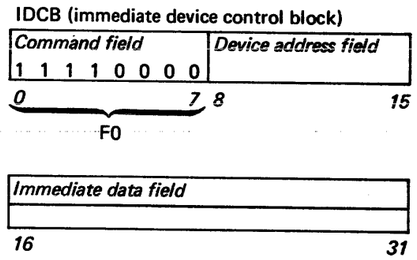
The Start command initiates a cycle-steal operation for the addressed device. The second word of the IDCB is transferred to the device. This word contains a 16-bit logical storage address of a device control block to be used by the device.

Start Cycle Steal Status



The Start Cycle Steal Status command initiates a cycle-steal operation for the addressed device. The purpose of this command is to collect status information relative to the previous cycle-steal operation. The second word of the IDCB, which is transferred to the device, contains a 16-bit logical address of a device control block.

Halt I/O



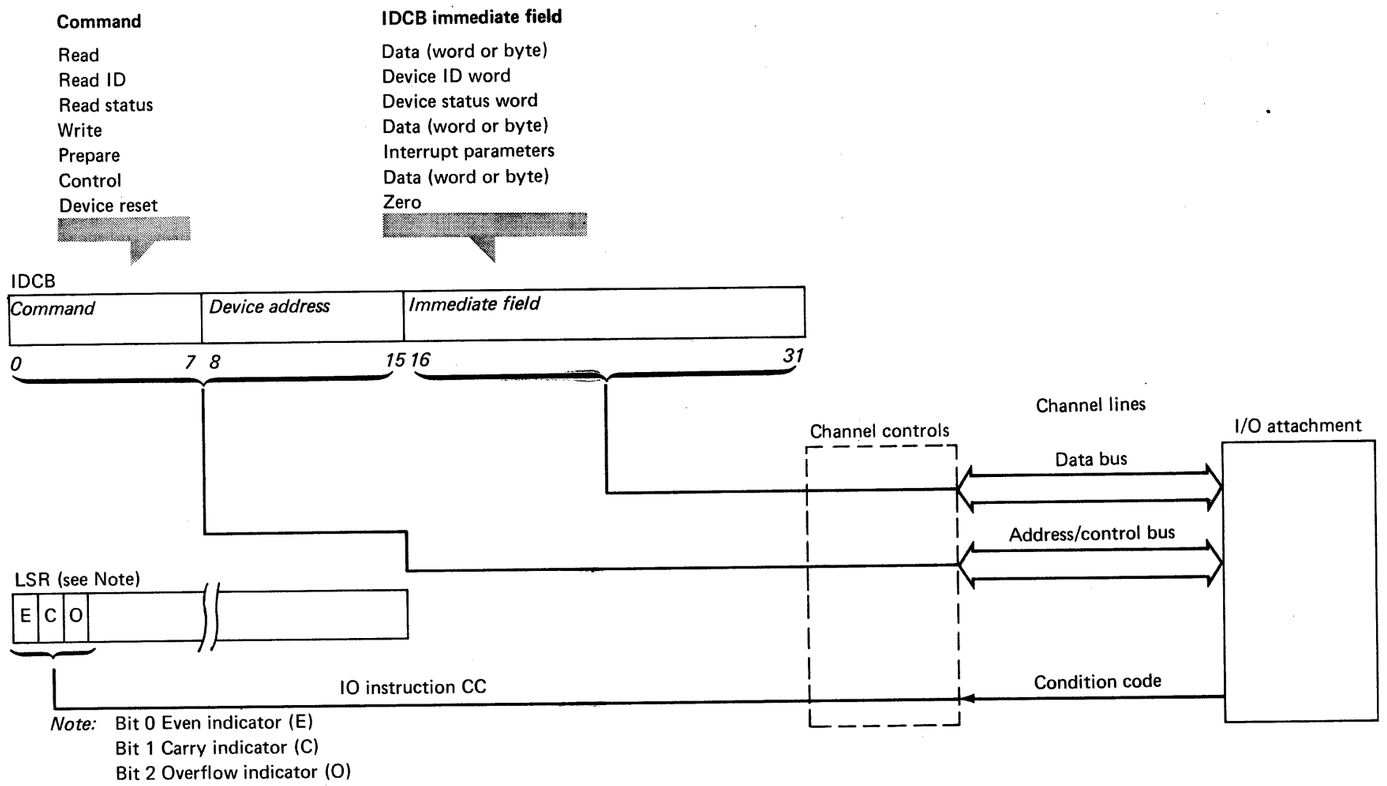
Halt I/O is a channel-directed command that causes a halt of all I/O activity on the channel. No data is associated with this command. All pending device interrupts are cleared. Device priority-interrupt-level assignments and device masks (I-bits) are unchanged.

Direct Program Control (DPC) Operation

A DPC operation involves an immediate transfer of data or control information to or from an I/O device. An Operate I/O instruction which must be executed for each data transfer, causes the following to occur:

1. The Operate I/O instruction points to an IDCB in main storage.
2. The I/O channel uses the IDCB to select the addressed device and to determine the operation to perform.
3. The channel gates data to or from the device.
4. The device presents a condition code to the processor.

Note: The DPC operation may terminate with a priority interrupt if the device has the capability. When the processor accepts the interrupt request, the device reports an interrupt condition code and interrupt ID word.



Cycle Steal

The cycle-steal mechanism allows data service to or from an I/O device while the processor is processing. This overlapped operation allows multiple data transfers to be initiated by one Operate I/O instruction. The processor executes the Operate I/O instruction; it then continues processing the instruction stream while the I/O device steals main storage data cycles when needed. The operation always terminates with a priority interrupt from the device. The channel resolves contention between multiple devices that request cycle-steal transfers. The channel also resolves contention for priority interrupts on the same level.

All cycle-steal operations are initiated by an Operate I/O instruction. The IDCB contains either a Start command or a Start Cycle Steal Status command. The immediate field of the IDCB contains the address of a device control block (DCB) that is fetched by the device. The DCB contains specific parameters of the cycle-steal operation.

The cycle-steal operation includes certain capabilities that are provided on a device-feature basis:

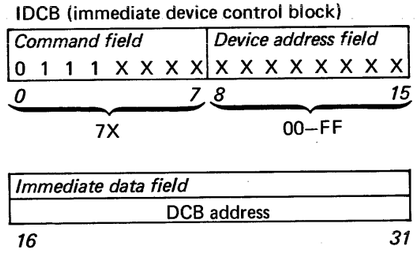
- Burst
- DCB command chaining
- Program-controlled interrupt (PCI)
- Suppress exception
- Storage addresses and data transfers by byte or word

All cycle-steal operations terminate with a priority interrupt. The device must have executed a successful Prepare command with the device mask (I-bit) enabled, before beginning the cycle-steal operation. Once a device is prepared, it remains prepared until it is reset by either a system reset or a power-on-reset.

Start Command

A cycle-steal operation begins after the successful execution of the Start command. The IDCB, pointed to by an Operate I/O instruction, has the format:

IDCB (immediate device control block)



The command modifier (X) is device-dependent. The DCB address always specifies a word boundary (even address), and is the starting storage address of the DCB. The address is used by the device, to fetch the DCB, using the cycle-steal mechanism.

Cycle-Steal Operation

A cycle-steal operation is presented in the following chart. Condition codes used in the chart are fully explained in "I/O Condition Codes and Status Information" in this chapter.

Cycle steal major steps	**Remarks**
Prepare I/O device instruction	1. Execute Operate I/O
DPC Operation	2. IDCB contains Prepare command and interrupt parameters.
	3. Device presents condition code 7 (satisfactory).
Start Cycle-Steal Status instruction	1. Execute Operate I/O
	2. IDCB contains Start command and points to a DCB.
	3. Device presents condition code 7 (satisfactory).

Cycle stealing starts when the device begins fetching the DCB from main storage.

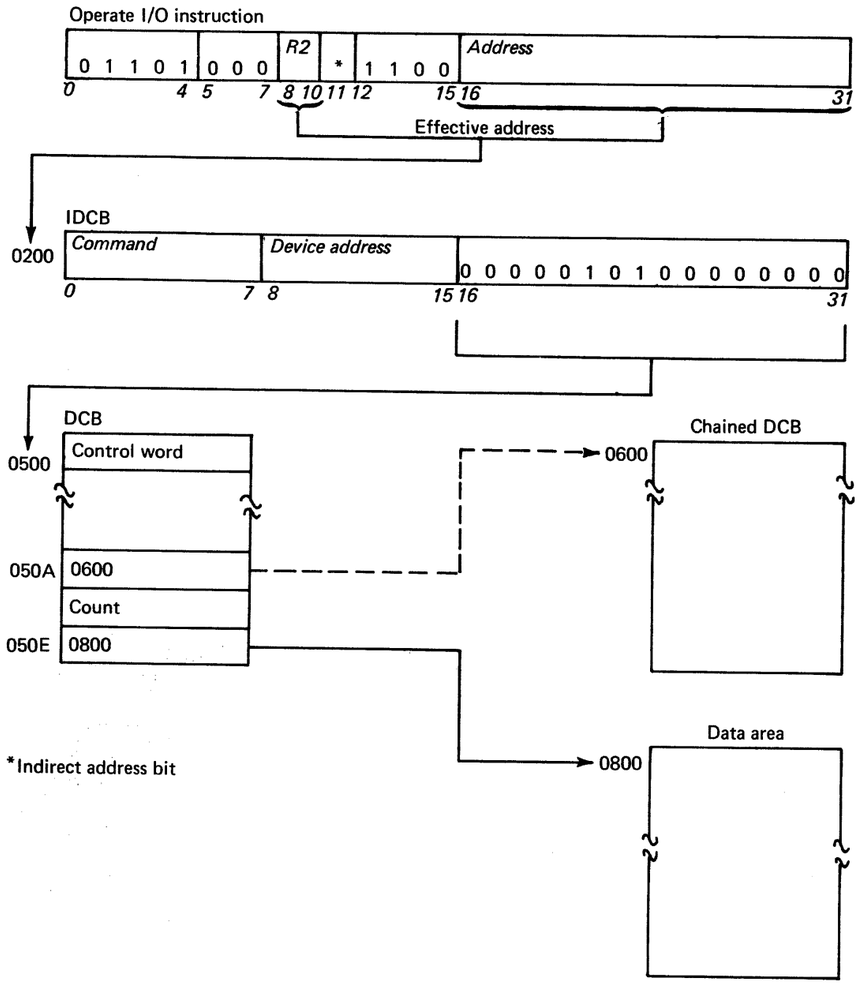
Device fetches DCB	1. Device uses the cycle-steal mechanism to fetch DCB.
	2. An active address key of 0 is used.
Data transfer	1. Data is transferred to or from the device in word or byte format.
	2. Transfer continues until the count in the DCB is exhausted.
	3. DCB specifies an address key for the data area.
Termination (no error-interrupt condition)	1. Device presents request.
	2. Channel sends a poll and accepts request.
	3. Device sends an interrupt ID word and interrupt condition code 3 (device end).
Termination (exception-interrupt condition)	1. Device presents request.
	2. Channel sends a poll and accepts request.
	3. Device sends an interrupt ID word and interrupt condition code 2 (exception).

Note: Other events that might occur during the cycle-steal operation are:

Chaining	1. Device completes the current DCB operation but does not present request.
	2. Device fetches the next DCB in the chain.
Program-controlled interrupt	1. Device fetches the DCB (PCI bit=1).
	2. Device initiates an interrupt ID word and interrupt condition code 1 (PCI).

Cycle-Steal Operation Data Flow

The following diagram shows the data flow for the cycle-steal operation.



Start Cycle-Steal Status Operation

This operation is initiated by a Start Cycle Steal Status command. The IDCB format is:

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 purpose of this operation is to obtain residual parameters from the device if the previous cycle-steal operation terminates due to an error or exception condition. The DCB format is the same as that for a normal cycle-steal operation, with some words and fields set to 0's.

DCB (device control block)

Word		
0	*Control word*	
	0 0 1 0 0 *Addr key*	0 0 0 0 0 0 0 0
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	
7	Data address	
	0	*15*

Residual Parameters

Data is transferred to main storage, starting at the data address specified in the DCB. This data consists of residual parameters and device-dependent status information, and has the following format:

Word 0	Residual address	
Word 1	Device cycle-steal status word 1	
Word 2	Device-dependent status word	
•	*0*	*15*
•		
•		

Residual Address. This word contains the main storage address of the last attempted cycle-steal transfer associated with a Start command. If an error occurs during a start cycle-steal status operation, this address is not altered. The residual address may be a data address, a DCB address, or a residual-status-block address, and is cleared only by a power-on reset. Following a power-on reset, the residual address is:

- 0000 (hex), for a byte-oriented device
- 0001 (hex), for a word-oriented device

This address is updated to the current cycle-steal storage address upon execution of cycle-steal transfers. For word transfers, the residual address points to the high-order byte of the word. Device reset, Halt I/O, machine check, and system reset have no effect on the residual address in the device.

Device Cycle-Steal Status Word 1. This word contains the residual byte count of a device. The residual byte count is initialized by the count field of a DCB associated with a Start command, and is updated as each byte of data is successfully transferred via a cycle-steal operation. The byte count is not updated by cycle-steal transfers into the residual status block, and it is not altered if an error occurs during a start cycle-steal status operation. It is reset by (1) power-on reset, (2) system reset, (3) device reset, (4) Halt I/O, or (5) machine-check condition.

Note: The contents of device cycle-steal status word 1 are device-dependent if the device does not (1) implement suppress incorrect length (SIL) or (2) store a residual byte count as part of its cycle-steal status.

Device-dependent. These are device-dependent status words. The number and contents of these words are specified by the individual device. Two conditions can cause bits to be set in the device-dependent status words:

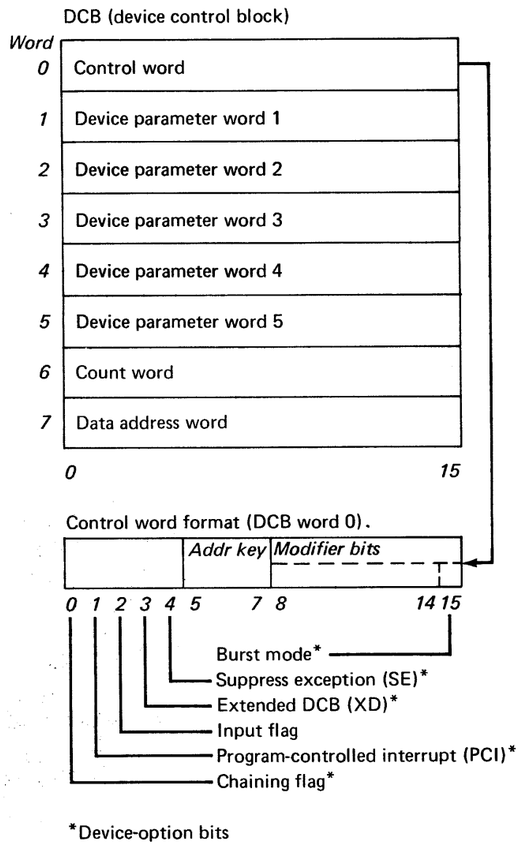
1. Execution of an I/O command that causes an exception interrupt
2. Asynchronous conditions in the device that indicate an error, an exception, or a state condition

These bits are reset as follows:

1. For condition 1, the bits are reset by the acceptance of the next I/O command (except Start Cycle Steal Status) following the exception interrupt. These bits are also reset by a power-on reset, system reset, or execution of a Halt I/O command.
2. For condition 2, the bits are reset on a device-dependent basis.

Device Control Block (DCB)

The DCB is an eight-word control block residing in the supervisor area of main storage that describes the specific parameters of the cycle stealing operation. The device fetches the DCB, using address protect key 0.



*Device-option bits

Control Word

Bit 0 *Chaining flag.* If this bit is equal to 1, a DCB chaining operation is indicated. After completing the current DCB operation, the device does not interrupt (excluding PCI interrupts). Instead, the device fetches the next DCB in the chain. Refer to "Chaining" in this chapter.

Bit 1 *Program-controlled interrupt (PCI).* If this bit is equal to 1, the device presents a program-controlled interrupt (PCI) at the completion of the DCB fetch. A pending PCI does not inhibit data transfers associated with the DCB. If the PCI is pending when the device encounters the next interrupt-causing condition, the PCI condition is discarded by the device and replaced with the new interrupt condition.

Bit 2 *Input flag.* The setting of this bit indicates, to the device, the direction of data transfers.

0 = output (main storage to device)

1 = input (device to main storage)

For bi-directional data transfers under one DCB operation, this bit must be set to 1. For control operations involving no data transfer, this bit must be set to 0.

Bit 3 *Extended DCB (XD).* This bit, when set to 1, specifies that the DCB is a non-standard type. The extended (non-standard) form of the DCB is used by certain devices attached to the I/O channel. The format of the device-dependent extended DCB is explained in the device publications.

Bit 4 *Suppress exception (SE).* If this bit is equal to 1, reporting of device-specified exception conditions are suppressed. The device continues the operation. The classes of exception conditions that can be suppressed are device-dependent (for example, an incorrect-length-record condition can be suppressed); an exception that occurs during a DCB fetch operation cannot be suppressed.

Bits 5–7 *Cycle-steal address key.* This key, presented by the device during data transfer, is used to ascertain storage access authorization.

Bits 8–14 *Modifier.* These bits may be used to describe functions unique to a particular device.

Bit 15 *Burst Mode.* If this bit is equal to 1, the transfer of data takes place in burst mode. This mode dedicates the channel to the device until the last data transfer associated with this DCB is completed. If burst mode is not supported by a device, this bit may be used for device-dependent information.

Device Parameter Words 1 and 2. These parameter words are device-dependent control words and are implemented as required. Refer to the individual device publications for definition.

Device Parameter Word 3. When PCI is specified, the high-order byte (bits 0–7) of this word is used for a DCB identifier. The device places the identifier in the interrupt information byte when the PCI is processed. The low-order byte (bits 8–15) is always device-dependent. The high-order byte is device-dependent when PCI is not specified.

Device Parameter Word 4. If suppress exception (SE) is used by a device, this word specifies a 16-bit main storage address called the status address. This address points to a residual status block that is stored by the device following completion of the DCB operation.

If suppress exception is not used by a device, a residual status block is not stored. In this case, parameter word 4 is device-dependent.

Note: Address key 0 is used when storing the residual status block.

Device Parameter Word 5. If the DCB chaining bit (bit 0 of the control word) is equal to 1, this word specifies a 16-bit main storage address of the next DCB in the chain. If chaining is not supported by the device, this parameter word is device-dependent.

Count Word. The count word contains a 16-bit unsigned integer that represents the number of data bytes to be transferred for the current DCB. Count is specified in bytes with a range of 0–65,535. The count specification must be even for word-only devices.

Data Address Word. This word contains the starting main storage address for the data transfer.

Residual Status Block. The size of the residual status block varies from two to sixteen words depending on the individual device. The first word contains the residual byte count. The second word contains device status flags. Additional words (maximum of two) contain device-dependent status information. Information contained within the residual status block may also be reported by the device during a start cycle-steal status operation. (Refer to the individual device publications for information pertaining to the device status.) If suppress exception is not used by a device, device parameter word 4 is device-dependent and has the same meaning as parameter words 1–3.

I/O Condition Codes and Status Information

Each time an Operate I/O instruction is issued, the device, the controller, or the channel reports to the processor a condition code pertaining to the execution of the I/O command. Three bits are used to encode a condition-code value (range 0–7). The bits are recorded in the even, carry, and overflow positions of the LSR, and may be interrogated by specific instructions such as Branch on Condition Code and Branch on Not Condition Code.

Condition codes are also reported during a priority interrupt. These codes pertain to an operation that continues beyond execution of the Operate I/O instruction. In this case, the device must be able to present an interrupt. Along with the interrupt condition code, the device also transfers an interrupt ID word to the processor. Bits 0–7 of the interrupt ID word contain status information related to the interrupt processing, and are called the interrupt information byte.

IO Instruction Condition Codes

These codes are reported during execution of an Operate I/O instruction.

Condition code (CC) value	LSR position Even	Carry	Overflow	Reported by	Meaning
	Bit 0	Bit 1	Bit 2		
0	0	0	0	Channel	Device not attached
1	0	0	1	Device	Busy
2	0	1	0	Device	Busy after reset
3	0	1	1	Chan/device	Command Reject
4	1	0	0	Device	Intervention required
5	1	0	1	Chan/device	Interface data check
6	1	1	0	Controller	Controller Busy
7	1	1	1	Chan/device	Satisfactory

CC=0 *Device not attached.* Reported by the channel when the addressed device is not attached to the system.

CC=1 *Busy.* Reported by the device when it is unable to execute a command because it is in the busy state. The device enters the busy state upon acceptance of a command that requires an interrupt for termination. It exits the busy state when the processor accepts the interrupt. Certain devices also enter the busy state when an external event that results in an interrupt occurs. When the busy condition code is reported, a subsequent priority interrupt from the addressed device always occurs.

CC=2 *Busy after reset.* Reported by the device when it is unable to execute a command because of a power-on reset, or a system reset, and the device has not returned to the quiescent state. No interrupt occurs to indicate termination of this condition.

CC=3 *Command Reject.* Reported by the device when:

1. A command that is outside the device command set is issued.
2. The device is in an improper state to execute the command.
3. The IDCB contains an incorrect parameter (for example, an odd byte DCB address or an incorrect function/modifier combination).

   When a cycle-steal device reports command reject, it does not fetch the DCB.

CC=4 *Intervention required.* Reported by the device when it is unable to execute a command because of a condition that requires manual intervention to correct.

CC=5 *Interface data check.* Reported by the device or the channel when a parity error is detected on the I/O data bus during a data transfer.

CC=6 *Controller busy.* This condition is reported by a device controller, and not the addressed device, when the controller is busy. It is reported only by controllers that have two or more devices attached (each device having a unique address). When this condition code is reported, a subsequent controller-end interrupt always occurs.

CC=7 *Satisfactory.* Reported by the device when it accepts the command.

These condition codes are mutually exclusive and have a priority sequence. That is, beginning with CC=7, each successive condition code through CC=0 takes precedence over the previous code. For example, if a device cannot accept a command because it is busy, it reports CC=1, regardless of error conditions encountered.

Note: The CC=6 (controller busy) condition code is the only exception. This code may have a variable priority depending on the particular controller.

Interrupt Condition Codes

These condition codes are reported by the device or controller during priority interrupt acceptance.

Condition code (CC) value	LSR position Even Bit 0	Carry Bit 1	Overflow Bit 2	Reported by	Meaning
0	0	0	0	Controller	Controller end
1	0	0	1	Device	Program-controlled interrupt (PCI)
2	0	1	0	Device	Exception
3	0	1	1	Device	Device end
4	1	0	0	Device	Attention
5	1	0	1	Device	Attention and PCI
6	1	1	0	Device	Attention and exception
7	1	1	1	Device	Attention and device end

CC=0 *Controller end.* Reported by a controller when *controller busy* (IO instruction condition code) has been previously reported one or more times. CC=0 signifies that the controller is now free to accept I/O commands for devices under its control. The device address reported with controller end is always the lowest address (numerical value) of the group of devices serviced by the controller. The interrupt information byte, in the interrupt ID word, is set to 0.

CC=1 *Program-controlled interrupt.* Reported when the interrupt indicates that a DCB, with the PCI bit set to 1, has been transferred to the device and no error or exception condition has occurred.

CC=2 *Exception.* Reported when an error or exception condition is associated with the interrupt. The condition is described in the interrupt status byte (ISB) or in device-dependent status words.

CC=3 *Device end.* Reported when no error, exception, or attention condition has occurred during the I/O operation, and the interrupt is not the result of a PCI.

Note: If the device has come to a normal end while supporting suppress exception (SE bit = 1), then bit 0 of the interrupt status byte is set to 1. This condition is called permissive device end (PDE), and indicates that errors or exceptions may have occurred and were suppressed. Pertinent status information is contained in the residual status block.

CC=4 *Attention.* Reported when the interrupt was caused by an external event rather than the execution of an Operate I/O instruction. Additional status information is not provided unless the event requires further definition (for example, code bits for a keyboard function).

CC=5 *Attention and PCI.* Reported when attention and PCI are both present.

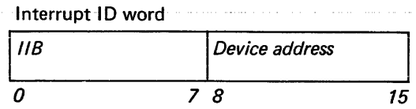
CC=6 *Attention and exception.* Reported when attention and exception are both present.

CC=7 *Attention and device end.* Reported when attention and device end are both present.

The interrupt condition codes are mutually exclusive with each other, but have no priority sequence.

I/O Status Information

Acceptance of an I/O interrupt causes the device to present an interrupt ID word to the processor. This word, placed in Register 7 when the interrupt is accepted, has the following format:



Bits 0–7 *Interrupt information byte (IIB).* For interrupt condition codes 2 and 6, the IIB has a fixed format. This is a special format of the IIB and is called an interrupt status byte (ISB). Refer to Interrupt Status Byte (ISB) in a subsequent paragraph. For most other interrupt condition codes, implementation of the IIB is device dependent. Exceptions are:

1. CC=3 or 7—bit 0 is set as described under CC=3.
2. CC=0—the IIB is set to 0.

Bits 8–15 *Device address.* This byte contains the address of the interrupting device.

Interrupt Status Byte (ISB). The ISB, a special format of the interrupt information byte (IIB), contains detailed information about the nature of the interrupt. The ISB is reported only for error or exception conditions (interrupt condition code 2 or 6). The ISB bits are normally set as a result of:

1. Status errors that occur during a DPC operation that cannot be indicated via a condition code
2. Status errors that occur during a cycle-steal operation

The ISB is never reported as 0 unless the condition code presentation of 2 or 6 is singular in meaning for devices that do not cycle steal. After the processor has accepted the interrupt request, the device resets the ISB.

Bits 0–7 of the two special formats are explained in the following paragraphs.

ISB (for devices that do not cycle steal):

Bit 0 *Device-dependent status available.* This bit, when set to 1, signifies that additional status information is available from the device. The information content and method of reading is described in the individual device publications.

Bit 1 *Delayed command reject.* This bit is set to 1 if the device cannot execute the command because of an incorrect parameter in the IDCB, or it cannot execute the command because of its present state. For example, (1) the IDCB specifies an incorrect function/modifier combination or (2) the device is temporarily not ready. The operation in progress is terminated. Interrupt condition code 2 or 6 is also reported. Command reject is set in the ISB only if the device cannot report I/O instruction condition codes for the condition.

Bits 2–7 *Device-dependent.* These bits, if used, are described in the individual device publications.

ISB (for cycle-stealing devices):

Bit 0 *Device-dependent status available.* This bit, when set to 1, signifies that (1) additional status information is available from the device or (2) the device is in an improper state to execute a function specified by a DCB.

The content and method of reading the additional status information is described in the individual device publications.

Note: When bit 0 of the ISB is equal to 1 and bits 2–7 are 0, the contents of the residual-address word (cycle-steal status) are defined by the device.

Bit 1 *Delayed command reject.* This bit is set to 1 if the device cannot execute the command because of one of the following conditions:

- The IDCB contains an incorrect parameter (for example, an odd-byte DCB address or an incorrect function/modifier combination).
- The present state of the device, such as a *not ready* condition, prevents execution of an I/O command specified in the IDCB or other command as specified in the DCB.

Delayed command reject is set in the ISB only if the device cannot report I/O instruction condition codes for the condition.

Bit 2 *Incorrect-length record.* This bit is set to 1 when the device encounters a mismatch between byte count and actual record length after beginning execution of the DCB. For example, the byte count is reduced to 0 (with chaining flag off) and no end-of-record is encountered. Incorrect-length record is not reported when the suppress exception bit is set to 1. Reporting of incorrect length record is a device-dependent feature, and may be implemented regardless of the suppress-exception feature.

Note: Refer to "Chaining" in this chapter for the effects of chaining operations on incorrect-length record reporting.

Bit 3 *DCB specification check.* This bit is set to 1 when the device cannot execute a command because of an incorrect parameter specification in the DCB. For example, (1) an odd-byte DCB chaining or status address, (2) an odd-byte count for a word-only device, (3) an odd-byte data address for a word-only device, (4) an invalid command or bit in the control word, or (5) an incorrect count.

Bit 4 *Storage data check.* This error condition applies to cycle-steal output operations only. If the bit is set to 1, it indicates that the main storage location accessed during the current output cycle contained bad parity. Parity in main storage is not corrected and the device terminates the operation. The bad parity data is not transferred to the I/O channel. No machine-check condition occurs.

Bit 5 *Invalid storage address.* This bit, when set to 1, indicates one of the following conditions:

1. During a cycle-steal operation, the device has presented a main storage address that is outside the storage size of the system.
2. A cycle-stealing device has attempted to access storage through a segmentation register and the valid bit in the segmentation register is set to 0. Note that address translation must be enabled before this condition can occur.

Invalid storage address can occur on a data transfer or on a DCB fetch operation. In either case, the cycle-steal operation is terminated.

Bit 6 *Protect check.* This bit, when set to 1, indicates that the I/O device attempted to access a main storage location and presented an incorrect address key, with the storage-protect mechanism enabled.

Bit 7 *Interface data check.* This bit, when set to 1, indicates that a parity error has been detected on the channel during a cycle-steal data transfer. The condition may be detected by the channel or the I/O device. In either case, the operation is terminated and an interrupt is presented to the processor.

Chaining

Chaining allows the programmer to sequence an I/O device through a set of operations. This chaining function is called ***DCB command chaining***, and is specified by the chaining flag (bit 0) in the DCB control word.

When the current DCB indicates a chaining operation (bit 0 of the control word set to 1), device parameter word 5 of the DCB contains a main storage address. This address points to the next DCB in the chain. The device completes the current operation, but does not present an interrupt request (excluding PCI) to the processor. Instead, the device fetches the next DCB in the chain and continues operation.

Note: The chaining operation has no effect on program-controlled interrupt (PCI).

DCB Command Chaining

When DCB command chaining is specified, each DCB fetched by the device is interpreted as a new operation (or function) to be performed. The DCB may be equal to, but not a continuation of, the operation specified by the previous DCB.

Chapter 2. Functional Description

Processor Cards

ROS Card

The ROS card contains the microprogram routines that control most of the processor functions. Microprogram routines control the execution of instructions, provide gating and control lines for the address and data cards, handle interrupt requests, and perform many other functions. The ROS card also generates:

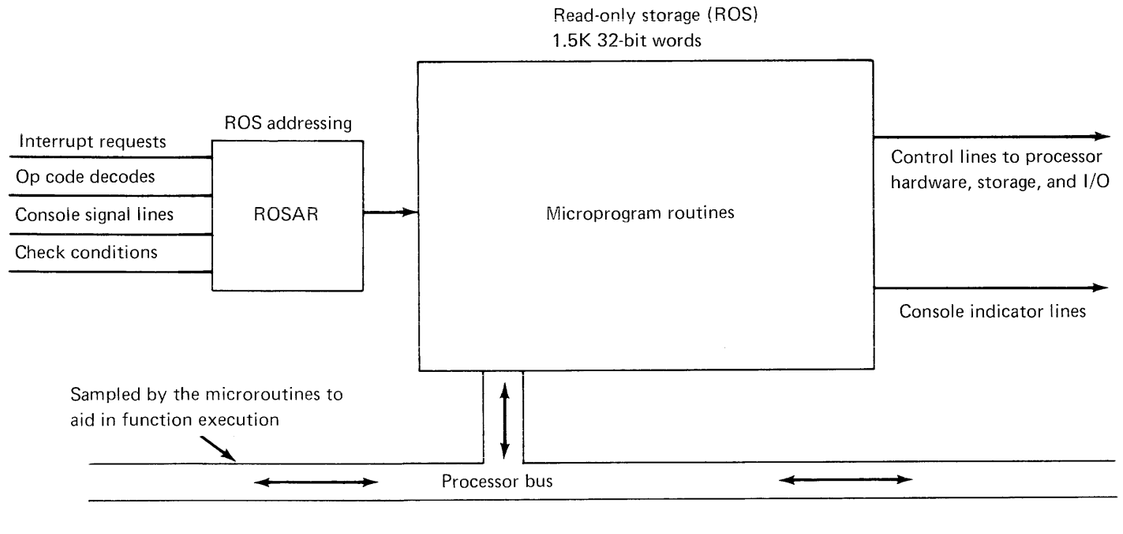
- System reset
- Processor reset
- Storage card select lines
- CSX and CSY lines for storage addressing on Models B, D, and E and storage address selection lines on Model F
- Storage refresh cycle (Model F only)
- Storage write controls
- Row lines to the consoles (see "Row- and Column-Line Operation" in this chapter)
- Console indicator lines (except data indicators)

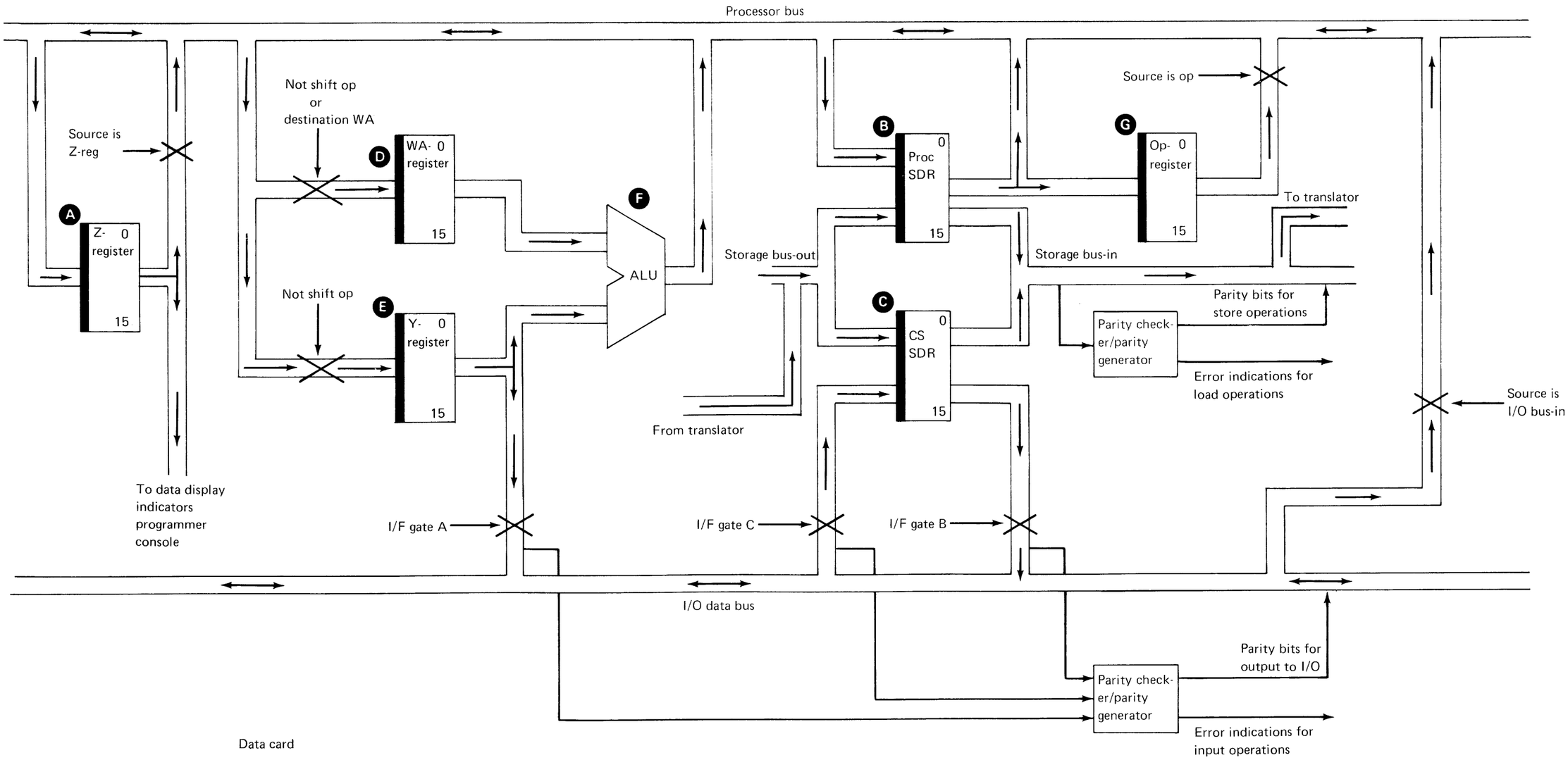
There are 1.5K 32-bit words in the ROS microprogram. These words are not addressable through any program or manual operation.

Addressing the proper starting point in the microprogram area is handled by the ROS address registers. Addresses are formed by the decoding of instruction op codes, signal lines from the consoles, interrupt requests, check conditions, and other machine status conditions.

The addressed routines respond by generating control lines that provide to the processor hardware the parameters necessary to complete the function that caused the ROS to be addressed.

Note: The ROS card for the 4955 Model F is not compatible with other 4955 models.




Processor bus
Source is Z-reg
A
Z- 0 register 15
Not shift op or destination WA
D
WA- 0 register 15
Not shift op
E
Y- 0 register 15
F
ALU
B
Proc SDR 0 15
C
CS SDR 0 15
Storage bus-out
From translator
Source is op
G
Op- 0 register 15
To translator
Storage bus-in
Parity bits for store operations
Parity check-er/parity generator
Error indications for load operations
Source is I/O bus-in
To data display indicators programmer console
I/F gate A
I/F gate C
I/F gate B
I/O data bus
Parity bits for output to I/O
Parity check-er/parity generator
Error indications for input operations
Data card

Data Card

Z-Register Ⓐ

- Primary purpose is to provide the drive lines for the data display indicators on the programmer console.
- Used as a temporary data register during processor functions.

Processor Storage Data Register (Proc SDR) Ⓑ

- All data, except cycle-steal data, coming from or going to storage, is gated through this register.
- The first word of every instruction that is fetched from storage is gated through the Proc SDR to the op register.
- Used as a temporary data register during processor functions.

Cycle-Steal Storage Data Register (CS SDR) Ⓒ

- All data involved in cycle-steal data cycles is gated through this register.
- I/F gate C, gates data from the I/O bus to the CS SDR, which in turn gates the data to storage.
- I/F gate B, gates data from the CS SDR to the I/O Data bus.

WA Register Ⓓ

- Primary input to the ALU for arithmetic and logical operations
- Used with the Y-register to perform doubleword shifting operations

Y-Register Ⓔ

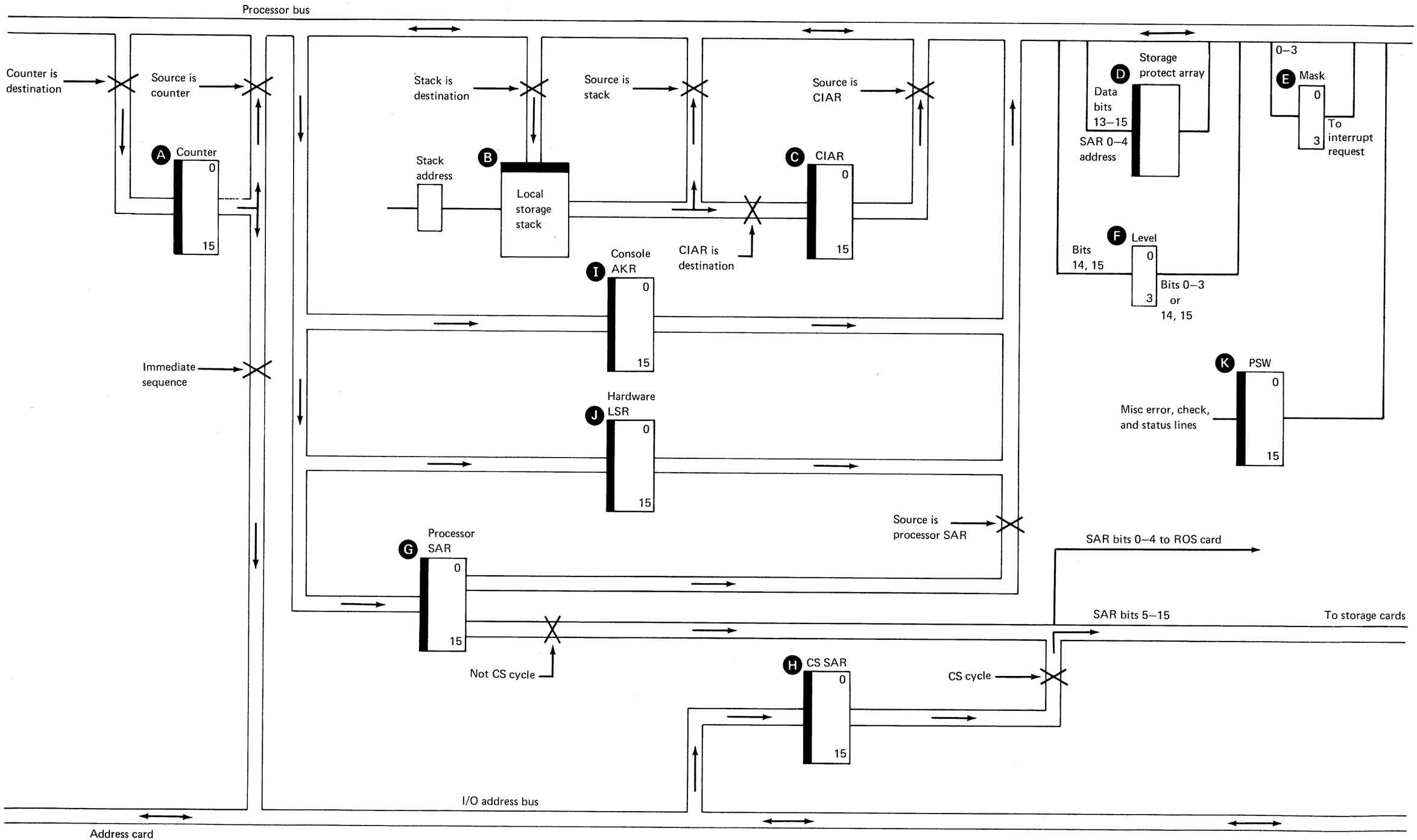
- Primary purpose is input to the ALU for arithmetic and logical operations.
- Used with the WA register to perform doubleword shifting operations.
- Provides the data path to the I/O data bus for DPC operations. I/F gate A gates the output of the Y-register to the I/O data bus.

Arithmetic Logical Unit (ALU) Ⓕ

- Performs the arithmetic and logical functions specified by the various instructions.
- The output of the ALU is gated to the processor bus, which is gated to the proper destination.

