IBM Maintenance Library

3125 Processing Unit General System Information

Third Edition (October 1973)

This manual is a major revision of, and makes obsolete, SY33-1059-0. Changes are continually made to the information in this manual; any such changes will be reported in subsequent revisions or Technical Newsletters.

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

Forms for readers' comments are provided at the back of the manual. If the forms have been removed, comments may be addressed to IBM Laboratories, Product Publications, Dept. 3179, 703 Boeblingen/Wuertt., P.O. Box 210, Germany. Comments become the property of IBM.

Preface

This manual is a reference manual for all the 3125 Processing Unit maintenance library manuals. Readers of the manual should have a basic understanding of IBM system concepts. The manual supplements the System/370 Model 125 CE course and serves also as a recall aid. This manual and the other MLMs are not intended for self-education.

The manual describes the operating principle of System/370 Model 125 and defines the purpose of the various subprocessors of the system. Bus concepts are also described. General maintenance information is given, such as physical locations and error handling techniques.

All the supporting 3125 maintenance library manuals are listed in Chapter 1, where the structure of these manuals is also explained. Definitions of the symbology and simplifications used throughout the associated manuals are given in Chapter 5. The common abbreviations and terms can be found in Chapter 7.

Associated Publications

System Library Manuals

IBM System/360 Principles of Operation, GA22-6821.
IBM System/370 Principles of Operation, GA22-7000.

Maintenance Library Manuals

IBM System/360 System/370 Logic Blocks, Automated Logic Diagrams, FEALD, ALD, and Component Circuits, Theory of Operation Manual, SY22-2798.
IBM MST Packaging, Tools, Wiring Change Procedure, Theory of Operation Manual, SY22-6739.

Details of other associated maintenance library manuals are given on page 1-040.

Contents

Safety

Personal Safety

Personal safety cannot be over-emphasized; it is a vital part of customer engineering. To ensure your safety and that of co-workers, always observe the safety precautions given during your safety training and adhere to the following:

General Safety Practices

Observe the general safety practices and the procedure for performing artificial respiration that are outlined in *CE Safety Practices* card, Order No. S229-1264 (shown here).

Grounding

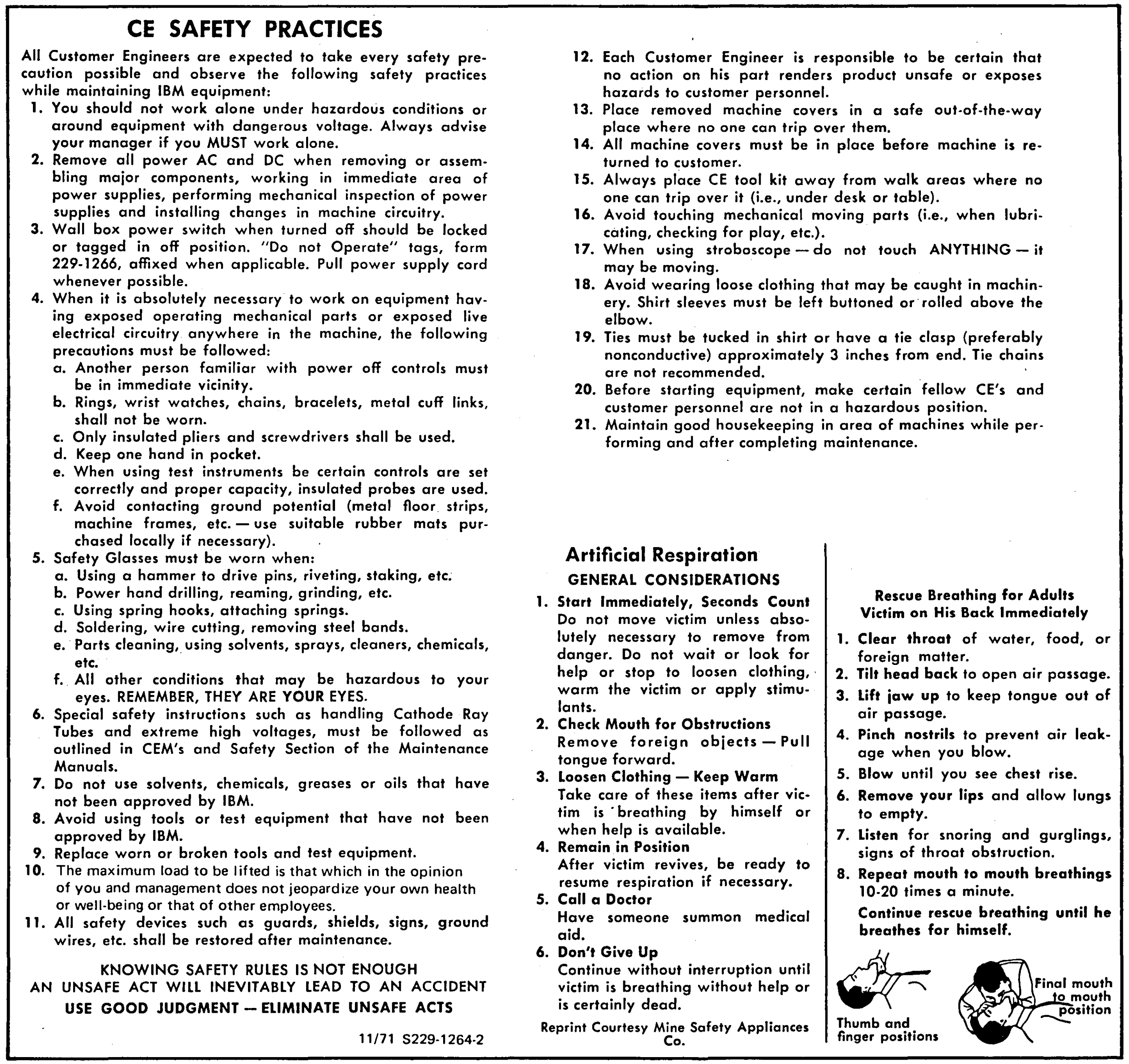
Ground current may reach dangerous levels. Never operate the system with the grounding conductor removed.

Line-Powered Equipment

Ground all line-powered test equipment through the third-wire grounding conductor in the power cord of the machine being tested.

Machine Warning Labels

Heed the warning labels placed in hazardous areas of the machines.



CE SAFETY PRACTICES

All Customer Engineers are expected to take every safety precaution possible and observe the following safety practices while maintaining IBM equipment:

1. You should not work alone under hazardous conditions or around equipment with dangerous voltage. Always advise your manager if you MUST work alone.
2. Remove all power AC and DC when removing or assembling major components, working in immediate area of power supplies, performing mechanical inspection of power supplies and installing changes in machine circuitry.
3. Wall box power switch when turned off should be locked or tagged in off position. "Do not Operate" tags, form 229-1266, affixed when applicable. Pull power supply cord whenever possible.
4. When it is absolutely necessary to work on equipment having exposed operating mechanical parts or exposed live electrical circuitry anywhere in the machine, the following precautions must be followed:
   a. Another person familiar with power off controls must be in immediate vicinity.
   b. Rings, wrist watches, chains, bracelets, metal cuff links, shall not be worn.
   c. Only insulated pliers and screwdrivers shall be used.
   d. Keep one hand in pocket.
   e. When using test instruments be certain controls are set correctly and proper capacity, insulated probes are used.
   f. Avoid contacting ground potential (metal floor strips, machine frames, etc. — use suitable rubber mats purchased locally if necessary).
5. Safety Glasses must be worn when:
   a. Using a hammer to drive pins, riveting, staking, etc.
   b. Power hand drilling, reaming, grinding, etc.
   c. Using spring hooks, attaching springs.
   d. Soldering, wire cutting, removing steel bands.
   e. Parts cleaning, using solvents, sprays, cleaners, chemicals, etc.
   f. All other conditions that may be hazardous to your eyes. REMEMBER, THEY ARE YOUR EYES.
6. Special safety instructions such as handling Cathode Ray Tubes and extreme high voltages, must be followed as outlined in CEM's and Safety Section of the Maintenance Manuals.
7. Do not use solvents, chemicals, greases or oils that have not been approved by IBM.
8. Avoid using tools or test equipment that have not been approved by IBM.
9. Replace worn or broken tools and test equipment.
10. The maximum load to be lifted is that which in the opinion of you and management does not jeopardize your own health or well-being or that of other employees.
11. All safety devices such as guards, shields, signs, ground wires, etc. shall be restored after maintenance.
12. Each Customer Engineer is responsible to be certain that no action on his part renders product unsafe or exposes hazards to customer personnel.
13. Place removed machine covers in a safe out-of-the-way place where no one can trip over them.
14. All machine covers must be in place before machine is returned to customer.
15. Always place CE tool kit away from walk areas where no one can trip over it (i.e., under desk or table).
16. Avoid touching mechanical moving parts (i.e., when lubricating, checking for play, etc.).
17. When using stroboscope — do not touch ANYTHING — it may be moving.
18. Avoid wearing loose clothing that may be caught in machinery. Shirt sleeves must be left buttoned or rolled above the elbow.
19. Ties must be tucked in shirt or have a tie clasp (preferably nonconductive) approximately 3 inches from end. Tie chains are not recommended.
20. Before starting equipment, make certain fellow CE's and customer personnel are not in a hazardous position.
21. Maintain good housekeeping in area of machines while performing and after completing maintenance.

KNOWING SAFETY RULES IS NOT ENOUGH
AN UNSAFE ACT WILL INEVITABLY LEAD TO AN ACCIDENT
USE GOOD JUDGMENT — ELIMINATE UNSAFE ACTS

11/71 S229-1264-2

Artificial Respiration

GENERAL CONSIDERATIONS

1. **Start Immediately, Seconds Count**
   Do not move victim unless absolutely necessary to remove from danger. Do not wait or look for help or stop to loosen clothing, warm the victim or apply stimulants.
2. **Check Mouth for Obstructions**
   Remove foreign objects — Pull tongue forward.
3. **Loosen Clothing — Keep Warm**
   Take care of these items after victim is breathing by himself or when help is available.
4. **Remain in Position**
   After victim revives, be ready to resume respiration if necessary.
5. **Call a Doctor**
   Have someone summon medical aid.
6. **Don't Give Up**
   Continue without interruption until victim is breathing without help or is certainly dead.

Reprint Courtesy Mine Safety Appliances Co.

Rescue Breathing for Adults
Victim on His Back Immediately

1. **Clear throat** of water, food, or foreign matter.
2. **Tilt head back** to open air passage.
3. **Lift jaw up** to keep tongue out of air passage.
4. **Pinch nostrils** to prevent air leakage when you blow.
5. **Blow** until you see chest rise.
6. **Remove your lips** and allow lungs to empty.
7. **Listen** for snoring and gurglings, signs of throat obstruction.
8. **Repeat mouth to mouth breathings** 10-20 times a minute.

Continue rescue breathing until he breathes for himself.

Chapter 1. Introduction

System Description

IBM System/370 Model 125 has a decentralized design and consists of several independent subprocessors grouped around the main storage. A specialized unit therefore exists for each main system function, and there is little interference between subprocessors within the system.

The four types of subprocessors located in the IBM 3125 Processing Unit are:

- Main Storage Controller (MSC)
- Instruction Processing Unit (IPU)
- Service Processor (SVP)
- Input/Output Processors (IOPs).

Each subprocessor (except MSC) has its own storage, work registers, and an arithmetic and logic unit (ALU), and is controlled by its own microprogram and timing device.

Main Storage

- Has a non-destructive readout.
- Storage cycle 480 nanoseconds (ns) per halfword.
- Main storage size: 128K max.

All storage has automatic correction of single-bit errors in a halfword, and detection of double-bit errors in units of data longer than a halfword.

Further details are given in *IBM 3125 Processing Unit, Main Storage,* SY33-1066.

Main Storage Controller

The main storage controller (MSC) regulates access to main storage. The MSC contains only hardware; it does not have a microprogram.

Subprocessors may request access at any time. At regular intervals, the MSC examines requests and accepts the one that has the highest priority.

Further details are given in *IBM 3125 Processing Unit, Main Storage Controller,* SY33-1061.

Main Storage Addressing

Main storage is addressed through address registers in the MSC local storage. Each subprocessor has two or more address registers. On a request from a subprocessor, the MSC uses the contents of the assigned register to address main storage.

During access, the MSC updates the main storage address, and later returns it to the original local storage register. Thus, a subprocessor provides only the start address of the data field.