Op Register Ⓖ

- Contains the first word of the instruction during the decode phase
- Used as a temporary data register when not holding the first word of the instruction
- Displayable from the programmer console


Processor bus
Counter is destination
Source is counter
A Counter
0
15
Immediate sequence
Stack is destination
Source is stack
Source is CIAR
Stack address
B
Local storage stack
C CIAR
0
15
CIAR is destination
I Console AKR
0
15
Hardware
J LSR
0
15
Source is processor SAR
G Processor SAR
0
15
Not CS cycle
H CS SAR
0
15
CS cycle
D Storage protect array
Data bits 13–15
SAR 0–4 address
0–3
E Mask
0
3
To interrupt request
F Level
Bits 14, 15
0
3
Bits 0–3 or 14, 15
K PSW
0
15
Misc error, check, and status lines
SAR bits 0–4 to ROS card
SAR bits 5–15
To storage cards
I/O address bus
Address card

Address Card

Counter Ⓐ

The counter is used by the microprogram as:

- A counter (low eight bits) for monitoring various processing functions.
- A temporary hold for the I/O address. Output gated to the I/O address bus with immediate sequence.
- A temporary 16-bit data register for other processing functions.

Local Storage Stack Ⓑ

The local storage stack contains the registers that make up the four level status blocks (one for each interrupt level), console data buffer, stop-on-address buffer, and other registers used by the microprogram during processing. The local storage stack is 64 X 16 bits. See "Local Storage Stack Map" in this chapter for detailed layout.

Current Instruction Address Register (CIAR) Ⓒ

The CIAR contains the address of the instruction being executed. The CIAR is loaded at the beginning of each instruction. During the execution of this instruction, the level IAR (there is one IAR for each interrupt level) is updated to the next instruction address. If a class interrupts stops the current instruction from being fully executed, the class interrupt routine uses the CIAR to determine the address of the failing instruction.

Storage Protect Array Ⓓ

The storage protect array consists of the 32 storage key registers. The array is enabled whenever the storage protect feature is enabled. Each register contains the storage key and the read-only bit for controlling a 2K block of storage. The Set Storage Key instruction sets the key and read-only bit into a specific storage key register. The Copy Storage Key instruction reads out a specific storage key register. Refer to "Storage Protection" in this chapter for details.

Mask Register Ⓔ

The mask register consists of four bits that are used to enable or disable interrupts on the four interrupt levels.

Bit 0=0, level 0 interrupts disabled
Bit 1=0, level 1 interrupts disabled
Bit 2=0, level 2 interrupts disabled
Bit 3=0, level 3 interrupts disabled
Bit 0=1, level 0 interrupts enabled
Bit 1=1, level 1 interrupts enabled
Bit 2=1, level 2 interrupts enabled
Bit 3=1, level 3 interrupts enabled

The mask register is set by the Set Interrupt Mask Register (SEIMR) instruction. The Copy Interrupt Mask Register (CPIMR) instruction is used to store the mask register into main storage.

Level Register Ⓕ

The level register consists of four bits that hold the current level indicator that is presently in effect. Set whenever the level is changed, either by the Level Exit instruction (LEX), the Set Level Block (SELB) instruction, acceptance of a priority interrupt, or by the Level keys on the programmer console. The level register is used in addressing the proper level status block in the local storage stack and is also used to determine if an interrupt may be accepted.

Processor Storage Address Register (Proc SAR) Ⓖ

The proc SAR is a 16-bit register that is used primarily to gate bits 5–15 of the effective storage address to the SAR lines (used in addressing main storage). Bits 0–4 of the effective storage address are gated to the ROS card, where they are decoded to form the CSX and CSY lines for storage addressing. The proc SAR is also used by the microprogram as a temporary data register when it is not needed for addressing main storage.

Cycle-Steal Storage Address Register (CS SAR) Ⓗ

The CS SAR is a 16-bit register that is used during cycle-steal I/O operations to gate bits 5–15 of the storage address on the SAR lines to storage. Bits 0–4 of the effective storage address are gated to the ROS card, where they are decoded to form the CSX and CSY lines for storage addressing. The CS SAR is loaded from the I/O address bus during cycle-steal operations.

Console Address Key Register (AKR) Ⓘ

The console AKR is a 16-bit register that contains the contents of the current level AKR during processing on that particular level. The console AKR provides the address key, which is compared against the storage key in the storage protect array. This is done for each storage access except for cycle-steal operations. The ISK field of the console AKR is also used as the console address key for any manual storage accesses from the programmer console.

Note: The console AKR is present in the processor even if the programmer console is not part of the configuration.

Refer to "Storage Protection" in this chapter for details of AKR usage.

AKR bit	Meaning
0	Equate Operand Spaces
1	Not used; always 0
2	Not used; always 0
3	Not used; always 0
4	Not used; always 0
5	Operand 1 key, bit 0
6	Operand 1 key, bit 1
7	Operand 1 key, bit 2
8	Not used; always 0
9	Operand 2 key, bit 0
10	Operand 2 key, bit 1
11	Operand 2 key, bit 2
12	Not used; always 0
13	Instruction space key, bit 0
14	Instruction space key, bit 1
15	Instruction space key, bit 2

Hardware Level Status Register (LSR)

The hardware LSR is a 16-bit register that contains the contents of the current level LSR. During processing on a specific level, the contents of the hardware LSR change as the result of arithmetic and logical operations. The current level LSR remains unchanged until a level switch occurs. At this time, the hardware LSR is placed into the LSR of the level being exited and the new level LSR is placed into the hardware LSR.

LSR bit	Meaning
0	Even indicator
1	Carry indicator
2	Overflow indicator
3	Negative result indicator
4	Zero result indicator
5	Not used; always 0
6	Not used; always 0
7	Not used; always 0
8	Supervisor state
9	In-process
10	Trace
11	Summary Mask
12	Not used; always 0
13	Not used; always 0
14	Not used; always 0
15	Not used; always 0

Bit 0—Even Indicator. Set to 1 if the low-order bit of the result is 0; otherwise, set to 0.

Bit 1—Carry Indicator. Set to 1 if the result of add or subtract operations cannot be represented as an unsigned number; otherwise, set to 0.

Bit 2—Overflow Indicator. Set to 1 if the result of an arithmetic operation cannot be represented as a signed number; otherwise, set to 0.

Bit 3—Negative Indicator. Set to 1 if bit 0 of the result is 1; otherwise set to 0.

Bit 4—Zero Indicator. Set to 1 if the result is all 0's; otherwise, set to 0.

Note: The processor does not regard numbers as either signed or unsigned, but performs the designated operation on the values presented. All indicators reflect the result of the operation. This allows the programmer to test results for the type of operation performed.

The Even, Carry, and Overflow indicators are also used by I/O operations to hold the condition codes sent to the processor by the I/O devices.

During I/O instruction execution the Even, Carry, and Overflow indicators are assigned the following condition-code values:

Condition code	Even	Carry	Overflow	Meaning
0	0	0	0	Device not attached
1	0	0	1	Busy
2	0	1	0	Busy after reset
3	0	1	1	Command reject
4	1	0	0	Intervention required
5	1	0	1	Interface data check
6	1	1	0	Controller busy
7	1	1	1	Satisfactory

During interrupt acceptance, all condition codes are reported by the device. The Even, Carry and Overflow indicators are assigned the following condition-code values:

Condition code	Even	Carry	Overflow	Meaning
0	0	0	0	Controller end
1	0	0	1	Program-controlled interrupt
2	0	1	0	Exception
3	0	1	1	Device end
4	1	0	0	Attention
5	1	0	1	Attention and PCI
6	1	1	0	Attention and exception
7	1	1	1	Attention and device end

Bit 8—Supervisor State. Set to 1 whenever the processor enters supervisor state. Supervisor state is entered when:

- A Supervisor Call instruction is executed
- A class interrupt occurs
- An IPL is completed
- An I/O interrupt is accepted
- System reset or power-on reset is completed

Bit 9—In-process. This bit is set or reset by the corresponding bit in the LSR of the storage LSB whenever the SELB instruction is executed. The SELB loads an LSB from storage into the designated level LSB in the local storage stack.

Bit 10—Trace. This bit is set or reset by the corresponding bit in the LSR of the storage LSB whenever the SELB instruction is executed. The SELB loads an LSB from storage into the designated level LSB in the local storage stack.

Bit 11—Summary Mask. The summary mask provides a masking facility for priority interrupts and certain class interrupts. The state of the summary mask (enabled or disabled) is controlled by bit 11 in the level status register (LSR) of the active priority level. When bit 11 is set to 0, the summary mask is disabled and prevents (1) all priority interrupts regardless of priority level, and (2) power/thermal and console class interrupts. All other class interrupts are not masked. When bit 11 is set to 1, the mask is enabled and the interrupts are allowed.

The summary mask is disabled and enabled as follows:

- Disabled (set to 0)
  - When a Supervisor Call (SVC) instruction is executed, the summary mask for the active level is disabled.
  - Execution of a Disable (DIS) instruction, with bit 15 of the instruction equal to 1, causes the summary mask for the active level to be disabled.
  - All class interrupts disable the active level summary mask.
  - The summary mask for a selected level is disabled by executing a Set Level Block (SELB) instruction with bit 11 of the LSR to be loaded equal to 0.
  - The summary mask bits for priority levels 1–3 are set to 0 by a system reset, power-on reset, or IPL.
- Enabled (set to 1)
  - Execution of an Enable (EN) instruction, with bit 15 of the instruction equal to 1, causes the active level summary mask to be enabled.
  - The summary mask for a selected level is enabled by executing an SELB instruction, with bit 11 of the LSR to be loaded equal to 1.
  - The level 0 summary mask is enabled by a system reset, power-on reset, or IPL.
  - The summary mask for the interrupted-to level is enabled by a priority interrupt.

Note: If the processor is in the wait state, the summary mask is enabled or disabled, as defined by bit 11 in the LSR of the last active priority level.

Processor Status Word (PSW) K

The PSW is a 16-bit register that contains error and exception information that caused a program check, machine check, soft-exception trap, or power/thermal-warning class interrupt to occur. Refer to "Class Interrupts" in this chapter for details of these check conditions. Three status flags are also contained in the PSW. The PSW is changed by hardware only, but may be displayed from the programmer console.

PSW bit	Meaning
Program check	
0	Specification check
1	Invalid storage address
2	Privilege violate
3	Protect check
4	Invalid function (either program check or soft-exception trap)
Soft-exception trap	
5	Floating-point exception
6	Stack exception
7	Not used; always 0
Machine check	
8	Storage parity check
9	Not used; always zero
10	CPU control check
11	I/O check
Status flags	
12	Sequence indicator
13	Auto IPL
14	Translator enabled
Power/thermal	
15	Power/thermal warning

Bit 0—Specification Check. Set when the storage address violates boundary requirements:

- All word and doubleword operand addresses must be on an even-byte boundary.
- For indirect addressing, the address operand must be on an even-byte boundary.
- When word displacements are added to a register, the register must contain an even-byte address.
- All instructions must be on an even-byte boundary.
- All effective instruction addresses must be on an even-byte boundary.
- Stack control blocks must be on an even-byte boundary.

Specification check is set if the effective address is odd when attempting to execute a floating-point instruction and the floating-point feature is not installed.

Bit 1—Invalid Storage Address. Set when an attempt is made to access a storage location not on the system, or an unvalidated storage location in a translated configuration.

Bit 2—Privilege Violate. Set when a privileged instruction is attempted, and the supervisor state bit (bit 8) in the LSR is not on. Privileged instructions are:

Copy Address Key Register (CPAKR)
Copy Console Data Buffer (CPCON)
Copy Current Level (CPCL)
Copy in Process Flags (CPIPF)
Copy Interrupt Mask Register (CPIMR)
Copy Instruction Space Key (CPISK)
Copy Floating Level Block (CPFLB)
Copy Operand 1 Key (CPOOK)
Copy Operand 2 Key (CPOTK)
Copy Level Status Block (CPLB)
Copy Processor Status and Reset (CPPSR)
Copy Segmentation Register (CPSR)
Copy Storage Key (CPSK)
Diagnose (DIAG)
Disable (DIS)
Enable (EN)
Interchange Operand Keys (IOPK)
Level Exit (LEX)
Operate I/O (IO)
Set Address Key Register (SEAKR)
Set Console Data Lights (SECON)
Set Floating Level Block (SEFLB)
Set Instruction Space Key (SEISK)
Set Interrupt Mask Register (SEIMR)
Set Level Status Block (SELB)
Set Operand 1 Key (SEOOK)
Set Operand 2 Key (SEOTK)
Set Segmentation Register (SESR)
Set Storage Key (SESK)

Bit 3—Protect Check. In problem state only, set when (1) an instruction is fetched from a storage location not assigned to the current operation, (2) an attempt is made to access an operand in a storage location that is not assigned to the current operation, or (3) the instruction attempts to change a main storage operand in violation of the read-only control.

Bit 4—Invalid Function. Set to 1 by one of the following:

1. Attempted execution of an illegal operation code or function combination. These are:

Op code	Function decode
00101	All (if register 7 is specified in R1 or R2 fields of the instruction)
00111	All
01000	0001,0010,0011,0101,0110,0111
01011	0101,0111 (also, 0001 and 1001 when the storage address relocation translator is not installed and the processor is in Supervisor state)
01100	111
01110	11000,11010,11011,11100,11110,11111
01111	1X1XX,01XXX,1X011,10001
10110	All
11011	All
11101	1100,1101,1110,1111

Note: The preceding illegal conditions cause a *program check* class interrupt to occur.

2. The processor attempts to execute an instruction with an uninstalled feature. These are:

Op code	Function decode
00100	All (if the floating-point feature is not installed)
01011	0011,1011 (if the floating-point feature is not installed and the processor is in supervisor state)

Note: The preceding condition causes a soft-exception-trap class interrupt to occur.

Bit 5—Floating-Point Exception. Set if the floating-point feature is installed and an arithmetic exception condition is detected by the feature. Refer to Chapter 3, "Floating-Point Feature," for further details.

Bit 6—Stack Exception. Set when an attempt is made to pop an operand from an empty stack, or to push an operand into a stack that is full. This bit may be used by the programmer to determine the status (full or empty) of the stack. Stack exception results in a soft-exception trap.

Bit 8—Storage Parity. Set when a parity error has been detected on data being read out of main storage by a processor operation.

Bit 10—CPU Control Check. A control check occurs if no level is active, and instruction execution is continuing.

Bit 11—I/O Check. Set when a hardware error has occurred in the channel; further communications with all I/O devices is prevented. An I/O check cannot be caused by a software error. The conditions that set an I/O check are:

- An Operate I/O instruction and Set Interface Request (SIR) active simultaneously. The SIR microcode command is only used during the Diagnose operation, power-on reset, and several floating-point instructions.
- 'Service gate' and 'address gate' active simultaneously.
- 'Service gate return' and 'address gate return' active simultaneously.
- Parity error on interrupt. Processor detected a parity error when I/O interrupt parameters are passed from the device to the processor.
- Poll time-out. No device response ('poll return') within approximately 19 microseconds of the processor-initiated poll.
- Service gate time-out. No device response ('service gate return') within approximately 19 microseconds of processor-initiated 'service gate.'

Bit 12—Sequence Indicator. This bit is used to further identify the I/O check. If bit 12 is 0, the I/O check occurred during an Operate I/O instruction. If bit 12 is 1, the I/O check occurred during either a cycle-steal or an interrupt-accept sequence. The conditions that set the sequence indicator are:

- Poll time-out
- Service gate time-out
- Parity error on interrupt

Bit 13—Auto IPL. If the Mode switch on the console is in the Auto IPL position, any power on causes an IPL to occur and this bit to be set.

Bit 14—Translator Enabled. This bit is set on when an Enable instruction (with bit 14 on) is executed, and the translator feature is installed. It is reset by the execution of a Disable instruction (with bit 14 on), by an Enable instruction (with bit 12 on), or by a processor reset.

Bit 15—Power/Thermal Warning. Set when a power failure is about to occur, or when a thermal condition is going to cause power to drop. From the time this bit is set, the software program has at least 20 milliseconds to perform any data saving necessary before power drops.

Local Storage Stack Map

Addresses 0, 1, 32 and 48 contain temporary data used by the microprogram during normal processing.

Addresses 5–15 make up the level status block for interrupt level 0.

Address 16 contains the manually entered address to be used for stop-on-address operations.

Address 17 contains the console storage key in the low three bits; the remaining bits are 0's.

Addresses 21–31 make up the level status block for interrupt level 1.

Address 33 is the console data buffer. The console data buffer may be stored into from the programmer console, and may be read by using the Copy Console Data Buffer instruction. In run state, the contents of the console data buffer are constantly displayed.

Address 34 is a sequence-flag location used by the microprogram during some operational sequences when in stop-on-error mode.

Addresses 37–47 make up the level status block for interrupt level 2.

Address 49 contains the level that was active when stop state was entered.

Address 50 contains a copy of the current level LSR.

Address 51 contains a copy of the current level AKR.

Addresses 53–63 make up the level status block for interrupt level 3.

0	Temp 2	
1	Temp 3	
2		
3		
4		
5	IAR	LSB Level 0*
6	AKR	
7	LSR	
8	General register 0	
9	General register 1	
10	General register 2	
11	General register 3	
12	General register 4	
13	General register 5	
14	General register 6	
15	General register 7	
16	SOA 1	
17	SOA 2	
18		
19		
20		
21	IAR	LSB Level 1*
22	AKR	
23	LSR	
24	General register 0	
25	General register 1	
26	General register 2	
27	General register 3	
28	General register 4	
29	General register 5	
30	General register 6	
31	General register 7	

32	Temp	
33	Console data buffer	
34	SOE sequence control	
35		
36		
37	IAR	LSB Level 2*
38	AKR	
39	LSR	
40	General register 0	
41	General register 1	
42	General register 2	
43	General register 3	
44	General register 4	
45	General register 5	
46	General register 6	
47	General register 7	
48	Temp 1	
49	Current level save	
50	Current LSR save	
51	Current AKR save	
52		
53	IAR	LSB Level 3*
54	AKR	
55	LSR	
56	General register 0	
57	General register 1	
58	General register 2	
59	General register 3	
60	General register 4	
61	General register 5	
62	General register 6	
63	General register 7	

*Displayable from programmer console.

Example of LSB and Hardware Register Activity

Assume that a program is executing on level 3 with no interrupt pending. Note that the hardware LSR (A) changes during program execution, but the level 2 and 3 LSRs remain unchanged. Level LSRs are updated when levels are switched.

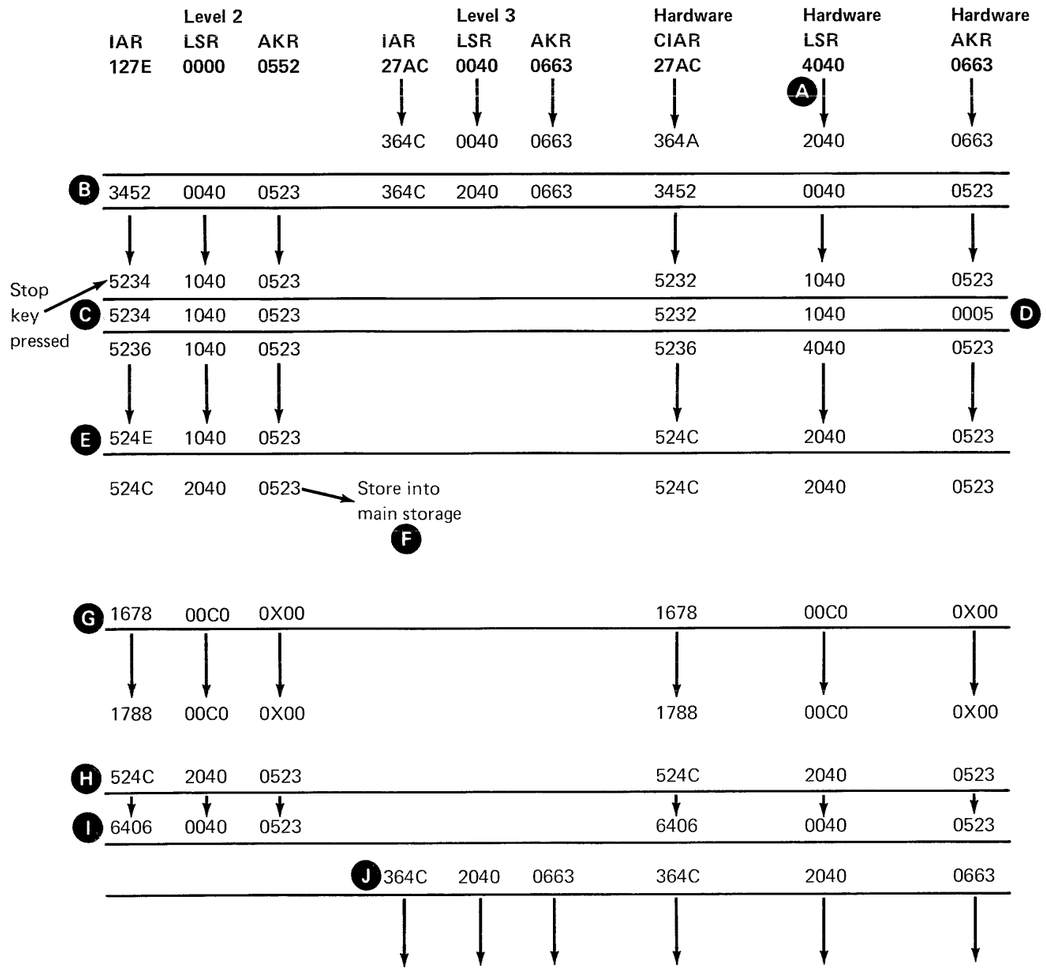
(B) A Set Level Status Block (SELB) instruction is executed addressing level 2 and the in-process flag in the main storage LSR is on. This causes a level switch to occur after the level 2 LSB is loaded from main storage. Level 2 is now the active level and level 3 is pending.

During the execution of the level 2 program, the operator wishes to stop the processor (C) and display a main storage location. Assume that address translation is enabled. During the manual display operation, the hardware AKR (D) is altered from the console to form part of the main storage address. After the display operation is completed, the Start key is pressed to restart the processor. The hardware AKR is updated by the contents of the level 2 AKR and processing resumes on level 2.

(E) A class interrupt occurs before the instruction at main storage address 524C is completed. The hardware CIAR contains the address of this instruction and the level 2 IAR has been updated to the address of the next instruction. The hardware LSR contains the current status for level 2, and the level 2 LSR contains the old status information. The microprogram proceeds to load the hardware CIAR into the level 2 IAR, and the hardware LSR into the level 2 LSR. The level 2 LSB is then stored into the designated area in main storage (F).

The class interrupt routine is executed on level 2 (G) and the level 2 LSB area is used by the class-interrupt routine. When the class-interrupt routine is completed, it executes an SELB instruction to restore the level 2 LSB from main storage. The level 2 program that was interrupted begins execution at address 524C (H) and continues to completion, where an LEX instruction is executed (I).

The pending level 3 program takes control, the level 3 LSR and AKR are loaded into the hardware LSR and AKR, and the level 3 IAR is used to address the level 3 program at the point it exited (J). Level 3 continues in effect until it executes an LEX instruction. If there are no levels pending, the processor enters wait state.



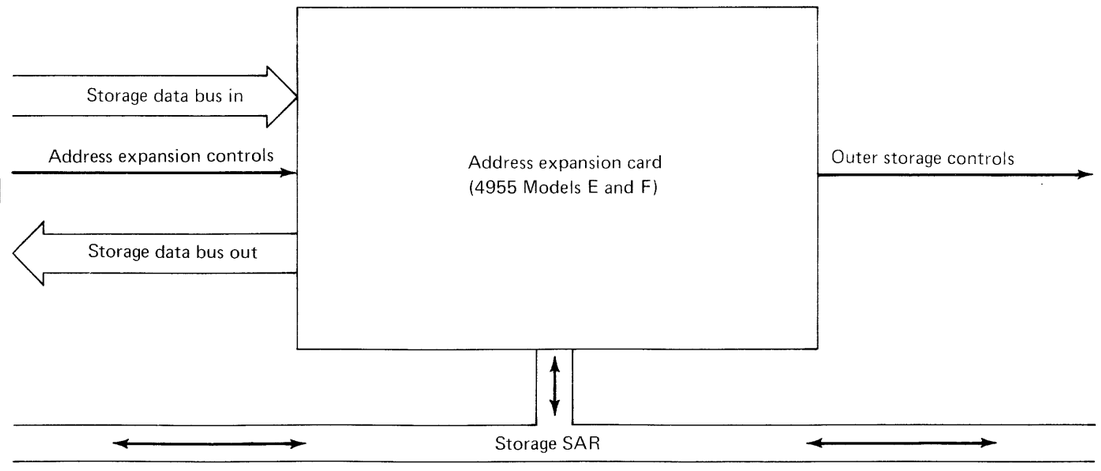
Address Expansion Card

The address expansion card is installed in the 4955 processor Models E and F. It is the fourth basic card of the Models E and F.

- Contains the logic to control outer storage (addresses above 64K bytes and up to 256K bytes for Model E; addresses above 64K bytes and up to 512K bytes for Model F)
- Controls SAR bits 0-4 for translated inner storage cycles (addresses less than 64K bytes)
- Contains 256 segmentation registers
- Performs storage protection for all translated storage cycles
- Controls storage refresh timing and addressing for the Model F only

Note: The address expansion cards for the Model E and Model F are not compatible.

Refer to "Segmentation Register Format" and "Storage Address Relocation Translator" in this chapter for a functional description of the operation of the address expansion card.



Main Storage

- The 4955 Models A and B have 16K bytes of main storage as a standard feature. Additional storage cards may be added for a maximum of 64K bytes of main storage for the Model A; a maximum of 128K bytes of storage for the Model B. The storage address relocation translator is required for the Model B if storage exceeds 64K.
- The 4955 Models C and D have 32K bytes of main storage as a standard feature. Additional storage cards may be added for a maximum of 64K bytes of main storage for the Model C; a maximum of 128K bytes of storage for the Model D. The storage address relocation translator is required for the Model D if storage exceeds 64K.
- The 4955 Model E has 64K bytes of main storage as a standard feature. Additional storage cards may be added for a maximum of 256K bytes of main storage. Address translation beyond 64K bytes is a basic feature.
- The 4955 Model F has 128K bytes of main storage as a standard feature. Additional storage cards in 128K-byte increments may be added for a maximum of 512K bytes of main storage. Address translation beyond 64K bytes is a basic feature.

Main storage is physically contained on storage cards. Each card contains 16K, 32K, 64K, or 128K (Model F only) bytes of storage. The addressing range of each card is determined by its card socket position in the 4955.

For the 4955 Models B and D, with the translator installed, and Models E and F, the address ranges shown are effective addresses. The storage address relocation translator feature card, for Models B and D, or the address expansion card, for Models E and F, selects either the lower 64K or a 64K segment above the lower 64K boundary. The physical addressing within that 64K segment is done in much the same manner as on a system with 64K or less of storage. Refer to "Storage Address Relocation Translator" in this chapter for a description of the translation process.

For the 4955 Models C and D, only one 16K-byte storage card can be installed, and it must reside in the last installed storage position. For the 4955 Model E, only one 32K-byte storage card can be installed, and it must reside in the last installed storage position. For example, if a 4955 Model E has two storage cards installed, and one is a 32K-byte card, the 32K-byte card must be plugged into the M-socket location. The 32K-byte card cannot be installed in the L-socket (the basic 64K-byte card location).

Note: Storage cards for Model F are not compatible with the other 4955 processors.

Storage Address Ranges

Hexadecimal address ranges for each storage socket location are listed below:

4955 Model A (64K installed)

Socket	Address range	Card name
M	0000–3FFF	16K card
N	4000–7FFF	32K card
P	8000–BFFF	48K card
Q	C000–FFFF	64K card

4955 Model B (64K installed)

Socket	Address range	Card name
G	0000–3FFF	16K card
H	4000–7FFF	32K card
J	8000–BFFF	48K card
K	C000–FFFF	64K card

4955 Model B (128K installed)

Socket	Address range	Card name
G	0000–3FFF	16K card
H	4000–7FFF	32K card
J	8000–BFFF	48K card
K	C000–FFFF	64K card
L	10000–13FFF	80K card
M	14000–17FFF	96K card
N	18000–1BFFF	112K card
P	1C000–1FFFF	128K card
Q	Translator card	

4955 Model C (64K installed)

Socket	Address range	Card name
P	0000–7FFF	32K card
Q	8000–FFFF	64K card

4955 Model D (64K installed)

Socket	Address range	Card name
L	0000–7FFF	32K card
M	8000–FFFF	64K card

4955 Model D (128K installed)

Socket	Address range	Card name
L	0000–7FFF	32K card
M	8000–FFFF	64K card
N	10000–17FFF	96K card
P	18000–1FFFF	128K card
Q	Translator card	

4955 Model E (64K installed)

Socket	Address range	Card name
L	0000–FFFF	64K card
Q	Address expansion card	

4955 Model E (256K installed)

Socket	Address range	Card name
L	0000–FFFF	64K card
M	10000–1FFFF	128K card
N	20000–2FFFF	192K card
P	30000–3FFFF	256K card
Q	Address expansion card	

4955 Model F (128K installed)

Socket	Address range	Card name
L	0000–1FFFF	128K card
Q	Address expansion card	

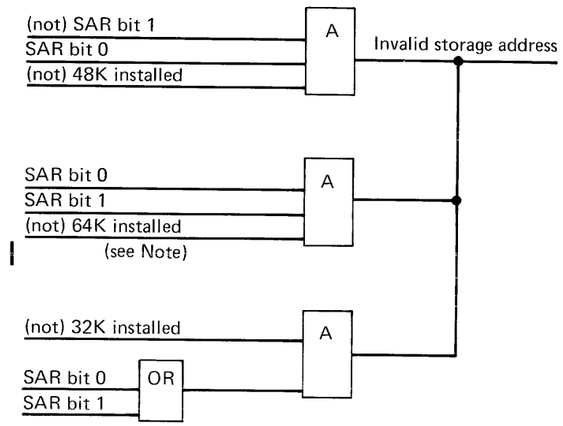
4955 Model F (512K installed)

Socket	Address range	Card name
L	0000–1FFFF	128K card
M	20000–3FFFF	256K card
N	40000–5FFFF	384K card
P	60000–7FFFF	512K card
Q	Address expansion card	

Invalid Storage Address (ISA)

An ISA results in a program-check class interrupt with bit 1 set in the PSW. An ISA is any storage address that attempts to access storage outside the range of the physically installed storage. An ISA could indicate a logical or a physical error.

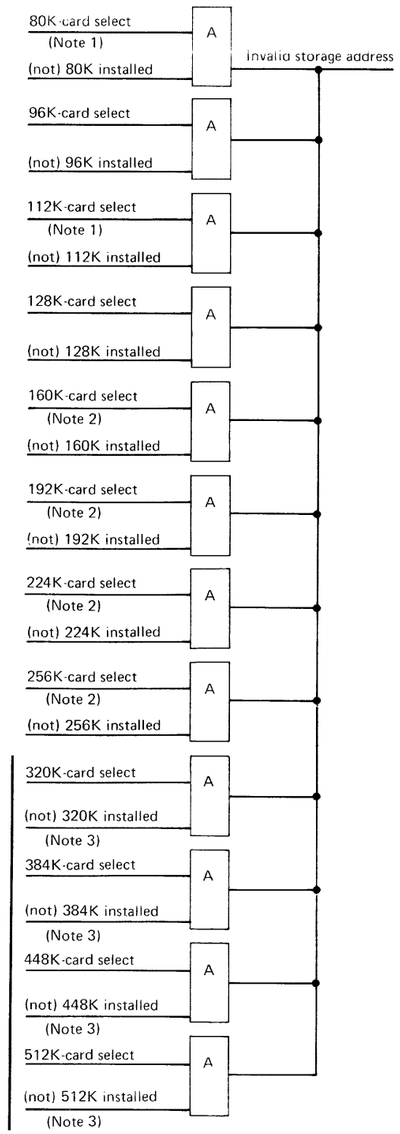
Inner Storage



Note: The 4955 Models E and F are not available without 64K bytes of storage; therefore, storage selection within the first 64K bytes does not produce an invalid address.

Outer Storage ISA

An outer storage ISA is detected by the storage address relocation translator for 4955 Models B and D, and by the address expansion card for the Models E and F.



Notes:

1. 4955 Models B and D only.
2. 4955 Models E and F only.
3. 4955 Model F only.

Storage Address Wrap

A storage address wrap is only allowed when 64K bytes of storage are installed, or when more than 64K bytes are installed and storage address translation is not enabled. For any other combination of installed storage, an ISA is flagged.

Storage Interface 0-64K Bytes (Inner Storage) for Models A, B, C, D, and E

Addressing the first 64K bytes of main storage, without the relocation translator installed or enabled, is the same for all models of the 4955 except Model F. Refer to "Storage Interface 0-64K Bytes (Inner Storage) for Model F" for addressing inner storage on the Model F.

The 16K-byte storage card (used in 4955 Models A, B, C, and D) is effectively divided into eight 2K-byte sections. A particular 2K-byte block is addressed by the coincidence of a CSY and a CSX line. The particular word within the block is addressed by SAR bits 5–14. If the operation is a write-byte type, the 'write byte 0' or 'write byte 1' line is activated to select the proper byte to be written into. On read operations, a word is always read from storage, and any byte selection is done by the processor logic. Storage data is gated to the processor on the storage data bus out (SDBO), and data is gated to storage on the storage data bus in (SDBI).

The 32K-byte storage card (used in the 4955 Models C, D, and E) and the 64K-byte storage card (used in the 4955 Model E) are addressed essentially the same as the 16K-byte card, except that additional card select lines (SAR bits 0 and 1) are provided to the cards to select the appropriate 16K-byte section.

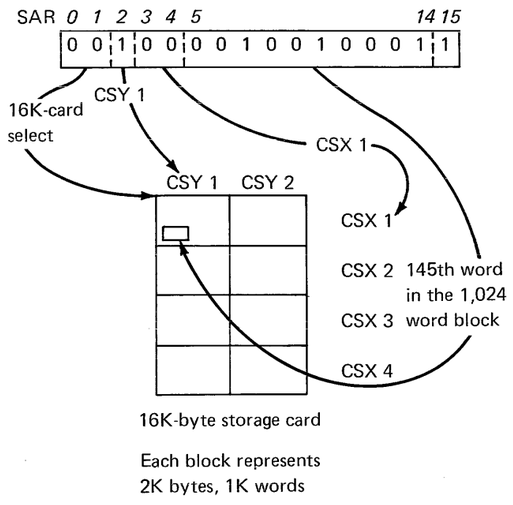
The 32K-byte storage card consists of two 16K-byte sections selected by SAR bits 0 and 1 equal to 00 and 01 for the first installed card, and SAR bits 0 and 1 equal to 10 and 11, respectively, for the second installed card (64K bytes, total inner storage).

The 64K-byte storage card consists of four 16K-byte sections selected by SAR bits 0 and 1 equal to 00, 01, 10, and 11, respectively (64K bytes, total inner storage).

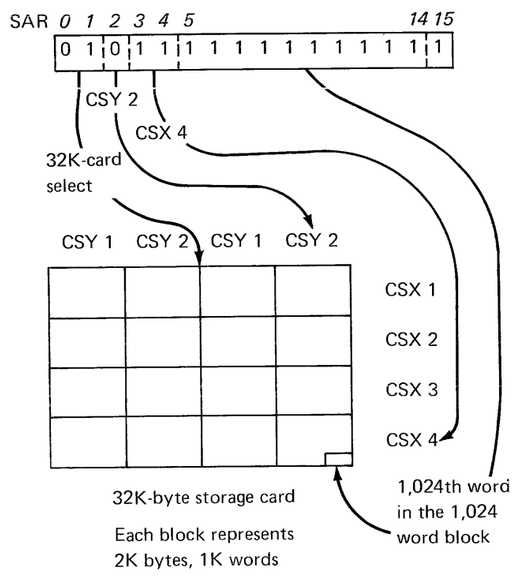
For storage addressing above 64K bytes, refer to "Storage Address Relocation Translator" in this chapter.

The following examples illustrate how the interface lines address storage:

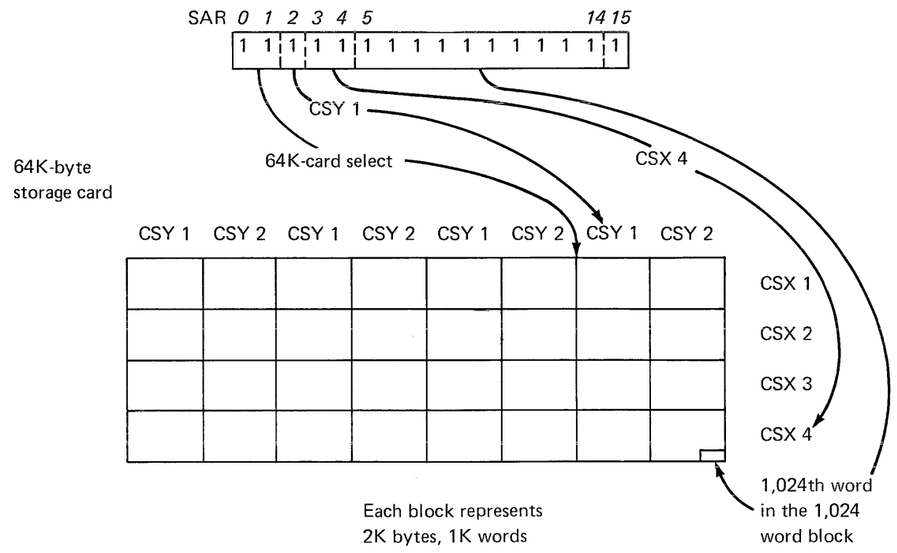
Hexadecimal address 2123



Hexadecimal address 5FFF



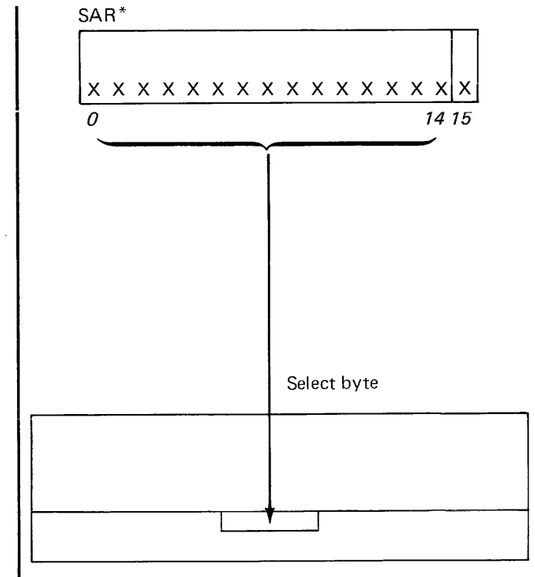
Hexadecimal address FFFF



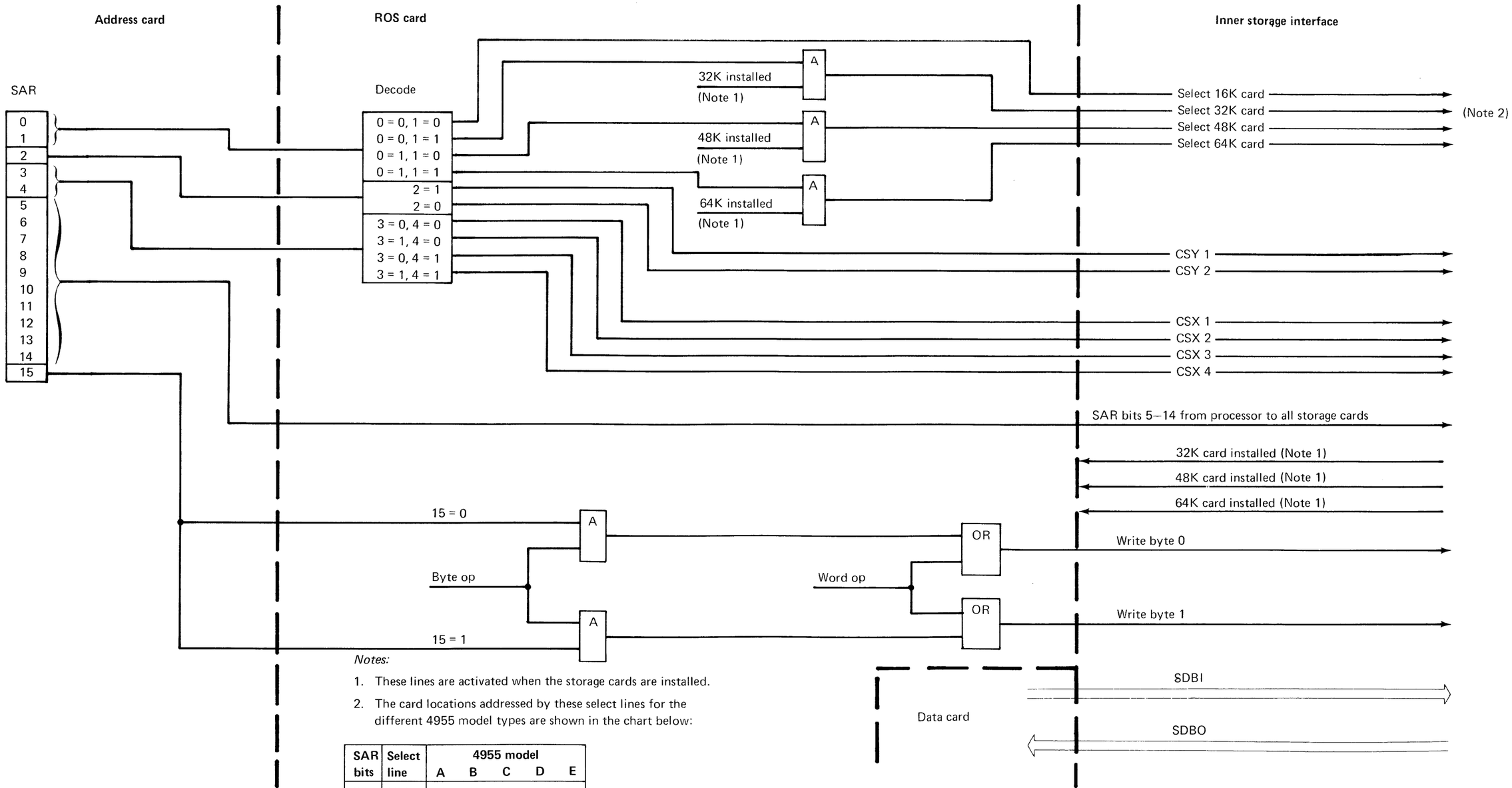
Storage Interface 0-64K Bytes (Inner Storage) for Model F

The first 64K bytes of main storage, with the relocation translator disabled, is addressed by SAR. If the operation is a write-byte type, 'write byte 0' or 'write byte 1' is activated to select the proper byte to be written into. On read operations, a word is always read from storage, and any byte selection is done by the processor logic. Storage data is gated to the processor on the storage data bus out (SDBO), and data is gated to storage on the storage data bus in (SDBI).

For storage addressing above 64K bytes, refer to "Storage Address Relocation Translator" in this chapter.



*Relocation translator disabled



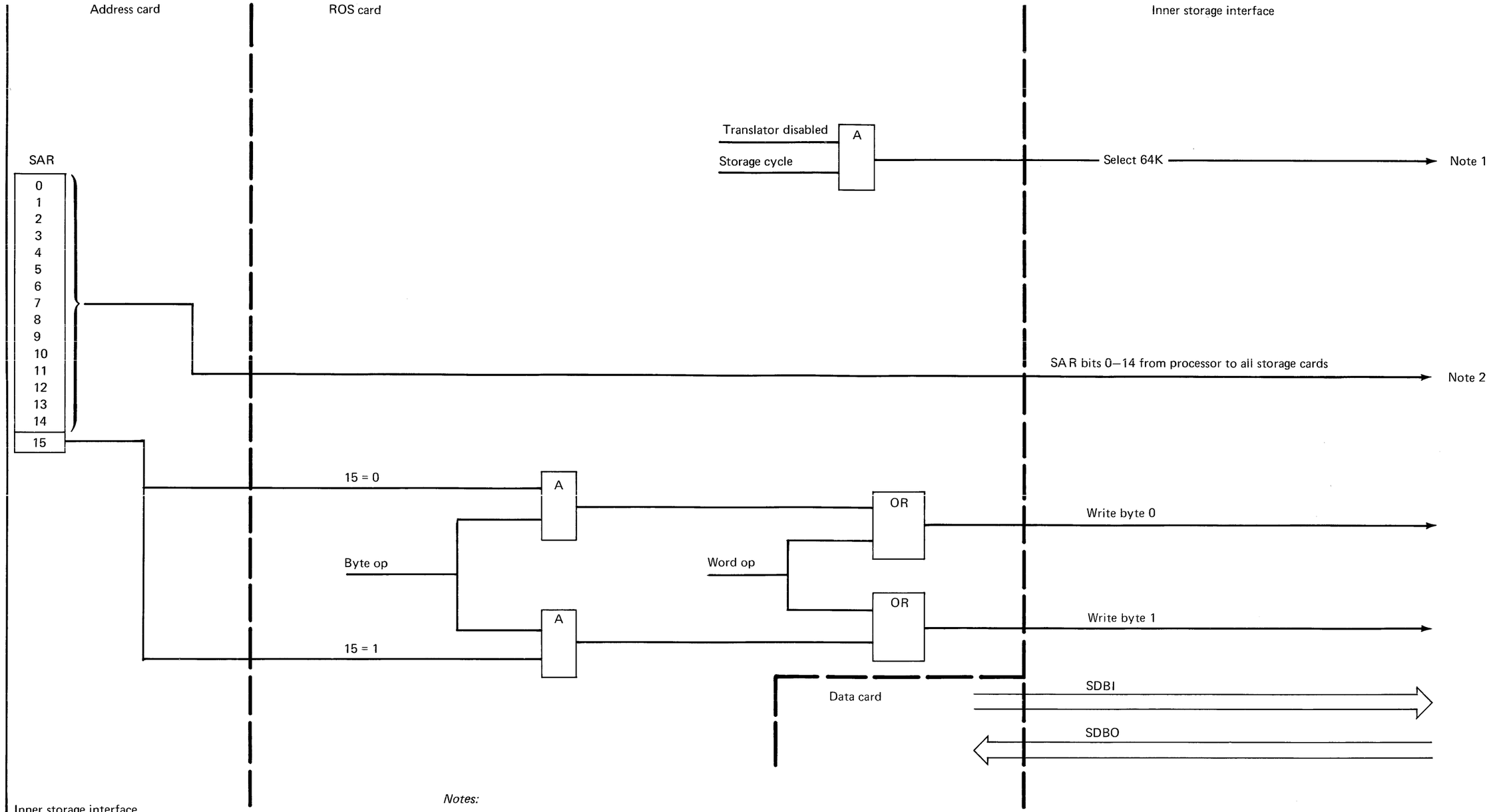
Notes:

1. These lines are activated when the storage cards are installed.
2. The card locations addressed by these select lines for the different 4955 model types are shown in the chart below:

SAR bits	Select line	4955 model A	B	C	D	E
00	16K	M	G	P	L	L
01	32K	N	H	P	L	L
10	48K	P	J	Q	M	L
11	64K	Q	K	Q	M	L

Card location

Inner storage interface from processor with relocation translator disabled (Models A, B, C, D, and E)

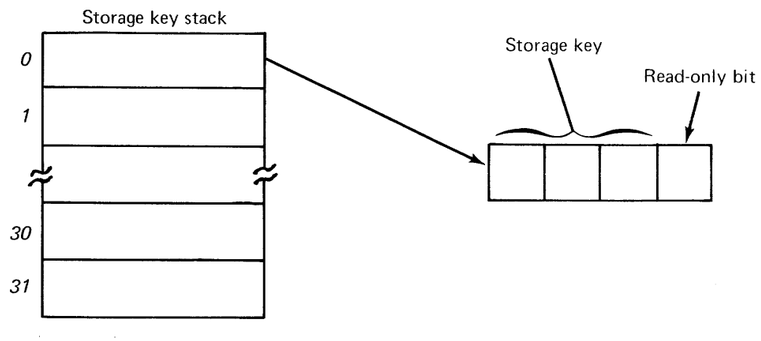


Notes:

1. Select 64K is on the 128K basic storage card located in socket L.
2. SAR bits 0-14 select the first 64K bytes of storage.

Storage Protection

- Storage protection is a standard feature on all models of the 4955.
- Protects against inadvertent accesses of storage locations not assigned to present program.
- Does not allow writing into storage specified as read-only.
- Protects against instruction fetching from storage areas not assigned to instructions.



Main storage is divided into thirty-two 2K-byte blocks. Associated with each 2K byte block is a storage key register. The 32 storage key registers reside in a stack that is located on the Address card of the processor. Each storage key register contains a 3-bit storage key and a read-only bit.

The storage key registers are set by the Set Storage Key (SESK) instruction. The SESK instruction is privileged, and is only executed while the processor is in supervisor state. The storage key registers may be examined through the use of the Copy Storage Key (CPSK) instruction. The CPSK instruction takes the contents of the addressed storage key register and places it into a designated storage location. The storage location containing the contents of the storage key register may then be displayed from the programmer console. The CPSK instruction is a privileged instruction, and is only executed while the processor is in supervisor state.

Enabling/Disabling Storage Protection

Although storage protection is a standard feature on the 4955 processor, it must be enabled to function. The state of the storage-protection mechanism is controlled by the Enable (EN) and Disable (DIS) instructions. The following chart describes storage-protection control through the execution of the Enable and Disable instructions:

Enable Instruction Parameter Field Bits

12	14	Meaning
0	0	No action
0	1	Storage protection disabled if storage address relocation translator feature is installed and enabled for 4955 Models B and D, or if the address translation function of the address expansion card is enabled for the 4955 Models E and F.
1	0	Storage protection enabled
1	1	Storage protection enabled (bit 14 ignored)

Disable Instruction Parameter Field Bits

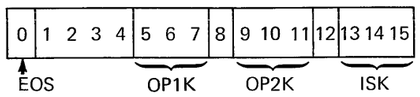
12	14	Meaning
0	0	No action
0	1	No action on storage protection; translator feature disabled
1	0	Storage protection disabled

When storage address relocation translation is enabled, the standard storage-protection mechanism is disabled. The translator provides the storage protection while it is active.

Storage-Protection Operation

Each time an access to storage is attempted, the processor compares the storage key associated with the storage address with the active address key, which is located in the active level address key register. A successful compare allows the storage access to occur. A mismatch does not allow the access to take place, and a program check is taken with protect check (bit 3) set in the PSW. See "I/O Storage Access Protection" in a subsequent paragraph in this section for I/O protect operation.

Each priority level has an Address Key Register (AKR). This register has the following format:



EOS — Equate operand spaces. When set to 1, all data operands use the OP2K address key. Inactive when set to 0.

OP1K — These three bits contain the address key for operand 1 storage accesses.

OP2K — These three bits contain the address key for operand 2 storage accesses.

ISK — These three bits contain the address key associated with the instruction space in main storage.

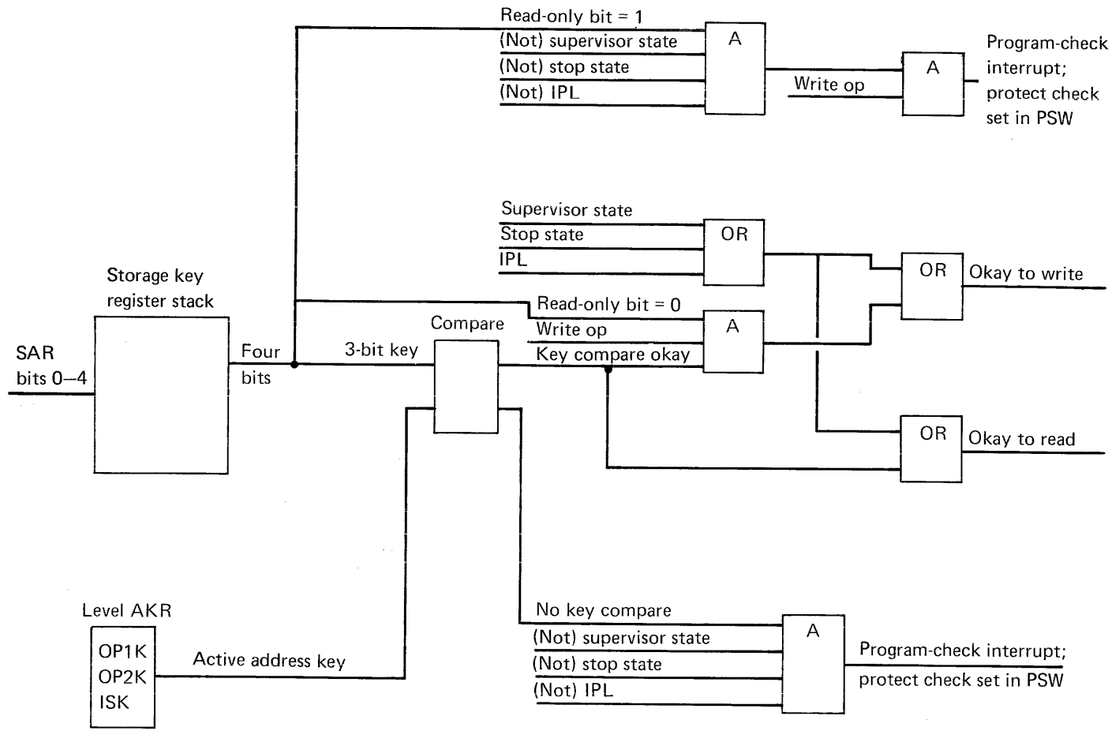
The particular key used for the comparison operation depends on the type of operand access, or whether the access is an instruction fetch attempt. When a key from the AKR is selected, it becomes the active address key and is used to compare against the key from the storage key register.

To allow storage accesses, one of the following conditions must be satisfied:

1. The processor is in supervisor state, or
2. The storage key of the address block is 7. If the access is a write, the read-only bit must be 0, or
3. The storage key matches the active address key. If the access is a write, the read-only bit must be 0.

If none of these conditions is met:

- The storage access is prevented and storage is unchanged.
- A program-check interrupt occurs with protect check (bit 3) set in the PSW.



Storage protect - logical operation chart

Suppressing Storage Protection

Storage protection is suppressed and no protect checking is performed for the following cases:

- Supervisor state
- During initial program loading
- While the processor is in Stop state, and a main storage access is performed from the programmer console
- While level status blocks are being stored during class interrupts
- When the storage-protection mechanism is disabled

I/O Storage Access Protection. I/O devices obey the same protection rules in terms of storage keys; however, the read-only bit is ignored during cycle-steal accesses to main storage. The I/O cycle steal address key is specified in the device control block.

Storage Protection During IPL. The storage protection mechanism is suppressed during an IPL operation. After the IPL is successfully completed, the processor enters supervisor state on priority level 0 with all address keys in the level 0 AKR set to 0. All I/O devices assume protect key 0 for DCB fetches; therefore, all DCBs must reside in a storage partition assigned storage key 0.

Address Key Values After Interrupts

When priority or class interrupts occur, certain values are set into the address keys of the affected AKR. These values anticipate the address spaces that the programmer might need for interrupt processing. The following chart shows the resulting AKR for each type of interrupt:

	Resulting AKR values			
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.

All interrupt service routines are presumed to reside in address space 0; therefore, the ISK and OP2K are set to 0 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.

Storage Address Relocation Translator

- The translator can be installed on the IBM 4955 Models B and D.
- For the 4955 Models E and F, storage address relocation translation is basic and is a function of the address expansion card. Operating characteristics and program control of the address translation function for the Models E and F are the same as for Models B and D.
- The translator increases the addressing capability of the 4955 Models B and D from 64K bytes to 128K bytes.
- The storage address relocation translation function of the address expansion card for the 4955 Model E allows an addressing capability up to 256K bytes; up to 512K bytes for the Model F.
- When installed and enabled, all main storage addresses are translated by this feature card.
- When installed and enabled, basic storage protection method is inhibited and the translate storage protect mechanism takes over.
- This feature is housed on one 4-wide, 6-high card, which is always plugged in to socket Q.
- Jumpers on the Model E address expansion card indicate the storage range of that processor. There are no jumpers on the Model F address expansion card; the processor storage range is electronically detected.

The purpose of storage address relocation translation is to allow for the installation and addressing of main storage above 64K bytes.

The 16-bit address generated by the processor is treated as a logical address. During translation, this address is translated into a set of select and address lines capable of addressing up to 128K bytes of main storage for 4955 Models B and D, up to 256K bytes of main storage for 4955 Model E, and up to 512K bytes of main storage for 4955 Model F.

For purposes of translation and control, main storage is divided into two areas. The first 64K-byte area is called inner storage; storage beyond 64K bytes is called outer storage.

Main storage is mapped into smaller sections through the use of segmentation registers, which are located on the translator card for 4955 Models B and D, and on the address expansion card for the 4955 Models E and F.

There are 256 segmentation registers, each controlling a 2K-byte block of main storage. The segmentation registers are arranged in eight groups of 32 registers. Each group is addressed by one of the eight possible active address keys. The specific segment register within the group is addressed by the first five bits of the logical address from SAR (bits 0–4).

Instructions Affecting the Translator

Set Segmentation Register	This privileged instruction loads one 32-bit segmentation register.
Copy Segmentation Register	This privileged instruction reads the contents of one 32-bit segmentation register and places the contents into a doubleword main storage location.
Enable	This privileged instruction sets bit 14 of the PSW, enabling the translator when accesses to main storage are made.
Disable	This privileged instruction resets bit 14 of the PSW, disabling the translator.

Refer to the *IBM Series/1 4955 Processor and Processor Features Description* manual, GA34-0021, for detailed descriptions of these instructions.

I/O Storage Access With Translator

All storage-access requests from I/O devices are handled by the translator exactly like storage-access requests from the processor. Because the DCBs for cycle-stealing devices must be located in the supervisor address space, all I/O devices must use address key 0 to gain access to the DCBs. The address keys and the logical addresses used by the cycle-steal operations reside in the DCBs. The I/O device presents these to the translator through the normal cycle-steal address and data paths.

Storage Protection With the Translator Installed and Enabled

When the translator is active, the standard storage-protection mechanism is disabled. The storage-protect stack is not accessed by the processor hardware. Instead, protection is provided by the segmentation registers located on the translator card. The active address key and the five high-order bits of the storage address are used to select a particular segmentation register.

Two bits in the segmentation registers provide both read-only and no-access protection. Bit 13, the valid bit, is set to 1 if the contents of the segmentation register are valid, and can be used for address translation. If bit 13 is set to 0, the contents of the segmentation register are invalid, and the segmentation register cannot be used for storage access.

This provides the facility of no-access protection, because a program-check interrupt, with *invalid storage address* set in the PSW, is taken when the register is accessed. Bit 14 is the read-only bit and, when set to 1, inhibits the writing into the 2K-byte block of main storage controlled by the segmentation register accessed. If a write operation is attempted, a program-check interruption, with protect-check set in the PSW, is taken. When bit 14 is set to 0, no protection is in effect for that particular 2K block. Bit 14 is ignored when the processor is in supervisor state or during a cycle-steal access.

When the translator is disabled, the standard storage-protection mechanism is in a ready state, and may be enabled using the Enable instruction.

Processor/Translator/Storage Interface Line Descriptions

The storage data out bits 0–15 are used to transfer data from storage to the processor. This bus is also used to transfer the contents of a segmentation register to the processor when a Copy Segmentation Register instruction is executed.

The SDR bits 0–15 to storage are used to transfer data from the processor to storage. This bus is also used to transfer data to the segmentation register that is addressed by the Set Segmentation Register instruction.

SAR bits 0–4 are used by the processor when inner storage is accessed. These bits are sent to the translator when the translator is installed and enabled. The translator decodes these bits and addresses a particular segmentation register. After the segmentation register is read out and decoded, the translator sends bits 8–12 of the segmentation register back to the processor on the 'SAR bits 0–4' line if the decode specifies an inner storage cycle. If the decode specifies an outer storage cycle, the bits from the segmentation register are used to generate the 'select,' 'T CSY,' and 'T CSX' lines for outer storage on Models B, D, and E and storage address selection lines on Model F. SAR bits 5–14 do not require any translation, and are sent directly to all storage cards.

The 'end of cycle' line is generated by the processor to signal the translator that the processor is finishing its storage cycle.

'Gate translator SAR' is generated by the translator to tell the processor to remove SAR bits 0–4 from the bus. This signal is generated when the translator decodes the access as being an inner-storage access. Fifty-five nanoseconds after this signal is sent, the translator places the five bits from the segmentation register (bits 8–12) on the SAR 0–4 bus, to be used by the processor to access inner storage.

The 'inner stg cycle' line is generated by the translator to inform the processor that the storage access is to be to the inner-storage space.

Protect key bits 1, 2, and 4 make up the active address key that is used by the translator to select a stack of segmentation registers.

The 'seg reg cycle' line is generated by the processor to let the translator know that the segmentation registers are to be accessed.

'Stg reg to translator' is generated by the processor to request that the translator take the logical address on the storage address bus and translate it. One microcycle (220 nanoseconds) is automatically skipped to allow the translator to access the proper segmentation register, obtain the physical address, and determine if the storage access is to be an inner or outer storage access.

For write operations to main storage, the 'stg write op 0' and 'stg write op 1' lines are generated by the processor to indicate to the translator that bytes are to be written into. 'Stg write op 0' indicates a write into byte 0 of the addressed word; 'stg write op 1' indicates a write into byte 1 of the addressed word. When the 'seg reg cycle' line is active, these two lines indicate the type of access to the segmentation registers. 'Stg write op 0' and 'stg write op 1' equal to 00 means that the operation is to be a read from the segmentation registers (Copy Segmentation Register instruction). 'Stg write op 0' and 'stg write op 1' equal to 11, means that the operation is to be a write into the segmentation registers (Set Segmentation Register instruction).

The 'time A, B, C, and D' lines are 55-ns pulses generated by the processor to provide synchronism between the translator and the processor.

The 'translator enabled' line is generated by the processor to inform the translator that the enable bit in the PSW has been set.

The 'translator installed' line from the translator indicates to the processor that the translator card is plugged into location Q of the 4955.

'Translator ISA' (invalid storage address) is generated by the translator. This results in a program-check interruption with the ISA bit set in the PSW. This can be caused by one of two conditions:

1. The address sent to the translator results in a translated address that points to a storage location that is not installed.
2. The addressed segmentation register contents are invalid, (segmentation register bit 13=0).

'Translator storage busy' is not implemented at this time; however, if an inadvertent attempt is made to access storage greater than physical 128K bytes for processor Models B and D, or 256K bytes for Model E, this line stays active until the translator times out and generates ISA.

The 'supervisor state' or 'cycle steal cycle' line is generated by the processor to indicate to the translator that bit 14 of the segmentation registers is to be ignored while this line is active. Bit 14 is the read-only access bit.

The translator generates the 'translator protect check' line when an attempt is made to write into a 2K-byte block of storage whose segmentation register has bit 14 set to 1.

The 'power on reset' line, generated when the processor is powering up, inhibits the translator from attempting to access outer storage.

The following refresh information applies to the Model F only:

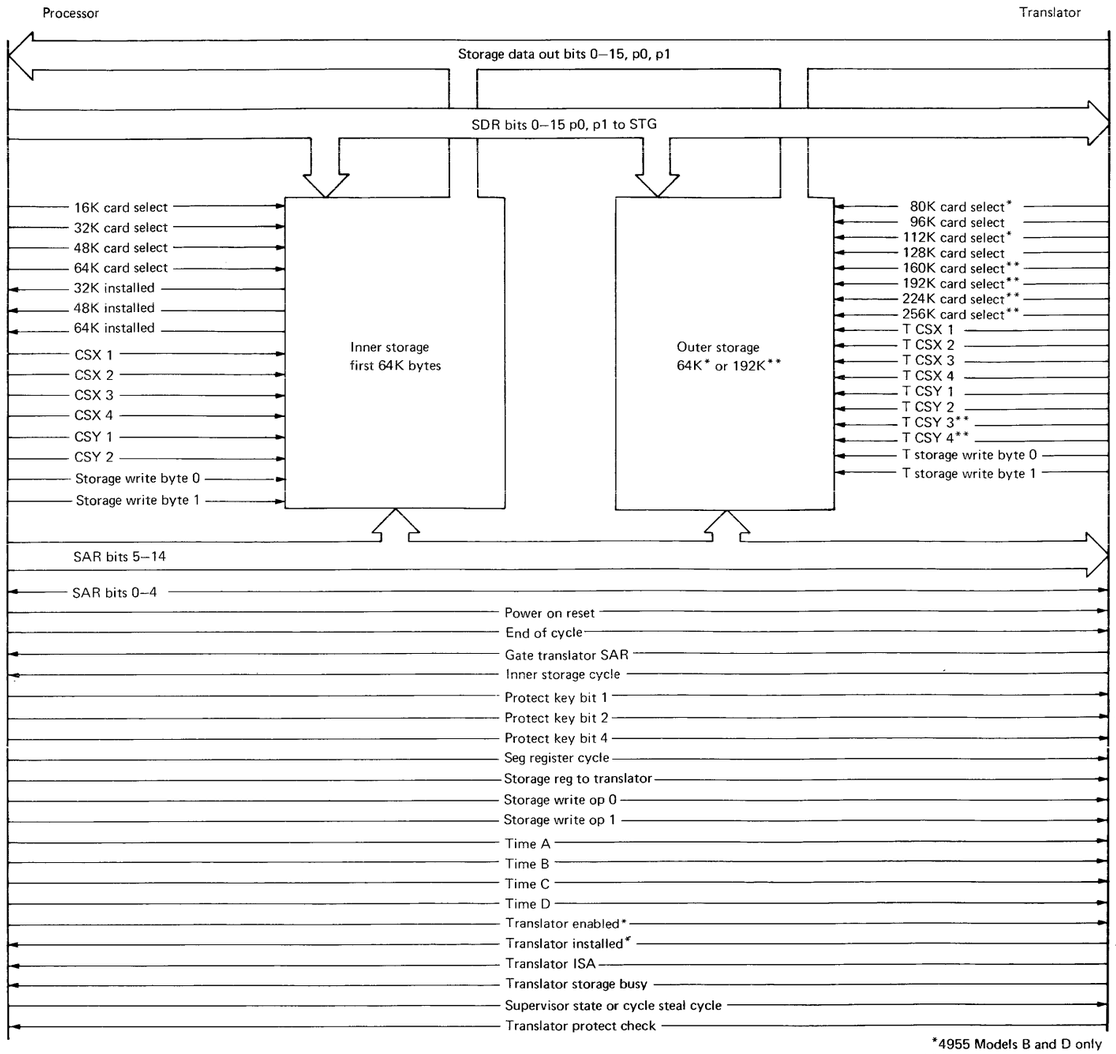
The 'refresh request' line indicates to the storage contention logic that storage must be refreshed. 'Refresh request' becomes active every 13.64 microseconds and remains active until the storage contention logic acknowledges the request.

The 'refresh cycle' line is activated when 'refresh request' has been acknowledged by the processor.

The 'translator refresh cycle' line is activated by 'refresh cycle' and refreshes all storage.

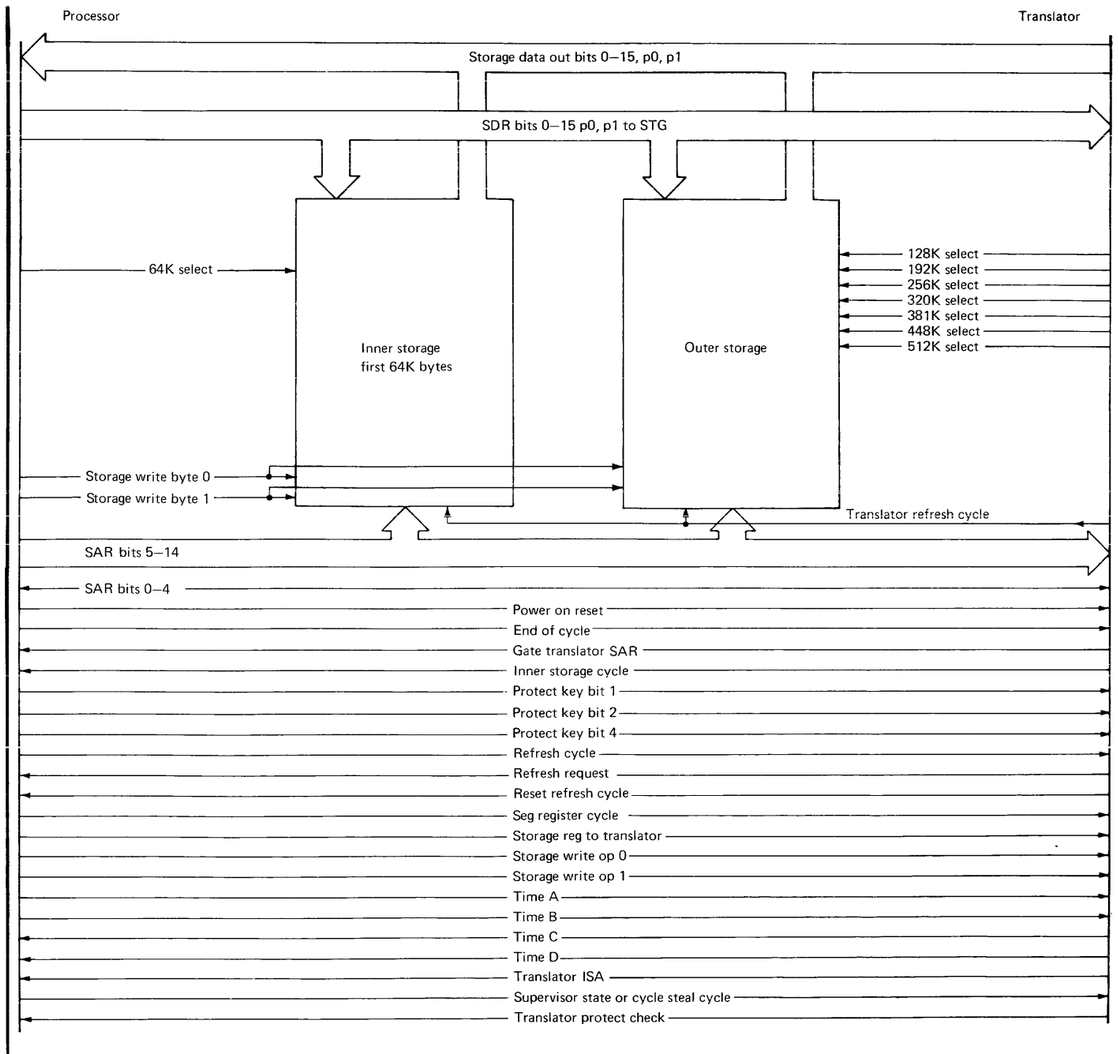
The 'reset refresh cycle' line is activated by the address expansion card when refresh is completed.

Note: The term refresh is used to describe the method of maintaining storage data for the Model F only.



*4955 Models B and D only
**4955 Model E only

Processor/Translator/Storage Interface
(Models B, D, and E)



Processor/Translator/Storage Interface
(Model F)

Segmentation Register Format

The segmentation register formats for 4955 Models B, D, E, and F are shown in the following illustrations.

If any of the bits that have the designation 'must be 0's' are set to 1's, a program-check interrupt is taken with invalid address (bit 1) set to 1 in the PSW.

Some of the bits designate the outer or inner storage cycle selection for a translated address.

Other bits designate storage size selection and storage address selection.

Note: The storage size selection is active for installed storage only.

Bit 13 Valid Bit. When this bit is set to 1, it indicates that this segmentation register contains a valid bit pattern, and that this register may be used in a translation process.

When this bit is set to 0, and an attempt is made to use this register in a translation access, a program check, with an invalid address (bit 1) set to 1 in the PSW (logical ISA), occurs.

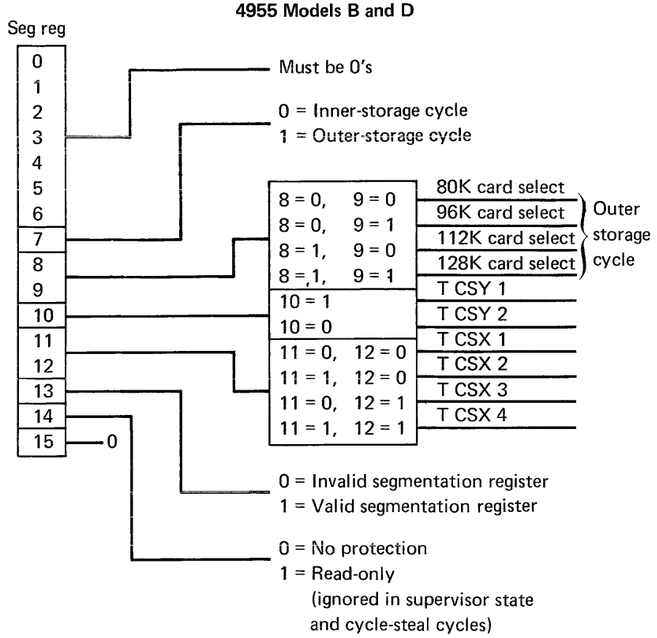
The segmentation registers must be initialized by an operating system or customer-written routine before they become valid.

Bit 14 Read-Only Bit. When this bit is set to 1, it prevents writing into the 2K-byte block of main storage designated by this segmentation register. If the translator is not enabled, and an attempt is made to write into this read-only area, the protect-check bit (bit 3) of the PSW is set to 1. When the translator is enabled and an attempt is made to write into this read-only area, the protect-check bit of the PSW is set to 1 and a program check results.

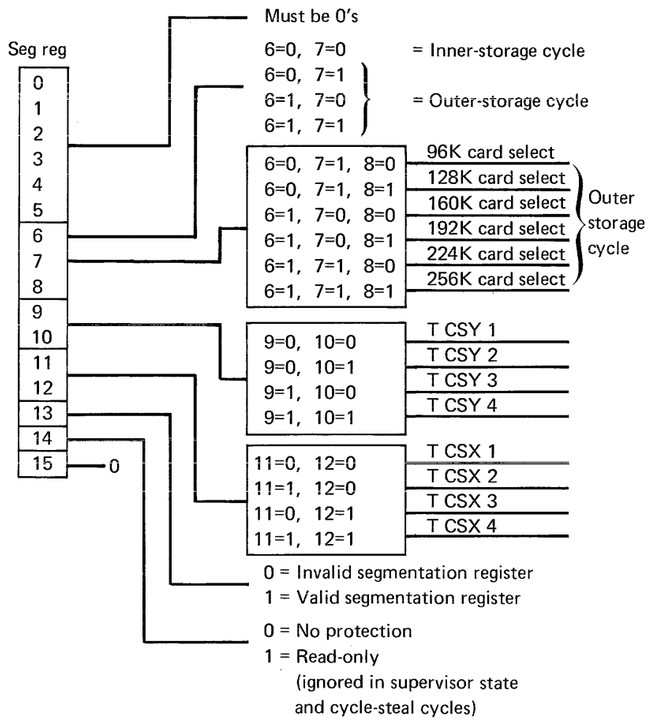
Bit 14 is ignored during the supervisor state or during a cycle-steal access.

Bit 15. Should always be 0.

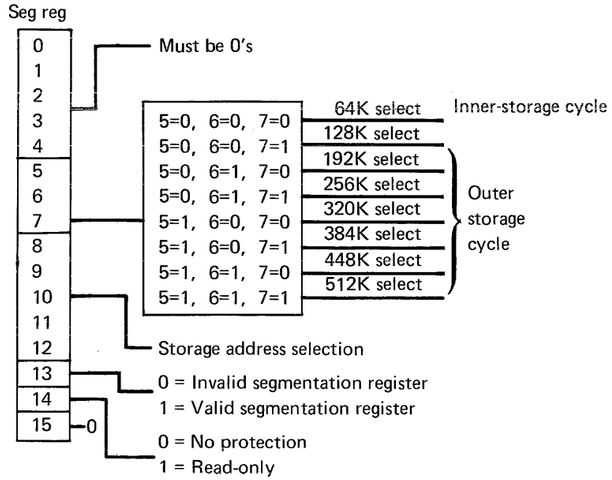
4955 Models B and D



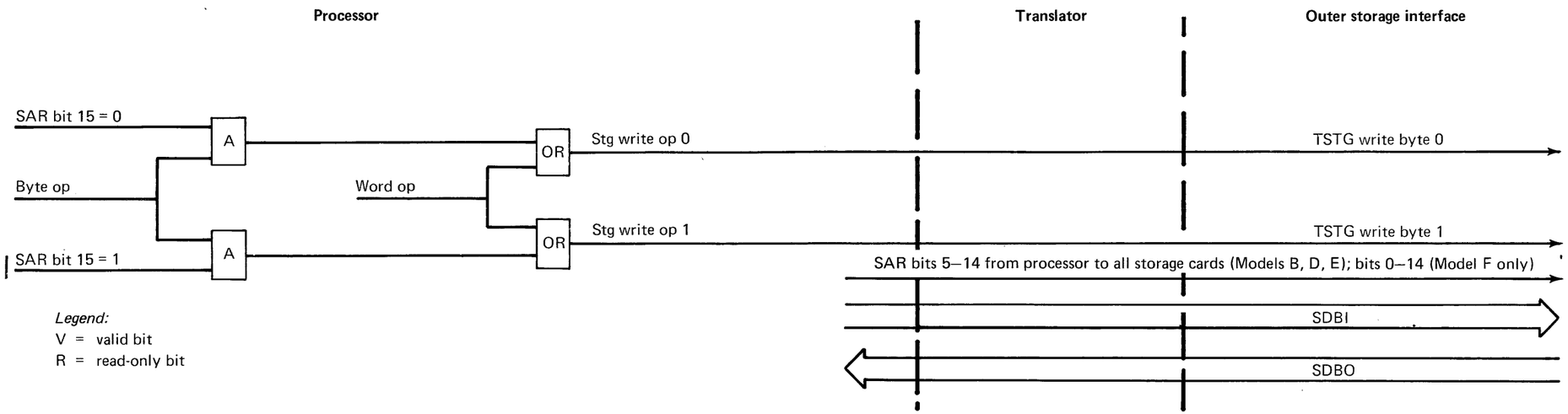
4955 Model E



4955 Model F



Note: The storage size selection is active for installed storage only.



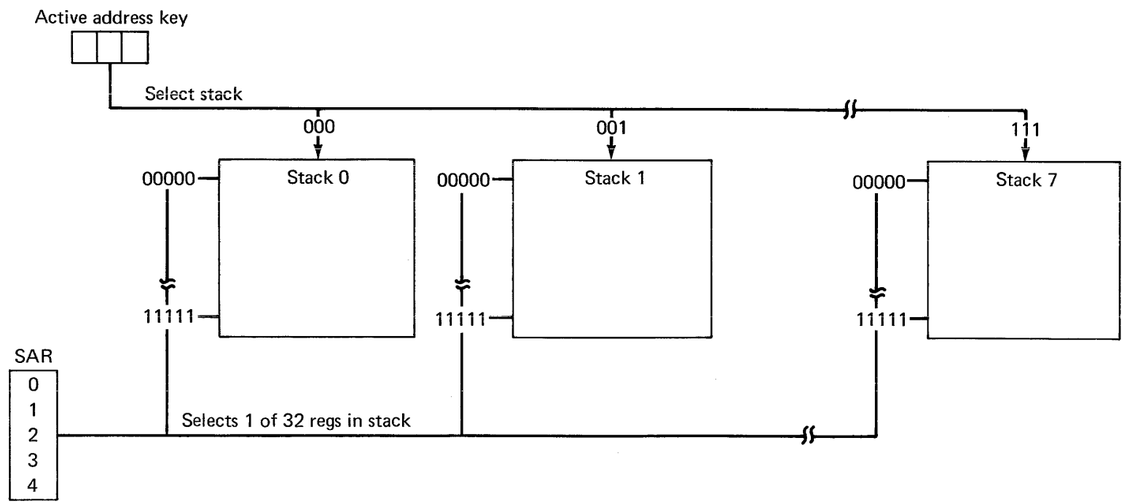
This page intentionally left blank.

This page intentionally left blank.

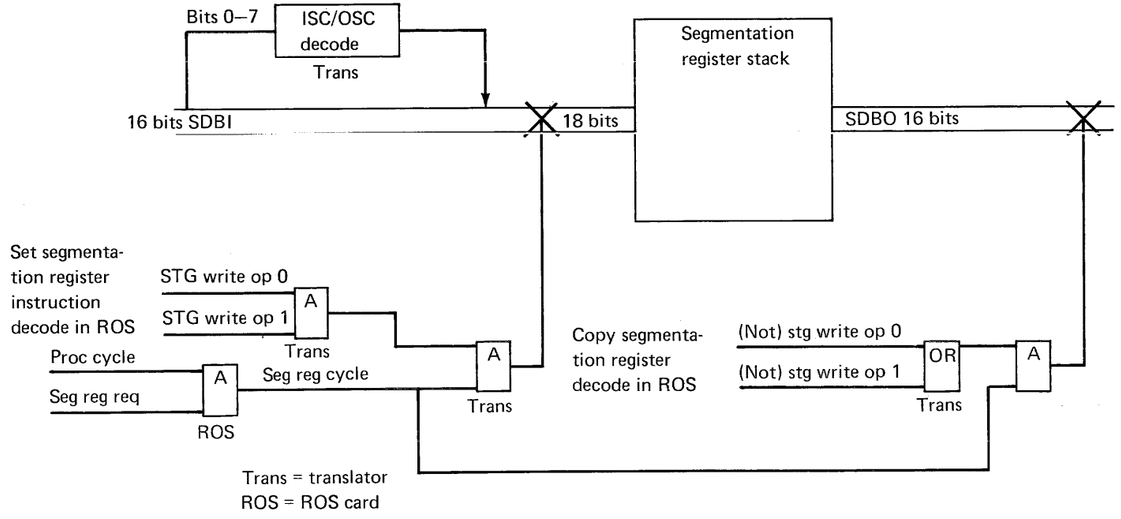
Storage Mapping

The mapping of main storage from logical addresses to physical addresses is accomplished with eight stacks of thirty-two 16-bit segmentation registers and one of five possible active address keys (OP1K, OP2K, ISK, console, or cycle-steal). Each segmentation register controls a 2K-byte segment of storage.

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



Segmentation register selection



Segmentation register input/output gating

Interrupts and Level Switching

Interrupt priority is established by four priority levels of processing: levels 0, 1, 2, and 3, with level 0 having the highest priority. Interrupt levels are assigned to I/O devices via program control. This provides flexibility for reassigning device priority as the application changes.

Processor level switching, under program control, may be accomplished by use of the Set Level Block (SELB) instruction.

Each of the four priority levels has its own set of registers. These consist of an address key register (AKR), a level status register (LSR), eight general registers (R0–R7), and an instruction address register (IAR). Information pertaining to a level is preserved automatically in these hardware registers when an interrupt occurs.

I/O and class interrupts include automatic branching to a service routine. Fixed locations in main storage are reserved for branch addresses or pointers, which are referenced during interrupt processing. This storage allocation is shown in "Automatic Interrupt Branching" in this chapter.

Interrupt masking facilities provide additional program control over the four priority levels. System and level masking are controlled by the summary mask and the interrupt level mask register. Device masking is controlled by the device mask. Manipulation of the mask bits can enable or disable interrupts on all levels, a specific level, or for a specific device. See "Interrupt Masking Facilities" in this chapter.

Interrupt Scheme

Each I/O device is assigned to one of the four existing priority interrupt levels, depending on the application. When an interrupt on a given level is accepted, that level remains active until:

1. A Level Exit (LEX) instruction is executed,
2. A Set Level Block (SELB) instruction causes a level switch, or
3. A higher priority interrupt is accepted

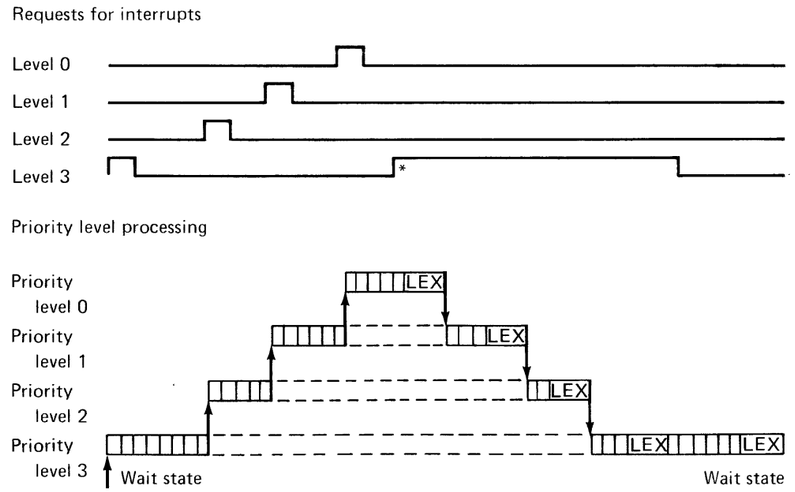
In the latter case, the processor switches to the higher level, completes execution (including an LEX instruction), and then automatically returns to the interrupted-from level. This automatic return can be delayed by other higher priority interrupts.

If an interrupt request is pending on the current active level, it is not accepted until after execution of an LEX instruction by the current program. If no other level of interrupt is pending when an LEX instruction is executed, the processor enters the wait state. In the wait state, no processing is performed, but the processor can accept interrupts that are expected to occur.

Supervisor state is entered upon acceptance of all priority interrupts. The priority interrupt algorithm is:

1. The summary mask must be on (enabled).
2. The mask bit (interrupt level mask register) for the interrupting level must be on (enabled).
3. For I/O interrupts, the device must have its device mask bit on (enabled).
4. The interrupt request must be the highest priority of the outstanding requests and higher than the current level of the processor.
5. The processor must not be in the stop state.

Class interrupts do not change priority levels; they are processed at the currently active level. If the processor is in the wait state when a class interrupt occurs, priority level 0 is used to process the interrupt.



*This interrupt request cannot be honored until the first shown LEX, on priority level 3, has been executed.