Instruction Processing Unit

- Fetches and executes program instructions.
- Performs arithmetical and logical instructions.
- Analyzes I/O instructions so that the IOP can be selected.
- Calculates addresses, sets condition codes, updates program status words (PSWs), and handles interrupts.

The instruction processing unit (IPU) is a microprogram-controlled subprocessor with a data flow and an ALU one halfword wide. The microprogram is loaded into the IPU control storage by the service processor. A microinstruction is processed in 480 ns.

Information is processed through two data registers that may be loaded from the MSC, the IPU local storage, the shift unit, external data, or from immediate data in a microinstruction. The data is processed in the ALU, and the results are sent back to main storage and/or IPU local storage.

Further details are given in *IBM 3125 Processing Unit, Instruction Processing Unit,* SY33-1062.

Service Processor

- Loads microprograms into all subprocessors (including itself).
- Provides the link between the operator and the system.
- Reloads microprograms upon request from any IOP, logs error conditions and reads them out later for program analysis.

The service processor (SVP) is a microprogram-controlled subprocessor with a data flow and an ALU one byte wide. A small *bootstrap* program in read-only storage allows the SVP to load its own main microprogram from the console file. This file also stores the microprograms of the other subprocessors, and provides space for error logging.

The SVP contains local storage for handling data, and the circuitry for operating the console disk file and the operator console (keyboard and display unit).

Further details are given in *IBM 3125 Processing Unit, Service Processor Subsystem,* SY33-1065.

Input/Output Processors

- Execute I/O commands.
- Supervise data transfer between the addressed I/O device and the MSC.

An input/output processor (IOP) is a subprocessor having its own microprogram, an ALU, internal and external work registers, and a clock. The IOP operates with a 450 ns cycle that is synchronous to the remainder of the system.

All of the IOPs have the same design. To meet the special needs of a connected I/O device, the IOPs are supplemented by one or more *front ends* that are compatible with the I/O interface, over which signals pass to and from the device.

Special microprograms are loaded to service attached I/O devices, and several microprograms can run concurrently in one IOP in *time slicing* mode. Each identical IOP thus performs a different task, representing an attachment, adapter, or the multiplexer channel.

Further details are given in *IBM 3125 Processing Unit, Input/Output Processor,* SY33-1063.

Compatibility

- Compatibility with IBM System/360.
- Upward compatibility with IBM System/370.
- Features available for compatibility with IBM System/360 Model 20 and IBM 1401, 1440, and 1460 Data Processing Systems.
- Features available for compatibility with Disk Operating System (DOS) and IBM 2311 Disk Storage Drive, Model 1.

For more information on compatibility see *IBM System/370 Model 125 Functional Characteristics,* GA33-1510.

Front Ends

Front ends are to'be considered as individual supplements to the standard IOPs. They serve as matching links between IOPs and I/O devices and do not have their own microprogram.

Further details are given in *IBM 3125 Processing Unit, xxxx Attachment, Front End.* Order numbers for these manuals are listed on page 1-040.

Magnetic Tape Adapter

The magnetic tape adapter is to be considered as a link between the 3125 Processing Unit and the IBM 3410 Magnetic Tape Unit and IBM 3411 Magnetic Tape Unit and Control. The magnetic tape adapter does not have its own microprogram and therefore it is controlled by either the IPU microprogram or by the tape control unit microprogram.

Further details are given in *IBM 3125 Processing Unit, Magnetic Tape Adapter,* SY33-1064.

Starting the System

1. Press POWER ON. The key lights up red and remains red until the power-on sequence is completed, when it changes to white. During the power-on sequence, the display unit screen remains dark and the keyboard is locked.

2. At the completion of the power-on sequence, the display unit gives the following message:

   INITIAL MICROPROGRAM LOAD IN PROGRESS

   This message indicates that the various subprocessors are now being loaded with their individual microprogram routines.

   During the load operation, different messages appear on the display unit. These messages indicate which subprocessor is checked out, loaded, and operational. This is a fully automatic procedure.

3. At the completion of the load operation, the display unit gives the following message:

   PROGRAM LOAD DISPLAY

   This message indicates that the system is now in manual mode and is ready to receive the address of the I/O device from which it is intended to load the job program.

4. Key in the I/O device address. Observe each step on the display unit.

5. If incorrect data is keyed in, this may be corrected by moving the cursor either to the left or to the right until the position to be corrected is reached, and then pressing the desired key. After correction the normal key-in operation may be continued.

6. At the completion of the key-in operation, ensure that the loading device is ready.

7. When the loading device is ready, press ENTER. This transfers the keyed-in data to the SVP registers where it is analyzed; the loading device is then addressed and the loading operation begins. During this program load phase the display unit gives the following message:

   PROGRAM LOAD IN PROGRESS

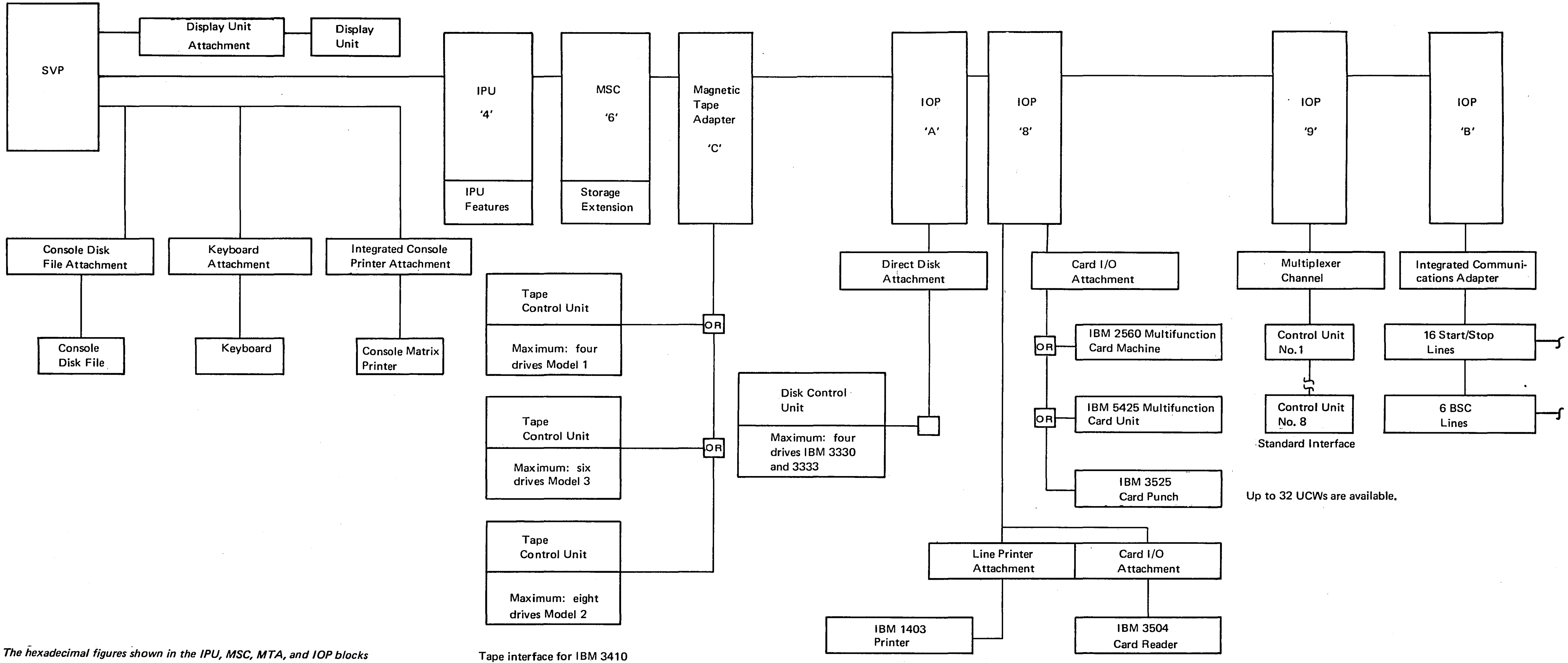
8. When the program is loaded, the IPU initiates program execution. The current PSW is loaded and analyzed; specific bits give the necessary information for program execution and point to the address of the next sequential instruction.

Manual Operations

- Manual operations allow the operator to interrupt a running program at any time so that the system status can be checked.
- Pressing MODE SELECT while the system is running does not stop program execution, but all possible modes are indicated on the display unit.
- After selection and keying-in of the desired mode, the system stops and the display unit indicates a message that is pertinent to the desired mode.
- The following manual operations are available, dependent upon the selected mode:
  System reset
  Address comparison
  Program load
  Check control
  Timer enable/disable
  Alter/display
  Instruction step
  PSW restart
  Log analysis
  Maintenance (CE use only).

Full operating instructions are given in *IBM System/370 Model 125 Operating Procedures,* GA33-1509.

Configuration of Maximum-Equipped System



Up to 32 UCWs are available.

The hexadecimal figures shown in the IPU, MSC, MTA, and IOP blocks represent those addresses that are used by the SVP (via the high-order part of the SVP address bus) for
- *Logging*
- *Sense operations*
- *Control operations.*

Tape interface for IBM 3410 Magnetic Tape Unit and IBM 3411 Magnetic Tape Unit and Control.

System Documentation

The following documentation is shipped with each system:

- Maintenance automated logic diagrams (MALDs).
- Microprogram listings.
- Machine history card.
- Maintenance library manuals.

MALDs

MALDs are specially prepared automated logic diagrams for maintenance purposes. The MALDs are simplified, but still show the circuitry of a unit per logic block.

The MALDs also show line levels and pin designations. The pin addressing scheme is shown on page 4-040 of this manual.

MALD logic block information and a description of card size and card identification is given on page 4-050.

Microprogram Listings

Details about microprogram listings are given in Appendix A of this manual. Details on microinstructions are given in *IBM 3125 Processing Unit, Microinstructions,* SY33-1058.

Machine History Card

The machine history card is provided to keep a record of the machine life history. All engineering changes (ECs) and features that are installed (whether during manufacture or in the field) must be recorded.

The machine history card is a very important document and must be kept up-to-date at all times.

Maintenance Library Manuals

The maintenance library manuals are intended to provide maintenance information and give sufficient information to enable the CE to recall what he previously learned on the System/370 Model 125 CE course. The manuals are *not* intended for self-education.

Each manual starts with an explanation of the principles of each subprocessor or adapter and goes deeper and deeper into details, step by step.

Operation-oriented logic diagrams and microprogram flowcharts are provided in each manual to serve as a guide through the ALDs and the microprogram listings. They are intended to explain machine functions and operations.

The operation-oriented logic diagrams show the circuitry of individual units. The symbology used is explained on pages 5-010 through 5-030 of this manual. The microprogram flowcharts show the manner in which machine functions are controlled and executed. Flowchart symbology is explained on page 5-060.

Structure of Manuals

Most of the Model 125 maintenance library manuals follow the same format.

Chapter 1. Introduction
System data flow
Functional principles
General statements
Physical locations

Chapter 2. Principles of Operations
Data and control overview
Operations
Microprogram summary

Chapter 3. Operational Details
Instructions and formats
Commands
Basic timings
Microprogram
Operation-oriented logic

Chapter 4. Functional Units
Descriptions
Timing

Chapter 5. Error Conditions
Error types
Checks

Chapter 6. Maintenance Information
Adjustments

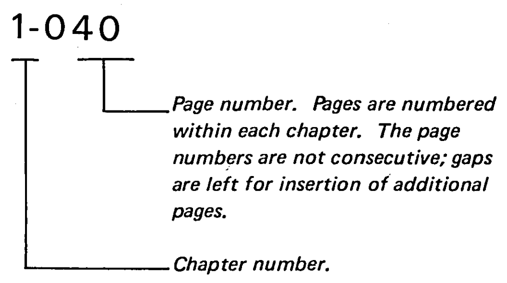
Chapter 7. Reference Information
Abbreviations and Glossary
Other reference data

Appendixes

Index

All lower pages within a chapter have a tab (or thumb index) which provides quick and easy entry to the required chapter.

The page numbering system is as follows:


1-040
Page number. Pages are numbered within each chapter. The page numbers are not consecutive; gaps are left for insertion of additional pages.
Chapter number.

The index is intended to be used as the *main* entry point to particular pages.

List of Manuals

IBM 3125 Processing Unit, Microinstructions, SY33-1058.

IBM 3125 Processing Unit, Power Supplies, SY33-1060.

IBM 3125 Processing Unit, Main Storage Controller, SY33-1061.