Interrupt priority scheme

Automatic Interrupt Branching

Hardware processing of an interrupt includes automatic branching to a service routine. The processor uses a reserved storage area, which begins at main storage address 0000, for branch information. The total size of the area depends on the number of interrupting devices attached. One word (two bytes) is reserved for each interrupting device. The device area begins at address 0030 (hex); the reserved area is 0000 through 022F (hex) if 256 devices (maximum number) are attached. These storage locations and contents are shown here:

Main storage address (hex)	Contents of word
022E	Device FF DDB pointer
≈	≈
0032	Device 01 DDB pointer
0030	Device 00 DDB pointer
002E	Reserved
002C	Reserved
002A	Reserved
0028	Reserved
0026	Reserved
0024	Reserved
0022	Soft-exception-trap SIA
0020	Soft-exception-trap LSB pointer
001E	Console-interrupt SIA
001C	Console-interrupt LSB pointer
001A	Trace SIA
0018	Trace LSB pointer
0016	Power-failure SIA
0014	Power-failure LSB pointer
0012	SVC SIA
0010	SVC LSB pointer
000E	Program-check SIA
000C	Program-check LSB pointer
000A	Machine-check SIA
0008	Machine-check LSB pointer
0006	Reserved
0004	Reserved
0002	Restart instruction word 2
0000	Restart instruction word 1

Addresses used for I/O interrupts. The device data block (DDB) pointer is the address of the first word of a device data block. This word is used to obtain the start instruction address for the service routine. Refer to "I/O Interrupts" in this chapter.

Address used for class interrupts. The level status block (LSB) pointer is the first address of an area where a level status block will be stored. The start instruction address (SIA) points to the first instruction of a service routine.

Restart instruction. Following IPL, a forced branch is made to address 0000.

Reserved storage locations

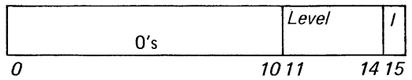
Note: The area reserved for I/O devices varies in size depending on the number of devices. The device address determines the fixed location to be accessed. For example, interrupts for device 01 always vector to main storage address 0032. Device addresses range from 00 through FF (hex).

I/O Interrupts

Prepare I/O Device for Interrupt

I/O device interrupt parameters are established via program control. The Operate I/O (IO) instruction initiates the device operation and, in conjunction with the Prepare I/O command, indicates to the device (1) if the device can interrupt and (2) what priority level to use for interrupts.

Execution of the Prepare command transfers a word to the addressed device that controls its interrupt parameters. This word has the format:



Bits	Contents
0–10	Set to 0's.
11–14	*Level.* A four-bit encoded field that assigns an interrupt priority level to the device (see Note). *Example:* 0000 is level 0, 0001 is level 1, 0010 is level 2, and 0011 is level 3.
15	*Device mask or I-bit.* This bit sets the interrupt mask in the device. When set to 1, the device can interrupt. When set to 0, the device cannot request an interrupt.

Note: The 4955 does not recognize priority levels other than 0 through 3; therefore, bits 11 and 12 must always be set to 0's or the interrupt is lost.

An interrupting device can always accept and execute a Prepare command, even if it is presently busy or has an interrupt request pending from a previous command. This allows the software to change the device mask and interrupt level at any time.

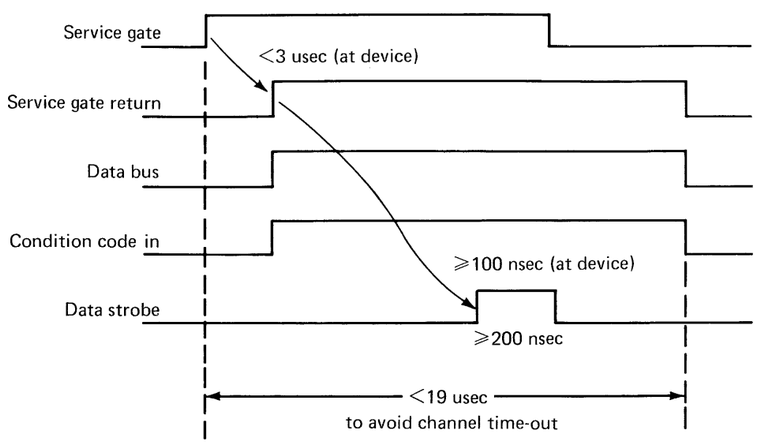
Present and Accept Interrupt

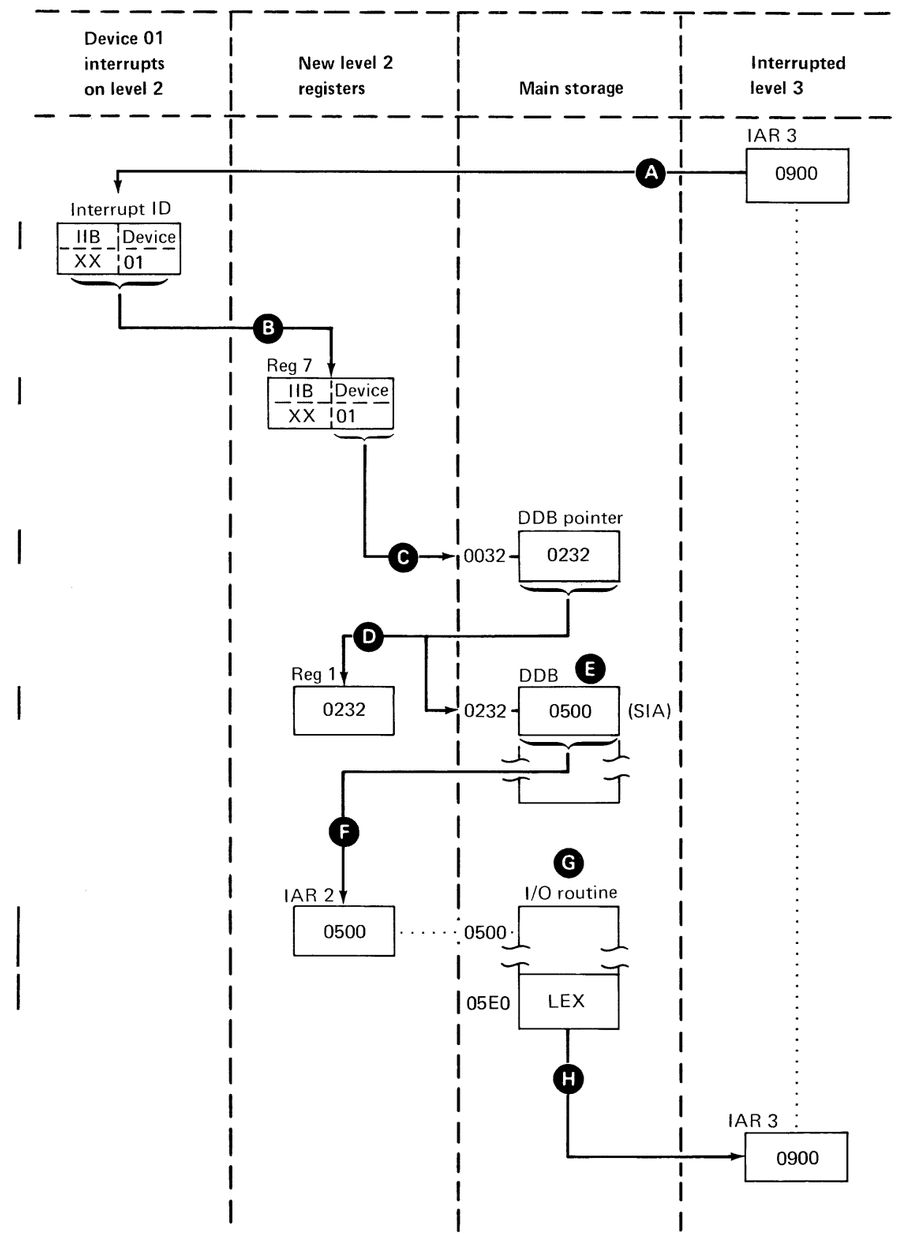
The I/O device presents an interrupt request on its assigned priority level. This request is applied to the interrupt algorithm for acceptance determination. Following acceptance, the device sends an interrupt ID word and a condition code to the processor. The interrupt ID word consists of an interrupt information byte (IIB) and the device address. Bits 0–7 of this word contain the interrupt information; bits 8–15 contain the device address. The condition code is placed in the Even, Carry, and Overflow indicators for the interrupted-to level.

The following events occur after the processor receives the interrupt ID word:

1. The processor hardware switches from the registers and status of the interrupted-from level to those of the interrupted-to level Ⓐ.
2. The interrupt ID word is placed in register 7 of the interrupted-to level Ⓑ.
3. The processor executes an automatic branch.
   a. The device address is used by hardware to fetch the device data block (DDB) pointer from reserved storage Ⓒ.
   b. The DDB pointer is placed in register 1 of the interrupted-to level Ⓓ.
   c. The DDB pointer is used by hardware to fetch the start instruction pointer Ⓔ.
   d. The start instruction address (SIA) is loaded into the IAR of the interrupted-to level Ⓕ.
4. Execution begins on the new level Ⓖ.
   a. Execution of the LEX instruction returns control to the interrupted level Ⓗ.

Interrupt ID presentation





Example of I/O interrupt with automatic branching

Class Interrupts

System error or exception conditions can cause seven types of class interrupts. All class interrupts cause the processor to enter supervisor state (LSR bit 8 = 1). Certain class interrupts are further defined in the PSW. (Each class interrupt is described in subsequent paragraphs.) The seven types are:

- Machine check, caused by a hardware error
- Program check, caused by a programming error
- Power/thermal warning, caused by a power or thermal irregularity
- Supervisor call, caused by execution of an SVC instruction
- Soft-exception trap, caused by software
- Trace, caused by instruction execution (trace enabled in the current LSR)
- Console, caused by a console interrupt when the optional programmer console is installed

Software can refer to the processor status word for the specific condition and any related status information. The Copy Processor Status and Reset (CPPSR) instruction can be used for this purpose. Power/thermal warning and console interrupts are controlled by the summary mask (LSR bit 11); all other class interrupts cannot be disabled. If the optional programmer console is installed and check restart is selected, machine-check and program-check interrupts do not occur. If stop-on-error mode is selected, the processor stops and then takes the class interrupt when the Start key is pressed, provided that no system reset occurred.

A class interrupt does not cause a change in priority level. However, supervisor state is entered, trace is reset, and all priority-interrupt requests are automatically disabled (summary mask bit set to 0). The address key register (AKR) is set as follows: (1) equate operand spaces bit is set to 0, (2) ISK and OP2K address keys are set to 0's, and (3) OP1K address key is set depending on the type of class interrupt. The class interrupt is serviced on the level that is active when the condition occurs. If the processor is in the wait state, the interrupt is serviced on priority level 0.

Note: Address key values are set in anticipation of the address spaces required by the interrupt service routine.

Priority of Class Interrupts

Although class interrupts are serviced on the current priority level, they are serviced according to exception condition priority.

Priority	Exception condition	Class interrupt routine
0	CPU control check I/O check	Machine check
1	Invalid function (Note 1)	Program check
2	Privilege violate	
3	Invalid function (Note 2)	
4	Protect check Specification check	
5	Invalid storage address Specification check	
6	Storage parity	Machine check
7	Power warning Thermal warning	Power/thermal warning
8	Supervisor call	Supervisor call
9	Invalid function (Note 3)	Soft-exception trap
10	Floating-point exception	
11	Stack exception	
12	Trace	Trace
13	Console	Console

Notes:

1. Caused by an illegal operation code or function combination.
2. A Copy Segmentation Register (CPSR) or Set Segmentation Register (SESR) instruction is attempted and the translator feature is not installed.
3. A floating-point instruction is attempted and the floating-point feature is not installed.

Execution of Class Interrupts

Occurrence of a class interrupt causes the processor to enter supervisor state, to disable trace, and to set the summary mask to 0 (disable). Reference is made to the reserved area of main storage to:

1. Store current level IAR, AKR, registers, and LSR into a level status block (LSB) in main storage.
2. Branch automatically to a service routine by using the start instruction address (SIA).

Note: Priority level 0 is forced active when a class interrupt occurs in the wait state. The level 0 LSB is stored into main storage. The in-process flag (LSR bit 9) is 0 in the stored LSB.

Each type of class interrupt has an associated LSB pointer and SIA. Contents of the level status block are as follows:

Main storage address (LSB) pointer

	Instruction address register
	Address key register
	Level status register
	Register 0 (R0)
	Register 1 (R1)
	Register 2 (R2)
	Register 3 (R3)
	Register 4 (R4)
	Register 5 (R5)
+14 (hex)	Register 6 (R6)
	Register 7 (R7)

0 *15*

The instruction address (contents of IAR) stored in the LSB depends on the type of class interrupt, as shown in the following chart:

Type of class interrupt	Contents of IAR (stored in LSB)
Program check Soft-exception trap	Address of the instruction that caused the interrupt
Supervisor call Trace Console Power/thermal warning	Address of the next instruction
Machine check (with sequence indicator off)	Address of the instruction that caused the interrupt
Machine check (with sequence indicator on)	Address of the instruction that was being executed at the time of the error

Machine Check. A machine-check interrupt is caused by a hardware malfunction and is considered a system-wide incident. The three types are:

- Storage parity check (PSW bit 8)
- CPU control check (PSW bit 10)
- I/O check (PSW bit 11)

An LSB is stored, starting at the location in main storage designated by the machine-check LSB pointer (contents of storage location hex 0008 and 0009). The contents of the storage address register (SAR) are loaded into register 7. The last active processor address key is loaded into the OP1K address key of the AKR. The machine check SIA (contents of storage locations hex 000A and 000B) is then loaded into the IAR, and it becomes the address of the next instruction to be fetched.

Note: When the error condition occurs:

- The IAR contains the true address of the first word of the instruction; it is not incremented if the error occurs in the second or third word of a long instruction.
- For a storage parity check, the last active processor key defines the space corresponding to the storage address loaded into register 7. For a CPU control check or an I/O check, this key provides no useful information.

Program Check. A program-check interrupt is caused by a programming error. The types are:

- Specification check (PSW bit 0)
- Invalid storage address (PSW bit 1)
- Privilege violate (PSW bit 2)
- Protect check (PSW bit 3)
- Invalid function (PSW bit 4)

An LSB is stored, starting at the location in main storage designated by the program-check LSB pointer (contents of storage locations hex 000C and 000D). The contents of the storage address register (SAR) are loaded into register 7. The last active processor address key is loaded into the OP1K address key of the AKR. The program-check SIA (contents of storage locations hex 000E and 000F) is then loaded into the IAR, and it becomes the address of the next instruction to be fetched.

Notes:

1. A program-check interrupt condition on one priority level does not affect software on other levels.
2. For a specification check, an invalid storage address, and a protect check, the last active processor key defines the address space corresponding to the storage address loaded into register 7. For privilege violate, this key provides no useful information.

Power/Thermal Warning (PSW Bit 15). A power/thermal warning class interrupt is initiated by:

1. A power warning signal that is generated when the power line decreases to about 85% of its rated value, or
2. A thermal warning that occurs if the temperature limits inside the enclosure are exceeded

In both cases, the instruction address that is stored in the LSB points to the next instruction to be executed. The starting address for the LSB is designated by the power-failure LSB pointer (contents of storage locations hex 0014 and 0015). The OP1K address key in the AKR is set to 0. The power-failure SIA (contents of storage locations hex 0016 and 0017) is then loaded into the IAR, and it becomes the address of the next instruction to be fetched.

A power/thermal-warning interrupt can occur when the system is running or in the wait state, assuming that (1) the summary mask is enabled and (2) the programmer console is not set to check restart or stop-on-error mode. These interrupts are not taken by the processor if the summary mask is disabled.

If the optional battery backup unit is installed and a power warning occurs, PSW bit 15 remains on as long as power is supplied by the battery. If a thermal warning occurs, the processor powers down regardless of the battery backup unit. The minimum time before the processor powers down is 20 milliseconds.

Power/thermal warning interrupts are not taken by the processor until the first instruction is executed following a power-on reset, an IPL, or exit from stop state.

Note: If the processor is in the wait state when the power/thermal condition occurs:

- The interrupt is serviced on priority level 0. The level 0 LSB is stored into main storage. Additional power/thermal interrupts are disabled at this time because the summary mask is set to 0 by the class interrupt.
- The instruction address stored in the LSB is unpredictable.

Supervisor Call. A supervisor-call class interrupt is initiated by executing an SVC instruction. (The SVC instruction is described in the *IBM Series/1 Common Features Theory Diagrams*, SY34-0091.) An LSB is stored, starting at the main storage location designated by the supervisor call LSB pointer (contents of storage locations hex 0010 and 0011). The OP2K address key is placed into the OP1K address key in the AKR; then OP2K, EOS bit, and ISK are set to 0's. The supervisor-call SIA (contents of storage locations 0012 and 0013) is then loaded into the IAR, and it becomes the address of the next instruction to be fetched.

Soft-Exception Trap. A soft-exception-trap interrupt is caused by software. The types are:

1. Invalid function (PSW bit 4)
2. Floating-point exception (PSW bit 5)
3. Stack exception (PSW bit 6)

These exception conditions may be handled by software; therefore, they do not constitute an error condition.

An LSB is stored, starting at the location in main storage designated by the soft-exception-trap LSB pointer (contents of storage locations hex 0020 and 0021). The contents of the storage address register (SAR) are loaded into register 7. The OP2K address key is placed into the OP1K address key in the AKR. The soft-exception-trap SIA (contents of storage locations hex 0022 and 0023) is then loaded into the IAR, and it becomes the address of the next instruction to be fetched.

Note: For invalid function, the main storage address loaded into register 7 is either: (1) the calculated effective address of data operand 2, or (2) the address of the attempted instruction for register-to-register operations.

Trace. The trace class interrupt provides an instruction trace facility for software debugging. Instruction tracing may occur on any priority level, and is enabled by the trace bit (LSR bit 10). The tracing occurs when bit 10 of the current LSR is set to 1. When trace is enabled, a trace class interrupt occurs at the beginning of each instruction. A level status block is stored, starting at the location in main storage designated by the trace LSB pointer (contents of storage locations hex 0018 and 0019). The trace SIA (contents of storage locations hex 001A and 001B) is then loaded into the IAR, and it becomes the address of the next instruction to be fetched.

Note: After the LSB is stored, and before the next instruction is fetched, supervisor state is set on (LSR bit 8), trace is turned off (LSR bit 10), and the summary mask is disabled (LSR bit 11).

The contents of the address key register (AKR) are set as follows:

1. EOS (bit 0) is set to 0 (disabled).
2. OP1K address key (bits 5–7) is set to the value of the ISK address key.
3. OP2K address key (bits 9–11) and ISK address key (bits 13–15) are set to 0's.

Programming Note: When trace is enabled, a trace class interrupt occurs prior to executing each instruction. Hardware processing of the interrupt provides an automatic branch to the programmer's trace routine. To prevent retracing the same instruction, the program should exit the trace routine by using the Set Level Block (SELB) instruction with the inhibit-trace (IT) bit set to 1. The inhibit-trace bit prevents a trace interrupt from occurring for the duration of one instruction. (The SELB instruction is described in the *IBM Series/1 Common Features Theory Diagrams*, SY34-0091.) A double trace of an instruction can also occur when the instruction is interrupted and must be reexecuted. For example, a class interrupt occurs during execution of a variable-field-length instruction. Under this condition, exit from the class interrupt routine should be via an SELB instruction, with the inhibit-trace bit set to 1.

The occurrence of any class interrupt or priority interrupt causes the trace bit (LSR bit 10) to be set to 0. This action permits tracing only problem state code. If it is desired to trace supervisor code, the programmer must make provisions within the service routine to enable the trace bit.

The following two conditions inhibit a trace class interrupt:

1. A Set Level Block (SELB) sets the trace bit on and the in-process bit on in the LSR of a selected level lower than the current level; then, when the selected level becomes active, the first instruction executed is not preceded by a trace interrupt.
2. The programmer console is in diagnose mode and a stop instruction is encountered while tracing; then, when the Start key is pressed, a trace interrupt does not occur prior to executing the first instruction.

Console. A console interrupt function is provided when the optional programmer console is installed. To recognize the interrupt, the processor must have the summary mask enabled and be in the run state or wait state. An LSB is stored, starting at the main storage location designated by the console-interrupt LSB pointer (contents of storage locations hex 001C and 001D). The OP1K address key is set to 0. The console interrupt SIA (contents of storage locations hex 001E and 001F) is then loaded into the IAR, and it becomes the address of the next instruction to be fetched.

Note: If the processor is in the wait state when a console interrupt occurs, the interrupt is serviced on priority level 0.

Program-Controlled Level Switching

Level switching under program control may be accomplished by using the Set Level Block (SELB) instruction. This instruction, in general:

- Specifies the location of a level status block (LSB) at an effective address in main storage
- Specifies a selected priority level associated with the main storage LSB
- Loads the main storage LSB into the hardware LSB for the selected level

Note: The hardware LSB consists of the following hardware registers for the selected level:

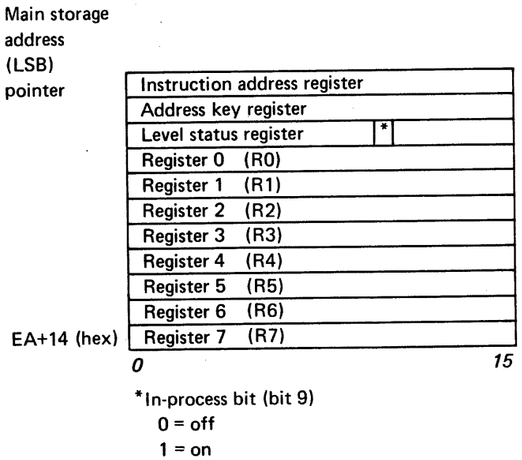
- Instruction address register
- Address key register
- Level status register
- Eight general registers (0–7)

The system programmer should become thoroughly familiar with other effects on the processor caused by execution of the SELB instruction. These effects are determined by three factors:

- The current execution level
- The selected level specified in the SELB instruction
- The state of the in-process bit (LSR bit 9) contained in the main storage LSB

Note: Interrupt masking, provided by the summary mask and the interrupt-level mask register, does not apply to program-controlled level switching.

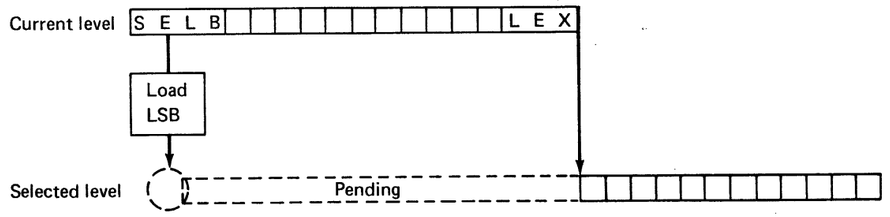
The main storage LSB and the location of the in-process bit are shown in the following diagram:



Execution of the SELB instruction may result in level switching or a change in the pending status of a level, as described in the following paragraphs:

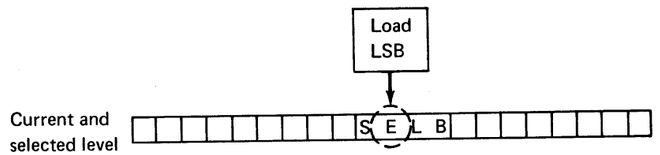
Selected Level Lower Than Current Level and In-process Bit On

These conditions cause the selected level to be pending. The main storage LSB is loaded into the hardware LSB for the selected level. Execution of an LEX instruction on the current level causes the selected level to become active providing no higher priority interrupts are being requested.



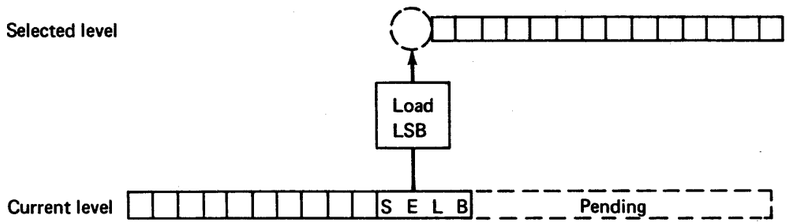
Selected Level Equal to Current Level and In-process Bit On

These conditions cause the hardware LSB to be overwritten by the LSB from main storage, with no level change. The effect is a task-switch on the current level. The main storage LSB is loaded into the hardware LSB for the selected level.



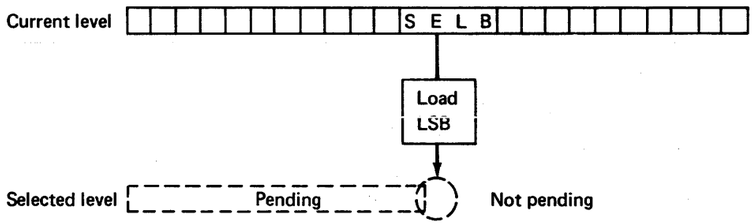
Selected Level Higher Than Current Level and In-process Bit On

These conditions cause the selected level to become the current level. The main storage LSB is loaded into the hardware LSB for the selected level. This is a level switch to the higher (selected) level and causes the lower level to be pending.



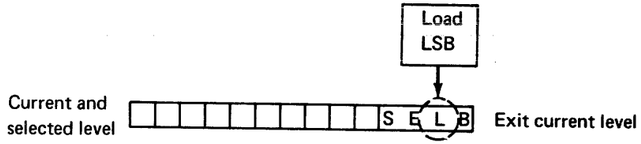
Selected Level Lower Than Current Level and In-process Bit Off

These conditions cause the selected level to be not pending. The main storage LSB is loaded into the hardware LSB for the selected level.



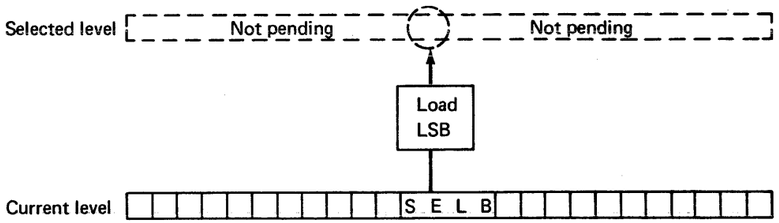
Selected Level Equal to Current Level and In-process Bit Off

These conditions cause an exit from the current level. The main storage LSB is loaded into the hardware LSB for the selected level.



Selected Level Higher Than Current Level and In-process Bit Off

The main storage LSB is loaded into the hardware LSB for the higher (selected) level.



Interrupt Masking Facilities

Three levels of priority interrupt masking are provided to the programmer for control of the interrupt processing. These consist of:

- Summary mask (LSR bit 11)
- Interrupt level mask register
- Device mask (I-bit)

Each masking facility has specific control, as explained in the following sections.

Summary Mask

The summary mask provides a masking facility for priority interrupts and certain class interrupts. The state of the summary mask (enabled or disabled) is controlled by bit 11 in the level status register (LSR) of the active priority level. When bit 11 is set to 0, the summary mask is disabled and prevents (1) all priority interrupts regardless of priority level, and (2) power/thermal and console class interrupts. All other class interrupts are not masked. When bit 11 is set to 1, the mask is enabled and the interrupts are allowed.

The summary mask is disabled and enabled as follows:

- Disabled (set to 0)
  - When a Supervisor Call (SVC) instruction is executed, the summary mask for the active level is disabled.
  - Execution of a Disable (DIS) instruction, with bit 15 of the instruction equal to 1, causes the summary mask for the active level to be disabled.
  - All class interrupts disable the active level summary mask.
  - The summary mask for a selected level is disabled by executing a Set Level Block (SELB) instruction with bit 11 of the LSR to be loaded equal to 0.
  - The summary mask bits for priority levels 1–3 are set to 0's by a system reset, power-on reset, or IPL.
- Enabled (set to 1)
  - Execution of an Enable (EN) instruction, with bit 15 of the instruction equal to 1, causes the active level summary mask to be enabled.
  - The summary mask for a selected level is enabled by executing a Set Level Block (SELB) instruction with bit 11 of the LSR to be loaded equal to 1.
  - The level 0 summary mask is enabled by a system reset, power-on reset, or IPL.
  - The summary mask for the interrupted-to level is enabled by a priority interrupt.

Note: If the processor is in the wait state, the summary mask is enabled or disabled, as defined by bit 11 in the LSR of the last active priority level.

Interrupt Level Mask Register

The interrupt level mask register is a four-bit register used to control interrupts on specific priority levels. Each level is controlled by a separate bit of the mask register, as shown here:

Interrupt level mask register

Bit position	0	1	2	3
	□	□	□	□
Priority level	0	1	2	3

With a bit position set to 1, the corresponding priority level is enabled and permits interrupts. With a bit position set to 0, the corresponding priority level is disabled. The Set Interrupt Mask Register (SEIMR) instruction is used to control bit settings in the interrupt level mask register. The Copy Interrupt Mask Register (CPIMR) instruction may be used to interrogate the register.

Note: All levels are enabled (set to 1's) by a system reset, power-on reset, or IPL.

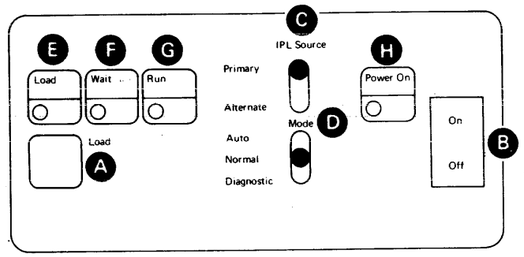
Device Mask (I-bit)

Each interrupting device contains a one-bit mask called the device mask or interrupt bit (I-bit). Interrupts by the device are permitted when its device mask is enabled (set to 1). With the device mask bit disabled (set to 0), that device cannot cause an interrupt. The device mask is controlled by a Prepare command in conjunction with an Operate I/O instruction.

Basic Console

Each IBM 4955 Processor contains the standard Basic Console, that provides the following capabilities:

- Power On/Off switch for the processor card file
- Load key for IPL (initial program load)
- Load, Wait, Run, and Power On indicators
- Mode switch to select: diagnostic, auto IPL, or normal mode
- IPL source switch to select primary or alternate IPL device



Keys and Switches

A Load — Pressing this key causes a system reset; then 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 is completed, instruction execution begins at location 0 on level 0.

B Power On/Off — When this switch is set to On, power is applied to the processor card file. After all power levels are up, the Power On indicator comes on. When this switch is set to Off, power is removed from the processor unit and the Power On indicator goes off.

C IPL Source — This switch selects the I/O device to be used for program loading. In the Primary position, the device that was prewired as the primary IPL device is selected. In the Alternate position, the device that was prewired as the alternate IPL device is selected.

D 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.
- Normal—This position is used when the system is being attended and programmed stops are not desired.
- Diagnostic—This position places the processor in diagnostic mode if the programmer console is attached. This position has no function without the programmer console. When the processor is in diagnostic mode, Stop instructions cause the processor to enter stop state. Stop instructions are treated as No-ops in any other mode.

Indicators

E Load — This indicator is on when the machine is performing an initial program load (IPL) or external IPL.

F Wait — This indicator is on when the active level has been exited and no other levels or interruptions are pending.

G Run — This indicator is on when the machine is executing instructions.

H Power On — This indicator is on when the proper power levels are available to the processor.

Programmer Console

The programmer console is an optional feature that can be ordered with the IBM 4955 Processor or may be field-installed at a later date. The programmer console:

- Starts and stops the processor
- Displays or alters any storage location
- Selects any of the four interrupt levels for display or alter purposes
- Displays or alters the storage address register (SAR), instruction address register (IAR), console address key register (AKR), console data buffer, or any general purpose register
- Displays, but does not alter, the level status register (LSR), current instruction address register (CIAR), op register, level address key register (AKR), or processor status word (PSW)

The programmer console also provides:

- A stop-on-address
- A stop-on-error
- An instruction step
- A check restart
- A request a console interrupt
- A check indicator, on when a machine-check or program-check class interrupt has occurred

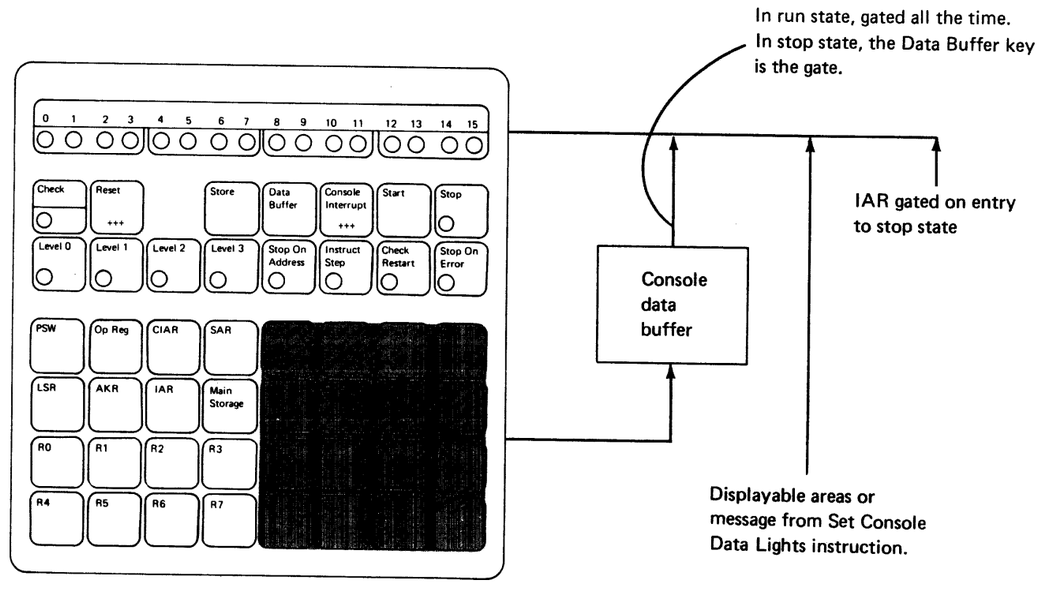
The programmer console keys are touch-sensitive. An audio-tone generator is installed to provide a response when the information resulting from pressing a key has been accepted and serviced by the processor.

Console Display

When the processor is in run state, the console data buffer is displayed in the data display indicators. The only exception to this is when a Set Console Data Lights instruction writes a message to the data display. This message remains displayed until the processor enters stop state or the Data Buffer key is pressed. When the Data Buffer key is pressed, the console data buffer is again displayed in the data display indicators.

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 while in stop state. Once data has been entered into the console data buffer, it remains there until other data is entered. The console data buffer can be displayed at any time, during either run or stop state, by pressing the Data Buffer key.

After a power-on reset, the data display indicators are all set on and the level indicators are set off.



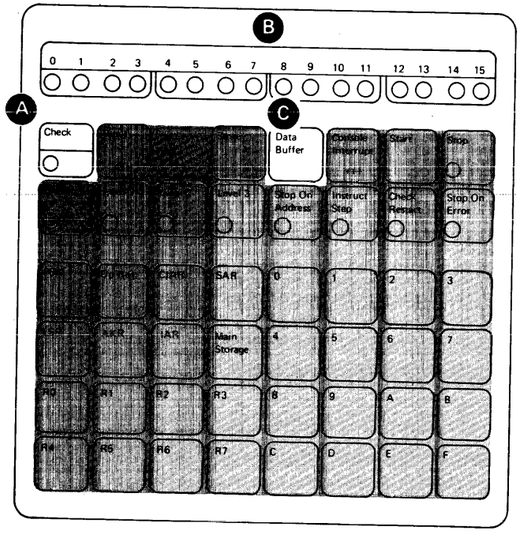
Indicators

Ⓐ Check

Turned on when a machine-check or program-check class interrupt has occurred. The Check indicator remains on until either the check condition is cleared or any console key is pressed while in stop state. The check condition is cleared by pressing the Reset key or the Load key, or by the execution of a Store PSW instruction (which resets the check bits). If a main storage display of a location with a parity error is performed, the Check indicator is turned on, or appears to remain on.

Ⓑ Data Display

- When the processor is in run state, the console data buffer is displayed in the data display indicators, unless the Set Console Data Lights instruction is executed and looped.
- 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 Ⓒ.



Combination Keys/Indicators

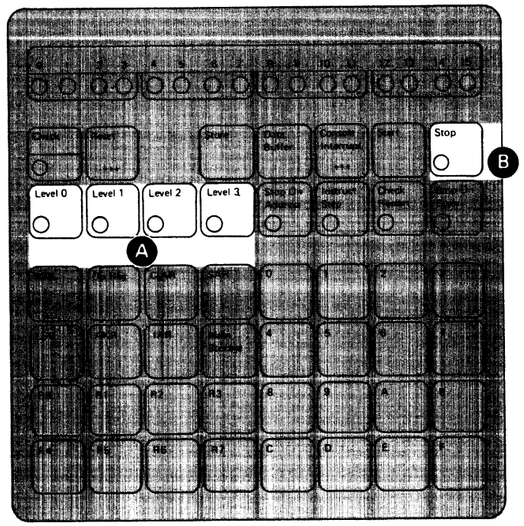
There are nine combination key/indicators:

- Level 0, 1, 2, and 3
- Stop
- Stop on Address
- Instruct Step
- Check Restart
- Stop On Error

● **Level 0–3** — The current active level is always displayed by one of the level indicators. When in the Stop state, pressing any of the level keys causes that level to be selected and the associated indicator to be turned on.

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

- 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 Stop key
  - In run state, the current instruction is completed
  - In load state, the IPL is completed and the first instruction is executed
  - In wait state, stop state is entered directly
  - In stop state, the contents of the IAR, upon entering 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 pressing the Reset key
- When a power-on reset occurs, and not in auto IPL mode



The Stop On Address Key and the Instruct Step key work in toggle fashion. When one is pressed, the other is reset if it is on.

- Stop on Address — This key places the processor in stop-on-address mode. When the Stop On Address key is pressed again; it resets stop-on-address mode and turns off the indicator.

Stop On Address Mode

The processor must be in stop state to set or change the comparison address.

1. If the translator is not installed and enabled, go to step 2, otherwise:
   a. Press AKR key (selects console AKR)
   b. Key in the address key (ISK bits of AKR).
   c. Press Store key.
2. Press Stop On Address key.
3. Key in selected address.
4. Press Store key. The selected address is placed in the stop-on-address buffer.
5. Press Start key. Execution begins at current IAR address on the current level.

When the selected address is loaded into the IAR, the processor enters stop state. To exit stop state, press the Start key; execution begins at the next sequential address.

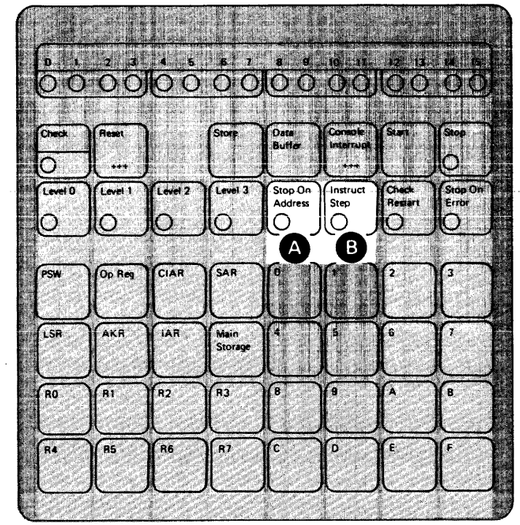
When in stop-on-address mode, each instruction time is increased by 1.54 microseconds (2.2 microseconds if the address translator is enabled).

- Instruct Step — Pressing the Instruct Step key places the processor in instruction step mode and turns the Instruct Step indicator on. The Stop On Address indicator is turned off.

  If the processor is in run 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 in 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, the processor reenters stop state. The IAR is updated to the next instruction address; this address is displayed in the data display indicators.
4. Each time the Start key is pressed, it causes one instruction to be executed and the IAR to be updated to the next instruction address.



The Check Restart key and the Stop On Error key work in toggle fashion. When one is pressed, the other is reset if it is on.

Ⓐ **Check Restart** When running in check restart mode, any program check, machine check, or power/thermal warning, causes the processor to be reset, and execution resumes at address 0 on level 0.

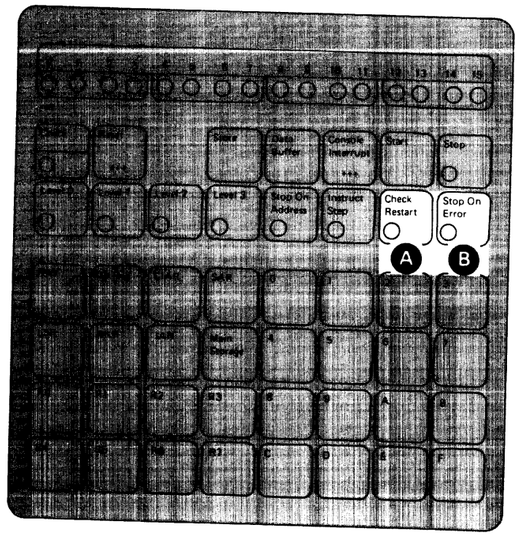
Ⓑ **Stop On Error** Pressing this key places the processor in stop on error mode. Any program check, machine check, or power/thermal warning causes the processor to enter Stop state. To determine the cause of the error, display the PSW (see PSW Table). To restart the processor, press the Reset key and then the Start key. Pressing only the Start key allows the processor to proceed with the class interrupt as if it were in process mode. (This occurs even if the check indicator was 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. Note that 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.

PSW Table

Bit	Meaning	Category
0	Specification check	Program check
1	Invalid storage address	Program check
2	Privilege violate	Program check
3	Protect check	Program check
4	Invalid function (may be program check)	Soft exception
5	Floating-point exception	Soft exception
6	Stack exception	Soft exception
7	Not used; always 0	
8	Storage parity check	Machine check
9	Not used; always 0	
10	CPU control check	Machine check
11	I/O check	Machine check
12	Sequence indicator	Status flag
13	Auto IPL	Status flag
14	Translator enabled	Status flag
15	Power/thermal warning	

Power/Therm

Note: Bits not used are always 0's.



Keys and Switches

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

- Processor, channel, and translator hardware reset.
- IAR on level 0 set to 0's
- AKR on level 0 set to 0's
- Console AKR set to 0's
- Interrupt mask set to all levels enabled
- LSR on level 0—indicators set to 0's, summary mask enabled, supervisor state and in-process bit turned on, trace disabled.
- LSRs for levels 1–3 set to 0's
- PSW set to 0's (not auto IPL; not power/thermal warning)
- SAR set to 0's
- CIAR set to 0's

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:

- GP registers (all levels)
- IARs (levels 1–3)
- AKRs (levels 1–3)
- Storage key stack
- Main storage
- Console data buffer
- Segmentation registers (translator feature)
- Floating point registers (floating-point feature)
- Stop-on-Address buffer

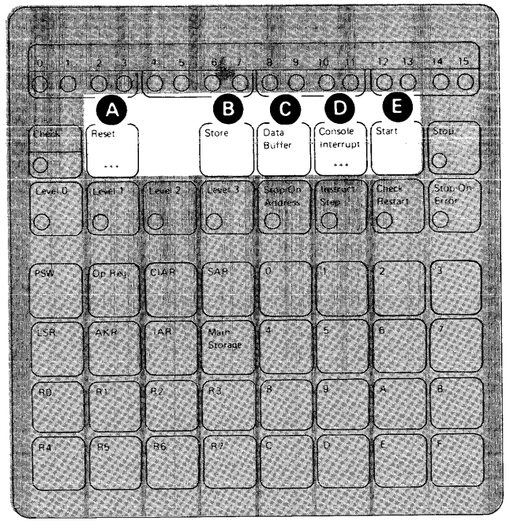
B Store — This key is effective only when the processor is in stop state. Pressing this key causes the last data entry to be stored in the last selected resource. If the last selected resource was main storage, pressing the Store key increments the SAR by 2.