IBM 3125 Processing Unit, Instruction Processing Unit, SY33-1062.

IBM 3125 Processing Unit, Input/Output Processor, SY33-1063.

IBM 3125 Processing Unit, Magnetic Tape Adapter, SY33-1064.

IBM 3125 Processing Unit, Service Processor Subsystem, SY33-1065.
Part 1: Service processor
Part 2: Console disk file
Part 3: Display unit and keyboard.

IBM 3125 Processing Unit, Main Storage, SY33-1066.

IBM 3125 Processing Unit, Multiplexer Channel, SY33-1067.

IBM 3125 Processing Unit, 2560 Attachment, Front End, SY33-1068.

IBM 3125 Processing Unit, 3525 Attachment, Front End, SY33-1070.

IBM 3125 Processing Unit, 3504 Attachment, Front End, SY33-1071.

IBM 3125 Processing Unit, 1403 Attachment, Front End, SY33-1072.

IBM 3125 Processing Unit, Direct Disk Attachment, SY33-1073.

IBM 3125 Processing Unit, Integrated Console Printer Attachment, SY33-1074.

IBM 3125 Processing Unit, Integrated Communications Adapter, Part B/M 1876075.

IBM 3125 Processing Unit, Installation Instructions, Part 4014001.

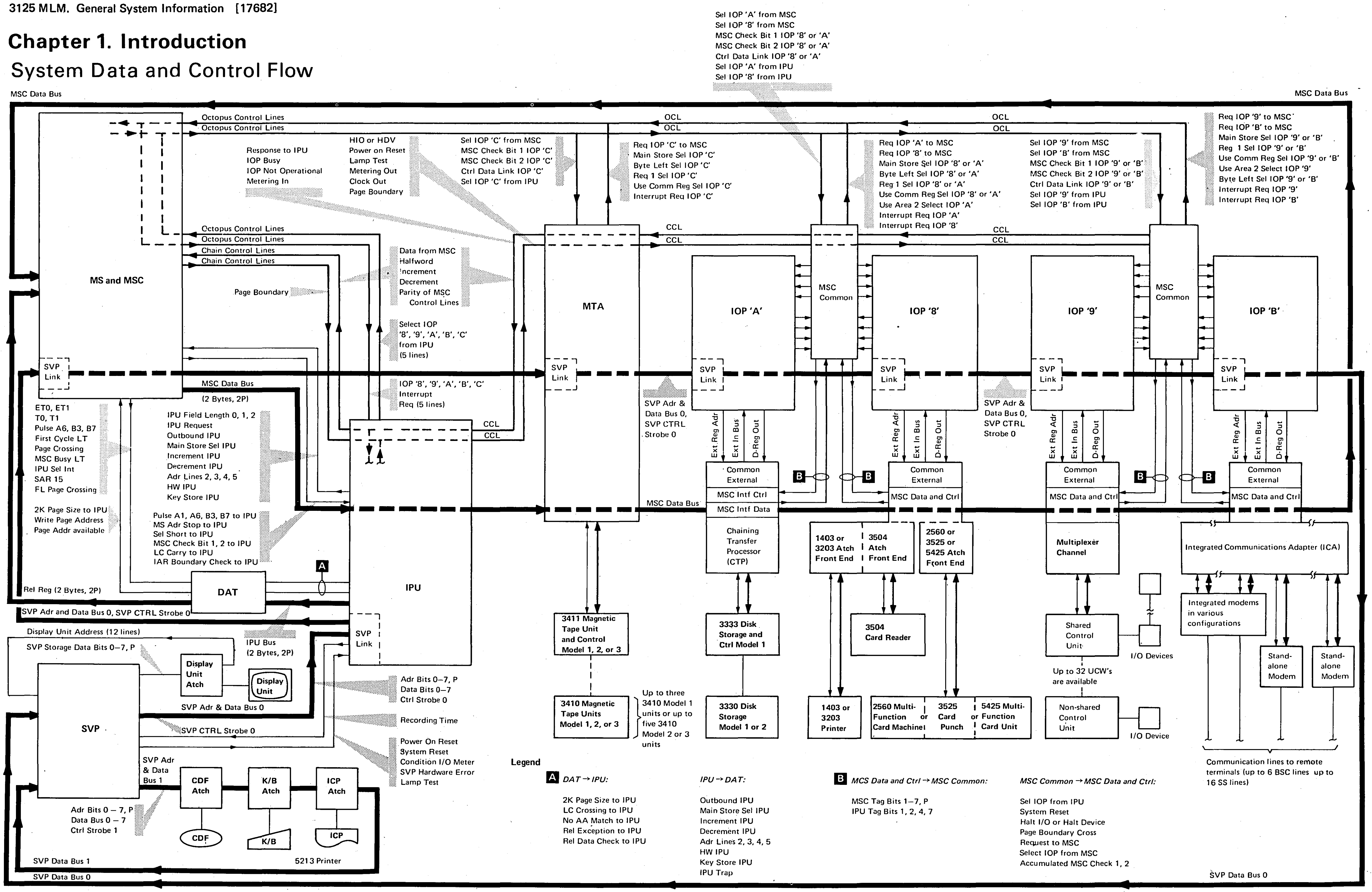
IBM 3125 Central Test Manual. Contains pages appropriate to the individual 3125 Processing Unit and provides detailed information about:

- Logging
- Log display
- Log analysis
- Diagnostic tests
- Application of diagnostics
- Diagnostic techniques.

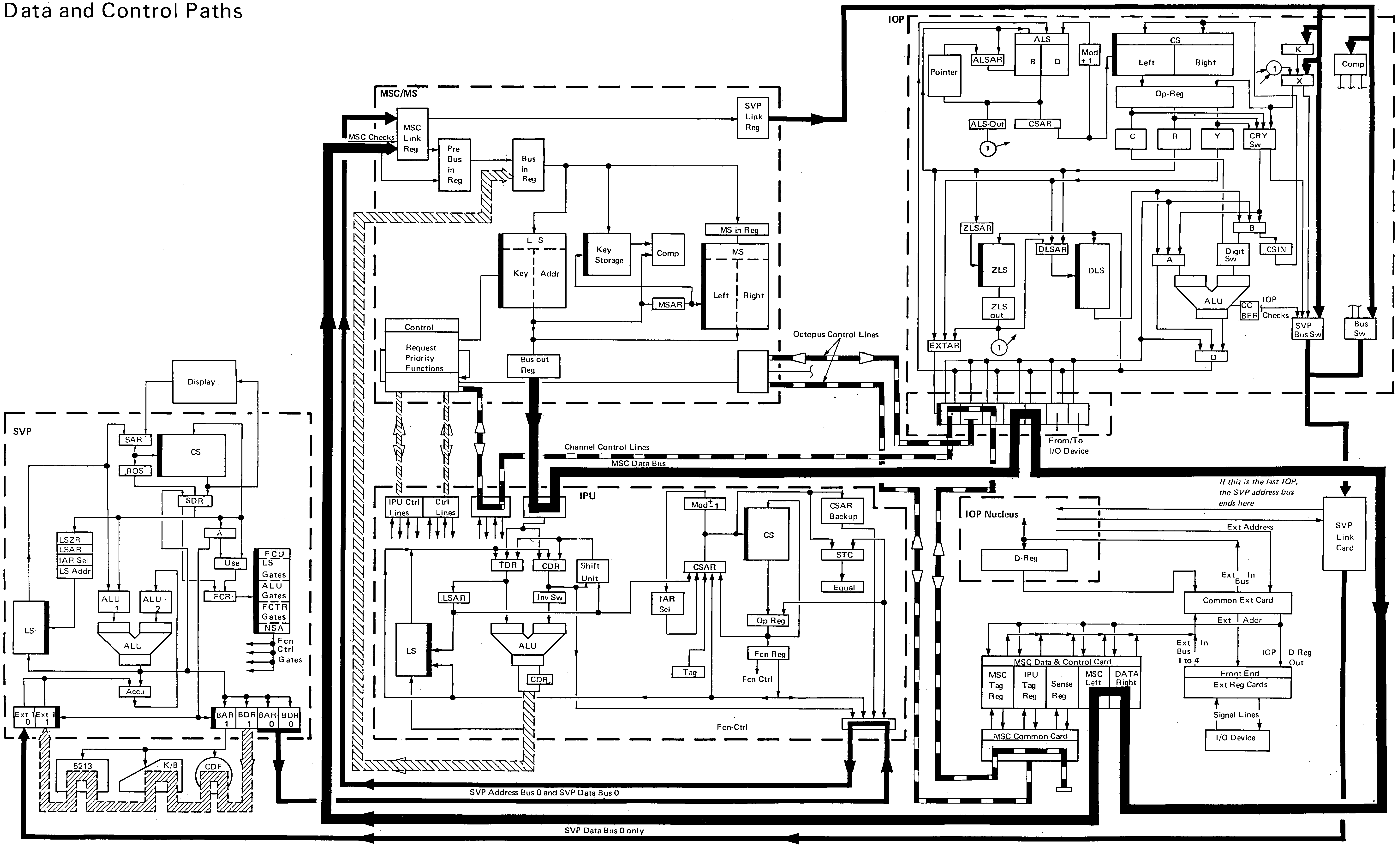
This page is intentionally left blank.

Chapter 1. Introduction

System Data and Control Flow



Data and Control Paths



Concept and Mission of Subprocessors

- Each subprocessor (that is, SVP, IPU, MSC, and IOP) has the following:
  Storage
  ALU (except the MSC)
  Work registers
  Clock
  Microprogram (except the MSC).
- All storages have automatic single-bit error correction within one halfword.
- Communication between subprocessors is via the system internal bus system, which consists of MSC Data Bus, SVP Data and Address Bus, Chain Control Lines, and Octopus Control Lines.
- Communication between system and operator is via the operator station, which consists of a display unit, a keyboard, and a console-printer (which is an optional feature).

SVP (Service Processor)

- The SVP is considered to be a general-purpose processor.
- Cycle time is 450 nanoseconds.
- Internal data flow is one byte wide.
- The SVP normally controls the following:
  Microprogram loading
  Manual operations
  Keyboard and CRT as I/O Processor control panel.
- In the event of an error, the SVP controls the following:
  Logging of error conditions
  Microdiagnostic Function Tests.

After power on, a system reset is performed under control of the *picoprogram,* which resides in a read-only storage. This picoprogram switches to a bootstrap routine, which loads the first record from the console disk file into SVP storage. This first record is the microprogram that initializes the SVP (LS registers, etc.). The next part of the microprogram fetched from the flexible magnetic disk (diskette) is used to load and to start all other microprograms.

Manual system control is also handled by the SVP. The conventional operator console is replaced by the operator station (consisting of a display unit, keyboard, and console printer).

During system operation, the SVP monitors all subprocessors for error conditions and logs the error information. If an operation cannot be performed successfully, the SVP requests a machine check interruption. In the event of an error, the logged error information is analyzed under control of SVP microprogram and the results are displayed on the screen.

Microdiagnostic function tests are also loaded, under control of the SVP, into the SVP or other subprocessor. The microdiagnostic function tests that are stored in the SVP storage are also executed by the SVP.

Programs that are necessary for operating and maintaining the system are stored on two magnetic disks:
System diskette
Service diskette.

The system diskette contains the following:
Subprocessor and adapter microprograms
Inline tests (ILTs)
Error log and log analysis programs (including log area).

The service diskette contains the following:
Automatic system checkout program (ASCP)
Microdiagnostic function tests (MFTs).

Refer to *IBM 3125 Processing Unit, Service Processor Subsystem,* SY33-1065.

IPU (Instruction Processing Unit)

- The IPU controls the execution of customer programs.
- Cycle time is 480 nanoseconds.
- Internal data flow is two bytes wide.

Under control of PSWs and the IPU microprogram, machine language instructions are fetched, one at a time, from the MSC. After an instruction has been analyzed (that is, its op code and addresses), associated operands are fetched (either from IPU storage or the MSC), the instruction is executed, and the result (if applicable) is returned to IPU storage or to the MSC, as specified. Thus, arithmetic and logic instructions are executed by, and in, the IPU.

I/O instructions are analyzed up to the point where an IOP can be selected. After the IOP is selected, all necessary information is transferred to that IOP, whereby
1. The IOP starts running under control of its own microprogram.
2. The IPU is released to continue the execution of the customer program.