C Data Buffer — Pressing this key causes the contents of the console data buffer to be displayed in the data display indicators.

D Console Interrupt — The function 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, a console class interruption occurs.

E Start — This key is effective in stop state only. The level that was current on entry to stop state is made current; then 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.

Note: The Reset and Console Interrupt keys have an indication (+++) on the face of the keys. Additional pressure must be used to activate these keys. This is to prevent the operator from inadvertently activating these functions.



Keys and Switches (continued)

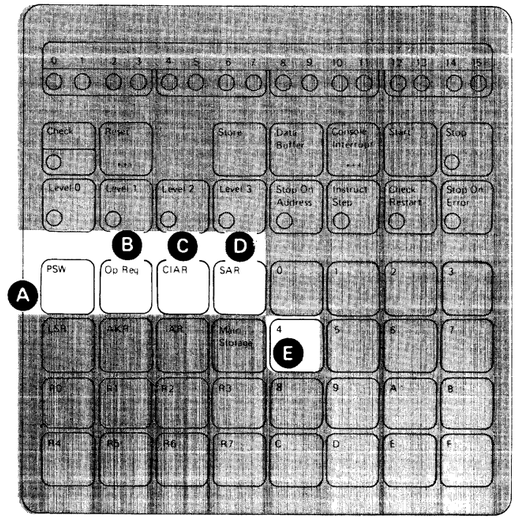
A PSW — Pressing this key causes the processor status word to be selected. The contents of the PSW are displayed in the data display indicators.

B Op Reg — Pressing this key causes the operations register to be selected and its contents to be displayed in the data display indicators. Data cannot be stored into the op register from the console.

C CIAR — Pressing this key after entering stop state causes the address of the instruction last executed to be displayed. Data cannot be stored into the CIAR from the console.

D SAR — Pressing this key while in stop state causes the contents of the storage address register to be displayed. An address can be stored into the SAR to address main storage for display or store operations. Bit 15 of the SAR cannot be set from the console.

E Main Storage — Pressing this key causes main storage to be selected as the facility to be accessed by the console. When this key is pressed, the contents of the two-byte main storage location addressed by the SAR is displayed in the data display indicators. After the Main Storage key is pressed once, each subsequent operation of the Main Storage key causes the SAR to be incremented by 2. The first time the Store key is pressed after main storage is selected, the SAR is decremented by 2, the data stored into the main storage location last displayed, and the SAR is incremented by 2.



Level-Dependent Keys

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

- LSR
- AKR
- IAR
- General purpose registers 0–7

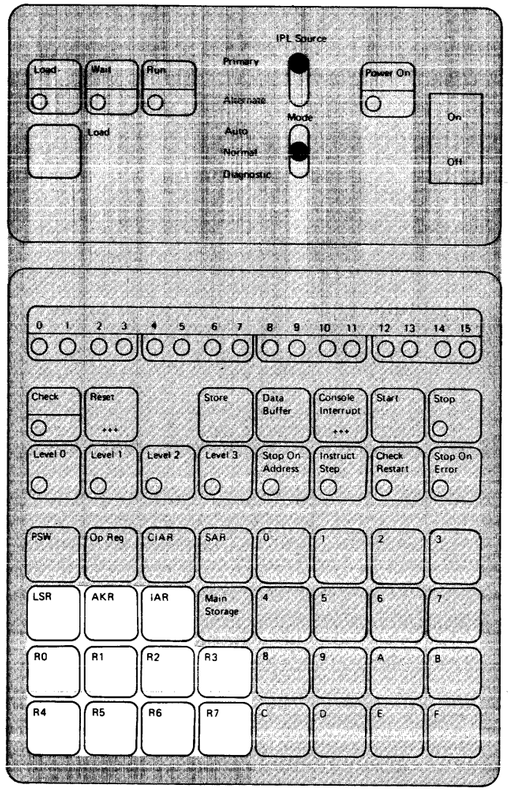
Pressing any of these keys, once a level has been selected, causes the contents of that register to be displayed in the data display indicators.

The level status register (LSR) is displayable only; data cannot be stored into this register.

To display an AKR for a given level, press the AKR key; the console AKR is displayed in the data display indicators, and the level indicators are reset. Press the desired Level key; the contents of the AKR for that level are now displayed. The level AKRs are displayable only; the console AKR can be stored into or displayed from.

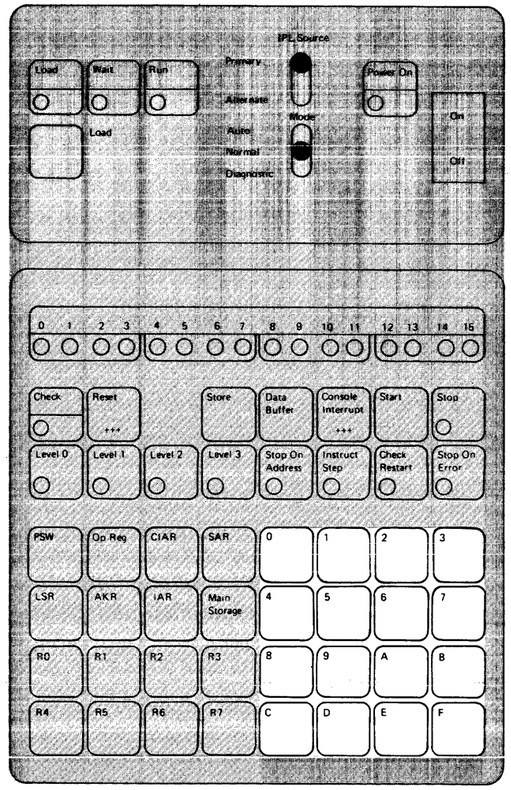
Bit 15 of the IARs can be changed from the console.

Pressing the Store key after selecting an LSR or AKR (except as noted above) results in no action taken and no audio-tone response.



Data Entry Keys

The 16 data entry keys are used to enter data into the console data buffer. This data can then be stored at any of the selected register or storage locations. Data present in the console data buffer is displayed in the data display indicators.



Displaying Main Storage Locations

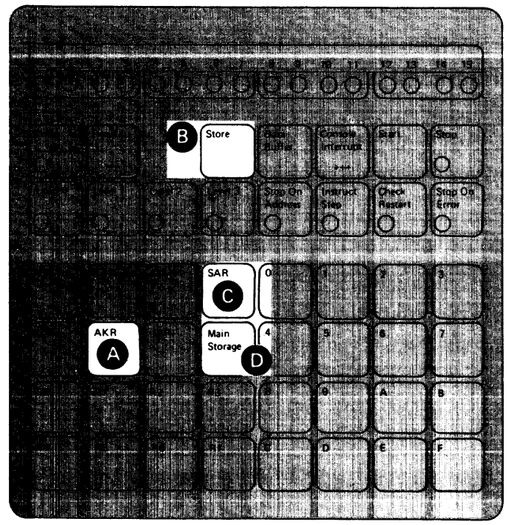
The processor must be in stop state to display main storage locations. If the translator is enabled, start at step 1; otherwise, start at step 4.

1. Press the AKR key (A). The contents of the console AKR are displayed in the data display indicators.
2. Key in one hex character (new address key). This character is displayed in bits 12–15 of the data display indicators. Bit 12 is ignored when the address is stored (the key is stored in bits 13–15).
3. Press the Store key (B) to store the new address key into the console AKR.
4. Press the SAR key (C). The contents of the SAR are displayed in the data display indicators.
5. Key in the selected address (four hex characters). This address is displayed in the data display indicators. Bit 15 of SAR cannot be set from the console.
6. Press the Store key (B). The address that is displayed is stored into the SAR.
7. Press the Main Storage key (D). The contents of the addressed storage location are displayed in the data display indicators. To display sequential main storage locations, press the Main Storage key repeatedly. The storage address is incremented by 2 each time the Main Storage key is pressed, and the addressed location is displayed.

Storing Into Main Storage

The processor must be in stop state to store into main storage. If the translator is enabled, start at step 1; otherwise, start at step 4.

1. Press the AKR key (A). The contents of the console AKR are displayed in the data display indicators.
2. Key in one hex character (new address key). This character is displayed in bits 12–15 of the data display indicators. Bit 12 is ignored when the address key is stored (the key is in bits 13–15).
3. Press the Store key (B) to store the new address key into the console AKR.
4. Press the SAR key (C). The current contents of the SAR are displayed in the data display indicators.
5. Key in the selected address (four hex characters). The address is displayed in the data display indicators.
6. Press the Store key (B). The address displayed in the data display indicators is stored into the SAR.
7. Press the Main Storage Key (D). The contents of the addressed storage location are displayed in the data display indicators.
8. Key in the data that is to be stored into main storage. This data is displayed in the data display indicators.
9. Press the store key (B). The data that is displayed is stored at the selected storage location. Each time the Store key is pressed, without pressing the Main Storage key, the SAR is incremented by 2, and the data that is displayed is stored at that location.



Displaying Registers

The processor must be in stop state.

1. Select the proper level by pressing the appropriate Level key Ⓐ.
2. Press the selected register (R0–R7) key. The contents of that register are displayed in the data display indicators Ⓑ. The contents of any register associated with the selected level can now be displayed by pressing the register key.

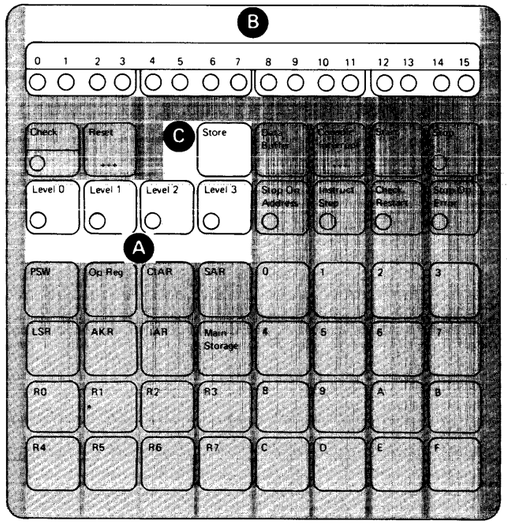
   To display the same register on each level, select the register; then press each Level key. Each level selection causes the selected register for that level to be displayed in the data display indicators.

Storing Into Registers

The processor must be in stop state.

1. Select the proper level by pressing the appropriate Level key Ⓐ.
2. Press the key for the register (R0–R7) 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.

To store into the corresponding register on another level, select the level and proceed starting with step 3; or, if the same data is to be stored, select the level and press the Store key.



Row- and Column-Line Operation

The row and column lines are used by the processor logic to identify the key being operated on the console.

The row lines are generated by the ROS card. They are constantly being sequenced, A through D, while the power is on. Pressing a key causes an associated column line to be activated; the processor logic then determines from the column line and the row counter the key that was pressed.

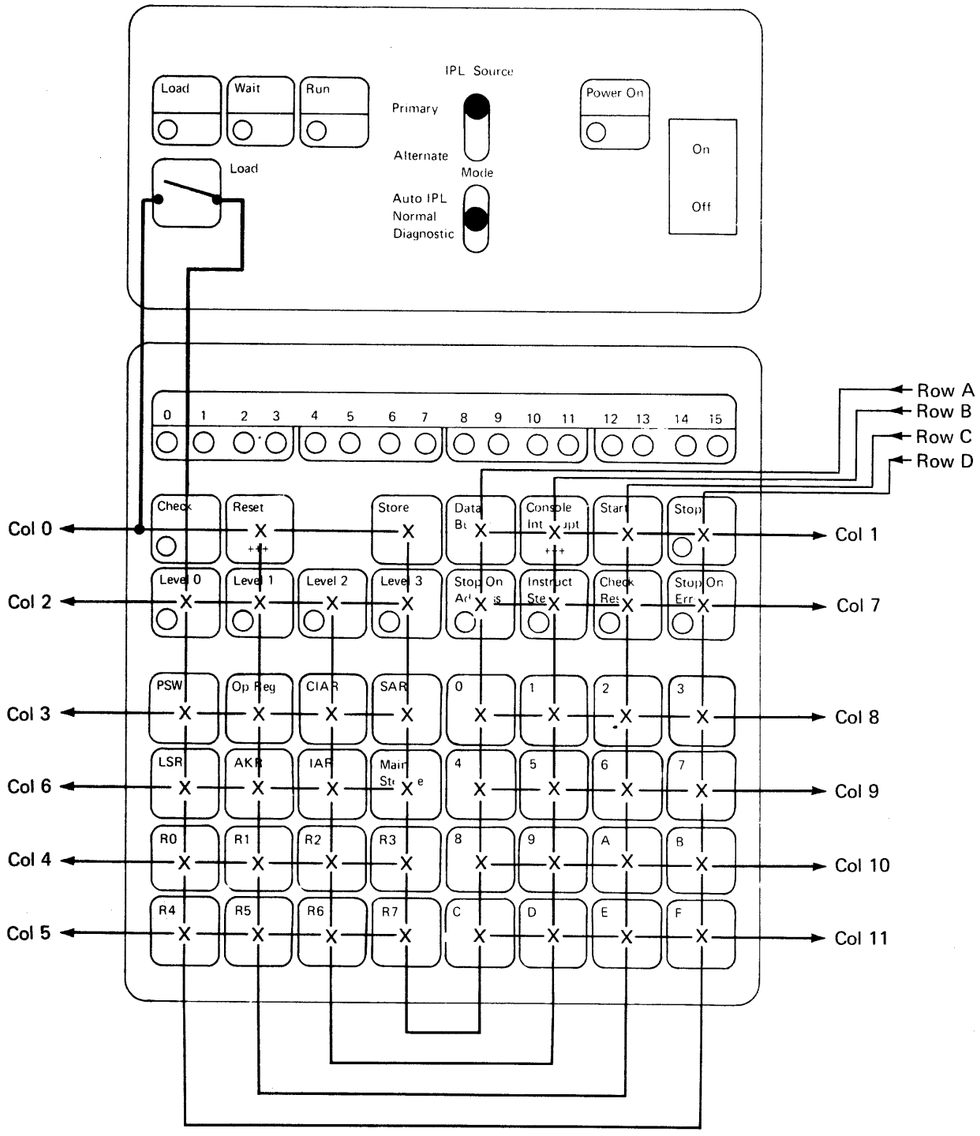
The IPL Source switch, Mode switch, and power On/Off switch have direct lines and need not be decoded.

Example

The processor is in stop state.

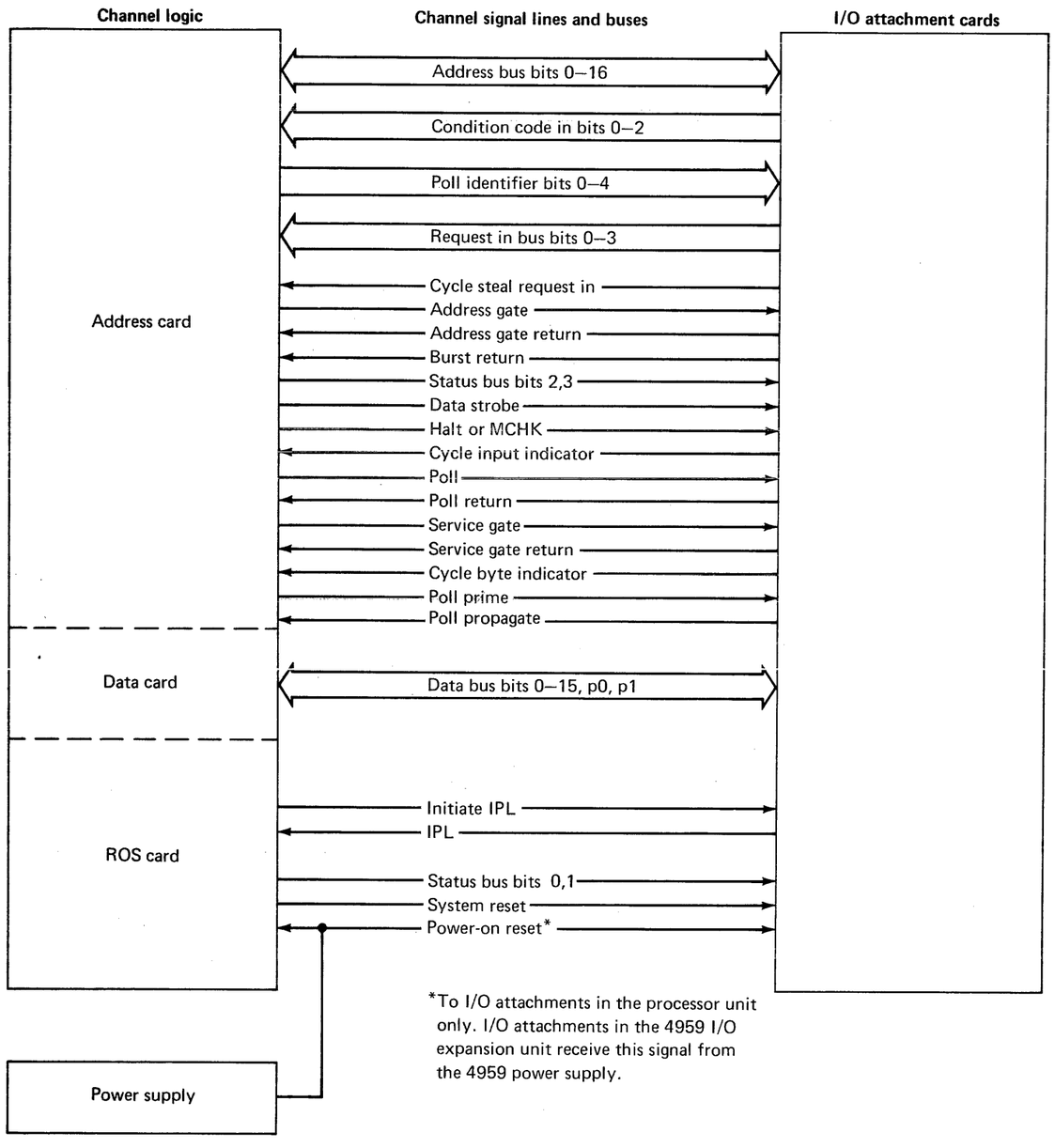
1. The level 2 key is pressed.
2. When the row counter is at B, the column 2 line is enabled; the row counter freezes at B.
3. The coincidence of the row B and column 2 lines identifies the Level 2 key to the processor microcode. The row counter is released at the end of a time-out.
4. The processor microcode determines if level 2 can be set. If so, level 2 is set and the level 2 indicator line is activated; the level 2 indicator is turned on.

Note: The data display indicators are driven from the data card. The level indicators are driven from the address card. Column lines 2–6 connect to the address card; column lines 1 and 7 connect to the data card. All other console lines connect to the ROS card.



Channel

The channel comprises 68 lines. The channel is common to all the I/O sockets in the processor card file and any I/O expansion card file attached to the processor. Not all I/O attachments use all 68 lines. All but three channel lines ('burst return,' 'halt' or 'MCHK,' and 'power on reset') are described in subsequent operational diagrams. These three lines are described in this chapter following "Initial Program Load (IPL)."

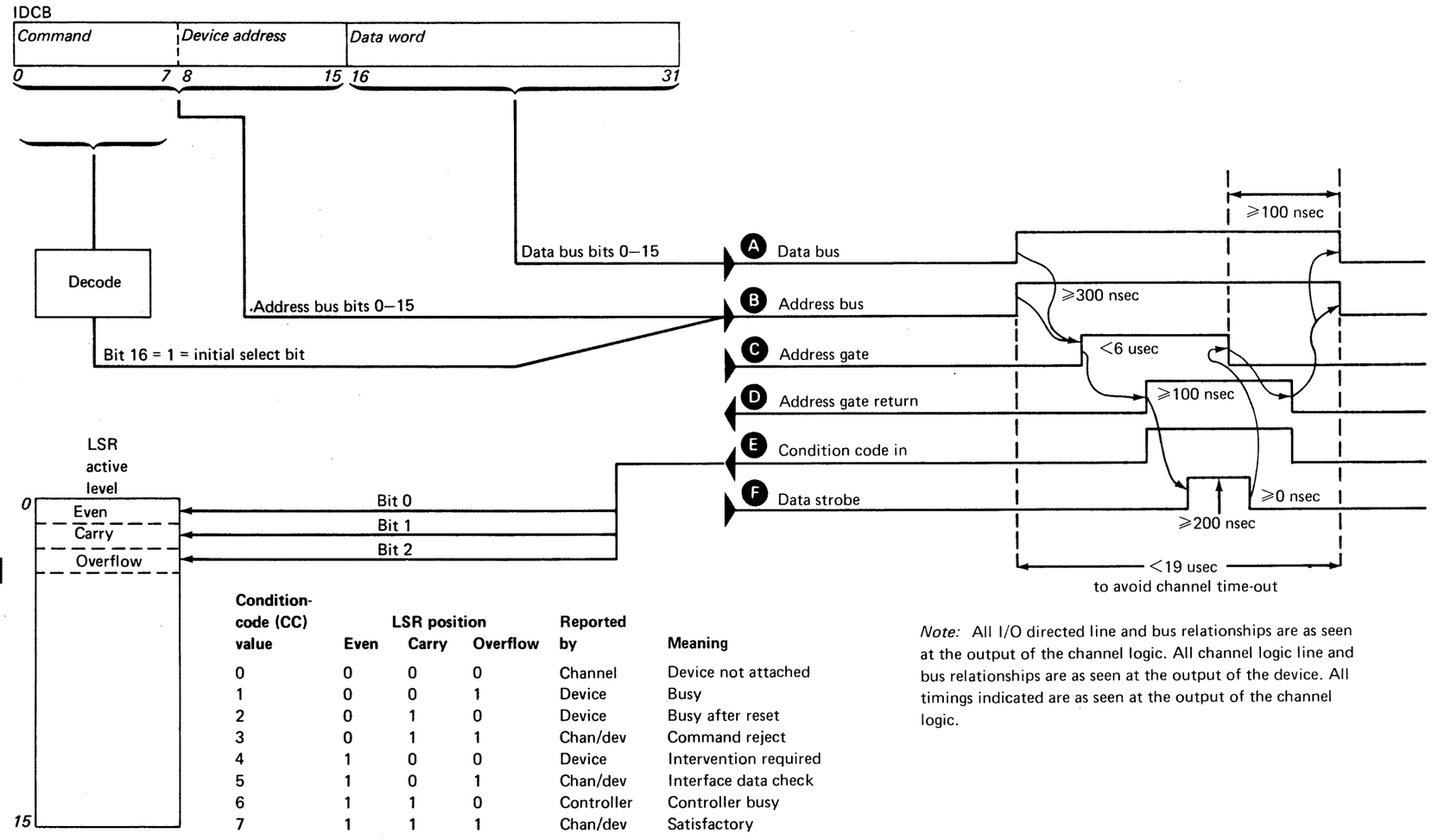


DPC Write Operation

DPC Write Operation Line Definitions

- **A** The data bus contains the data word from the second word of the IDCB. The data bus is active from 300 nanoseconds prior to the rise of 'address gate' until the fall of 'address gate return,' measured at the output of the channel logic.
- **B** Address bus bits 0–15 contain the first word of the IDCB.

  The address bus is active from 300 nanoseconds prior to the rise of 'address gate' until the fall of 'address gate return,' measured at the output of the channel logic. A compare of the device address and bits 8–15 of the address bus, with bit 16 on constitutes initial selection. At this point, the device may examine bits 0–7 for command acceptance. Upon command acceptance, the device examines the data bus for proper parity.
- **C** 'Address gate' is the outbound tag used to signal the device to respond to initial selection and begin the operation specified by the command (address bus bits 0–7).
- **D** 'Address gate return' is the tag raised by the device to signal the channel that it has received 'address gate' and has activated immediate status on the 'condition code in' bus. This tag must rise within 10 microseconds of the rise of 'address gate,' as seen at the output of the channel logic. If not, condition code 0 is returned to the channel logic and the sequence is terminated. 'Address gate' falls and the address bus is cleared.
- **E** The 'condition code in' bus is a binary-encoded three-bit field. The I/O device passes status to the channel logic on this bus during the 'address gate return' tag time. The condition code bits are placed into the Even, Carry, and Overflow indicators of the current level LSR.
- **F** 'Data strobe' is an outbound tag, and it may be used by the device to register data being sent to the device. 'Data strobe' is at least 200 nanoseconds in duration, and falls with the fall of 'address gate.'



Condition-code (CC) value	LSR position Even	Carry	Overflow	Reported by	Meaning
0	0	0	0	Channel	Device not attached
1	0	0	1	Device	Busy
2	0	1	0	Device	Busy after reset
3	0	1	1	Chan/dev	Command reject
4	1	0	0	Device	Intervention required
5	1	0	1	Chan/dev	Interface data check
6	1	1	0	Controller	Controller busy
7	1	1	1	Chan/dev	Satisfactory

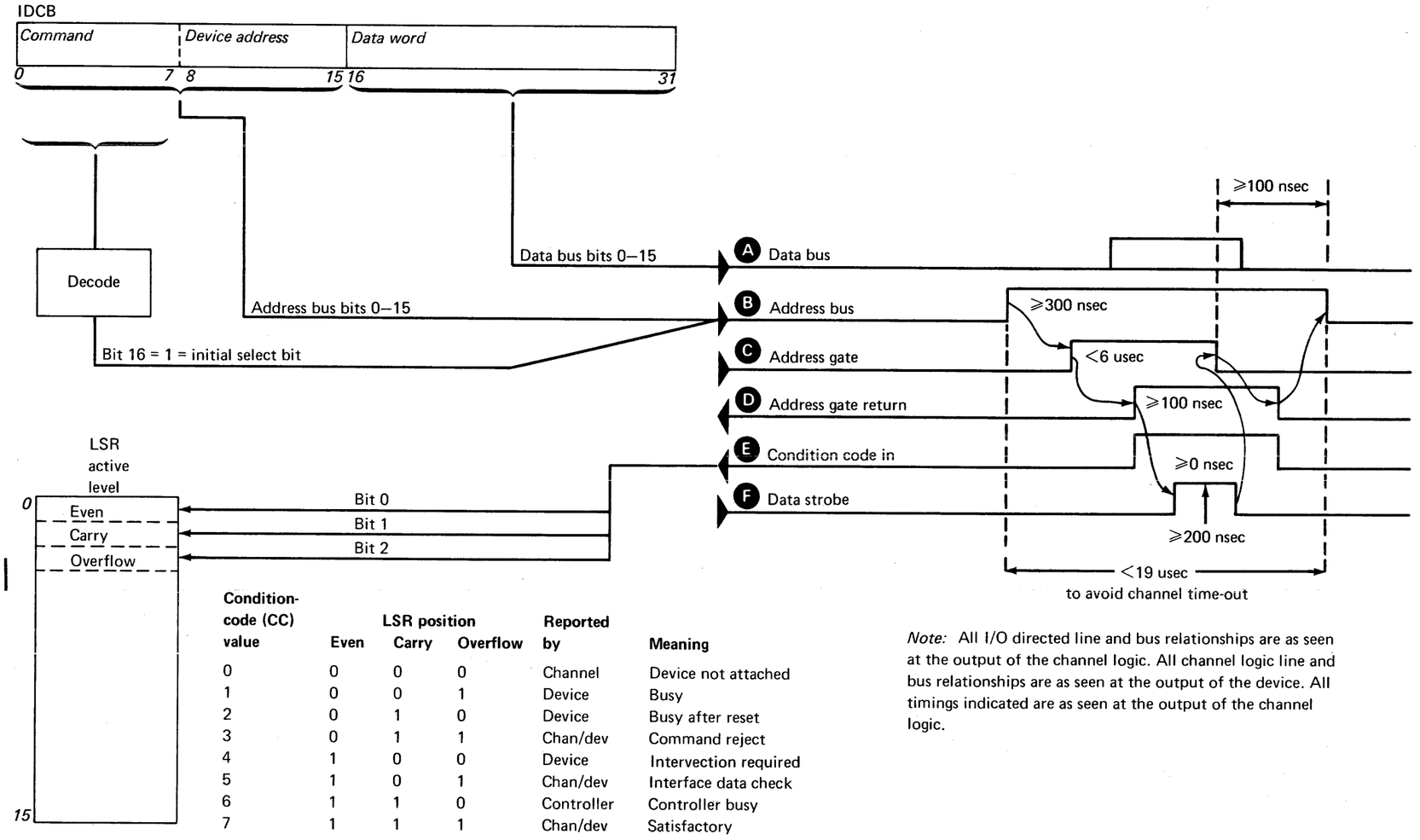
Note: All I/O directed line and bus relationships are as seen at the output of the channel logic. All channel logic line and bus relationships are as seen at the output of the device. All timings indicated are as seen at the output of the channel logic.

DPC Read Operation

DPC Read Operation Line Definitions

- Ⓐ The data bus contains the data word read from the device. The contents of the data bus are placed into the second word of the IDCB that was addressed by the Operate I/O instruction.
- Ⓑ Address bus bits 0–15 contain the first word of the IDCB.

  The address bus is active from 300 nanoseconds prior to the rise of 'address gate' until the fall of 'address gate return,' measured at the output of the channel logic. A compare of the device address and bits 8–15 of the address bus, with bit 16 on constitutes initial selection. At this point, the device may examine bits 0–7 for command acceptance.
- Ⓒ 'Address gate' is the outbound tag used to signal the device to respond to initial selection and begin the operation specified by the command (address bus bits 0–7).
- Ⓓ 'Address gate return' is the tag raised by the device to signal the channel that it has received 'address gate' and has activated immediate status on the 'condition code in' bus. This tag must rise within 10 microseconds of the rise of 'address gate,' as seen at the output of the channel logic. If not, condition code 0 is returned to the channel logic and the sequence is terminated. 'Address gate' falls and the address bus is cleared.
- Ⓔ The 'condition code in' bus is a binary-encoded three-bit field. The I/O device passes status to the channel logic on this bus during the 'address gate return' tag time. The condition code bits are placed into the Even, Carry, and Overflow indicators of the current level LSR.
- Ⓕ 'Data strobe' is an outbound tag, and it may be used by the device to register data being sent to the device. 'Data strobe' is at least 200 nanoseconds in duration, and falls with the fall of 'address gate.'



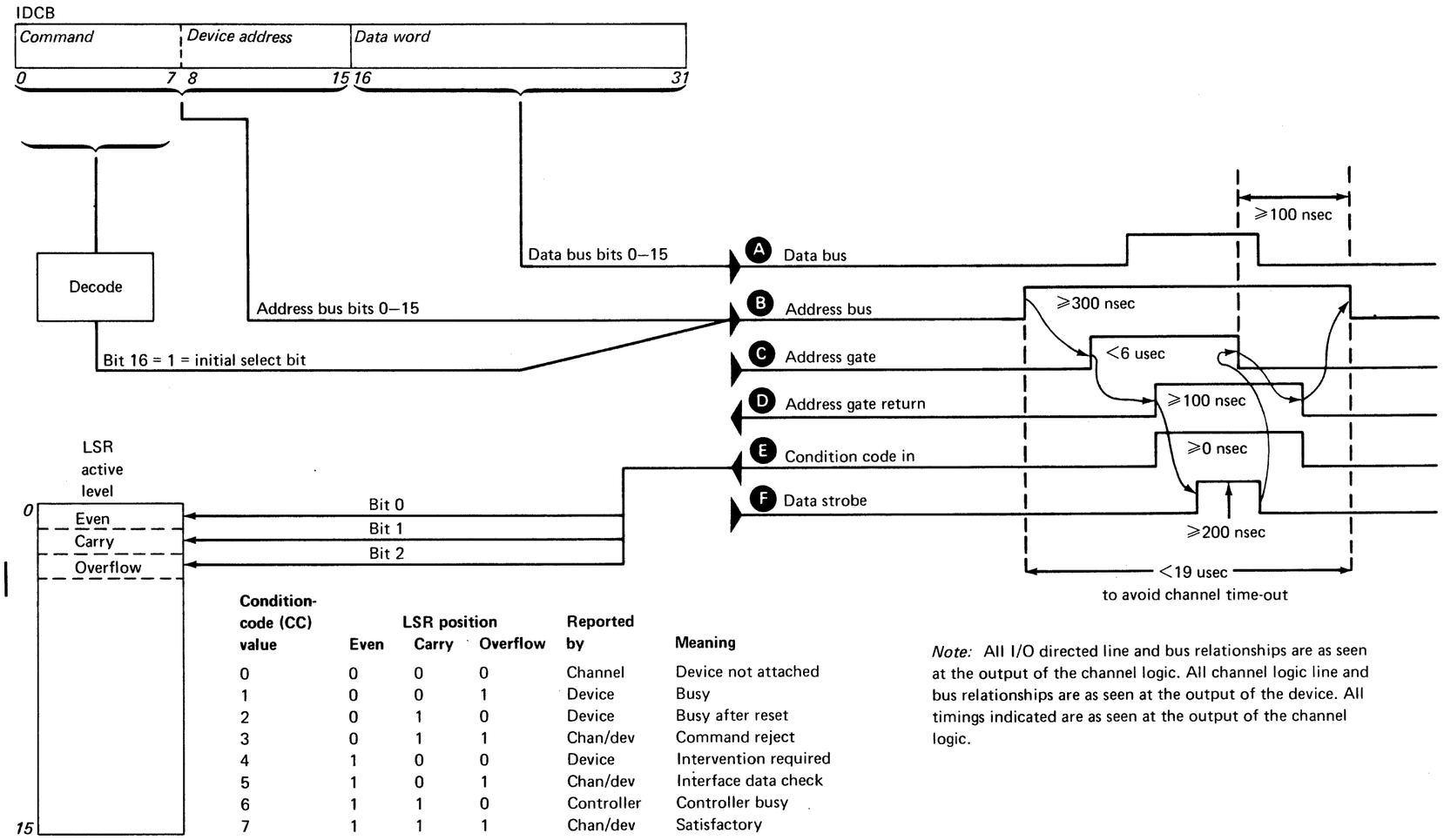
Condition-code (CC) value	LSR position Even	Carry	Overflow	Reported by	Meaning
0	0	0	0	Channel	Device not attached
1	0	0	1	Device	Busy
2	0	1	0	Device	Busy after reset
3	0	1	1	Chan/dev	Command reject
4	1	0	0	Device	Intervection required
5	1	0	1	Chan/dev	Interface data check
6	1	1	0	Controller	Controller busy
7	1	1	1	Chan/dev	Satisfactory

Note: All I/O directed line and bus relationships are as seen at the output of the channel logic. All channel logic line and bus relationships are as seen at the output of the device. All timings indicated are as seen at the output of the channel logic.

Start Command (Initiate Cycle-Steal Operation)

Start Command Line Definitions

- **A** The data bus contains the data word from the second word of the IDCB. This word is the address of the DCB to be fetched by the device. The data bus is active from 300 nanoseconds prior to the rise of 'address gate' until the fall of 'address gate return,' measured at the output of the channel logic.
- **B** Address bus bits 0–15 contain the first word of the IDCB. Bit 16 signals the device that the address bus contains the device address and the command field from the IDCB. The address bus is active from 300 nanoseconds prior to the rise of 'address gate' until the fall of 'address gate return,' measured at the output of the channel logic. A compare of the device address and bits 8–15 of the address bus, with bit 16 on constitutes initial selection. At this point, the device may examine bits 0–7 for command acceptance. If the command is a write, the device examines the data bus for proper parity.
- **C** 'Address gate' is the outbound tag used to signal the device to respond to initial selection and begin the operation specified by the command (address bus bits 0–7).
- **D** 'Address gate return' is the tag raised by the device to signal the channel that it has received address gate and has activated immediate status on the 'condition code in' bus. This tag must rise within 10 microseconds of the rise of 'address gate,' as seen at the output of the channel logic. If not, condition code 0 is returned to the channel and the sequence is terminated. Address gate falls and the address bus is cleared.
- **E** The 'condition code in' bus is a binary-encoded three-bit field. The I/O device passes status to the channel on this bus during the 'address gate return' tag time. The condition code bits are placed into the Even, Carry, and Overflow indicators of the current level LSR.
- **F** 'Data strobe' is an outbound tag, and it may be used by the device to register data being sent to the device. 'Data strobe' is at least 200 nanoseconds in duration, and falls with the fall of 'address gate.'



Condition-code (CC) value	LSR position Even	Carry	Overflow	Reported by	Meaning
0	0	0	0	Channel	Device not attached
1	0	0	1	Device	Busy
2	0	1	0	Device	Busy after reset
3	0	1	1	Chan/dev	Command reject
4	1	0	0	Device	Intervention required
5	1	0	1	Chan/dev	Interface data check
6	1	1	0	Controller	Controller busy
7	1	1	1	Chan/dev	Satisfactory

Note: All I/O directed line and bus relationships are as seen at the output of the channel logic. All channel logic line and bus relationships are as seen at the output of the device. All timings indicated are as seen at the output of the channel logic.

Poll Capture

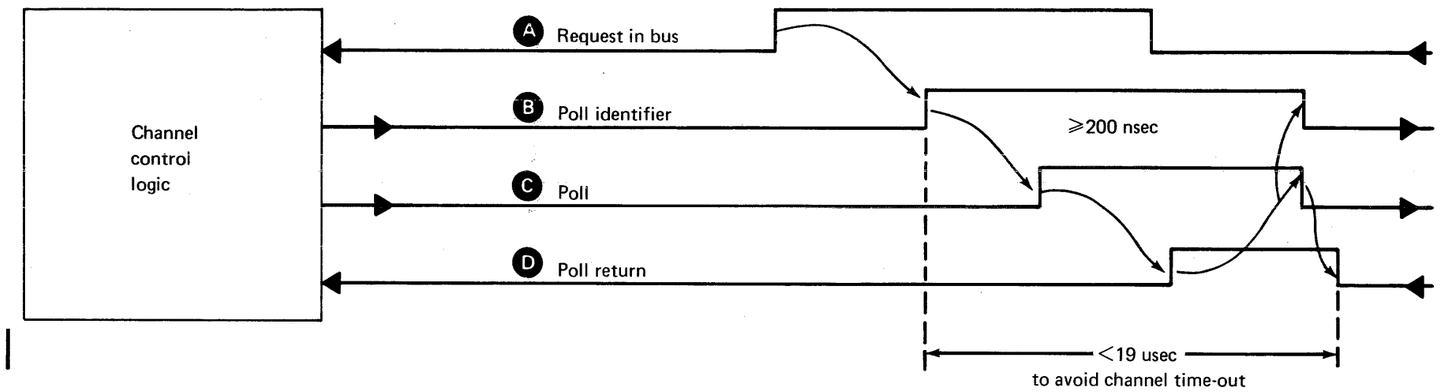
Ⓐ The 4955 processor channel recognizes bits 0–3 of the 'request in' bus and the 'cycle steal request in' line. Bits 0–3 indicate the interrupt level of the device, and the 'cycle steal request in' line indicates a cycle-steal request. This bus remains active until the device captures the poll.

Ⓔ The 'poll identifier' is a five bit bus used to indicate the type of poll being propagated over the channel lines. The poll identifier bit combination is determined by the type of request-in. A poll is started only when a request is made. The significance of the five bits is as follows:

Bits 0 1 2 3 4	Significance
0 0 0 0 0	Poll interrupt level 0
0 0 0 0 1	Poll interrupt level 1
0 0 0 1 0	Poll interrupt level 2
0 0 0 1 1	Poll interrupt level 3
0 0 1 0 0	Not used
⌇ ⌇ ⌇ ⌇ ⌇	
0 1 1 1 1	
1 0 0 0 0	Quiescent value
1 0 0 0 1	Not used
⌇ ⌇ ⌇ ⌇ ⌇	
1 1 1 1 0	
1 X X 1 1	'Poll' for cycle steal

Ⓒ The 'poll' tag is generated by the channel to resolve contention between I/O devices requesting interrupts on the same level and cycle-stealing requests. Each I/O device receives the 'poll' tag and drives it to the next device by sending 'poll propagate' if the device does not capture the poll. If the device captures the 'poll,' it responds with 'poll return' or 'burst return.' In this case, the 'poll' tag is not propagated to any other device.

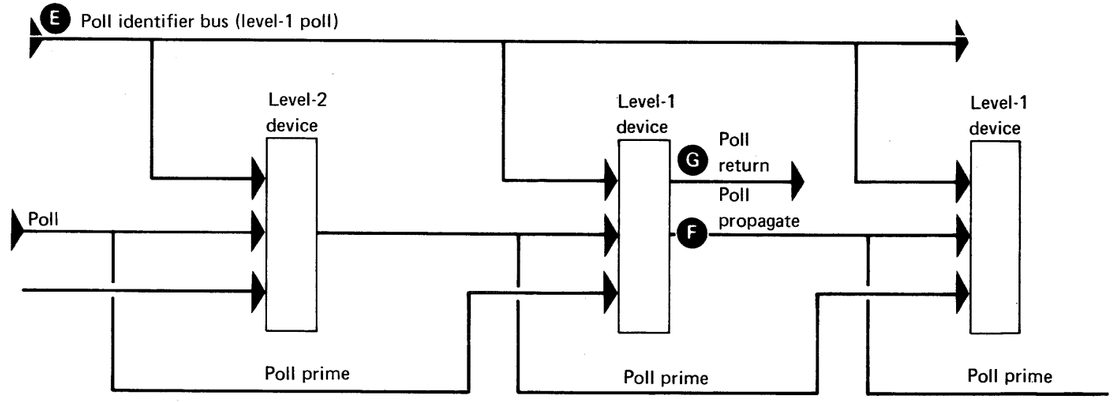
Ⓓ 'Poll return' is raised by the I/O device to signal the channel that a 'poll' capture has taken place. If a 'poll' is not captured by any of the devices on the channel within approximately 19 microseconds after the processor initiated the poll, a machine check is set with PSW bits 11 and 12 on.



Example of Poll Capture

For the following example assume that all three devices have interrupt pending:

The 'poll identifier' bus Ⓔ specifies a 'poll' for level-1 devices. The first level-1 device captures the 'poll,' inhibits 'poll propagate' Ⓕ and generates a 'poll return' tag Ⓖ to the channel. The channel responds with 'service gate.' The device then responds to that with 'service gate return' and starts the data transfer.