The IPU also handles the interrupts. After completion of a data transfer, the IOP activates its 'interrupt request' line. If request is accepted (that is, none waiting with higher priority and not masked off), the IPU fetches necessary information from the MSC, analyzes this information and switches to the interrupt handling routine by means of exchanging PSWs. PSWs are used to control the execution of the customer programs. The major parts of each PSW are as follows:

System Mask — Defines which interrupts are allowed and which interrupts are "masked off".

Key — Is compared with the key that is assigned to a storage area. If the keys match, main storage operations are allowed. If the keys do not match, access to that storage area is not permitted and a 'protection check' is raised.

Interrupt Code — Contains the address of the instruction that is currently being executed, or terminated.

Instruction Address — Indicates the address of the next instruction that is to be executed.

MSC (Main Storage Controller)

- The MSC represents the system storage unit; the maximum main storage size is 128K bytes.
- The storage contains customer programs and data.
- Cycle time for one MSC request is 480 nanoseconds.
- Internal data flow is two bytes wide.
- The MSC is not microprogram controlled.
- Requests to fetch or store information are handled according to predetermined priorities.
- Storage "fetch" (or "read") operations are called *outbound* operations.
- Storage "store" (or "write") operations are called *inbound* operations.

Main storage positions are addressed by the contents of the MSC-local-storage registers. Two or more MSC-LS registers are assigned to each subprocessor. Local storage is not available to programmers.

If an interrupt request is accepted (that is, none present with higher priority and not masked off), the contents of the assigned local storage register are used to address main storage. This address is then updated (that is, incremented or decremented, as specified by the chain control lines) during main storage access, and is returned to the local storage register.

Thus, for storage operations, only a start address is necessary. Each subprocessor must ensure that the correct address is available in local storage at the correct time.

The MSC notifies the requesting subprocessor of the following information:

1. That the requesting subprocessor has been selected.
2. The validity of the transferred data.
3. Whether a protected storage area is being addressed.
4. Whether an area is being addressed outside the maximum storage size.

To ensure that the correct number of bytes is handled, a byte count is specified, together with the start address. This byte count is stored in the subprocessors and is reduced by either one or two (according to whether a byte or a halfword is being transferred) under control of the subprocessor microprogram.

When the byte count reaches zero, the transfer operation between the MSC and the subprocessor is terminated.

IOP (Input/Output Processor)

- The IOP links one or more I/O devices to the SVP, the IPU, and the MSC.
- The IOP executes I/O commands and supervises the data transfer between I/O devices and the MSC.
- Cycle time is 450 nanoseconds.
- Internal data flow is one byte wide.

To cater for the special requirements of I/O devices, special front ends are provided. These front ends are located between the I/O device(s) and the controlling IOP. All signals to and from the I/O device have to pass through the front end, and are converted if necessary.

Each IOP is controlled by its own device-oriented microprograms. The IOPs therefore operate independently and thus save the IPU from directly serving the I/O devices. These microprograms (or their routines) may run in either *normal mode* or *time slice mode.*

Normal mode is used if only one microprogram (or routine) is to be executed. Time slice mode is used if several microprograms (or routines) are to be executed simultaneously.

Note: Time slice mode, in this context, means that the execution of one or more microinstructions of one routine is followed by the execution of one or more microinstructions of another routine in a predetermined sequence.

The execution of one or more microprogram routines is controlled by *index words.* Index words consist of a *pointer* and a *link.* The pointer always points to the microinstruction that is to be executed next; the link leads to the next index word.

This chain of index words (called the *time slice period*) may be altered during the execution of microprogram routines. This alteration is done in a predetermined way according to time requirements, which depend upon the I/O device types being used.

The alteration of the time slice period may be performed in two ways:

1. By using special microinstructions that allow the index words to be overwritten in their storage.
2. By using special *trap bits* that are ORed to the link portion of appropriate index words in the IOP circuitry.

The last index word in a chain always leads back to the first index word. This ensures that the microinstructions and routines are executed in a proper and predetermined sequence.

The IOP idles in its basic loop waiting for select or service requests. This basic loop mainly consists of a number of branch instructions.

After selection of an IOP by the IPU, the IOP fetches, under control of its own microprogram, all necessary information that is required to execute an I/O instruction. Subsequently, the IOP releases the IPU for further processing.

For data transfers, an IOP front end requests service from the MSC. If the request is accepted, the MSC selects the IOP and the data transfer takes place. After completion of the data transfer, inactivation of the 'MSC select' line causes the IOP to continue in its microprogram and prepares the next data transfer.

After the last byte has been transferred, the IOP activates its 'interrupt request' line to the IPU. If this request is accepted, the IPU selects the IOP, which causes the IOP to store the following into the MSC:

IOP address
I/O device address
CSW information
Condition code.

The IOP then signals to the IPU that the information transfer is completed. This causes the IPU to continue in its own program (normally the interrupt handling routine); the IOP returns to its basic loop and waits for the next select or service request.

Front Ends

An IOP may be considered as a common and standardized part of an I/O attachment. Front ends are those parts of the attachment that fulfil the individual needs of I/O devices, that do not have their own microprogram, to be attached to an IOP.

The front ends are able to:

1. Contain additional hardware to perform the functions of the I/O device.
2. Perform those functions that exceed the capacity of the IOP.

Magnetic Tape Adapter

The magnetic tape adapter (MTA), which allows the IBM 3410/3411 magnetic tape subsystem to be attached to the IBM System/370 Model 125, operates under the control of either the IPU or the TCU microprogram. The MTA performs two types of operation:

1. Read-type operation, which transfers data from the addressed TCU to the MSC.
2. Write-type operation, which transfers data from the MSC to the addressed TCU.

Opposite to the front ends the SVP has access to the MTA and may perform SENS or CTRL operations.

Bus Concept

Interconnection between subprocessors and adapters is accomplished by an internal bus system that consists of the following:
SVP data and address bus
MSC data bus
Chain control lines
Octopus control lines
SVP–IPU control lines
IPU–MSC control lines.

SVP Data and Address Bus

- The SVP data buses 0 and 1 and the SVP address buses 0 and 1 are all one byte wide. They do not have a parity bit.
- The purpose of the data and address buses is as follows:

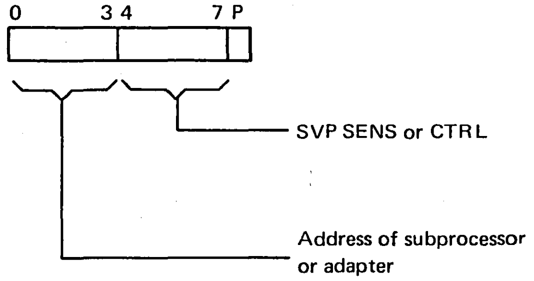
SVP Address Bus 0: Addresses all subprocessors and adapters. The bus ends at the last subprocessor. (Addresses are shown on page 1-030.)

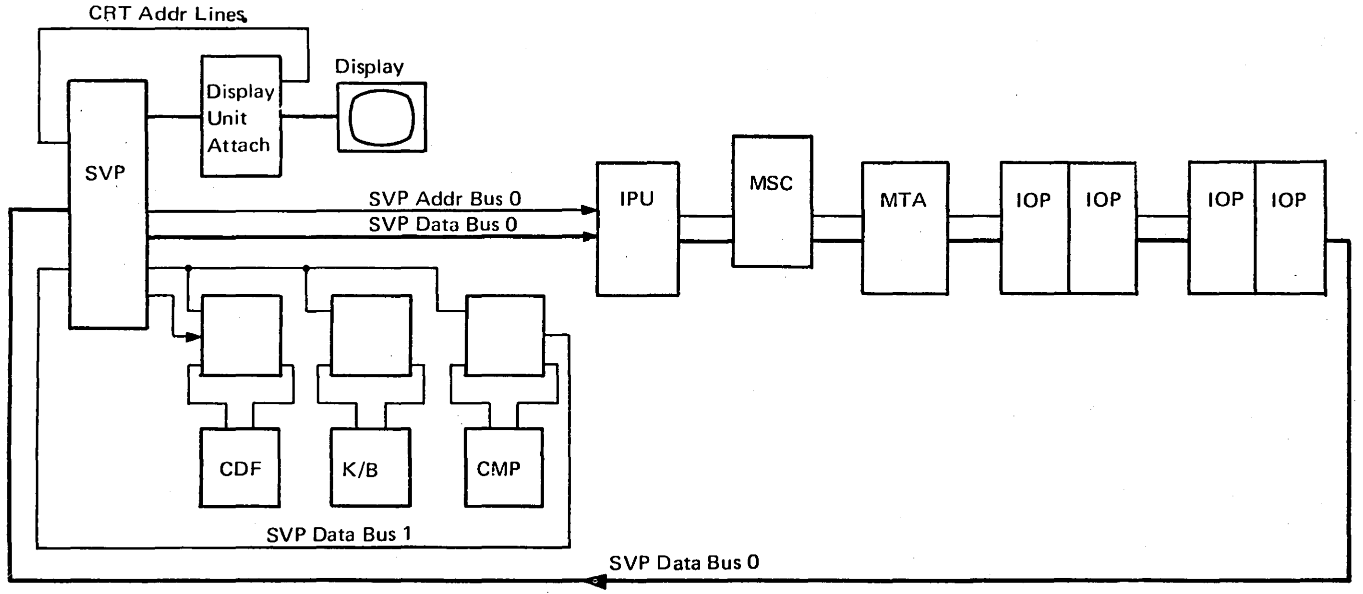
SVP Address Bus 1: Addresses all SVP console units (for example, console disk file, keyboard, console printer).

SVP Data Bus 0: Exchanges information between the SVP and the addressed subprocessor or adapter.

SVP Data Bus 1: Exchanges information between the SVP and its addressed SVP control unit.

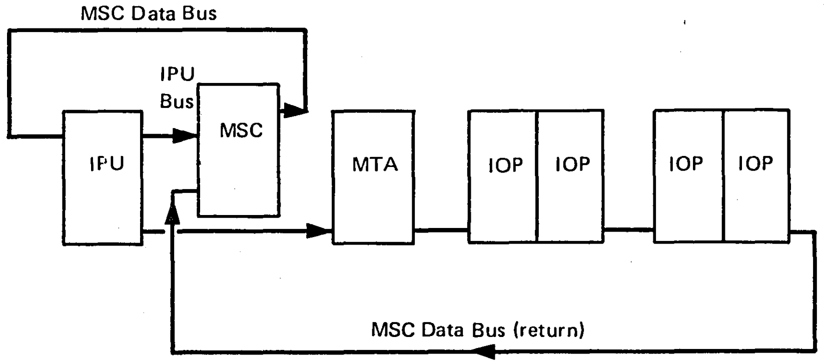
Bit Pattern on SVP Address Bus

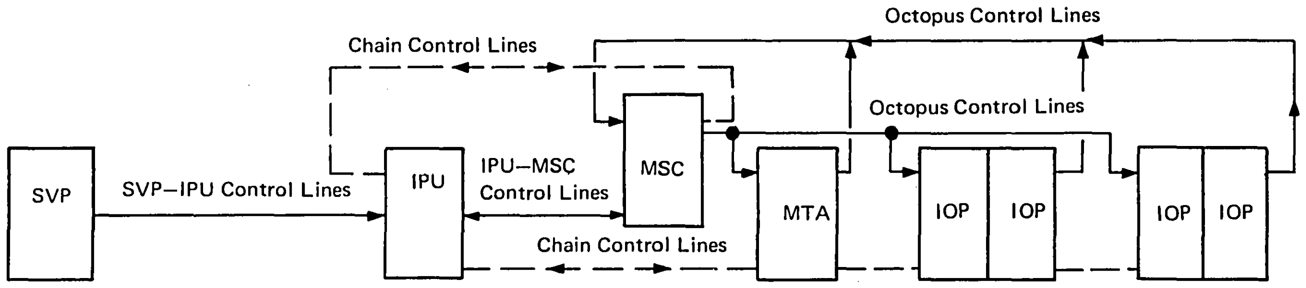




MSC Data Bus and IPU Bus

- The MSC data bus and the IPU data bus are both two bytes wide. Both bytes of these buses have their own parity bit.
- The MSC data bus exchanges information between the MSC and the selected subprocessor or adapter. Half-word transfers and single-byte transfers may be performed.
- The IPU bus is a separate-return bus from the IPU to the MSC. This bus allows fast access to main storage.





Chain Control Lines (CCL)

- Chain control lines are connected from one subprocessor or adapter to the next subprocessor or adapter.
- The chain control lines end at the last subprocessor or adapter.
- These lines specify the conditions of a data transfer and may be shared or non-shared.
- Shared lines are common to all the subprocessors and adapters; they are considered active only when 'select' is active.
- Non-shared lines are individual lines for the subprocessors or adapters.

Octopus Control Lines (OCL)

- Octopus control lines are individual lines for the subprocessor or adapter and are also used to specify transfer conditions.

SVP—IPU Control Lines

- There are five SVP—IPU control lines. Three lines are connected to all the subprocessors or adapters as part of the chain control lines. The other two lines indicate SVP conditions.
- The SVP—IPU control lines are as follows:
  Power on reset (CCL)
  System reset (CCL)
  Condition I/O meter
  SVP hardware error
  Lamp test (CCL)

IPU-MSC Control Lines

- The IPU—MSC control lines are a number of lines that are used by the IPU to address MSC-LS, and to specify IPU—MSC operations.
- The functions of these lines are similar to those of the chain and octopus control lines.

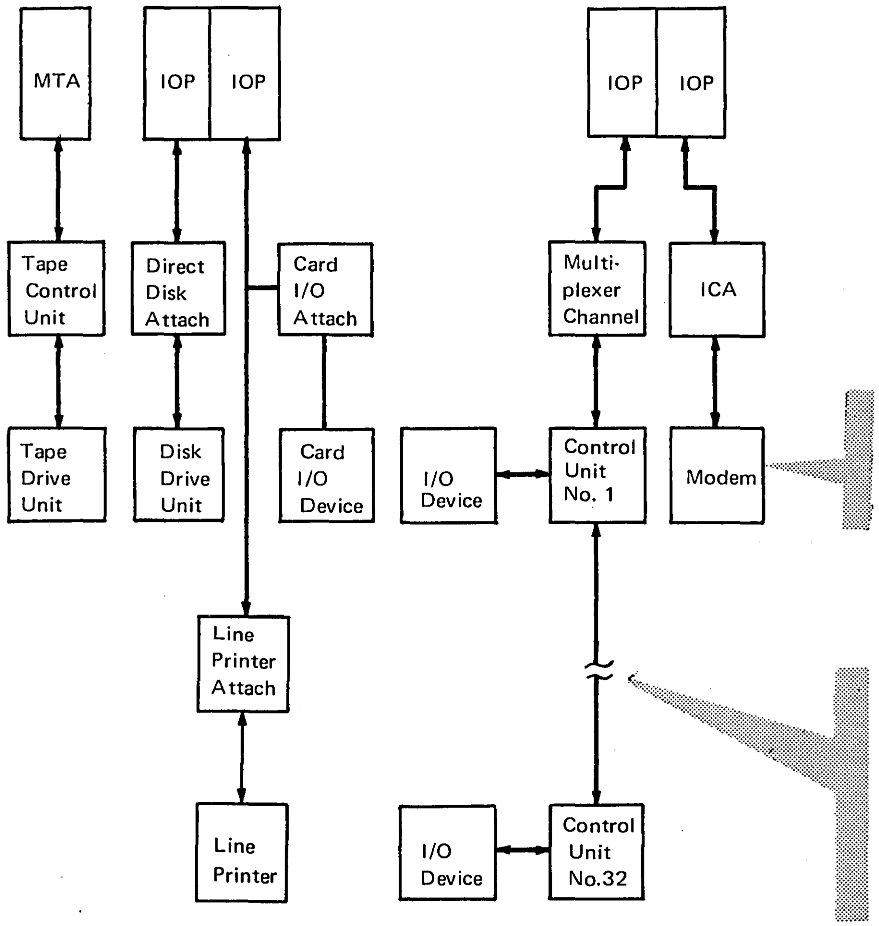
Connections Between the System and I/O Devices

Front ends are connected to the respective subprocessor via the following:

1. A one-byte wide external in bus.
2. A one-byte wide IOP D-register. (The D-register output is considered to be the bus out.)
3. External address lines.
4. Control lines.

Connection between front ends and I/O devices are device specific.

The tape control unit is directly connected to the bus system via the magnetic tape adapter. Therefore, no IOP is required.



The output from the integrated communications adapter is connected to a modem, of which many types are available.
The connections between two modems are the communication lines.

I/O devices and their control units are connected to the multiplexer channel via a standard interface.
The standard interface is a standardized bus system consisting of the following:
Bus in (one byte wide)
Bus out (one byte wide)
Tag in lines
Tag out lines

Chapter 3. Program and Microprogram Information

Two levels of program exist in the IBM System/370 Model 125:
Customer program
Microprogram.

Customer Program

- The customer program "tells" the machine "what to do"; it is also called the *job program.*
- This type of program uses machine language instructions. Machine language instructions specify typical operations that are to be executed by the machine.

Microprogram

- The microprogram specifies "how" the numerous machine language instructions of the customer program are to be performed.
- Three different microprograms are used:
  SVP microprogram
  IPU microprogram
  IOP microprogram.
- If the system is equipped with a magnetic tape adapter and/or a multiplexer channel, additional microprograms are used. They are located in the control units of the attached I/O devices.

Sense and Control Instructions

- Connection between machine language and program language (that is, customer program as well as microprogram) is made by using *sense* and *control* instructions.
- Sense instructions transfer information from hardware into programs. These instructions are used mainly for branch purposes. (Refer to Appendix B for further information.)
- Control instructions transfer information from programs into hardware. These instructions are used to set up a desired condition.
- To distinguish between the different sense and control instructions, the following nomenclature is used:
  1. SNS and CTL for machine language instructions.
  2. SENS and CTRL for microprogram instructions.
- Because each subprocessor (that is, the SVP, the IPU, and the IOPs) has its own microprogram, individual sense and control microinstructions are used to link microprogram and hardware.

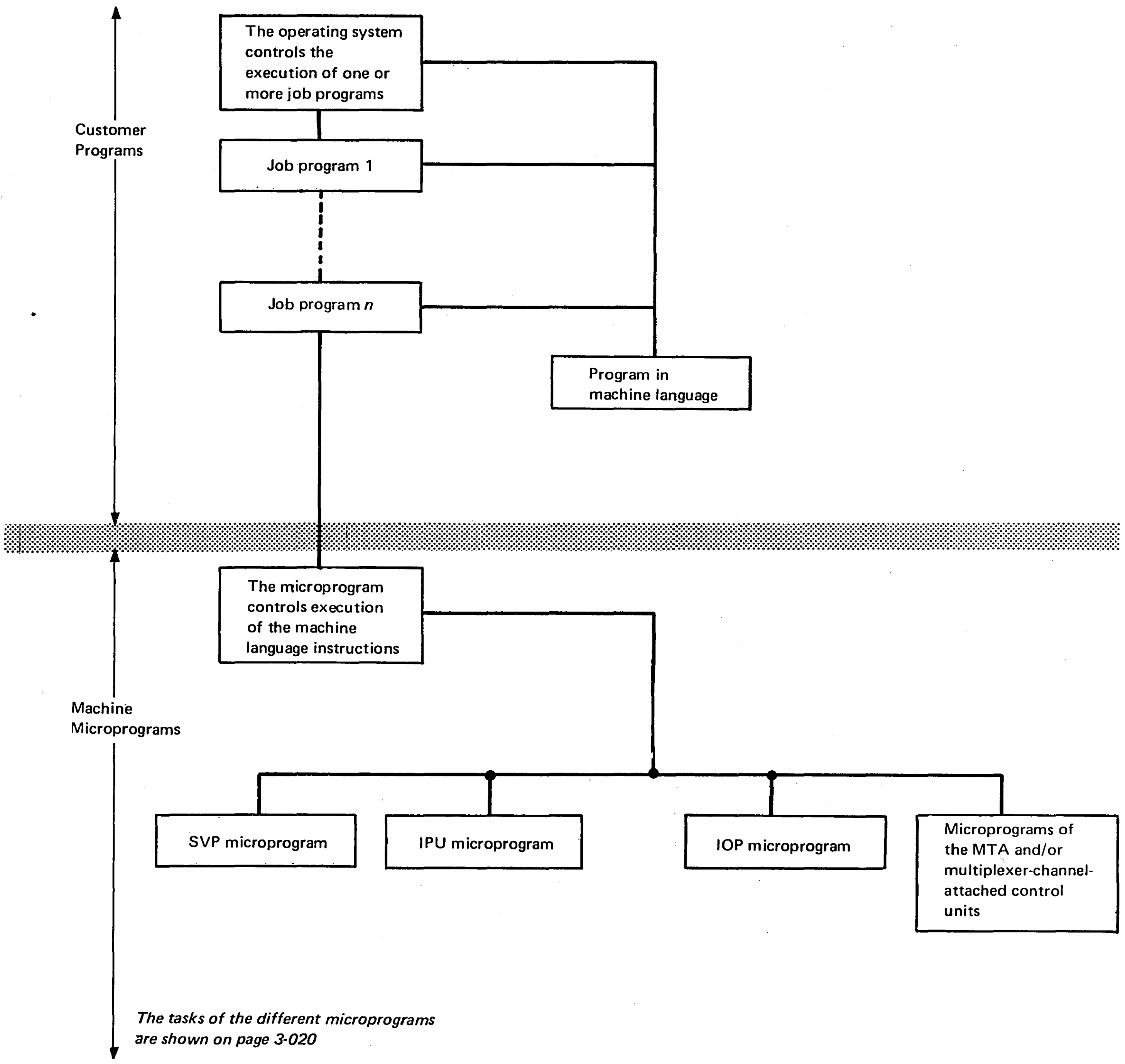
Sense Instructions

- The *SVP sense instruction* fetches information from a specified register of an addressed subprocessor or adapter.
- The *IPU sense instruction* fetches information from specified IPU or MSC locations and sets this information into CDR or TDR.
- The *IOP* does not use a particular op-code for its sense instruction; it uses the standard "move" op-code to sense specified Front End locations.

Control Instructions

- The *SVP control instruction* sets information into a specified register of an addressed subprocessor or adapter.
- The *IPU control instruction* sets information from CDR into a specified IPU or MSC location.
- The *IOP* does not use a particular op-code for its control instruction; it uses the standard "move" op-code to set information into a specific Front End location.

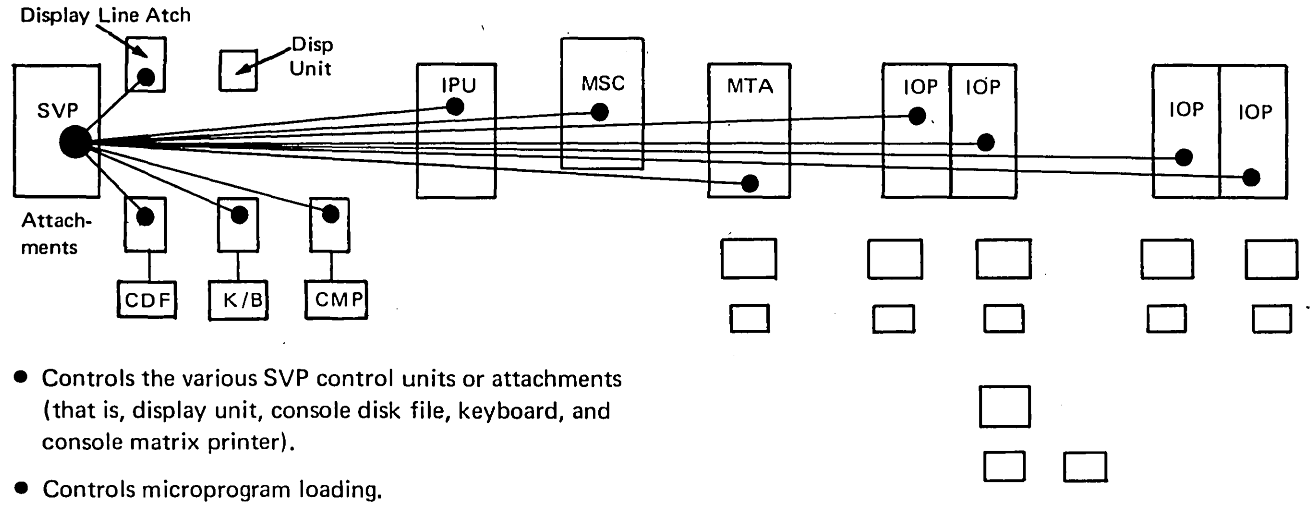
Program and Program Language Hierarchy


Customer Programs
The operating system controls the execution of one or more job programs
Job program 1
Job program *n*
Program in machine language
Machine Microprograms
The microprogram controls execution of the machine language instructions
SVP microprogram
IPU microprogram
IOP microprogram
Microprograms of the MTA and/or multiplexer-channel-attached control units