Poll Propagate Wiring

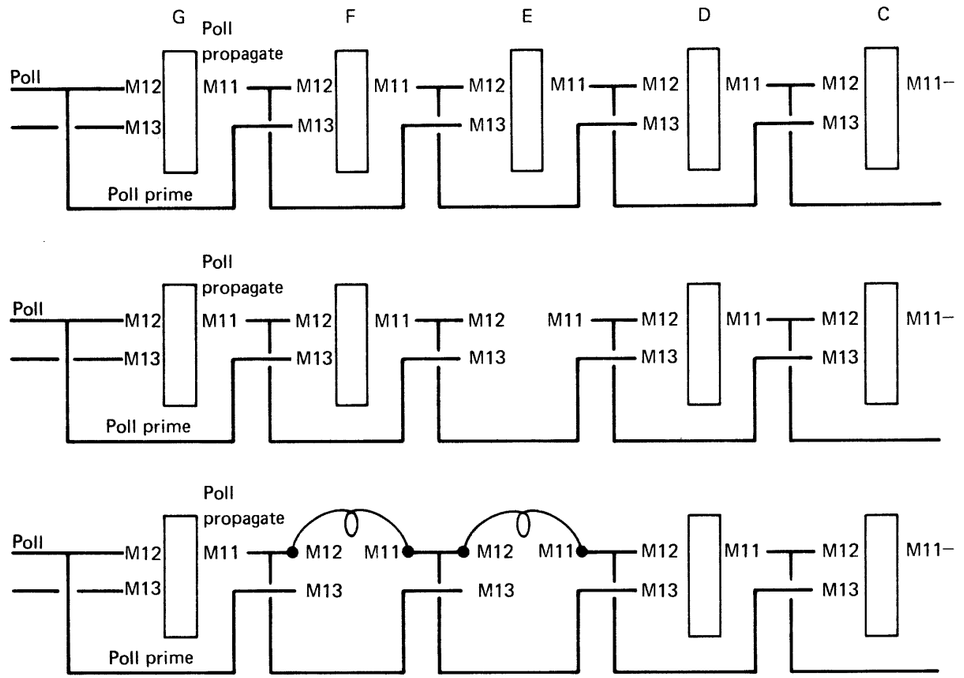
The 'poll' tag is a serially propagated line generated by the channel to resolve contention between I/O devices attempting to interrupt on the same level. This line also allows cycle-stealing devices to capture a poll that was generated as a result of a cycle-steal request.

The I/O card sockets are wired so that they allow the plugging of every other card socket without the use of manually installed jumpers.

When there are two or more sequential card sockets empty between plugged feature cards, jumpers must be installed to ensure proper channel operation. The jumper for each vacant socket is installed on the pin side of the back panel, and its intra socket connection is between pins M11 and M12.

If the Q-card socket of the 4959 I/O expansion unit is vacant, a jumper must be installed between pins M11 and M12 of the Q-socket. Also, all vacant I/O expansion unit sockets following the vacant Q-card socket must be jumpered according to the preceding description.

Socket locations



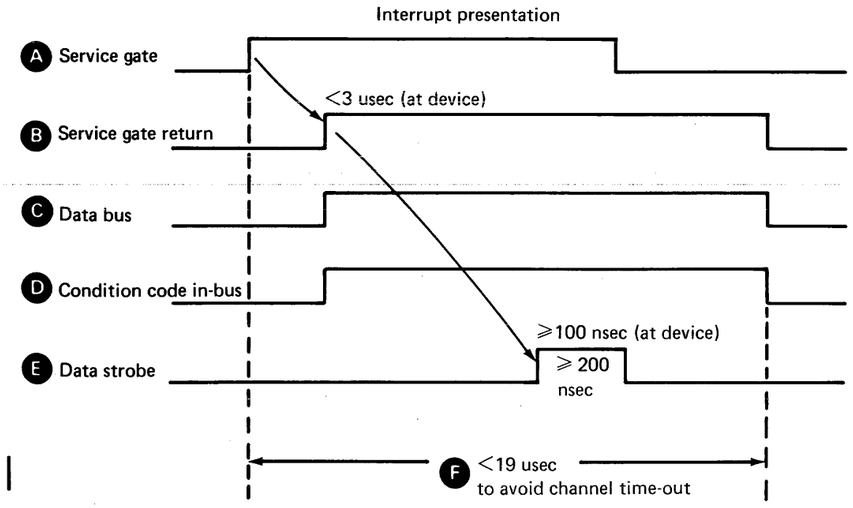
Wiring side of board

M11 = Poll propagate
M12 = Poll
M13 = Poll prime

Interrupt Presentation

A poll interrupt level is issued by the channel only after an interrupt request is made by an I/O device. The device recognizes its interrupt level in the poll identifier bit combination (bits 0–4) and raises 'poll return' to signal the channel that a 'poll' capture has taken place.

- (A) 'Service gate' is activated by the channel when it receives 'poll return' from a device. The device detecting the first 'service gate' after a 'poll' capture is the selected device for the service sequence.
- (B) The device raises 'service gate return' in response to 'service gate' to indicate to the channel that it has placed the necessary data on the data bus and 'condition code in' bus.
- (C) The data bus contains the Interrupt ID word, which consists of the interrupt information byte (IIB) in bits 0–7, and the device address in bits 8–15. This word is set into register 7 of the interrupted-to level.
- The 'condition code in' bus contains the interrupt condition code in bits 0–2, which is set into the Even, Carry, and Overflow indicators for the interrupted-to level.
- (E) 'Data strobe' is an outbound tag, and it may be used by the device to register data.
- The total duration of the interrupt service sequence is timed by the channel for error-detection purposes. The total duration is measured from the activation of 'service gate' to the deactivation of 'service gate return.' If, within the time-out period, 'service gate return' does not become active, or 'service gate' or 'service gate return' does not deactivate after being active, then a machine check occurs with bits 11 and 12 set in the PSW.



Cycle-Steal Output Operation—Word Transfer

Prior to this output operation, the device had sent a cycle steal request to the channel. The channel responded with the polling sequence, and this device captured that poll.

Ⓐ 'Service gate' is raised by the channel to indicate to the device that captured the poll that data transfers may begin.

Ⓑ When the device detects its 'service gate,' it sends 'service gate return' to the channel to indicate that it has placed the necessary data and control information on the channel lines. Any data provided by the device for the transfer is activated no later than the rise of this tag line. This tag line may fall no sooner than the fall of 'service gate' and 'data strobe,' as seen at the output of the I/O device.

Ⓒ The address bus contains the main storage address that is used to fetch the data word to be transferred to the device. The contents of the address bus are gated to the cycle-steal SAR in the address card. A storage fetch takes place and the word fetched is placed into the cycle-steal SDR in the data card; the word is then gated onto the data bus.

Ⓓ The data bus contains the word read from main storage.

Ⓔ The 'condition code in' bus contains the address key to be used during the main storage access. 'Condition code in' bus bits 0, 1, and 2 correspond to bits 5, 6, and 7 of the DCB control word. These bits are the cycle-steal address key. This bus is activated with the rise of 'service gate return' and is maintained until the fall of 'service gate.'

Ⓕ 'Data strobe' is an outbound tag, and it may be used by the device to register data being sent to the device. 'Data strobe' is 200 nanoseconds in duration and falls with the fall of 'address gate.'

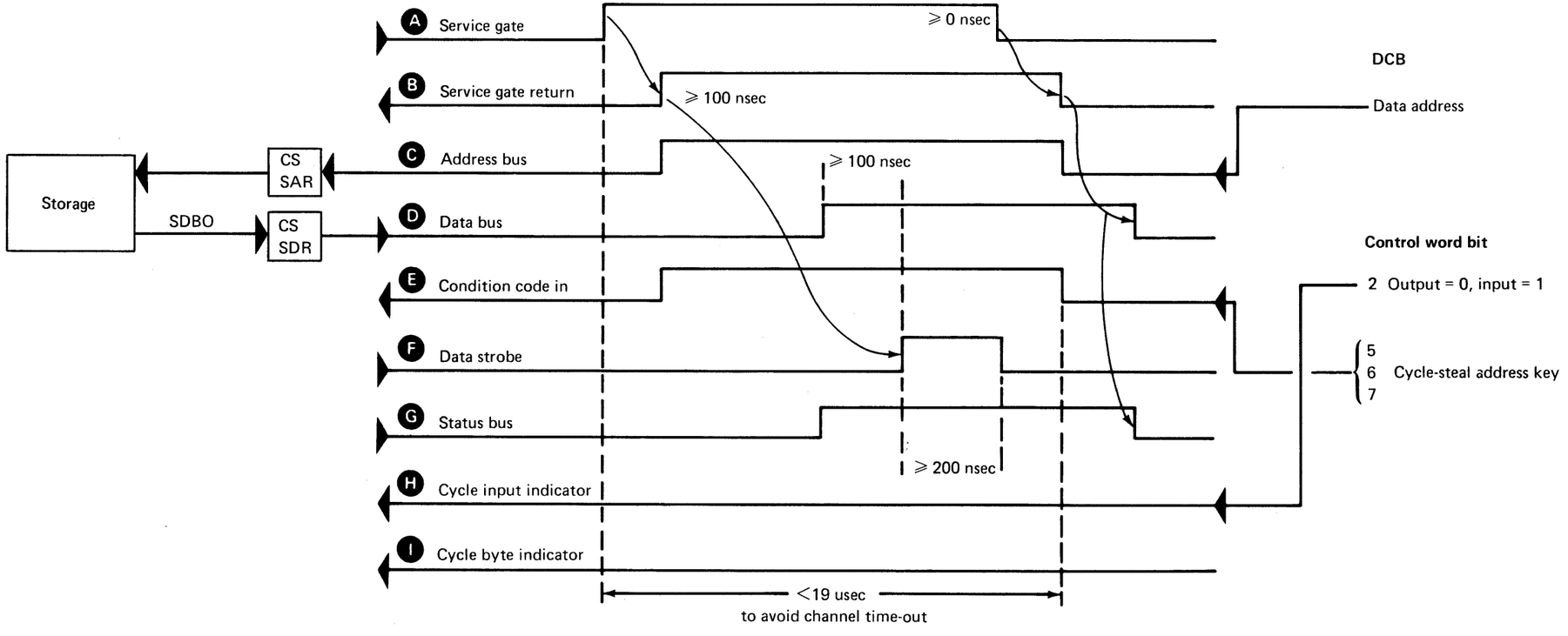
Ⓖ The status bus is used by the channel to signal the device if an error is detected. Status bus bits have the following meanings:

Bit 0	Storage data check
Bit 1	Invalid storage address
Bit 2	Protect check
Bit 3	Interface data check

If this bus is activated, the device retains the information for presentation in the interrupt status byte at interrupt time. The cycle-steal operation is terminated and the device presents end interrupt. If the device has already raised 'cycle-steal request in' for the next transfer, or if it is in burst transfer mode, it must complete one more "servicing" over the channel lines. This servicing is a dummy cycle where no device-held parameters are updated or any status bits are accumulated.

Ⓗ 'Cycle input indicator' tag equal to 0 indicates to the channel that the operation is an output from storage.

Ⓘ The 'cycle byte indicator' tag equal to 0 indicates to the channel that a word transfer is to take place.



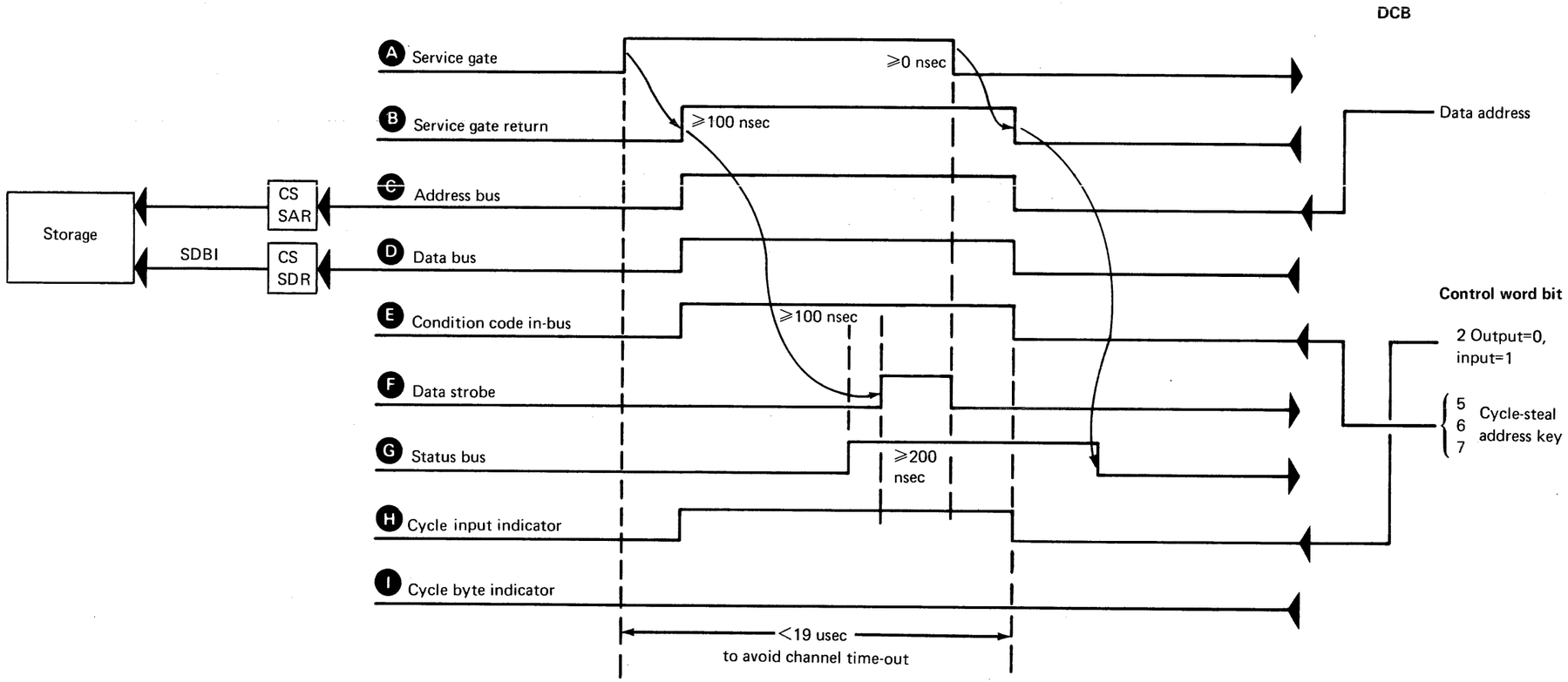
Cycle-Steal Input Operation—Word Transfer

Prior to this operation, the device had sent a cycle-steal request to the channel. The channel responded with the polling sequence, and this device captured the poll. The device then fetched the DCB from main storage, using cycle steal. At this point, the DCB resides in a space in the device attachment.

- (A) 'Service gate' is raised by the channel to indicate to the device that captured the poll that data transfers may begin.
- (B) When the device detects 'service gate' it sends its 'service gate return' to the channel to indicate that it has placed the necessary data and control information on the channel lines. Any data provided by the device for the transfer, is activated no later than the rise of this tag line. This tag line may fall no sooner than the fall of 'service gate' and 'data strobe,' as seen at the output of the I/O device.
- (C) The address bus contains the main storage address that points to the location where the word on the Data Bus is to be stored.
- (D) The data bus contains the word from the device that is to be placed at the location in main storage specified by the address on the address bus.
- (E) The 'condition code in' bus contains the address key to be used during the main storage access. Condition code bits 0, 1, and 2 correspond to bits 5, 6, and 7 of the DCB control word. These bits are the cycle-steal address key. This bus is activated with the rise of 'service gate return' and is maintained until the fall of 'service gate.'
- (F) 'Data strobe' is an outbound tag, and it may be used by the device to register data being sent to the device. 'Data strobe' is at least 200 nanoseconds in duration, and falls with the fall of 'address gate.'
- (G) The status bus is used by the channel to signal the device if an error is detected. Status bus bits have the following meanings:

Bit 0	Storage data check
Bit 1	Invalid storage address
Bit 2	Protect check
Bit 3	Interface data check

  If this bus is activated, the device retains the information for presentation in the interrupt status byte at interrupt time. The cycle-steal operation is terminated and the device presents end interrupt. If the device has already raised 'cycle steal request in' for the next transfer, or if it is in burst transfer mode, it must complete one more "servicing" over the channel lines. This servicing is a dummy cycle where no device-held parameters are updated or any status bits are accumulated.
- (H) 'Cycle input indicator' tag equal to 1 indicates to the channel that the operation is an input to storage.
- (I) 'Cycle byte indicator' tag equal to 0 indicates to the channel that a word transfer is to take place.



Initial Program Load (IPL)

Ⓐ The 'initiate IPL' tag is raised by the channel to signal the IPL device that the processor requires a program load. Once this tag is raised, it remains active until the device responds with the 'IPL' tag.

Ⓑ 'System reset' is a tag from the channel to all I/O devices. The I/O devices clear any status, states, requests, registers, and channel control logic, with the following exceptions:

- Residual address
- Output sensor points
- Timer values
- Logic not addressed by the software

Ⓒ 'IPL' is either:

- A tag from the device in response to 'initiate IPL,' or
- A tag from a device connected to a host system to signal the channel that it is initiating the IPL

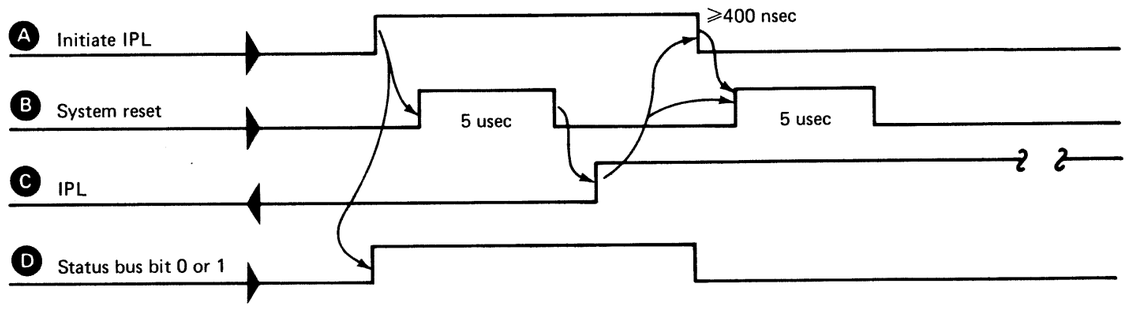
In either case, 'IPL' is held active until the IPL is completed. The device raises the tag and, following a system reset, begins the IPL, using the cycle-steal mechanism. Completion of the IPL is signaled by the device dropping the 'IPL' tag.

Ⓓ 'Status bus bit 0 or 1'—The channel raises bit 0 to signal the primary load device that it is selected to perform the IPL. The channel raises bit 1 to signal the alternate load device that it has been selected to perform the IPL.

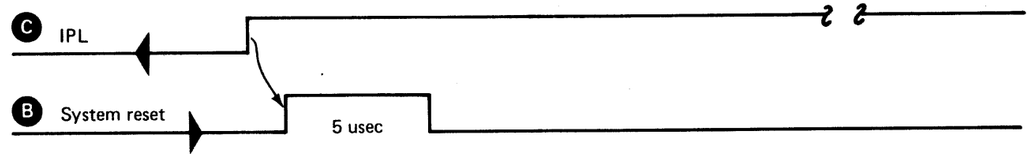
If the channel detects an error during IPL, the appropriate status bus bit is made active and the device terminates IPL data transfer, but leaves the 'IPL' tag active.

Note: If the Mode switch on the basic console is in the Auto IPL position, and a power-on sequence is completed, the IPL sequence is the same as the processor-initiated IPL, except that bit 13 of the PSW (auto IPL) is set.

Processor-initiated IPL



Host-initiated IPL



Burst Return

This is an inbound tag sent by an I/O device to signal the channel that a poll capture (cycle steal only) has taken place and that a burst transfer is required. Once activated, the next leading edge of a 'service gate' tag signals the beginning of the burst transfer. The entire channel is now dedicated to the I/O device. 'Service gate'/'service gate return' "handshaking" between the channel and the device continues until the 'burst return' tag is dropped. An I/O device deactivates the 'burst return' tag at the rise of the service gate tag for the last transfer. Burst mode is used only if it is specified in the DCB control word; this is under program control.

Halt or MCHK

This is a tag line generated by the channel logic and sent to all I/O devices attached to the channel. The tag means that either:

- A Halt I/O command has been issued by the program, or
- A machine-check class interrupt (excluding a storage-parity check) has occurred

When the tag is detected by an I/O device, the device must disable selection, block poll propagation, clear any status, states, requests, interface control logic, and registers, with the following exceptions:

- Residual address
- Prepare level and I-bit
- Output sensor points
- Timer values
- Those registers not addressable by the software

Power-On Reset

This is an outbound control line generated by the power supply to all system components. This line is activated on all power-on/off sequences. When the line is active, all system components are held in a system reset state. The residual address, output sensor points, and timer values are also reset.

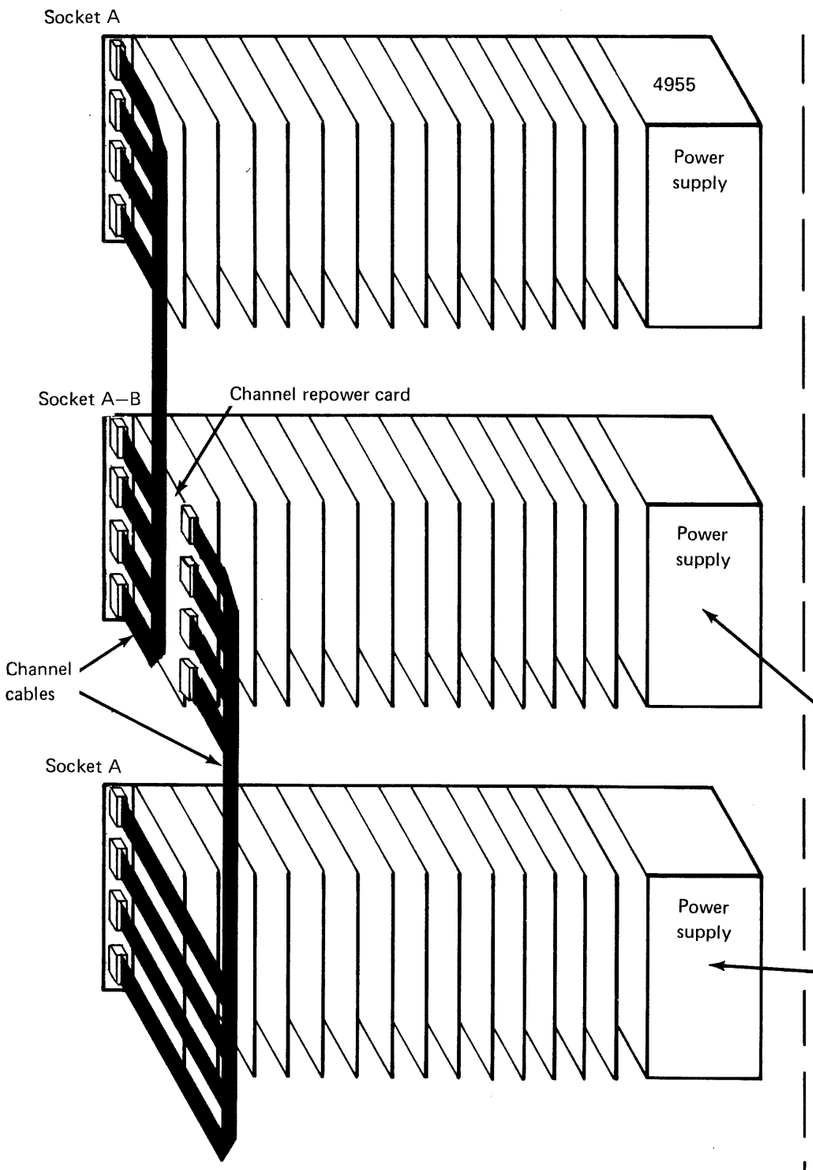
Power-on resets are not electrically connected between units.

Channel Repower Feature

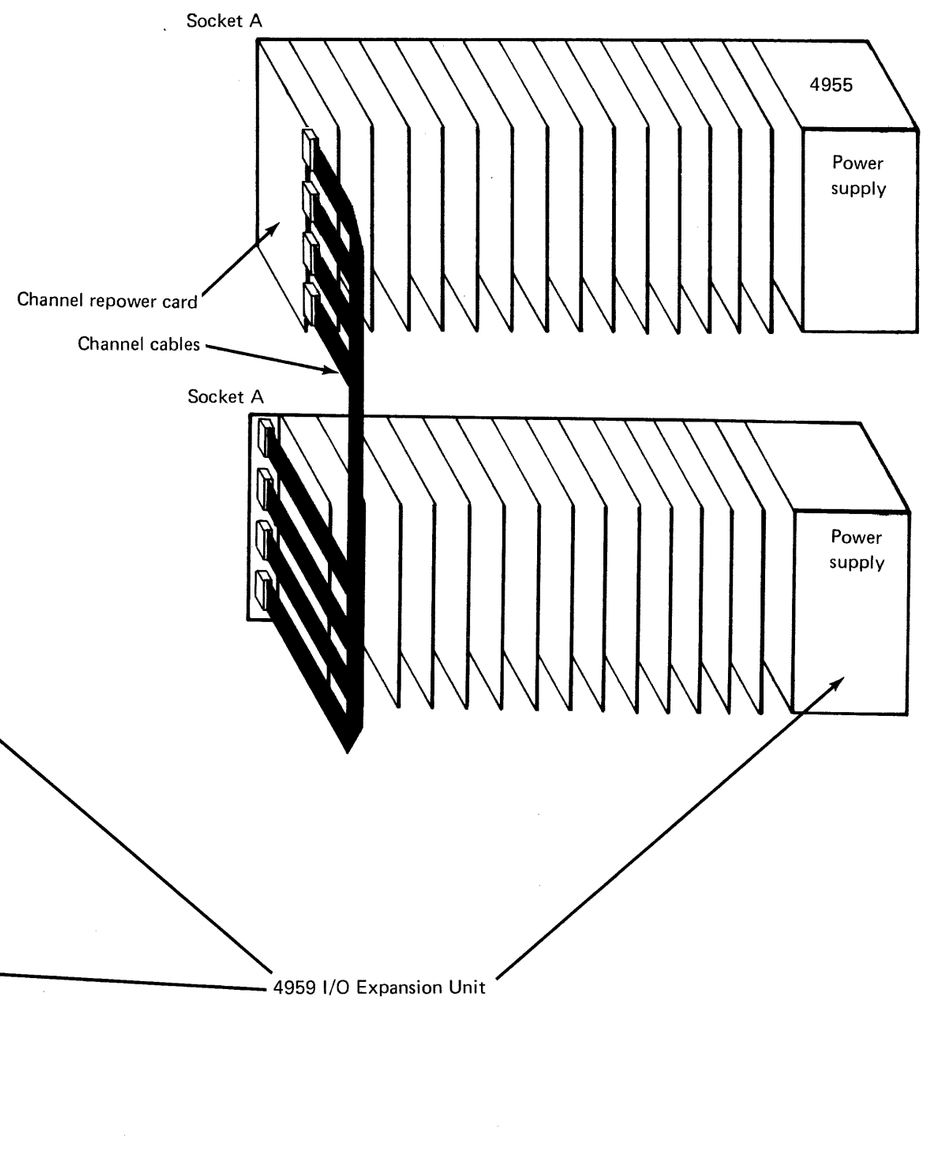
- The channel repower feature is to repower the channel signal lines and drives them to the next unit.
- This feature also provides power isolation between the units connected through this feature card.
- This feature card plugs into socket A of the 4955 Processor, or socket B of the 4959 I/O Expansion Unit.

The 4955 channel logic is capable of driving the channel lines to 21 I/O attachment cards over a distance of 5 feet. If the I/O configuration is going to exceed 21 I/O attachments, the channel repower feature is required. The channel repower card receives the channel lines on the base pins of the card. The channel lines are then powered and driven to the top edge connectors. The channel cable, which attaches to the edge connectors carries the I/O channel lines to the next unit.

The channel repower feature may also be used to provide power isolation between the processor and the I/O expansion unit. With the channel repower feature card plugged in socket A of the processor and the channel cable installed in the I/O expansion unit, the I/O expansion unit could be powered down without affecting the channel operation in the processor unit. In this case, the only indication to the processor that the 4959 is not operational is if an I/O device that is plugged into the 4959 is addressed. A condition code of 0 is returned when the Operate I/O instruction is executed.



Channel repower card used as repower feature



Channel repower card used as power isolation

300-Watt Power Supply

- The 300-watt power supply converts ac input voltages to dc voltages and distributes the dc voltages to the voltage planes on the processor board.
- The 300-watt power supply is used in the 4955 processor Models A, B, C, and D, and in the 4959 I/O expansion unit.

The major functional units of the 300-watt power supply are:

1. Sequence and control card
2. +8.5 Vdc, −5 Vdc regulator card
3. ±12 Vdc power supply card (installed with optional Communications Feature)
4. Power chassis

There are three indicators located on the top edge of the sequencer card. These indicators which are visible when the front of the power supply is exposed, are:

Load failure
Thermal shutdown
Power-on reset

If a power problem exists, these indicators are an aid to analyzing the type of failure that caused the power shutdown.

Load Failure. Indicates that an overvoltage, overcurrent, or undervoltage caused a power shutdown.

Thermal Shutdown. Indicates that either the power supply temperature exceeded 80°C or that the card area thermal sensor opened, caused the power shutdown.

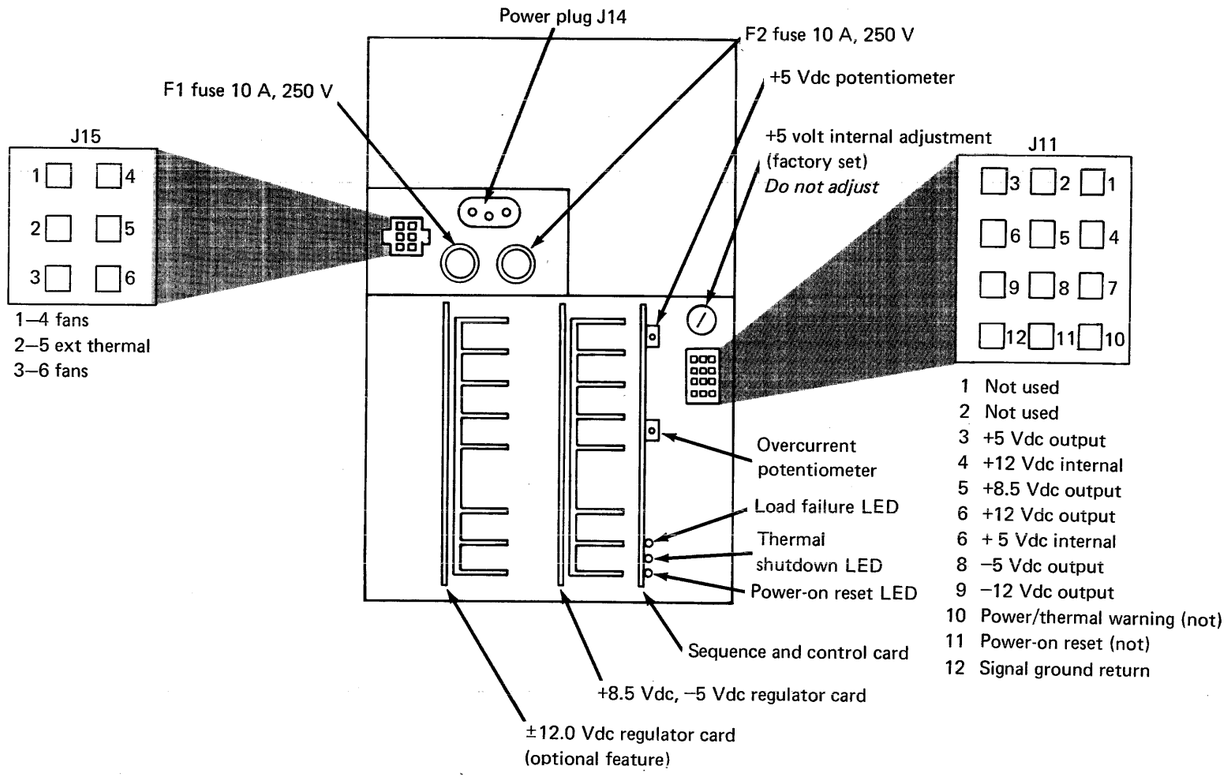
Power-On Reset. Indicates that the supply generated this signal before the power supply was sequenced off.

Input Voltage

The input voltage is 47 to 63 Hz, single-phase 100/110/115/123.5/ or 200/208/230, or 235, Vac ±10%.

Output Voltages

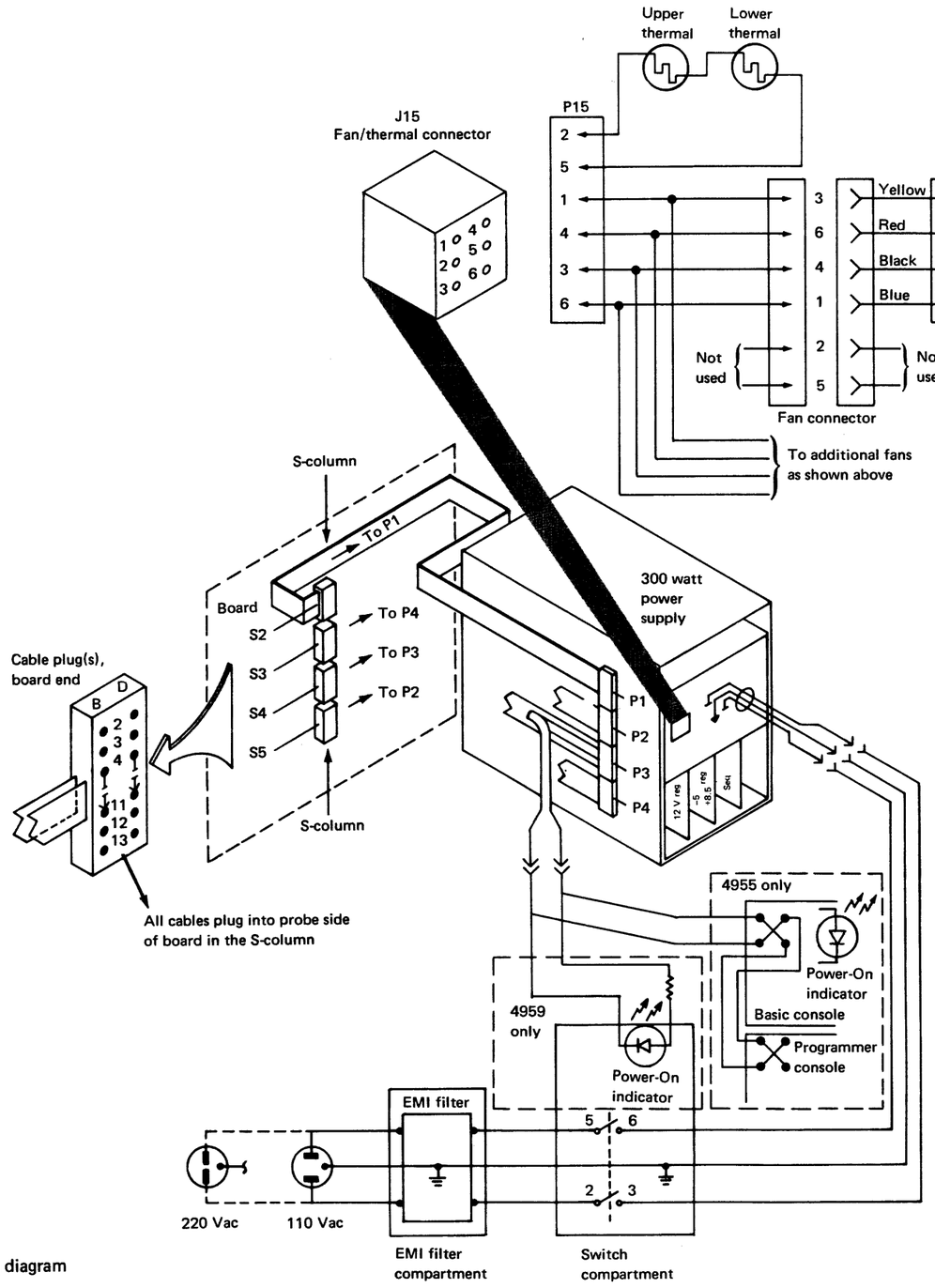
Output voltages are +5 Vdc, +8.5 Vdc, −5 Vdc, +12 Vdc, and −12 Vdc. The +5 Vdc is the only dc voltage that is adjustable. The +12 Vdc and −12 Vdc are the outputs of the ±12 Vdc power supply card, which is required for the optional Communications feature.
Test points for checking voltages and signals are on socket J11, which is located on the front of the supply.



Location diagram, front view

Cabling for DC Output

DC voltages are distributed to the board by flat-wire bus cables to the S-column on the board. Voltage distribution to the cards is achieved through the board internal planes, with each four-wide socket containing ±5 Vdc, +8.5 Vdc, ±12 Vdc, and ground. For specific pin locations and voltage distribution, refer to the Machine Logic Diagrams (MLDs). For removal/replacement procedures of power supply FRUs and checking and adjustment procedures, refer to the *IBM 4955 Maintenance Information* manual, SY34-0050.



Cabling diagram

400-Watt Power Supply

- The 400-watt power supply is used in the 4955 processor Model E only.
- The 400-watt power supply converts ac input voltages to dc voltages and distributes the dc voltages to the voltage planes on the processor board.

The major functional units of the 400-watt power supply are:

- Prime power assembly
- Power card assembly
- Low voltage card assembly

There are three indicators located on the front, top edge of the low-voltage card. These indicators which are visible when the front cover of the power supply is removed, are:

Load failure
Thermal
Power-on reset

If a power problem exists, these indicators are an aid to analyzing the type of failure that caused the power shutdown.

Load failure. Indicates that an overcurrent or undervoltage caused the power shutdown.

Thermal. Indicates that the thermal on the heat sink of the low-voltage card in the power supply exceeded 68°C (154.4°F), or that a processor thermal sensor is open, and caused the power shutdown.

Power-On Reset. Indicates that the power supply sensed an overvoltage condition or failed to start.

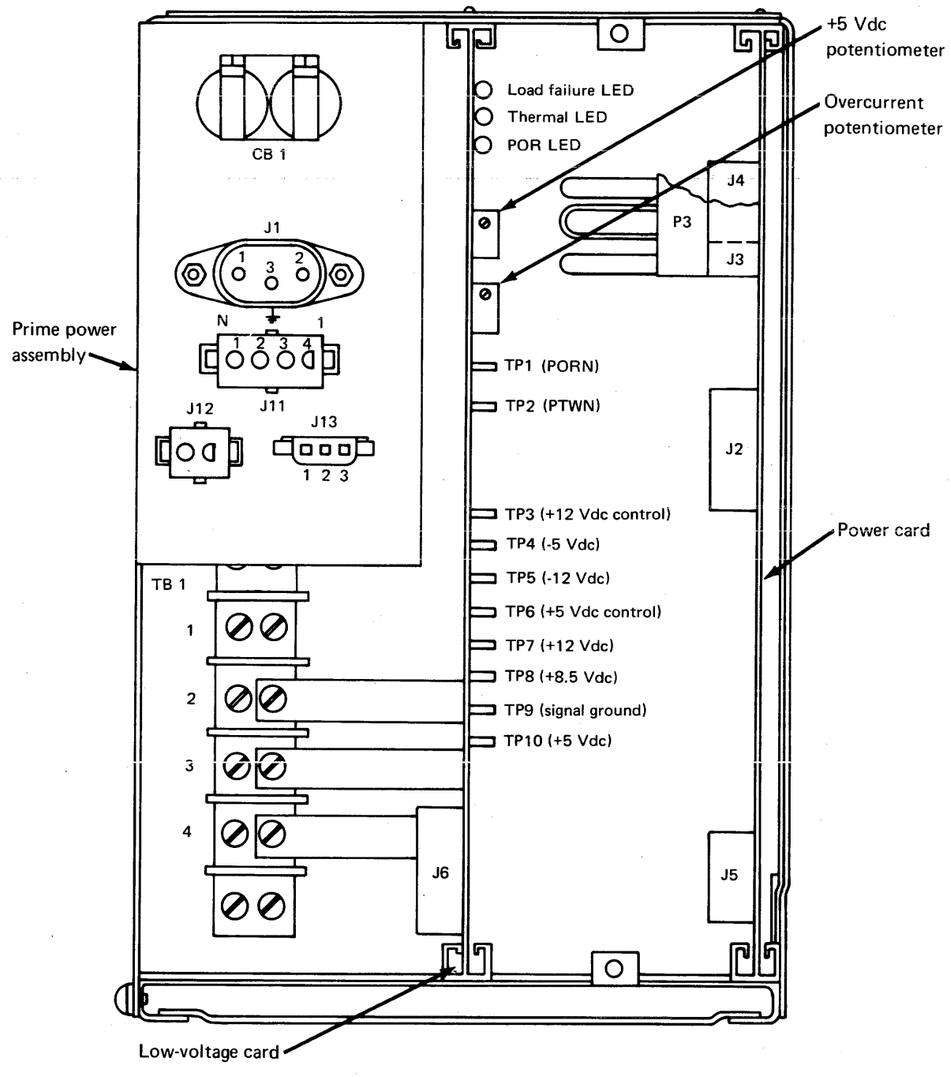
Input Voltage

The power supply input voltage is 47 to 63 Hz, single-phase, 100 to 127 Vac ±10%, or 200 to 240 Vac ±10%.

Output Voltages

The output dc voltages of the 400-watt power supply are +5, −5, +8.5, +12, and −12. The +5 Vdc is the only dc voltage that is adjustable.

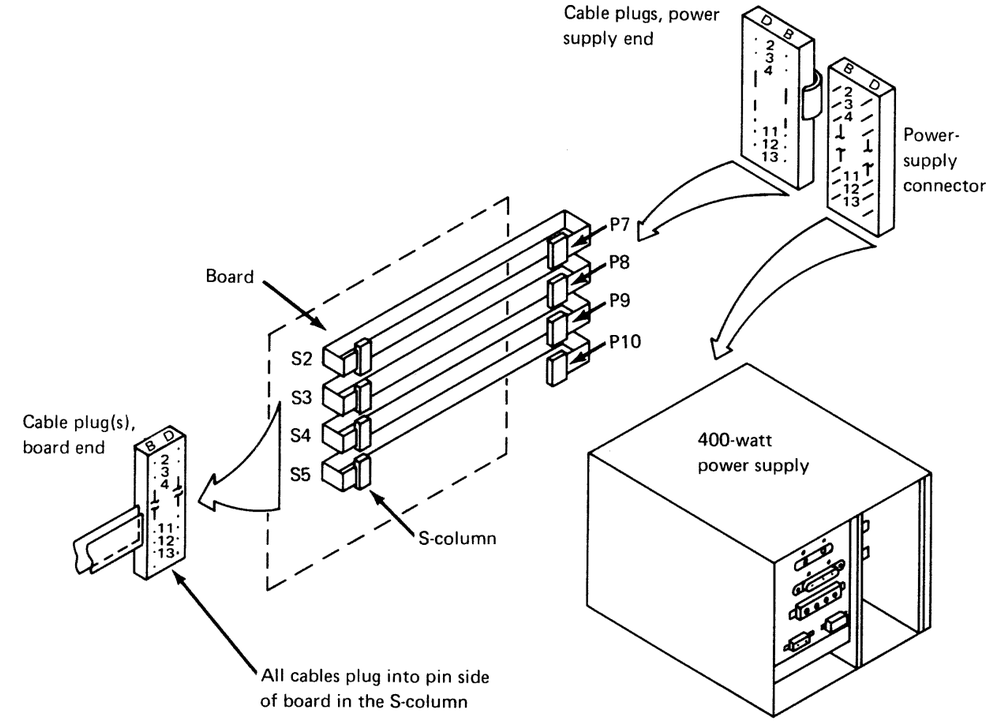
Test points for checking voltages and signals are located on the low-voltage card.