The tasks of the different microprograms are shown on page 3-020

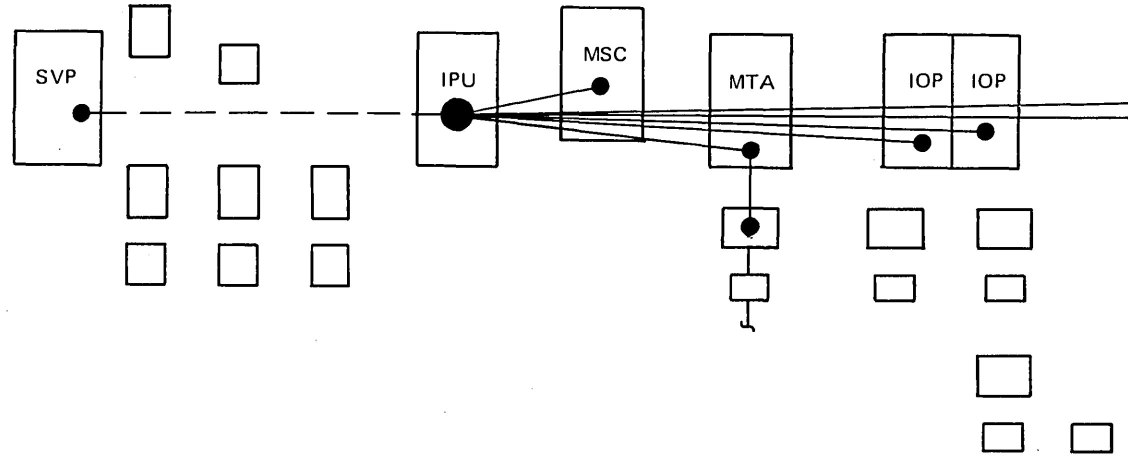
Microprogram Tasks

SVP Microprogram



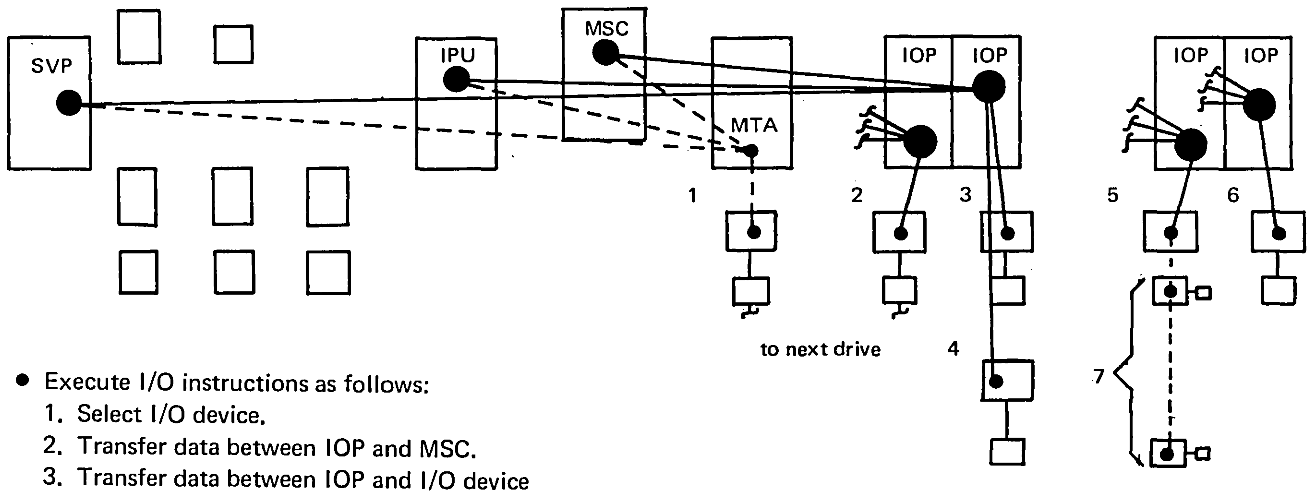
- Controls the various SVP control units or attachments (that is, display unit, console disk file, keyboard, and console matrix printer).
- Controls microprogram loading.
- Handles the service requests from the IPU, the MSC, the MTA, and all IOPs.
- Performs manual operations.

IPU Microprogram



- Controls the execution of the customer program's instructions. (I/O instructions are initiated only.)
- Handles interrupt requests after completion of an I/O instruction.
- Requests SVP service in the event of an error.
- The magnetic tape adapter does not have its own microprogram; it is considered as an adapter only.
- The tape drive units require a control unit, which is located in the first drive (also called the master drive).
- I/O instructions for the tape units are initiated by the IPU, which also handles the interrupt request.
- Data transfers between the MSC and the tape drive units are controlled by the microprogram of the control unit. "SVP service" is requested in the event of an error.

IOP Microprograms



- Execute I/O instructions as follows:
  1. Select I/O device.
  2. Transfer data between IOP and MSC.
  3. Transfer data between IOP and I/O device (or control unit).
- Request an interrupt after completion of an I/O instruction.
- Request "SVP service" in the event of an error.

Key:

1 Tape Control Unit
2 Direct Disk Attachment
3 Card I/O Attachment
4 Line Printer Attachment
5 Multiplexer Channel
6 Integrated Communications Adapter
7 Control Units (multiplexer channel)

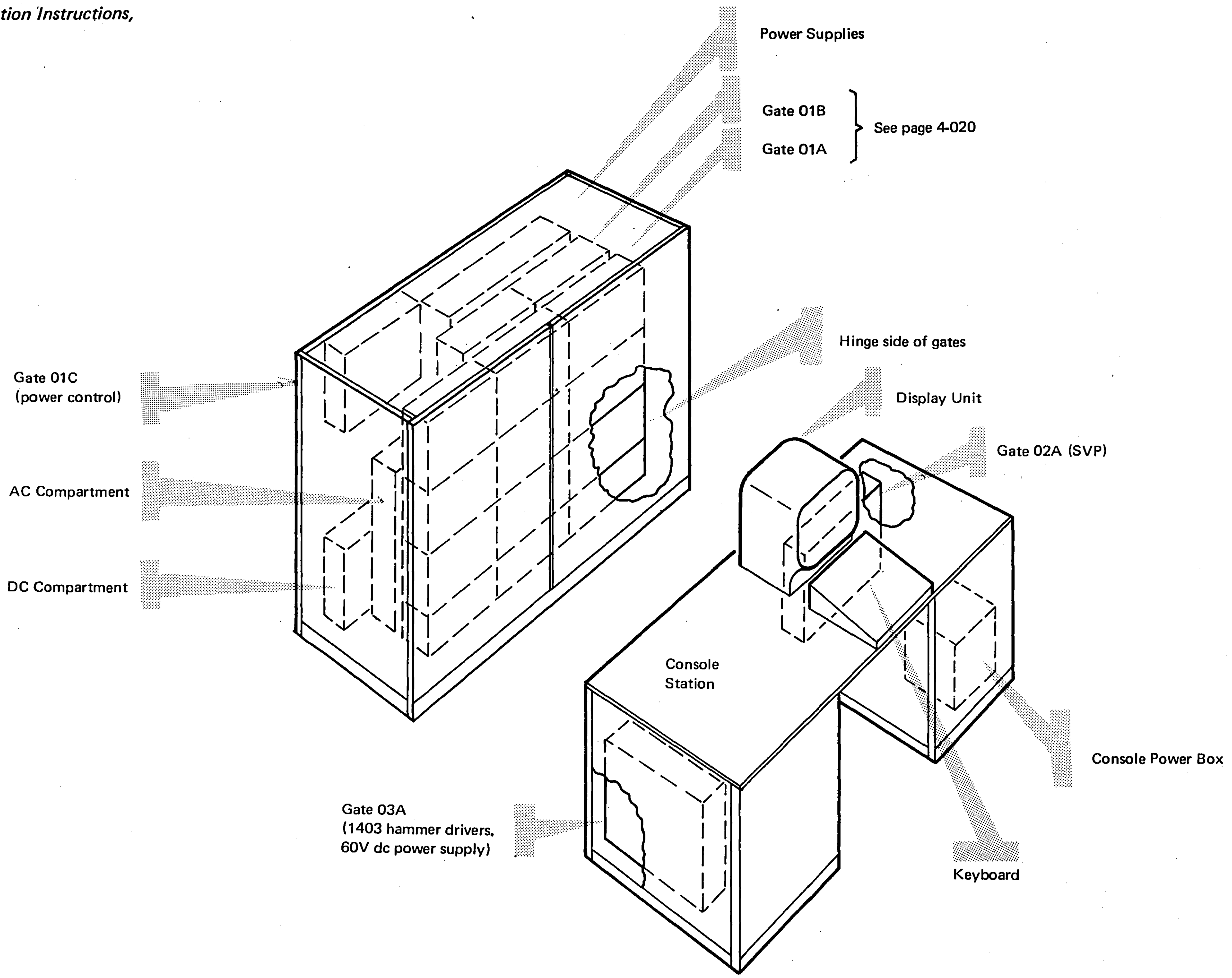
Exceptions

1. IOPs that control I/O devices that have high data transmission rates (for example, disk drive units).
   An I/O instruction is only initiated and interrupts are requested. Data transfers between the MSC and the I/O devices are performed by hardware; the IOP is bypassed.
2. The multiplexer channel.
   The control units (located between the multiplexer channel and the I/O device) usually have their own microprogram. Thus the multiplexer channel communicates with the control units, and the control units communicate with the I/O devices.

Chapter 4. Physical Layout

Physical Locations

- See also *IBM 3125 Processing Unit, Installation Instructions,* Part 4014001.



Gate Layout
(view from card side)

Gate 01A

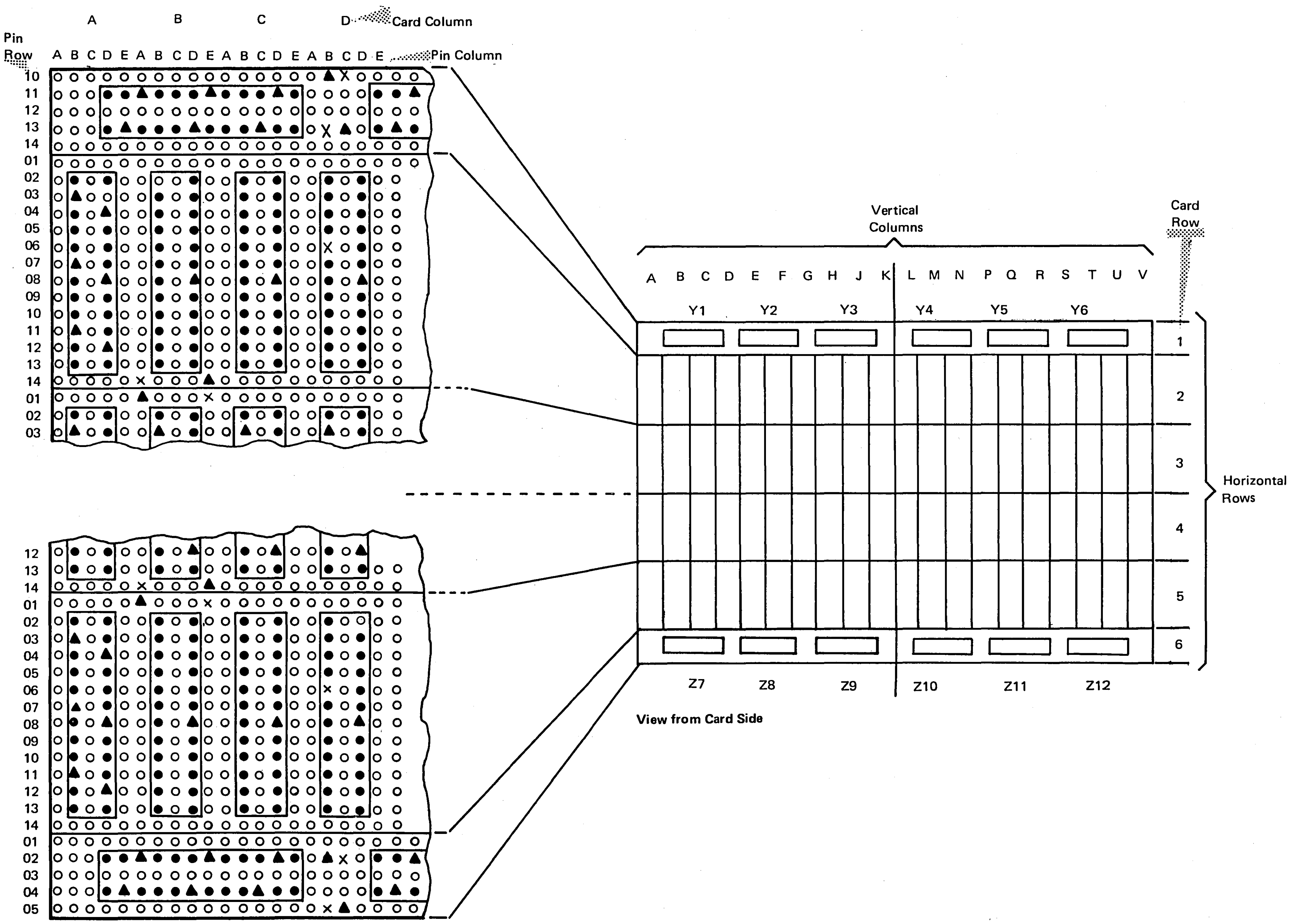
	A	B	C
1	Direct Disk Attachment	Magnetic Tape Adapter IPU Features	Card I/O Attachment (Front End) for 3504 Printer Attachment (Front End) for 1403
2	Instruction Processing Unit	IOP 'A' (controlling the Direct Disk Attachment) IOP 'B' (controlling card I/Os and printer)	Card I/O Attachment (Front End) for 2560 or 5425 or 3525
3	Main Storage Controller	Main Storage up to 64K	Main Storage Extension 96K 128K
	Blower		

Hinge Side

Gate 01B

	A	B
1	Multiplexer Channel (Not used)	(Not used)
2	IOP '9' (controlling the multiplexer channel) IOP 'B' (controlling the Integrated Communications Adapter)	Integrated Communications Adapter
3	(Not used)	Location for undercover modem
	Blower	

Board Layout



- Boards consist of four electrically independent planes.
- Two internal planes are used to distribute –4V and ground levels.
- Two surface planes carry the actual circuitry that interconnects the logic cards.
- To eliminate crossovers, only vertical printed lines are on the card side of the board, and only horizontal printed lines are on the probe side of the board. (For a connection between two pins the line can change from one side of the board to the other using so-called *via* pins.)
- All the outer socket positions

  Y1 to Y6
  Z7 to Z12
  A2 to A5
  V2 to V5

  are reserved for cable connectors (that is, 18 signals and six grounds.)
- The socket positions

  B2 to B5
  C2 to C5
  T2 to T5
  U2 to U5

  are designed in such a manner that either cable connectors or logic cards may be plugged into them. For these positions, voltage and ground connections are made on the surface planes.
- The socket positions

  D2 to D5
  .
  .
  .
  U2 to U5

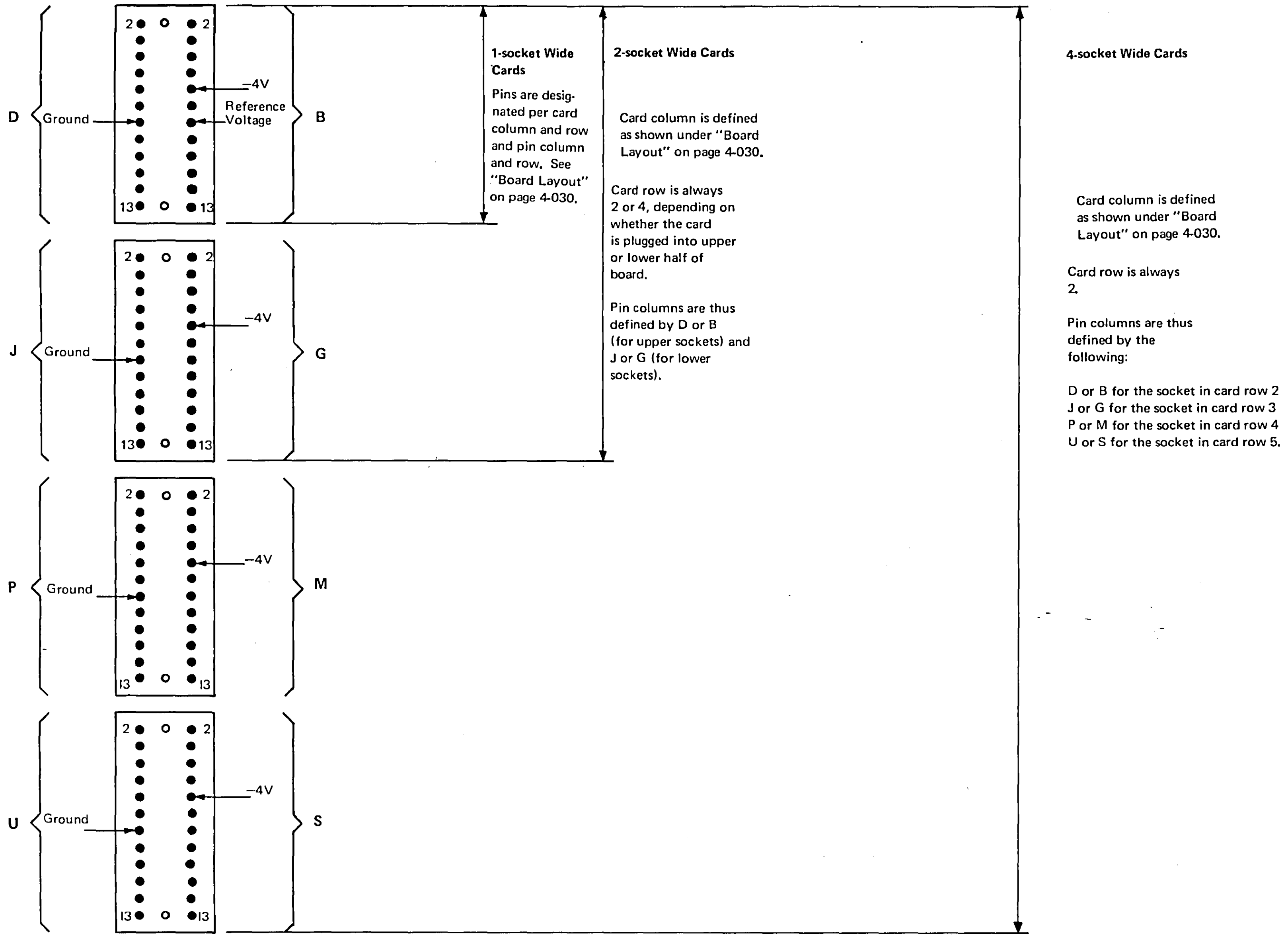
  are reserved for logic cards.
- Voltage and ground connections to the different positions are made via predetermined pins.

  ▲ represents a ground pin
  × represents a –4V pin.
- Each board is connected in the same way to the system voltage distribution network that is called the *laminar bus system.* (For voltage distribution details, refer to the system ALDs.)
- For further information, refer to *IBM System/360 and System/370 Logic Blocks, Automated Logic Diagrams, FEALD, ALD, and Component Circuits, Theory of Operations,* SY22-2798, and *IBM MST Packaging, Tools, Wiring Change Procedure, Theory of Operations,* SY22-6739.

Pin Addressing Scheme

The circled letters in the diagram below define the pin column from the *probe side*. This lettering includes the definition of Card or Socket row, respectively.

Note: Further information is given in *IBM System/360 and System/370 Logic Blocks, Automated Logic Diagrams, FEALD, ALD, and Component Circuits, Theory of Operations,* SY22-2798 and *IBM MST Packaging, Tools, Wiring Change Procedure, Theory of Operations,* SY22-6739.



1-socket Wide Cards

Pins are designated per card column and row and pin column and row. See "Board Layout" on page 4-030.

2-socket Wide Cards

Card column is defined as shown under "Board Layout" on page 4-030.

Card row is always 2 or 4, depending on whether the card is plugged into upper or lower half of board.

Pin columns are thus defined by D or B (for upper sockets) and J or G (for lower sockets).

4-socket Wide Cards

Card column is defined as shown under "Board Layout" on page 4-030.

Card row is always 2.

Pin columns are thus defined by the following:

D or B for the socket in card row 2
J or G for the socket in card row 3
P or M for the socket in card row 4
U or S for the socket in card row 5.

Legend

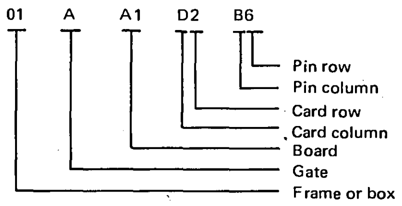
○ – represent *via* pins. Normally these pins are not accessible (see page 4-030).

● – represent *actual* pins, which are accessible from both sides of the board (see page 4-030).

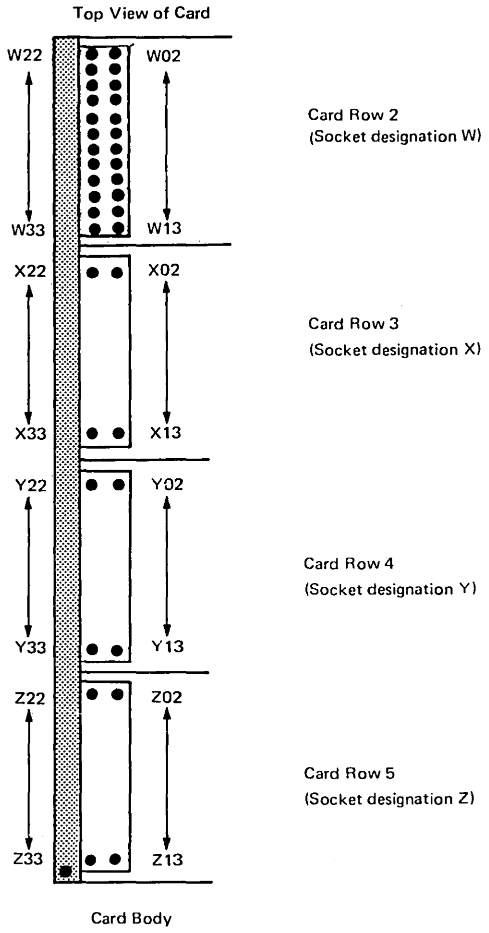
On the card side, these pins serve as connector pins to the logic cards.

On the probe side, these pins may be used for test purposes.