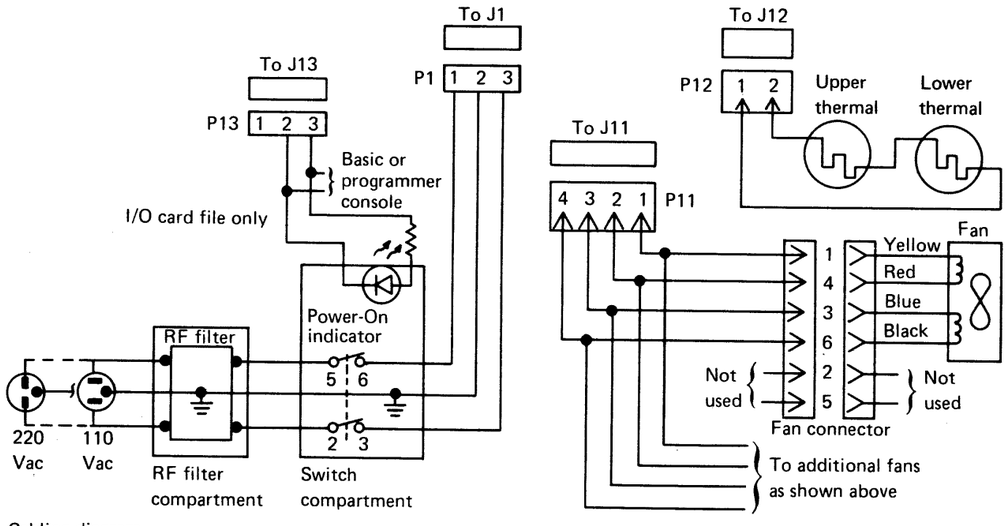


Location diagram, front view

Cabling for DC Output

The dc voltages are distributed to the processor board by flat-wire bus cables to the S-column on the board. Voltage distribution to the cards is achieved through the board internal planes, with each four-wide socket containing ±5 Vdc, +8.5 Vdc, ±12 Vdc, and ground. For specific pin locations and voltage distribution, refer to the Machine Logic Diagrams (MLDs). For removal/replacement procedures of power supply FRUs and check/adjustment procedures, refer to the *IBM 4955 Maintenance Information* manual, SY34-0050, and the power supply MAPs.





Cabling diagram

High-Frequency Power Supply

- The high-frequency power supply is used in the IBM 4955 Model F only.
- The high-frequency power supply converts ac input voltages to dc voltages and distributes the dc voltages to the voltage planes on the processor board.
- The high-frequency power supply plugs directly into the T-socket of the processor board.

The pins on the T-column of the processor board are test points. If a power problem exists, the test points aid in analyzing the power problem.

Input Voltages

The power supply input voltages have the following ranges:

60-Hz input voltage

Nominal	Minimum	Maximum
100	90	110
110	96.5	119
120	104	127
127	111	137
200	180	220
208	180	220
220	193	238
240	208	254

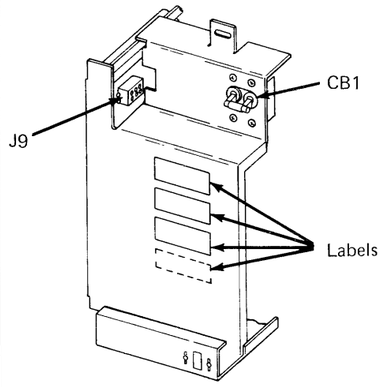
50-Hz input voltage

Nominal	Minimum	Maximum
100	90	110
110	96.5	119
200	180	220
220	193	238
230	202	249
240	210	259

Note: The IBM 4999 Battery Backup Unit can be used as a backup for the high-frequency power supply.

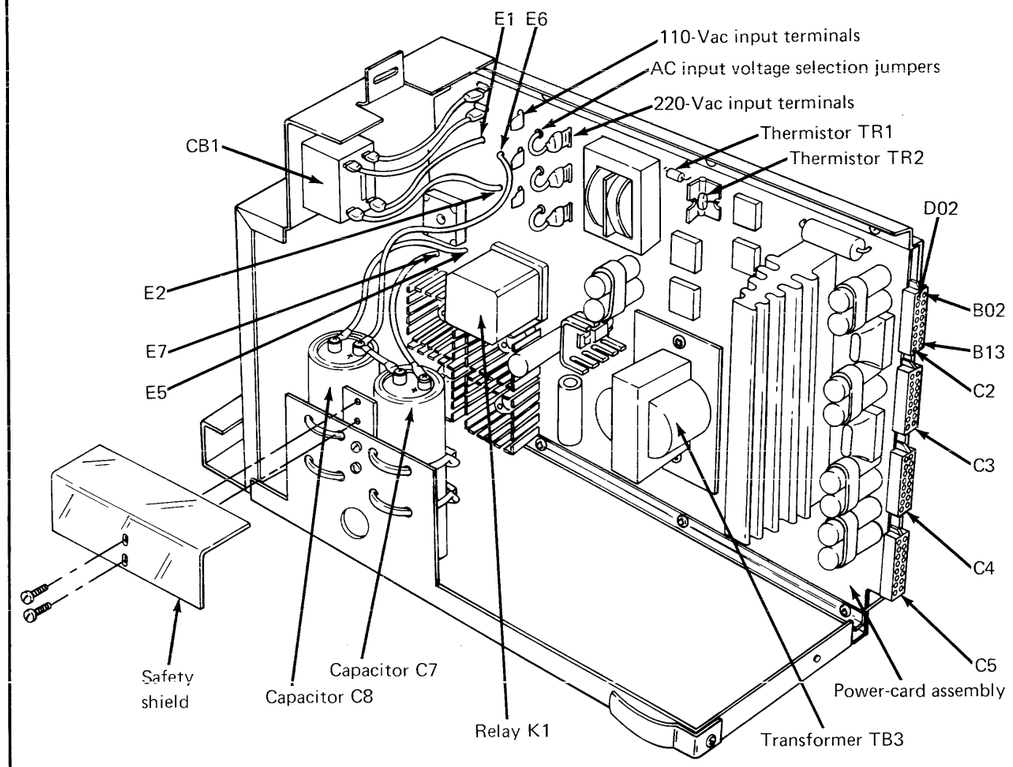
Output Voltages

The output dc voltages of the power supply are +5, −5, +8.5, +12, and −12 Vdc.



Front view

High-Frequency Power Supply Location Diagrams (4955 Model F)



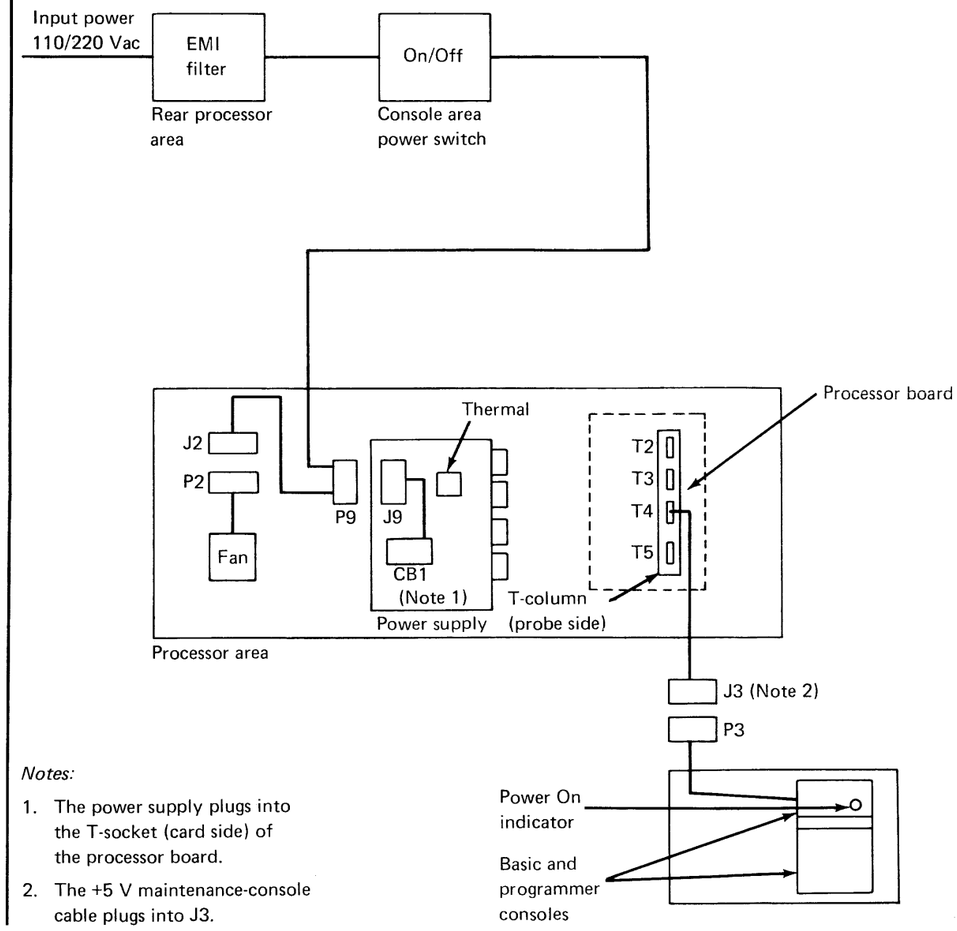
Rear view

Cabling for DC Output

The dc voltages are distributed directly to the T-column of the processor board from the pluggable power supply. Voltage is distributed to the other card sockets on the processor board via internal planes. Refer to the Machine Logic Diagrams (MLDs) for specific pin locations and voltage distribution.

For removal/replacement/adjustment procedures, refer to the *IBM 4955 Maintenance Information* manual, SY34-0050, and to the power supply MAPs.

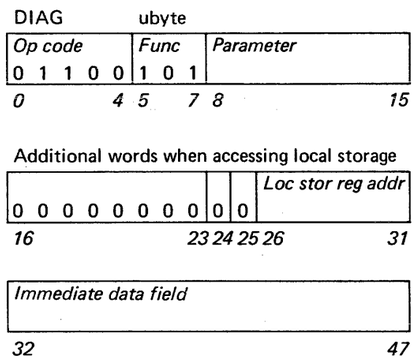
High-Frequency Power Supply Cabling Diagrams (4955 Model F)



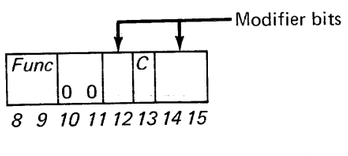
Notes:

1. The power supply plugs into the T-socket (card side) of the processor board.
2. The +5 V maintenance-console cable plugs into J3.

Diagnose (DIAG) Instruction



The Diagnose instruction is used for controlling or testing various hardware functions in a machine-dependent manner. The parameter field has the following general significance:



Bits 10 and 11 are not used, and should be set to 0's to avoid future code obsolescence.

- If the C-bit (bit 13) is set to 1, the number 0005 is loaded into register 0 of the current interrupt level. Software uses this number to determine that the processor is a 4955.
- If the C-bit is set to 0, the function bits have the following meanings:

Bits 8	9	Meaning
0	0	Diagnostic word storage error recovery
0	1	Diagnostic byte storage error recovery
1	0	Main storage to/from local storage
1	1	Enable/disable channel request lines

Diagnostic Storage Error Recovery. This function allows the inhibiting of storage parity generation and checking when using the processor SAR and SDR. The cycle-steal storage data register and storage address register can be selected but parity cannot be inhibited. Other bits in the parameter field are as follows:

Bits	Significance
12 = 0	Load from storage
12 = 1	Store storage
14 = 0	Inhibit parity check/generation
14 = 1	Enable parity check/generation
15 = 0	Select processor SDR/SAR
15 = 1	Select cycle steal SDR/SAR (ignore bit 14)

The storage address for this storage cycle is contained in register 7 while the data register is register 0.

Notes:

1. Functions selected by the parameter field apply only to the storage cycle initiated by the execution of this instruction.
2. Bit 9 provides the option of single-byte manipulation when using the processor SAR and SDR. Diagnostic byte operations are not supported when using the cycle-steal SAR and SDR; therefore, bit 9 is ignored when bit 15 is set to 1.
3. If bit 9 is on (byte operation), and bit 12 is off (load storage operation), the register that is loaded has bits 0–7 set to 0's.

Main Storage To/From Local Storage. This function permits the transfer of data between main storage and local storage by directly addressing local storage. Two additional words are appended to the Diagnose instruction when this function is specified.

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

Bits	Definition
16–25	Unused
26–31	Local storage address
32–47	Data to be transferred
Bit 12 of the parameter field specifies the direction of transfer.	
12 = 0	Load immediate data to local storage
12 = 1	Store local storage to immediate data

Programming Note: This function can change AKRs, IARs, and LSRs in local storage. The current level AKR and LSR in local storage are not continuously updated. Use of this instruction to load or store the current level AKR or LSR is not recommended.

Enable/Disable I/O Channel Request Lines. This function inhibits and logically isolates the interrupt and cycle-steal request lines between the channel and the device. Bit 14 of the parameter field is used as follows:

14 = 0, disable channel request lines
14 = 1, enable channel request lines

Indicators

No indicators are directly changed by this instruction; however, LSRs may be changed by the main storage to local storage function.

Program-Check Conditions

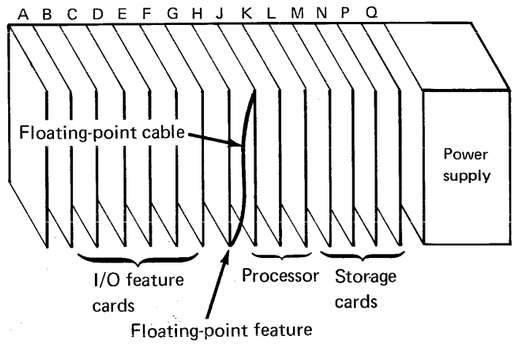
Privilege Violate. Privileged instruction.

Chapter 3. Floating-Point Feature

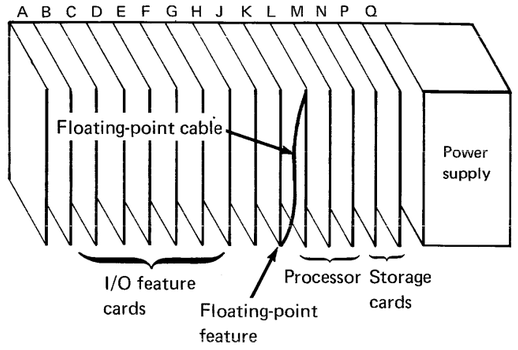
- The floating-point feature provides the floating-point arithmetic capability to the 4955 processor.
- The floating-point feature is packaged on a 4-wide, 6-high card.
- The communications between the floating-point feature card and the processor is carried out on the I/O channel and a floating-point feature cable. The cable connects the floating-point feature card to the data card of the processor.

The floating-point feature card contains a high speed floating-point arithmetic unit, four 64-bit floating-point registers for each level, instruction decode circuits, and the logic for communicating results.

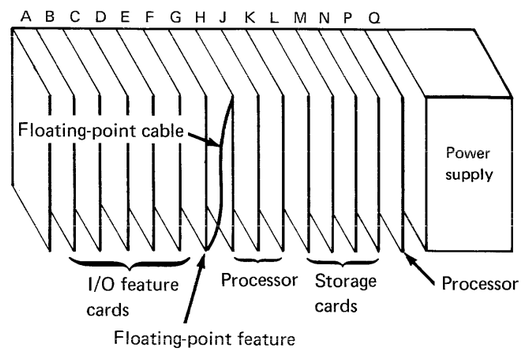
When the processor detects a floating-point instruction, it signals the floating-point feature card that communications between the feature card and the processor are about to begin. The floating-point instruction and the level information are sent to the feature card for further decode and execution. The feature card performs the designated operation and sends the appropriate indications back to the processor.



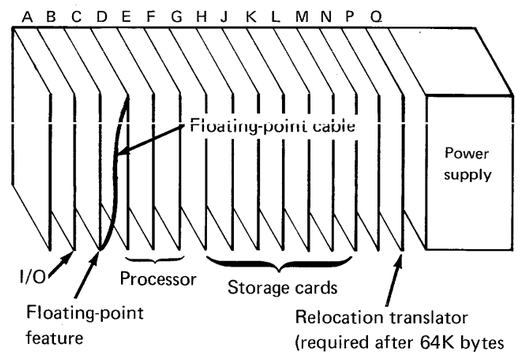
4955 Model A



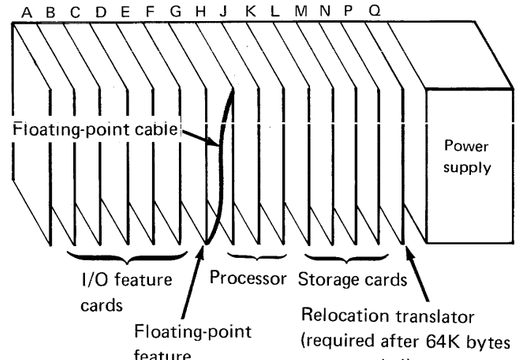
4955 Model C



4955 Models E and F



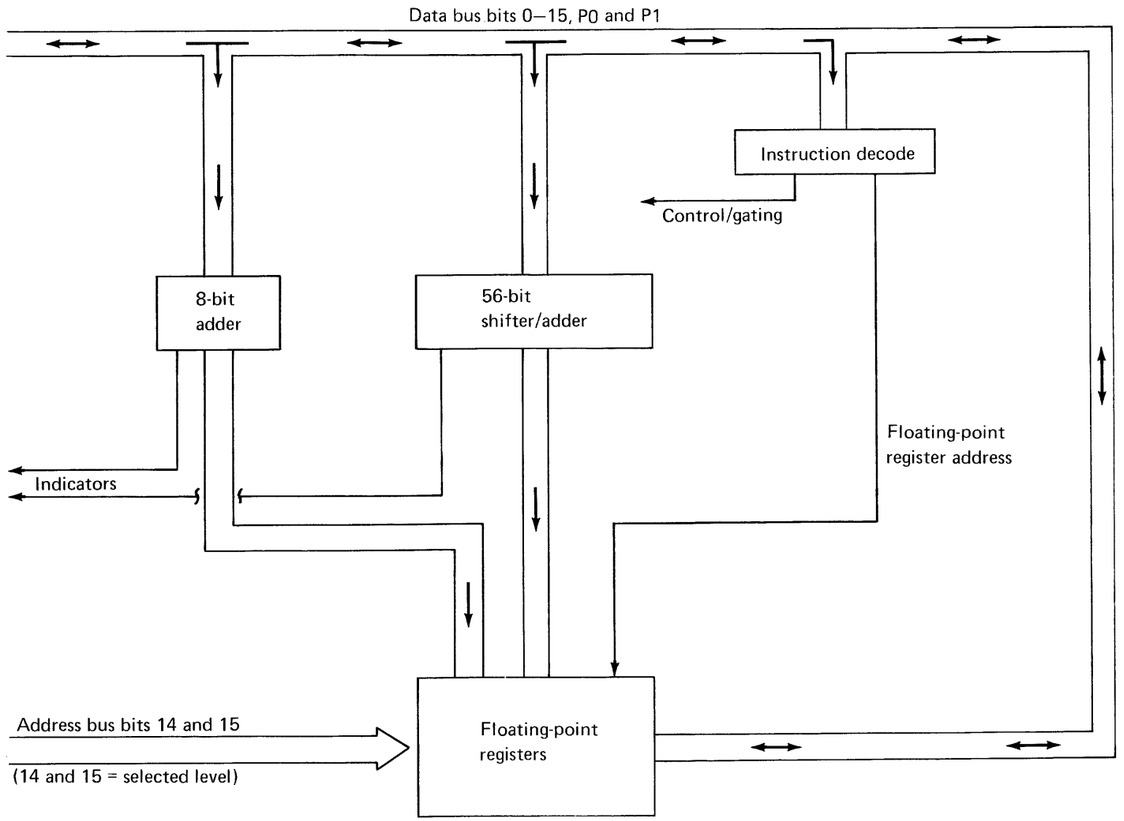
4955 Model B



4955 Model D

Floating-Point Feature Card Data Flow

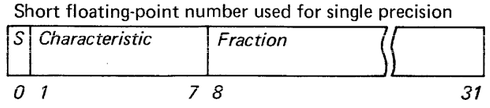
The following diagram shows the data flow for the floating-point feature card.



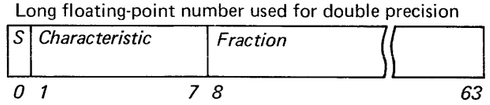
Data Format

Two fixed-length formats (short and long) may be used for floating-point data.

Short floating-point number used for single precision



Long floating-point number used for double precision



Both formats may be used in main storage and in the floating-point registers. The first bit in either format is the sign bit (S). The subsequent seven bit positions are occupied by the characteristic. The fraction field may have either 6 or 14 hexadecimal digits.

The entire set of floating-point instructions is available for both short and long operands. When single precision (short format) is specified, all operands and results are 32-bit floating-point words. With two exceptions, the rightmost 32 bits of the floating-point registers do not participate in single-precision operations, and are unchanged by the operations. The two exceptions are (1) the product in multiply operations (a 64-bit floating-point word that occupies a full register), and (2) a storage-to-register move (the low-order 32 bits are set to 0). When double precision (long format) is specified, all operands and results are 64-bit floating-point words.

Although final results in short precision have six fraction digits, intermediate results in add and subtract operations may extend to seven fraction digits. The low-order digit of a seven-digit fraction is called the guard digit, and it serves to increase the precision of the final result. Intermediate results in long precision may extend to 15 fraction digits, with the 15th digit being the guard digit.

Number Representation

Floating-Point Numbers

The fraction of a floating-point number is expressed in hexadecimal digits. The radix point of the fraction is assumed to be immediately to the left of the high-order fraction digit. To provide the proper magnitude for the floating-point number, the fraction is considered to be multiplied by a power of 16. The characteristic portion, bits 1–7 of both floating-point formats, indicates this power. The bits within the characteristic field can represent numbers from 0 through 127. To accommodate large and small magnitudes, the characteristic is formed by adding 64 to the actual exponent. The range of the exponent is thus −64 through +64. This technique produces a characteristic in "excess 64" notation.

Both positive and negative quantities have a true fraction, with the difference in sign indicated by the sign bit. The number is positive or negative if the sign bit is 0 or 1, respectively.

A floating-point number with 0 characteristic, 0 fraction, and a plus sign is called a true zero. A true zero may arise as the result of an arithmetic operation because of the particular magnitude of the operands. A result is forced to be true zero when an exponent underflow occurs or when a result fraction is 0.

Binary Integers in Main Storage

Signed binary integers occupy storage in one or two fixed-length formats:

- One word format (16 bits)
- Doubleword format (32 bits)

Both formats may be used in main storage, and both are automatically converted to single- or double-precision floating-point numbers during *floating move and convert* operations that move data from storage to a floating-point register. Negative-signed binary integers are in main storage in two's complement form, and they are converted to contain a true fraction. An integer may be moved from main storage to a floating-point register, without conversion, by using the floating move instruction. In this case, the integer is assumed to be a floating-point number.

Floating move and convert operations that move data from a floating-point register to storage accomplish the reverse process; the floating-point number in the register is converted automatically to an integer. This integer result is then placed in main storage.

Normalization

A quantity can be represented with the greatest precision by a floating-point number of given fraction length when that number is normalized. A normalized floating-point number has a non-0 high-order hexadecimal fraction digit. If one or more high-order fraction digits are 0's, the number is said to be unnormalized. The process of normalization consists of shifting the fraction left until the high-order hexadecimal digit is non-0, and reducing the characteristic by the number of hexadecimal digits shifted. A 0 fraction is not normalized.

Floating-point numbers in main storage are assumed to be normalized. Unnormalized numbers do not provide the same accuracy as normalized numbers. Normalization takes place after the multiply instruction and after add and subtract instructions if an actual subtraction has taken place. For example, A+(−B), A−(+B), (−A)−(−B). Normalization does not follow a true addition of operands that were unnormalized numbers.

Processor to Floating-Point Card Line Descriptions

When the floating-point processor is in the reset state, 'floating pt enable' becoming active causes the floating-point processor to examine address bus bits 8–11 for its encoded selection address (all bits 0's for the floating-point address). Once addressed, the floating-point card raises the 'floating pt installed' line to indicate to the processor that floating-point is selected.

Address bus bits 14 and 15 carry the binary-encoded current level of the processor, or the selected level of a privileged operation.

The data bus is used to carry data to and from the floating-point card during the execution of floating-point instructions. During the early part of the instruction cycle, the first word of the floating-point instruction is sent to the floating-point card on the data bus. The floating-point card decodes the instruction and brings up the necessary gating and control lines.

'System reset' is generated by the processor, and is used to reset the floating-point logic.

'Data strobe' is a multipurpose signal generated by the processor. Other than the first data strobe, which is not used, and the last, which is used to break the floating-point card out of its fetch loop, data strobe indicates that the processor is going to send data to the floating-point card. The data bus becomes active after the data strobe is sent to the floating-point card.

The timing pulses 'gate time A' and 'gate time C' are generated by the processor and used by the floating-point card to synchronize its operation with the processor.

'Floating pt installed' is generated by the floating-point card to signal the processor that it has been selected and floating-point operations may begin. This line is reset at the end of the operation. An exception to this is during store integer operations, where the floating-point card signals the processor to prepare to store the first word. The signal is the resetting of the 'floating pt installed' line. A system reset, program-check, or machine-check condition from the processor resets this line at any time.

'Floating pt sample' is a multipurpose signal generated by the floating-point card. If this line becomes active during the time 'floating pt enable' is active, it indicates that data is to be sent to the processor. On all non-privileged operations, this line is generated after 'floating pt enable' has become inactive to signify completion of the operation.

'Floating-Pt exception' is generated by the floating-point card whenever an exception is recognized during the execution of a floating-point instruction.

The 'floating-point indicators' ('floating pt zero,' 'floating pt carry,' 'floating pt negative,' 'floating pt overflow,' and 'floating pt even') are set as the result of the execution of the floating-point instructions. These indicators are used to determine the next operation to be performed.

The 'CPU check' line is generated by the processor when it detects a program-check or machine-check condition. The floating-point card treats the check as a system reset. This line remains active long enough to ensure that the channel is in the quiescent state when the check line is deactivated.

The 'floating pt enable' line is a multipurpose signal generated by the processor to gate the drivers and receivers in the floating-point card, select the floating-point card, and to control cycle-steal interleaving during data transfer.

Channel lines used by floating-point

	Processor	Direction	Floating-point card
Address bus bits 14, 15		→	
Data bus bits 0–15, P0, P1		↔	
System reset		→	
Data strobe		→	
Special floating-point lines (board-wired)			
Gate time A		→	
Gate time C		→	
Floating-point installed		←	
Top card connector to data card			
Floating-point sample		←	
Floating-point exception		←	
Floating-point zero		←	
Floating-point carry		←	
Floating-point negative		←	
Floating-point overflow		←	
Floating-point even		←	
CPU check		→	
Floating-point enable		→	

Floating-Point Instructions

The floating-point instruction set provides a variety of instructions that deal with single- or double-precision floating-point data. The main categories are:

- Arithmetic instructions (add, subtract, multiply, divide, and compare)
- Data movement instructions (with or without conversion of binary integers)

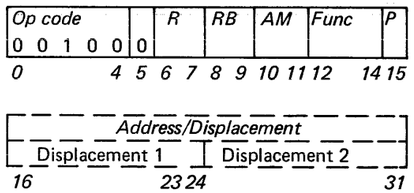
Two privileged instructions are also provided for interrogation of the floating-point registers: Copy Floating Level Block (CPFLB) and Set Floating Level Block (SEFLB).

All floating-point instructions use the floating-point registers. One group of instructions (storage/floating-point register) specifies a register for one operand and an effective main storage address for the other operand. Another group (floating-point register to floating-point register) specifies registers for both operands.

Instruction Formats

Arithmetic and data movement instructions use the following two formats:

Storage/Floating-Point Register



Op-code field. Specifies a floating-point operation.

R-field. Specifies a floating-point register.

Function field. Designates the function to be performed.

Function field	Instruction
000	Add
001	Subtract
010	Multiply
011	Divide
100	Move and convert (from storage)
101	Move (from storage)
110	Move and convert (to storage)
111	Move (to storage)

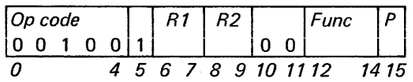
RB and AM fields. Designate the effective address argument.

P-field. Designates the precision of floating-point data.

0 = Single precision
1 = Double precision

Second word (bits 16-31). Address mode appended word for an AM field equal to 10 or 11.

Floating-Point Register to Floating-Point Register



Op-code field. Specifies a floating-point operation.

R1 and R2 fields. Specify floating-point registers.

Bits 10–11. Designate the function modifier. These bits are not used and must be set to 0's to avoid future code obsolescense.

Function field. Designates the function to be performed.

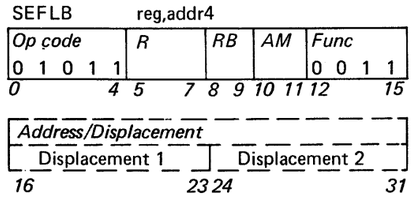
P-field. Designates the precision of floating-point data.

0 = Single precision
1 = Double precision

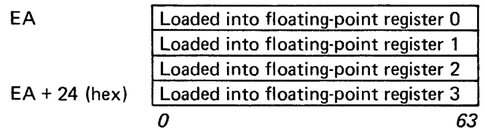
Function field	Instruction
000	Add
001	Subtract
010	Multiply
011	Divide
100	Move
101	Compare
110	(Not used)
111	(Not used)

Privileged Instructions

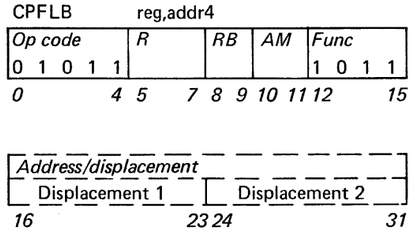
Set Floating Level Block (SEFLB)



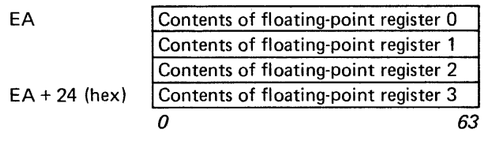
A floating level block in main storage is loaded into the floating-point registers for the level specified by the R-field register. The generated effective address (EA) specifies the beginning address of the floating level block. The contents of main storage and the R-field register remain unchanged. The floating level block appears in main storage as follows:



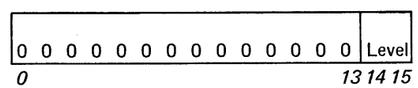
Copy Floating Level Block (CPFLB)



The contents of the floating-point registers (floating level block) for the level specified by the R-field register are stored into main storage locations beginning at the specified effective address (EA). All registers remain unchanged. After execution of this instruction, the floating level block appears in main storage as follows:



The general register specified by the R field has the format:



Exception Conditions

Program-Check Conditions

Invalid Storage Address

Instruction Word or Operand. One or more words of the instruction or the effective address are outside the installed storage size of the system. The register-to-register instructions are suppressed. The storage/register instructions are terminated.

A program-check class interrupt occurs, with invalid storage address (bit 1) set in the PSW.

Privilege Violate

Privileged Instruction. A privileged instruction is encountered while the processor is in problem state. The instruction is suppressed.

A program-check class interrupt occurs with privilege violate (bit 2) set in the PSW.

Protect Check

Instruction Fetch or Operand Access. When the processor is in the problem state, an instruction is fetched or data is accessed from a storage area not assigned to the current operation.

Operand Store. When the processor is in the problem state, the instruction attempts to change an operand in a storage area assigned as read-only.

The register-to-register instructions are suppressed. The storage/register instructions are terminated. A program-check class interrupt occurs, with protect check (bit 3) set in the PSW.

Specification Check

Operand Address. The generated effective address has violated an even-byte boundary requirement.

Indirect Address. When using addressing mode (AM=11), the indirect address is not on an even-byte-boundary.

The register-to-register instructions are suppressed. The storage/register instructions are terminated. A program-check class interrupt occurs, with specification check (bit 0) set in the PSW.

Soft-Exception Trap Condition

Floating-Point Exception

A floating-point underflow, overflow, or divide check has occurred. The instruction completes execution. A soft-exception-trap class interrupt occurs, with floating-point exception (bit 5) set in the PSW. The Overflow, Even, and Carry indicators are set as follows:

Overflow Indicator. Set to 1 by an overflow, underflow, or divide check.

Carry Indicator. Set to 1 by a divide check.

Even Indicator. Set to 1 by an underflow.

Invalid Function

An attempt is made to execute a floating-point instruction when the feature is not installed. The register-to-register instructions are suppressed. The storage/register instructions are terminated. A soft-exception-trap class interrupt occurs, with invalid function (bit 4) set in the PSW.

Note: The resulting class interrupt causes the contents of the storage address register (SAR) to be loaded into general register 7. SAR contains either (1) the calculated effective address of data operand 2 or (2) the address of the attempted instruction for register-to-register operations.

Chapter 4. Special Maintenance Equipment

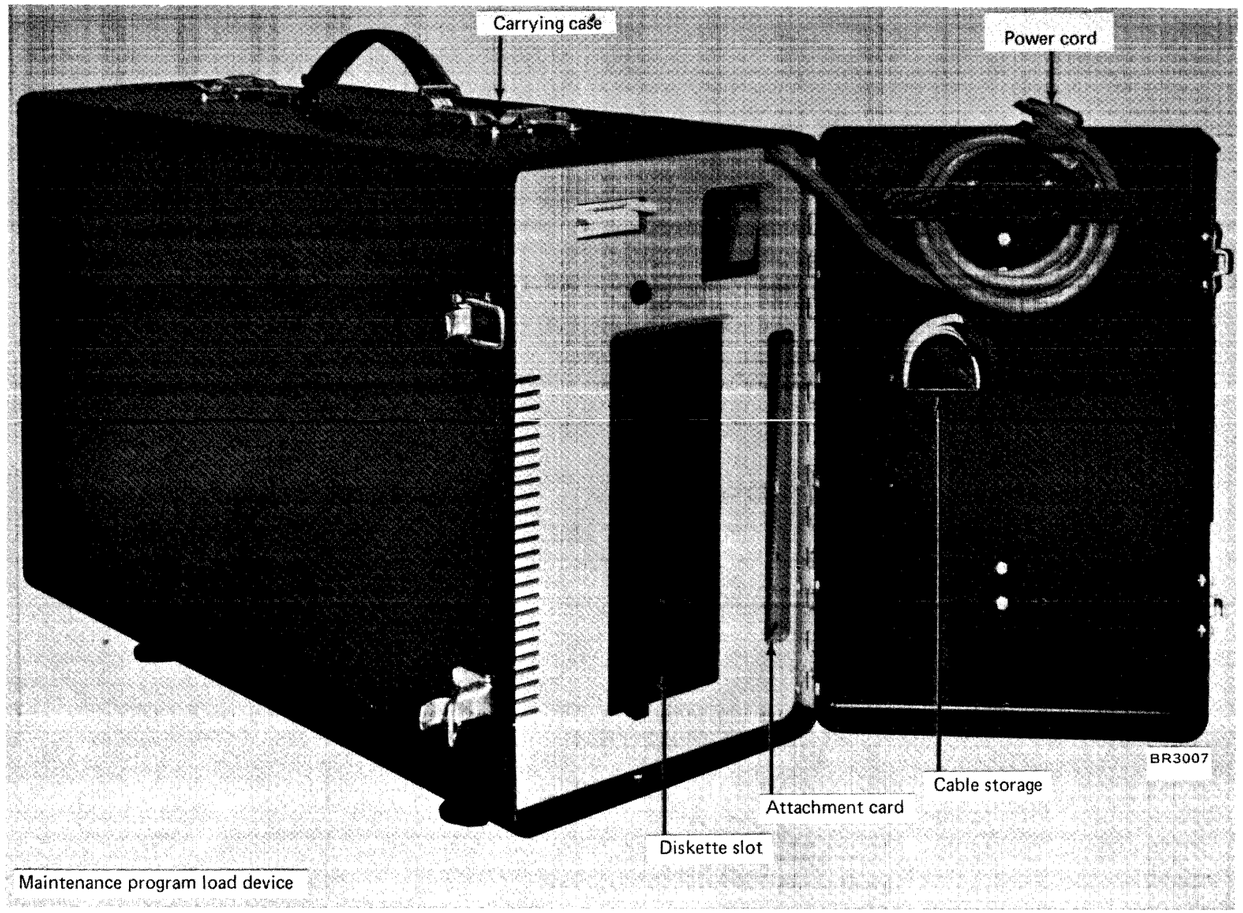
This chapter describes the three special maintenance tools that are needed, in certain cases, to service the 4955 or the 4959. Other special maintenance tools are covered in related I/O manuals.

Maintenance Program Load Device

Maintenance program load device may be used if the machine configuration being serviced does not include an IBM 4964 Diskette Unit, or if the 4964 is not capable of loading programs into processor storage.

The maintenance program load device contains an attachment card and cable, and a power cord. The attachment card should be plugged into the next available I/O slot in either the processor or the 4959 I/O expansion unit. If there is no slot available, one of the configuration attachment cards must be pulled to make room. The power cord plug may be plugged into any available ac outlet.

Refer to the *IBM Series/1 4955 Maintenance Information* manual, SY34-0050, for the attachment procedure.

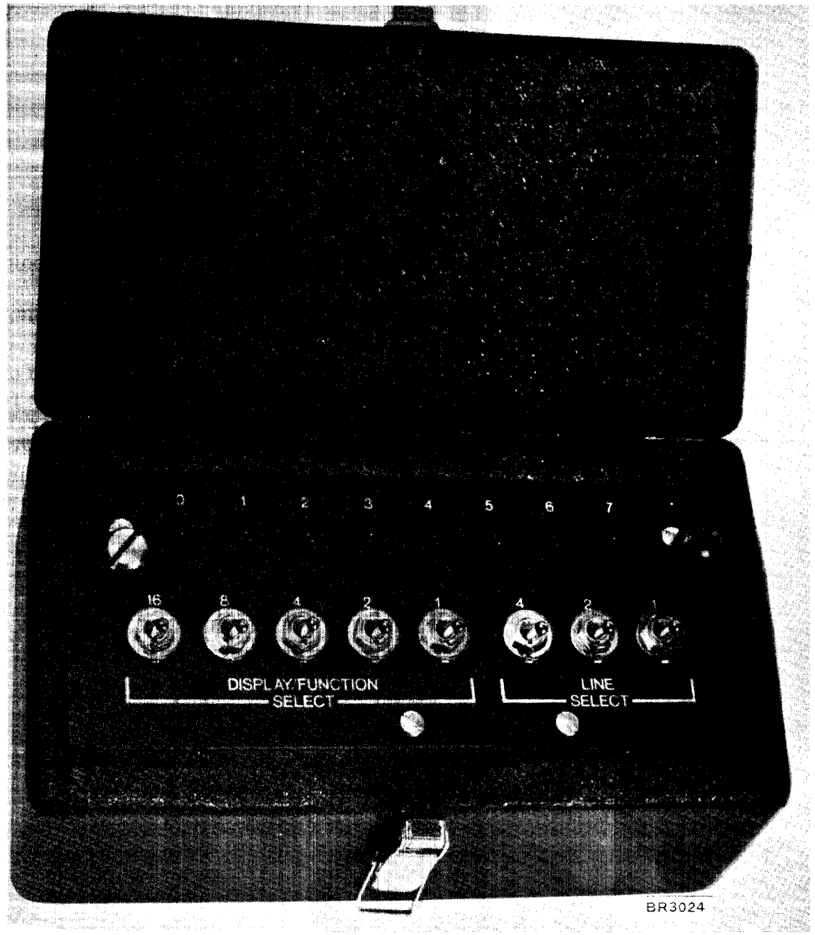


Maintenance program load device

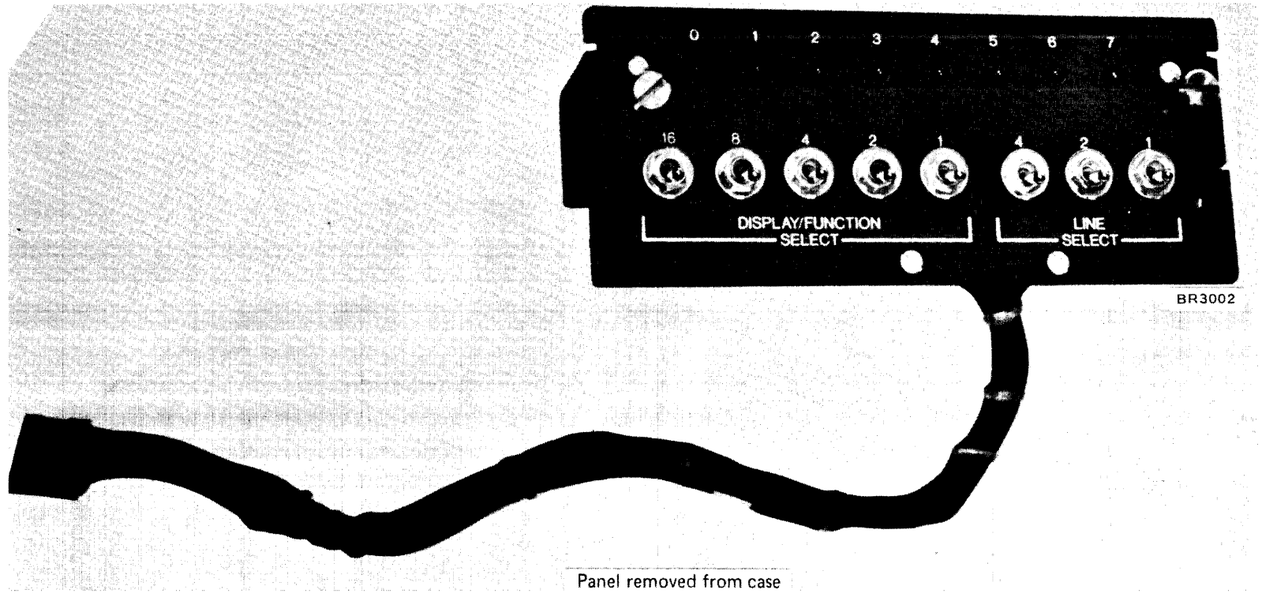
Maintenance Communications Panel

The maintenance communications panel is used to service communications feature problems on configurations that do not include the optional communications panel. The maintenance communications panel comes in a carrying case, and includes the panel and an attachment cable.

Refer to the *IBM Series/1 4955 Maintenance Information* manual, SY34-0050, for the attachment procedure.



Communications panel in carrying case



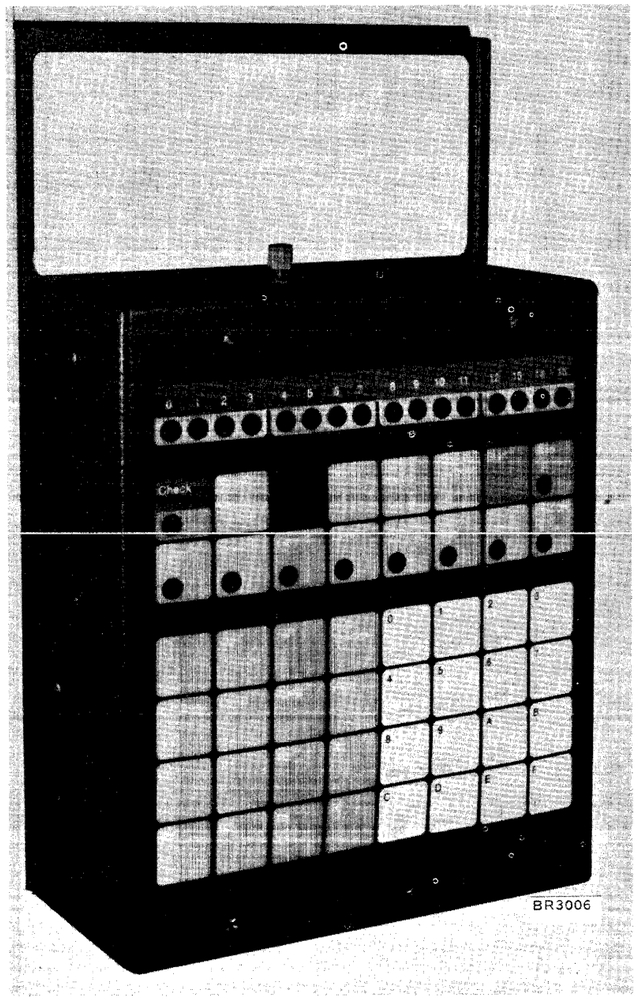
Panel removed from case

Maintenance Console

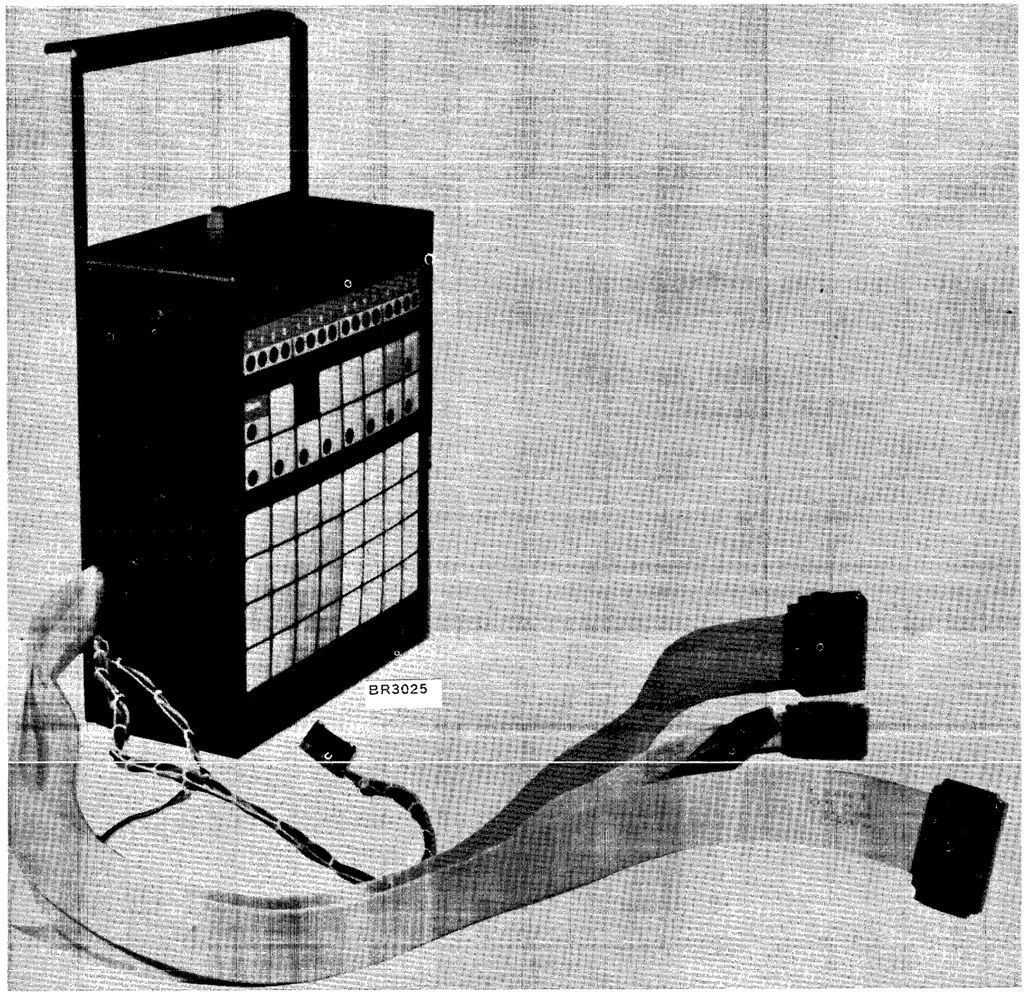
The maintenance console is necessary when the processor being serviced does not contain the optional programmer console feature. The tool is identical to the programmer console.

The maintenance console tool consists of the console, a protective case and cables.

Refer to the *IBM Series/1 4955 Maintenance Information* manual, SY34-0050, for the attachment procedure. Refer to Chapter 2 of this manual for a description of the console operation.



Maintenance console in case



Maintenance console ready to install

Appendix A. Instruction Formats

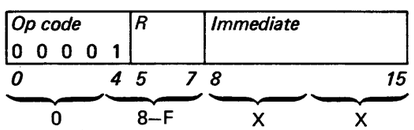
The following instruction formats are shown in ascending sequence based on operation code. Bits 0–4 of the first instruction word comprise the operation code field. Bit combinations and their hexadecimal representations are shown for each operation code.

Some instructions contain a function field that modifies the operation code to form individual instructions within a group. Each chart shows the function field bit combinations in hexadecimal and in ascending sequence. The assembler mnemonic, assembler syntax, and instruction name are listed for the individual instructions. The asterisk (*) shown with the assembler syntax indicates indirect addressing.

Refer to the *IBM Series/1 4955 Processor and Processor Features Description*, GA34-0021, for a description of the address mode (AM) appended words.

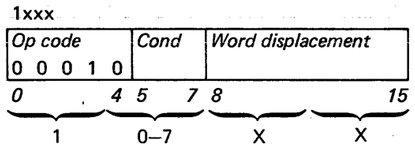
0xxx

0	0–7	X	X	ABI	byte,reg	Add Byte Immediate



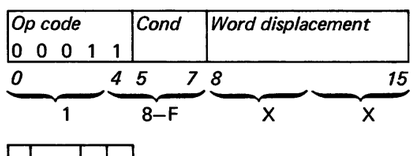
0	8–F	X	X	MVBI	byte,reg	Move Byte Immediate

1xxx



1	0–7	X	X	JC	cond,jdisp	Jump on Condition
				JC	cond,jaddr	Jump on Condition

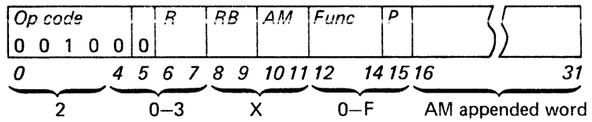
Extended mnemonics:
JCY, JE, JEV, JLE, JLLE, JLLT, JLT, JMIX
JN, JOFF, JON, JP, JZ



1	8–F	X	X	JNC	cond,jdisp	Jump on Not Condition
				JNC	cond,jaddr	Jump on Not Condition

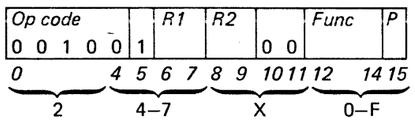
Extended mnemonics:
JGE, JGT, JLGE, JLGT, JNCY, JNE, JNEV,
JNMIX, JNN, JNOFF, JNON, JNP, JNZ

2xxx



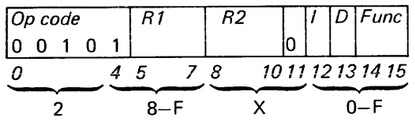
2	0–3	X				
			0	FA	addr4,freg	Floating Add
			1	FAD	addr4,freg	Floating Add Double
			2	FS	addr4,freg	Floating Subtract
			3	FSD	addr4,freg	Floating Subtract Double
			4	FM	addr4,freg	Floating Multiply
			5	FMD	addr4,freg	Floating Multiply Double
			6	FD	addr4,freg	Floating Divide
			7	FDD	addr4,freg	Floating Divide Double
			8	FMVC	addr4,freg	Floating Move and Convert
			9	FMVCD	addr4,freg	Floating Move and Convert Double
			A	FMV	addr4,freg	Floating Move
			B	FMVD	addr4,freg	Floating Move Double
			C	FMVC	freg,addr4	Floating Move and Convert
			D	FMVCD	freg,addr4	Floating Move and Convert Double
			E	FMV	freg,addr4	Floating Move
			F	FMVD	freg,addr4	Floating Move Double

2xxx



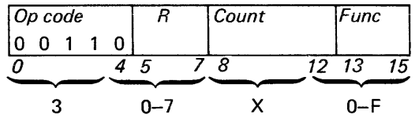
2	4–7	X				
			0	FA	freg,freg	Floating Add
			1	FAD	freg,freg	Floating Add Double
			2	FS	freg,freg	Floating Subtract
			3	FSD	freg,freg	Floating Subtract Double
			4	FM	freg,freg	Floating Multiply
			5	FMD	freg,freg	Floating Multiply Double
			6	FD	freg,freg	Floating Divide
			7	FDD	freg,freg	Floating Divide Double
			8	FMV	freg,freg	Floating Move
			9	FMVD	freg,freg	Floating Move Double
			A	FC	freg,freg	Floating Compare
			B	FCD	freg,freg	Floating Compare Double
			C	(must not be used)		Executes FMV
			D	(must not be used)		Executes FMVD
			E	(must not be used)		Indicators are reset
			F	(must not be used)		Indicators are reset

2xxx



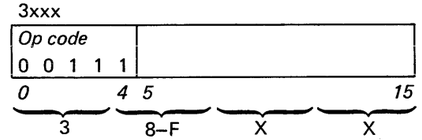
2	8–F	X				
			0	MVFD	(reg),(reg)	Move Byte Field and Decrement
			1	(unused)		
			2	CFNED	(reg),(reg)	Compare Byte Field Not Equal and Decrement
			3	CFED	(reg),(reg)	Compare Byte Field Equal and Decrement
			4	MVFN	(reg),(reg)	Move Byte Field and Increment
			5	(unused)		
			6	CFNEN	(reg),(reg)	Compare Byte Field Not Equal and Increment
			7	CFEN	(reg),(reg)	Compare Byte Field Equal and Increment
			8	FFD	reg,(reg)	Fill Byte Field and Decrement
			9	(unused)		
			A	SFNED	reg,(reg)	Scan Byte Field Not Equal and Decrement
			B	SFED	reg,(reg)	Scan Byte Field Equal and Decrement
			C	FFN	reg,(reg)	Fill Byte Field and Increment
			D	(unused)		
			E	SFNEN	reg,(reg)	Scan Byte Field Not Equal and Increment
			F	SFEN	reg,(reg)	Scan Byte Field Equal and Increment

3xxx



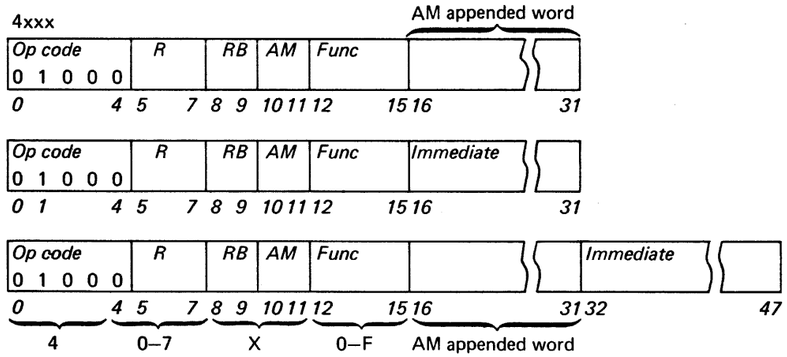
3	0–7	X				
			0,8	SLC	cnt16,reg	Shift Left Circular
			1,9	SLL	cnt16,reg	Shift Left Logical
			2,A	SRL	cnt16,reg	Shift Right Logical
			3,B	SRA	cnt16,reg	Shift Right Arithmetic
			4,C	SLCD	cnt31,reg	Shift Left Circular Double
			5,D	SLLD	cnt31,reg	Shift Left Logical Double
			6,E	SRLD	cnt31,reg	Shift Right Logical Double
			7,F	SRAD	cnt31,reg	Shift Right Arithmetic Double

3xxx



3	8–F	X	X	Illegal operation code (program-check condition)

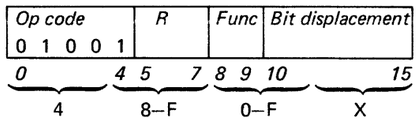
4xxx



4	0–7	X	0	MVA	raddr,addr4	Move Address
			0	MVWI	word,addr4	Move Word Immediate
			1	(invalid)		
			2	(invalid)		
			3	(invalid)		
			4	MVA	addr4,reg	Move Address (see Note)
			4	MVWI	word,reg	Move Word Immediate (see Note)
			5	(invalid)		
			6	(invalid)		
			7	(invalid)		
			8	STM	reg,addr4[,abcnt]	Store Multiple
			9	AWI	word,addr4	Add Word Immediate
			9	AA	raddr,addr4	Add Address
			A	LMB	addr4	Load Multiple and Branch (see Note)
			B	TWI	word,addr4	Test Word Immediate
			C	OWI	word,addr4	OR Word Immediate
			C	SBTWI	word,addr4	Set Bits Word Immediate
			D	RBTWI	word,addr4	Reset Bits Word Immediate
			E	SWI	word,addr4	Subtract Word Immediate
			E	SA	raddr,addr4	Subtract Address
			F	CWI	word,addr4	Compare Word Immediate
			F	CA	raddr,addr4	Compare Address

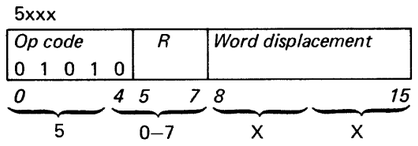
Note: Use format without immediate field.

4xxx



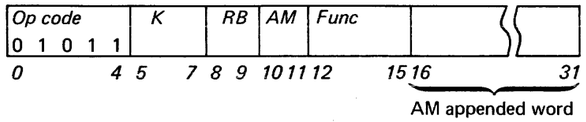
4	8–F	0–3	X	TBT	(reg,bitdisp)	Test Bit
		4–7		TBTS	(reg,bitdisp)	Test Bit and Set On
		8–B		TBTR	(reg,bitdisp)	Test Bit and Reset
		C–F		TBTV	(reg,bitdisp)	Test Bit and Invert

5xxx



5	0	0	0	NOP		No Operation
	0	X	X	J	jdisp	Jump Unconditional
				J	jaddr	Jump Unconditional
	1–7	X	X	BXS	(reg^{1-7},jdisp)	Branch Indexed Short
				BXS	(reg^{1-7})	Branch Indexed Short
				BXS	addr	Branch Indexed Short

5xxx

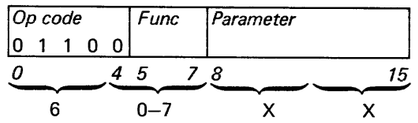


5	8–F	X	0	SEIMR	addr4	Set Interrupt Mask Register
			1	SESR	reg,addr4	Set Segmentation Register
			2	SEAKR	addr4	Set Address Key Register (Note 1)
			3	SEFLB	reg,addr4	Set Floating Level Block
			4	SESK	reg,addr4	Set Storage Key
			5	(invalid)		
			6	SELB	reg,addr4	Set Level Status Block
			7	(invalid)		
			8	CPIMR	addr4	Copy Interrupt Mask Register
			9	CPSR	reg,addr4	Copy Segmentation Register
			A	CPAKR	addr4	Copy Address Key Register (Note 2)
			B	CPFLB	reg,addr4	Copy Floating Level Block
			C	CPSK	reg,addr4	Copy Storage Key
			D	CPIPF	addr4	Copy In-Process Flags
			E	CPLB	reg,addr4	Copy Level Block
			F	CPPSR	addr4	Copy Processor Status and Reset

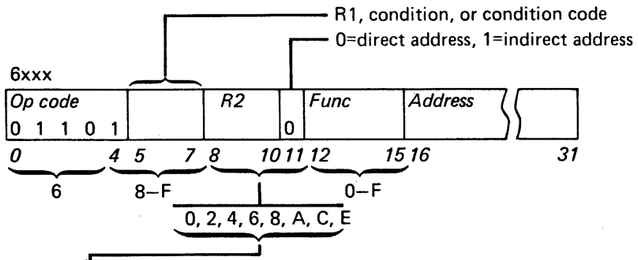
Notes:

1. Use format with K-field.
   Extended mnemonics: SEISK, SEOTK, SEOOK.
2. Use format with K-field.
   Extended mnemonics: CPISK, CPOTK, CPOOK.

6xxx



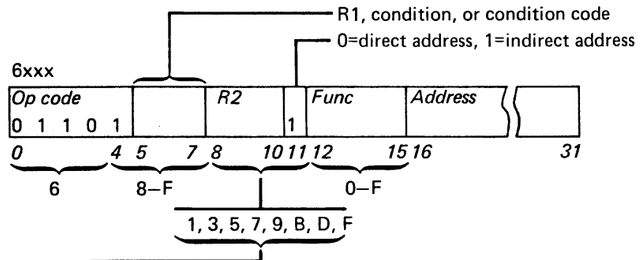
6	0	X	X	SVC	ubyte	Supervisor Call
	1			LEX	[ubyte]	Level Exit
	2			EN	ubyte	Enable
	3			DIS	ubyte	Disable
	4			STOP	[ubyte]	Stop
	5			DIAG	ubyte	Diagnose
	6			IOPK		Interchange Operand Keys
	7			(invalid)		



6	8–F				
		0	BC	cond,longaddr	Branch on Condition (Note 1)
		1	BNC	cond,longaddr	Branch on Not Condition (Note 2)
		2	B	longaddr	Branch Unconditional (Note 3)
		3	BAL	longaddr,reg	Branch and Link (Note 4)
		4	BCC	cond,longaddr	Branch on Condition Code (Note 5)
		5	BNCC	cond,longaddr	Branch on Not Condition Code (Note 6)
		6	BOV	longaddr	Branch on Overflow
		7	BNOV	longaddr	Branch on Not Overflow
		8	MVW	longaddr,reg	Move Word
		9	OW	longaddr,reg	OR Word
		9	SBTW	longaddr,reg	Set Bits Word
		A	RBTW	longaddr,reg	Reset Bits Word
		B	XW	longaddr,reg	Exclusive OR Word
		C	IO	longaddr	Operate I/O
		D	MVW	reg,longaddr	Move Word
		E	AW	longaddr,reg	Add Word
		F	SW	longaddr,reg	Subtract Word

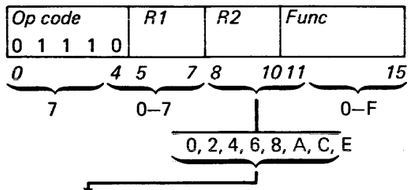
Notes:

1. Extended mnemonics: BCY, BE, BEV, BLE, BLLE, BLLT, BLT, BMIX, BN, BOFF, BON, BP, BZ.
2. Extended mnemonics: BGE, BGT, BLGE, BLGT, BNCY, BNE, BNEV, BNMIX, BNN, BNOFF, BNON, BNP, BNZ.
3. Extended mnemonic: BX.
4. Extended mnemonic: BALX.
5. Extended mnemonic: BNER.
6. Extended mnemonic: BER.



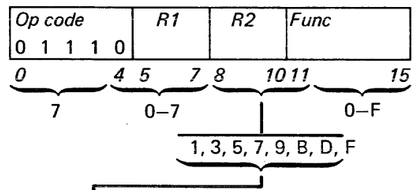
6	8–F				
		0	BC	cond,longaddr*	Branch on Condition
		1	BNC	cond,longaddr*	Branch on Not Condition
		2	B	longaddr*	Branch Unconditional
		3	BAL	longaddr*,reg	Branch and Link
		4	BCC	cond,longaddr*	Branch on Condition Code
		5	BNCC	cond,longaddr*	Branch on Not Condition Code
		6	BOV	longaddr*	Branch on Overflow
		7	BNOV	longaddr*	Branch on Not Overflow
		8	MVW	longaddr*,reg	Move Word
		9	OW	longaddr*,reg	OR Word
		9	SBTW	longaddr*,reg	Set Bits Word
		A	RBTW	longaddr*,reg	Reset Bits Word
		B	XW	longaddr*,reg	Exclusive OR Word
		C	IO	longaddr*	Operate I/O
		D	MVW	reg,longaddr*	Move Word
		E	AW	longaddr*,reg	Add Word
		F	SW	longaddr*,reg	Subtract Word

7xxx

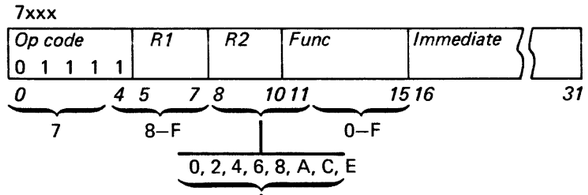


7	0–7				
		0	RBTW	reg,reg	Reset Bits Word
		1	OW	reg,reg	OR Word
		1	SBTW	reg,reg	Set Bits Word
		2	SCY	reg	Subtract Carry Indicator
		3	XW	reg,reg	Exclusive OR Word
		4	MVW	reg,reg	Move Word
		5	CW	reg,reg	Compare Word
		6	CMR	reg[,reg]	Complement Register
		7	IR	reg,reg	Interchange Registers
		8	AW	reg,reg	Add Word
		9	AWCY	reg,reg	Add Word With Carry
		A	SW	reg,reg	Subtract Word
		B	SWCY	reg,reg	Subtract Word With Carry
		C	ACY	reg	Add Carry Register
		D	VR	reg[,reg]	Invert Register
		E	CPLSR	reg	Copy Level Status Register
		F	SEIND	reg	Set Indicators

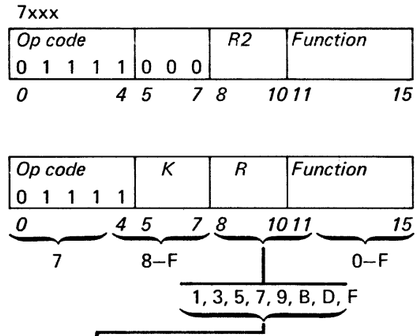
7xxx



7	0–7				
		0	SLC	reg,reg	Shift Left Circular
		1	SLL	reg,reg	Shift Left Logical
		2	SRL	reg,reg	Shift Right Logical
		3	SRA	reg,reg	Shift Right Arithmetic
		4	SLCD	reg,reg	Shift Left Circular Double
		5	SLLD	reg,reg	Shift Left Logical Double
		6	SRLD	reg,reg	Shift Right Logical Double
		7	SRAD	reg,reg	Shift Right Arithmetic Double
		8	(invalid)		
		9	SLT	reg,reg	Shift Left and Test
		A	(invalid)		
		B	(invalid)		
		C	(invalid)		
		D	SLTD	reg,reg	Shift Left and Test Double
		E	(invalid)		
		F	(invalid)		



7	8–F				
		0	NWI	word,reg[,reg]	And Word Immediate
		1	AWI	word,reg[,reg]	Add Word Immediate
		1	AA	raddr,reg[,reg]	Add Address
		2	SWI	word,reg[,reg]	Subtract Word Immediate
		2	SA	raddr,reg[,reg]	Subtract Address
		3	OWI	word,reg[,reg]	OR Word Immediate
		3	SBTWI	word,reg[,reg]	Set Bits Word Immediate
		4	RBTWI	word,reg[,reg]	Reset Bits Word Immediate
		5	XWI	word,reg[,reg]	Exclusive OR Word Immediate
		6	CWI	word,reg	Compare Word Immediate
		6	CA	raddr,reg	Compare Address
		7	TWI	word,reg	Test Word Immediate
		8	(invalid)		
		9	(invalid)		
		A	(invalid)		
		B	(invalid)		
		C	(invalid)		
		D	(invalid)		
		E	(invalid)		
		F	(invalid)		

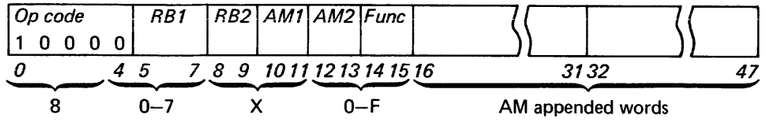


7	8–F				
		0	SECON	reg	Set Console Data Lights
		1	(invalid)		
		2	SEAKR	reg	Set Address Key Register (Note 1)
		3	(invalid)		
		4	(invalid)		
		5	(invalid)		
		6	(invalid)		
		7	(invalid)		
		8	CPCON	reg	Copy Console Data Buffer
		9	CPCL	reg	Copy Current Level
		A	CPAKR	reg	Copy Address Key Register (Note 2)
		B	(invalid)		
		C	(invalid)		
		D	(invalid)		
		E	(invalid)		
		F	(invalid)		

Notes:

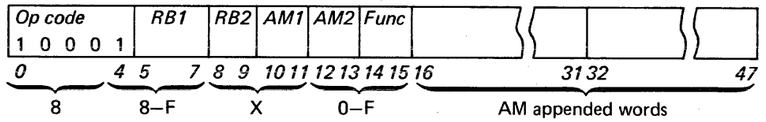
1. Use format with K-field.
   Extended mnemonics: SEISK, SEOTK, SEOOK.
2. Use format with K-field
   Extended mnemonics: CPISK, CPOTK, CPOOK.

8xxx



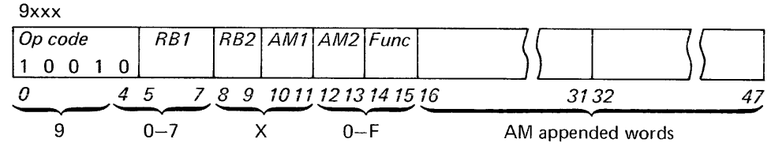
8	0–7	X	0,4 8,C	MVB	addr5,addr4	Move Byte
			1,5 9,D	OB	addr5,addr4	OR Byte
			1,5 9,D	SBTB	addr5,addr4	Set Bits Byte
			2,6 A,E	RBTB	addr5,addr4	Reset Bits Byte
			3,7 B,F	CB	addr5,addr4	Compare Byte

8xxx

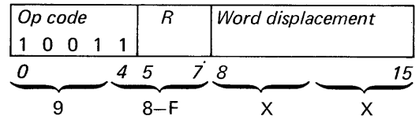


8	8–F	X	0,4 8,C	MVW	addr5,addr4	Move Word
			1,5 9,D	OW	addr5,addr4	OR Word
			1,5 9,D	SBTW	addr5,addr4	Set Bits Word
			2,6 A,E	RBTW	addr5,addr4	Reset Bits Word
			3,7 B,F	CW	addr5,addr4	Compare Word

9xxx

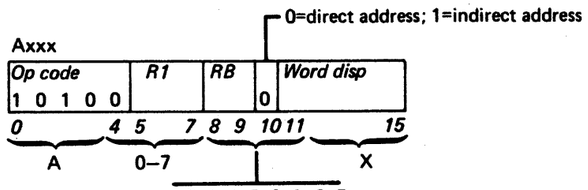


9	0–7	X	0,4 8,C	MVD	addr5,addr4	Move Doubleword
			1,5 9,D	OD	addr5,addr4	OR Doubleword
			1,5 9,D	SBTD	addr5,addr4	Set Bits Doubleword
			2,6 A,E	RBTD	addr5,addr4	Reset Bits Doubleword
			3,7 B,F	CD	addr5,addr4	Compare Doubleword

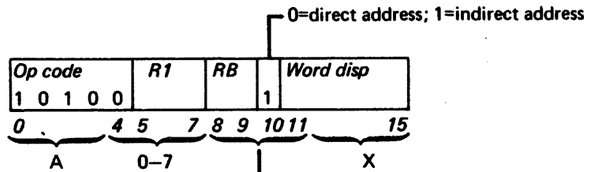


9	8–F	X	X	JAL	jdisp,reg	Jump and Link
				JAL	jaddr,reg	Jump and Link

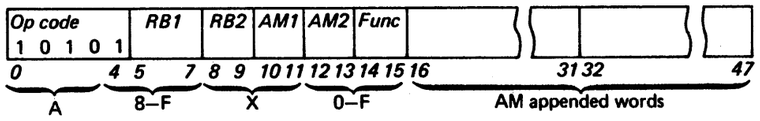
Axxx



A	0–7		X	MVWS	reg,shortaddr	Move Word Short

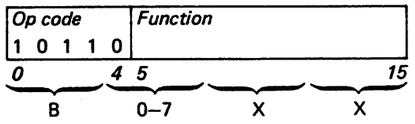


A	0–7		X	MVWS	reg,shortaddr*	Move Word Short



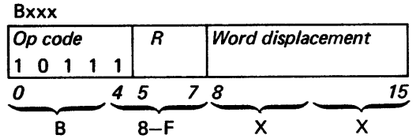
A	8–F	X	0,4 8,C	AW	addr5,addr4	Add Word
			1,5 9,D	SW	addr5,addr4	Subtract Word
			2,6 A,E	AD	addr5,addr4	Add Doubleword
			3,7 B,F	SD	addr5,addr4	Subtract Doubleword

Bxxx



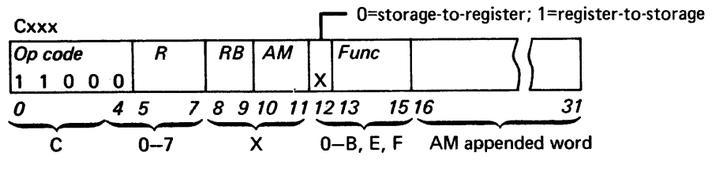
B	0–7	X	X	Illegal operation code (program-check condition)

Bxxx

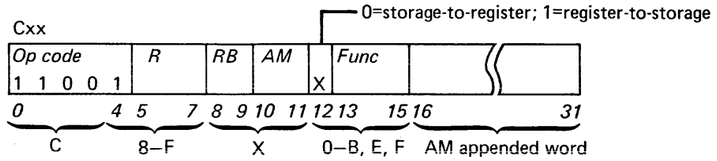


B	8–F	X	X	JCT	jdisp,reg	Jump on Count
				JCT	jaddr,reg	Jump on Count

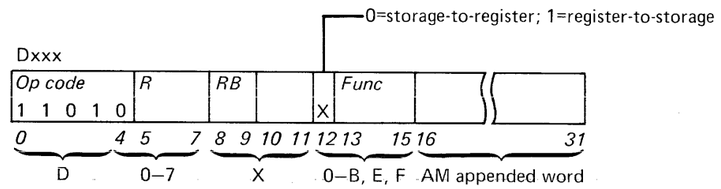
Cxxx



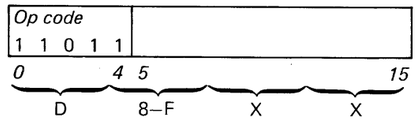
C	0–7	X	0	MVB	addr4,reg	Move Byte
			1	OB	addr4,reg	OR Byte
			1	SBTB	addr4,reg	Set Bits Byte
			2	RBTB	addr4,reg	Reset Bits Byte
			3	XB	addr4,reg	Exclusive OR byte
			4	CB	addr4,reg	Compare Byte
			5	MVBZ	addr4,reg	Move Byte and Zero
			6	AB	addr4,reg	Add Byte
			7	SB	addr4,reg	Subtract Byte
			8	MVB	reg,addr4	Move Byte
			9	OB	reg,addr4	OR Byte
			9	SBTB	reg,addr4	Set Bits Byte
			A	RBTB	reg,addr4	Reset Bits Byte
			B	XB	reg,addr4	Exclusive OR Byte
			E	AB	reg,addr4	Add Byte
			F	SB	reg,addr4	Subtract Byte



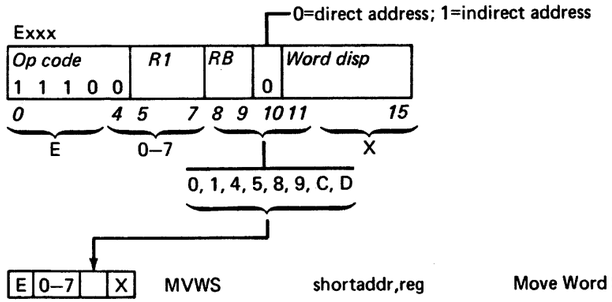
C	8–F	X				
		0	MVW	addr4,reg	Move Word
		1	OW	addr4,reg	OR Word
		1	SBTW	addr4,reg	Set Bits Word
		2	RBTW	addr4,reg	Reset Bits Word
		3	XW	addr4,reg	Exclusive OR Word
		4	CW	addr4,reg	Compare Word
		5	MVWZ	addr4,reg	Move Word and Zero
		6	AW	addr4,reg	Add Word
		7	SW	addr4,reg	Subtract Word
		8	MVW	reg,addr4	Move Word
		9	OW	reg,addr4	OR Word
		9	SBTW	reg,addr4	Set Bits Word
		A	RBTW	reg,addr4	Reset Bits Word
		B	XW	reg,addr4	Exclusive OR Word
		E	AW	reg,addr4	Add Word
		F	SW	reg,addr4	Subtract Word



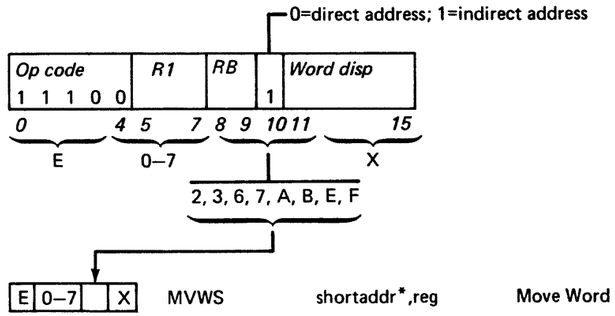
D	0–7	X			
		0	MVD	addr4,reg	Move Doubleword
		1	OD	addr4.reg	OR Doubleword
		1	SBTD	addr4,reg	Set Bits Doubleword
		2	RBTD	addr4,reg	Reset Bits Doubleword
		3	XD	addr4,reg	Exclusive OR Doubleword
		4	CD	addr4,reg	Compare Doubleword
		5	MVDZ	addr4,reg	Move Doubleword and Zero
		6	AD	addr4,reg	Add Doubleword
		7	SD	addr4,reg	Subtract Doubleword
		8	MVD	reg,addr4	Move Doubleword
		9	OD	reg,addr4	OR Doubleword
		9	SBTD	reg,addr4	Set Bits Doubleword
		A	RBTD	reg,addr4	Reset Bits Doubleword
		B	XD	reg,addr4	Exclusive OR Doubleword
		E	AD	reg,addr4	Add Doubleword
		F	SD	reg,addr4	Subtract Doubleword



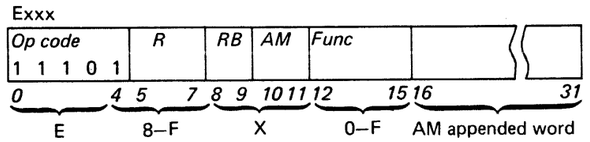
D	8–F	X	X	
				Illegal operation code (program-check condition)



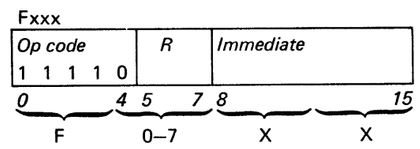
MVWS shortaddr,reg Move Word Short



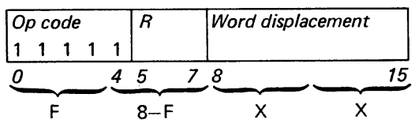
MVWS shortaddr*,reg Move Word Short



E	8–F	X				
			0	PSB	reg,addr4	Push Byte
			1	MB	addr4,reg	Multiply Byte
			2	DB	addr4,reg	Divide Byte
			3	PB	addr4,reg	Pop Byte
			4	PSW	reg,addr4	Push Word
			5	MW	addr4,reg	Multiply Word
			6	DW	addr4,reg	Divide Word
			7	PW	addr4,reg	Pop Word
			8	PSD	reg,addr4	Push Doubleword
			9	MD	addr4,reg	Multiply Doubleword
			A	DD	addr4,reg	Divide Doubleword
			B	PD	addr4,reg	Pop Doubleword
			C	(invalid)		
			D	(invalid)		
			E	(invalid)		
			F	(invalid)		



F	0–7	X	X			
				CBI	byte,reg	Compare Byte Immediate



F	8–F	X	X			
				BALS	(reg,jdisp)*	Branch and Link Short
				BALS	(reg)*	Branch and Link Short
				BALS	addr*	Branch and Link Short

Appendix B. List of Abbreviations

ac	alternating current
ACCA	asynchronous communications control attachment
AKR	address key register
ALU	arithmetic logic unit
amp	ampere
BBU	battery backup unit
BCC	block check character
bps	bits per second
BSC	binary synchronous communications
BSCA	binary synchronous communications attachment
CB	circuit breaker
CC	condition code
CIAR	current instruction address register
cm	centimeter
CPU	central processing unit
CRC	cyclic redundancy check
CS	cycle steal
CTS	clear to send
dc	direct current
DCB	device control block
DDB	device data block
DIAG	diagnose
DI	digital input
DIS	disable (instruction)
DO	digital output
DPC	direct program control
DSR	data set ready
DTR	data terminal ready
EA	enable (instruction)
EIA	Electronic Industry Association
EN	enable (instruction)
EOA	end of address
EOB	end of block
EOT	end of transmission
Ext	external
FET	field-effect transistor
FRU	field replaceable unit
GR	general register
I	interrupt
I-bit	interrupt bit
I/O	input/output
ID	identification
IDCB	immediate device control block
IIB	interrupt information byte
IPE	input parity error
IPL	initial program load
IPO	instant power off
ISA	invalid storage address
ISB	interrupt status byte
IT	inhibit trace
LCC	line control character
LEX	level exit (instruction)
LRC	longitudinal redundancy check
LSB	level status block
LSR	level status register
MCHK	machine check
MLD	machine logic diagram
MP	multipoint
ns	nanosecond
OPE	output parity error
OP1K	operand 1 address key
OP2K	operand 2 address key
PCI	program-controlled interrupt
PDE	permissive device end
PI	process interrupt
PSW	processor status word
PTTC	perforated tape transmission code
Reg	register
req	request
ROS	read only storage
RTS	request to send
SAR	storage address register
SDBI	storage data bus in
SDBO	storage data bus out
SDR	storage data register
SE	suppress exception
SEG	segmentation
serdes	serializer/deserializer
SIA	start instruction address
SIL	suppress incorrect length
STG	storage
SVC	supervisor call (instruction)
TCS	two-channel switch
TP	teleprocessing
TTL	transistor-transistor logic
VRC	vertical redundancy check

Index

IBM / Technical Newsletter

This Newsletter No.	SN34-0657
Date	March 20, 1981
Base Publication No.	SY34-0041-3
Previous Newsletters	None

IBM Series/1
4955 Processor and Processor Features
Theory Diagrams

This Technical Newsletter provides replacement pages for the subject publication. Pages to be inserted and/or removed are:

title page, ii	2-15 through 2-26
iii, iv	2-39, 2-40
1-1 through 1-4	2-60.1, 2-60.2 (added)
2-1, 2-2	3-1, 3-2
2-9 through 2-14	X-1 through X-4
2-14.1, 2-14.2 (added)	

A technical change to the text or to an illustration is indicated by a vertical line to the left of the change.

Summary of Amendments

This Technical Newsletter incorporates information about a new processor, the IBM Series/1 4955 Model F, a new power supply, and minor technical changes.

Note: Please file this cover letter at the back of the manual to provide a record of changes.

IBM Corporation, Information Development, Department 27T, P.O. Box 1328, Boca Raton, Florida 33432

Printed in U.S.A.

Cut Along Line

READER'S COMMENT FORM

IBM Series/1 4955 Processor and Processor Features
Theory Diagrams

SY34-0041-3

Your comments, please . . .

Your comments assist us in improving the usefulness of our publications; they are an important part of the input used in preparing updates to the publications. All comments and suggestions become the property of IBM.

Please do not use this form for technical questions about the system or for requests for additional publications; this only delays the response. Instead, direct your inquiries or requests to your IBM representative or to the IBM branch office serving your locality.

Corrections or clarifications needed:

Page Comment

What is your occupation? _____
Number of latest Technical Newsletter (if any) concerning this publication: _____
Please indicate your name and address in the space below if you wish a reply.

_____

_____

_____

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.)

READER'S COMMENT FORM

IBM Series/1 4955 Processor and Processor Features
Theory Diagrams

SY34-0041-3

Your comments, please . . .

Your comments assist us in improving the usefulness of our publications; they are an important part of the input used in preparing updates to the publications. All comments and suggestions become the property of IBM.

Please do not use this form for technical questions about the system or for requests for additional publications; this only delays the response. Instead, direct your inquiries or requests to your IBM representative or to the IBM branch office serving your locality.

Corrections or clarifications needed:

Page Comment

What is your occupation? _____
Number of latest Technical Newsletter (if any) concerning this publication: _____
Please indicate your name and address in the space below if you wish a reply.

_____

_____

_____

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.)

Your comments, please . . .

This manual is part of a library that serves as a reference source for IBM systems. Your comments on the other side of this form will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Fold

Fold

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:

IBM Corporation
Systems Publications, Dept 27T
P.O. Box 1328
Boca Raton, Florida 33432

Fold

Fold

IBM

SY34-0041-3
Printed in U.S.A.

Your comments, please . . .

This manual is part of a library that serves as a reference source for IBM systems. Your comments on the other side of this form will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Fold

Fold

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:

IBM Corporation
Systems Publications, Dept 27T
P.O. Box 1328
Boca Raton, Florida 33432

Fold

Fold

IBM

SY34-0041-3
Printed in U.S.A.

IBM

SY34-0041-3
Printed in U.S.A.