Example of Pin Designation

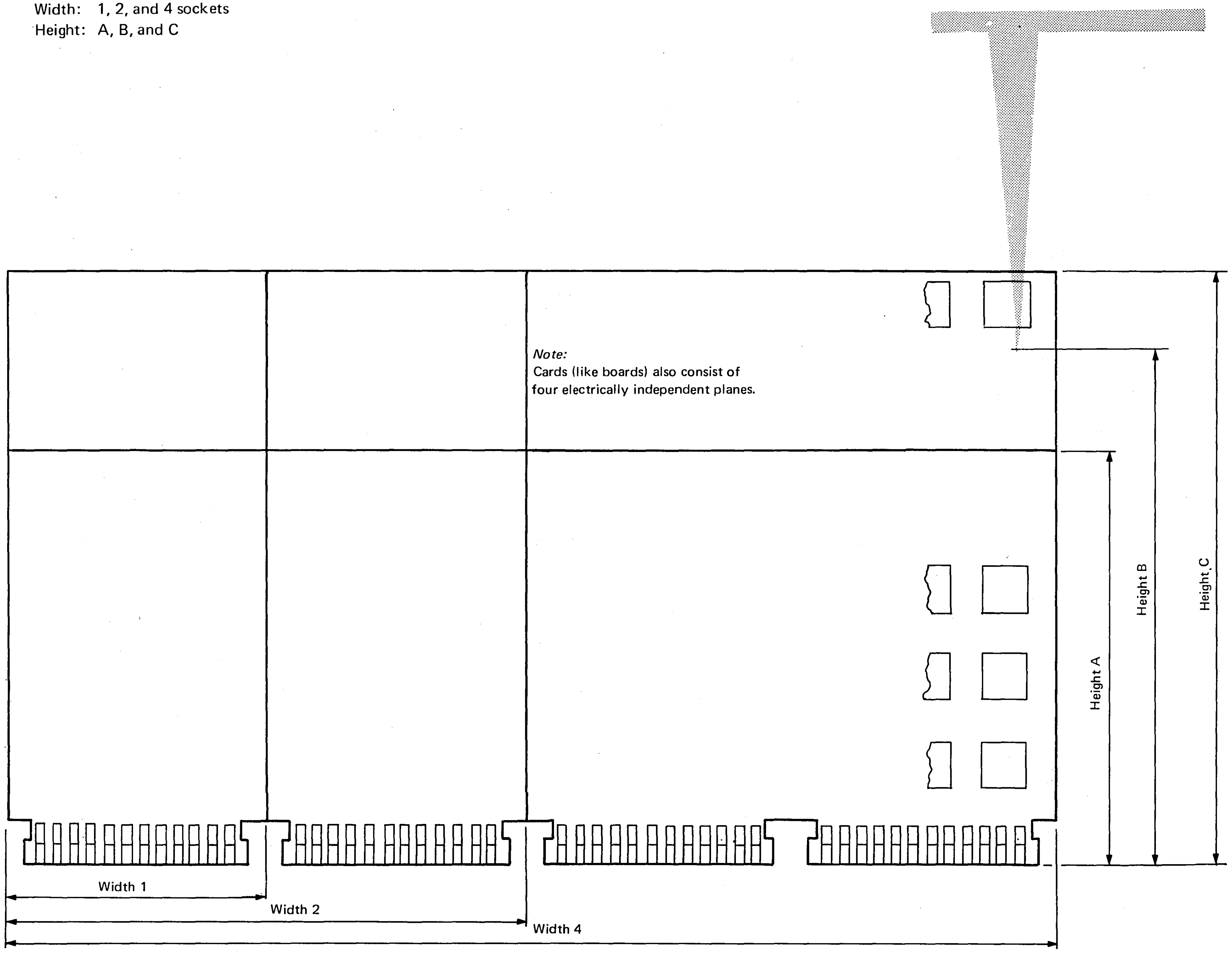


- Some cards carry *top connectors.* These top connectors can be mounted instead of modules in the upper row of cards that have height B (see page 4-050).
- Top connectors and their pins are counted as shown below.



Card Types

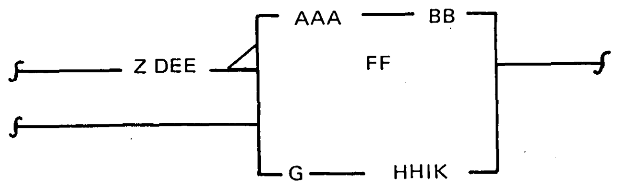
- Cards of different standard sizes (width and height) are available.
- Card sizes are as follows:
  Width: 1, 2, and 4 sockets
  Height: A, B, and C
- For some purposes it is practical to group card types according to the number of sockets and the maximum number of module positions.
- Cards of the height B can have top connectors instead of the upper row of modules (see page 4-040).



Note:
Cards (like boards) also consist of four electrically independent planes.

Card Identification

- Each card is identified by the following:
  Part number
  Current EC level
  Card code.
- The part number and the EC level are shown on each card and are required when ordering new cards from stock.
- The card code is used for general identification purposes. Respective card codes are shown on the bottom of all ALD pages, but the card code normally is not used by the CE.
- The logic blocks on the MALDs contain the following information:



Legend

AAA:	If these positions are blank this logic block is standard. If something is shown in these positions, the letters point out that these logic blocks belong to a particular optional feature.
BB:	Logic block serial number.
Z:	Defines MST logic. If this position is blank and the block does not belong to MST logic, levels are shown in connection with line names.
D:	Pin column (see Page 4-040)
EE:	Pin number (see Page 4-040)
⊿	Defines that the active line level is minus.
FF:	Function of logic block.
G:	Location; frame and gate.
HH:	Location; board (see Pages 4-010 and 4-020)
I:	Card column (see Page 4-030)
K:	Card row (see Page 4-030)

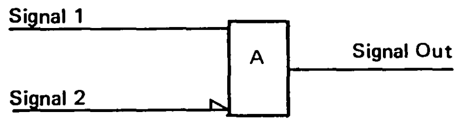
Chapter 5. Symbology

Logic Simplifications

- The operation-oriented logic diagrams in the 3125 Processing Unit maintenance library manuals use the *bottle* technique and are drawn according to *potential independent logic* (which does not show line levels).
- A bottle is used to show as many single AND/OR functions, simplifications and compressed into one logic block, as possible.
- Normally, a bottle consists of an upper and lower part, separated by the bottle-neck. Signals connected to the upper part are gating and timing signals for the signals connected to the lower part.
- If no timing or gating signals are present, the block is drawn without the bottle-neck.

Wedges

- Wedges are used in potential independent logic to indicate that an input line must be inactive to satisfy the logic block input conditions.

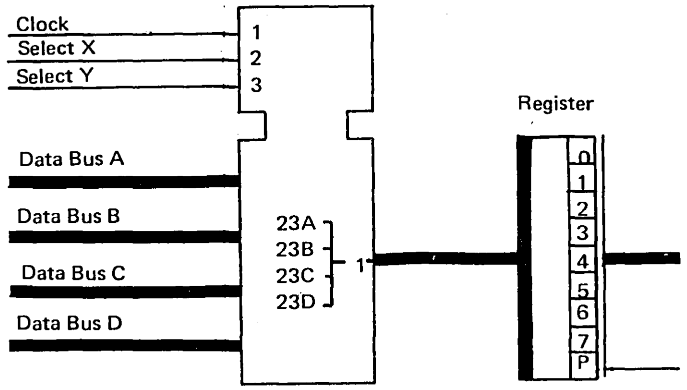


To satisfy the AND switch,
Signal 1 must be active
Signal 2 must be inactive.

- **Wedges do not define minus levels (as in the ALDs).**

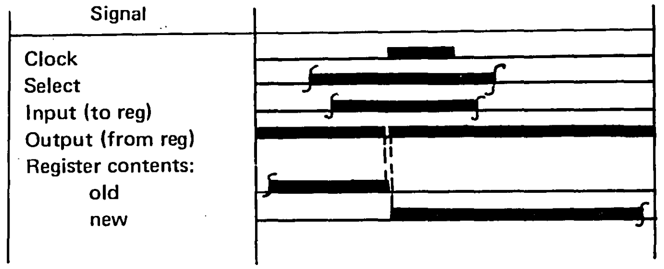
Registers

Registers may be considered as interim storages

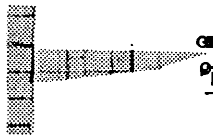


The two 'select' lines X and Y are decoded, and one of the four data buses A, B, C, and D is gated into the register under control of the clock signal.

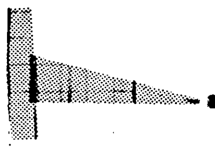
The active clock signal resets the register. While the clock signal is still active, a new bit pattern may be set into the register. The register then holds its new contents until the next active clock pulse.



There is another type of register: this uses the two select lines directly. One out of two buses is selected and is indicated in the bottle that precedes the register.



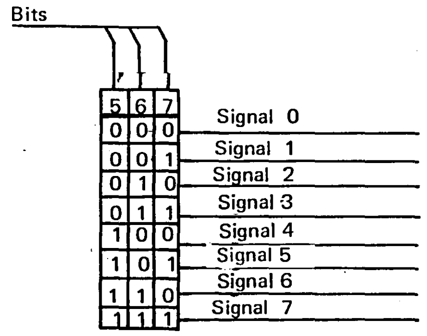
The presence of open connectors on the signal lines indicates that the signals are connected to card pins and that they are measurable



The absence of open connectors on the signal lines indicates that the signals have card internal connections and that they are not measurable

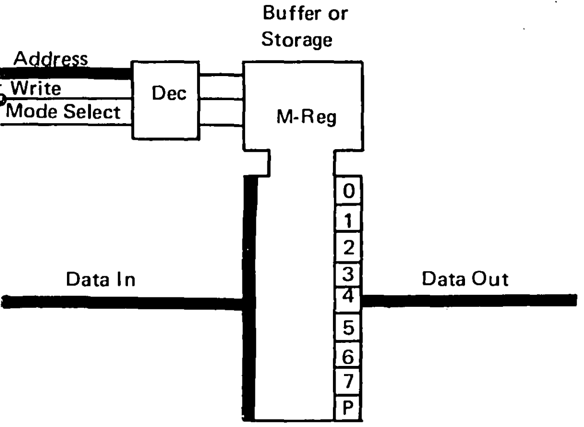
Table (or Decoder)

The following example shows a table (or decoder) that uses three binary-coded bits. These bits are used to generate eight different lines.



5	6	7	
0	0	0	Signal 0
0	0	1	Signal 1
0	1	0	Signal 2
0	1	1	Signal 3
1	0	0	Signal 4
1	0	1	Signal 5
1	1	0	Signal 6
1	1	1	Signal 7

Buffers and Storages



Information can be stored for later use. The storage capacity is variable, that is,
the number of addressable storage positions
and
the number of bits per addressable storage position.
Control lines enter the upper part of the logic block.

Data to be stored enters the lower part of the logic block.

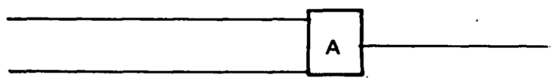
The 'address' line in conjunction with the 'module select' line is used to address or select the storage positions. The storage positions are often referred to as registers.

Storage read or write operations are identified by the 'write' line ('write pulse' or 'write gate').

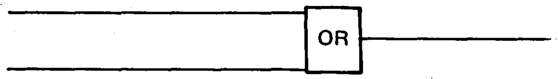
Store Operation
(Also known as *storage write* or *storage read in* operation.)
With the 'write' line active, data is set into the addressed storage position.

Fetch Operation
(Also known as *storage read* or *storage read out* operation.)
With the 'write' line inactive, data is read out of the addressed storage position.

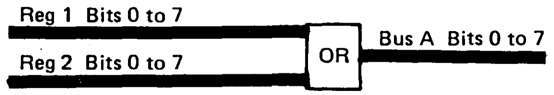
Logic Symbology



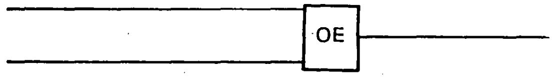
AND Switch



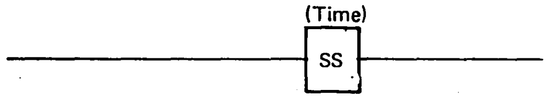
OR Switch



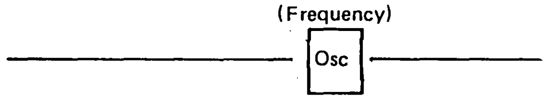
This circuit shows that the exits of registers 1 and 2 are ORed together by eight independent OR switches, each having two inputs.



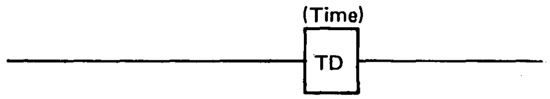
Exclusive OR Switch (sometimes represented by XOR)



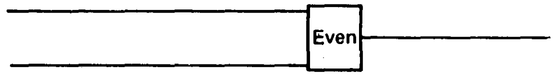
Singleshot



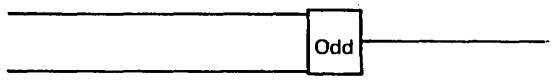
Oscillator



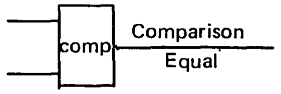
Time Delay



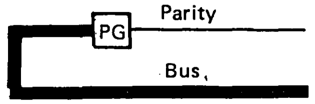
Even
The output is active when an even number of inputs is active



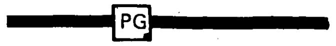
Odd
The output is active when an odd number of inputs is active



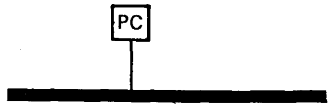
Compare



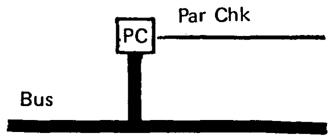
Parity Generation



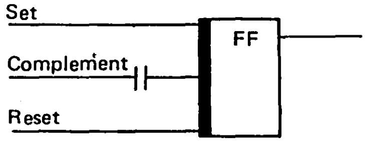
Parity Generator



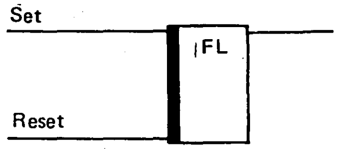
Parity Check



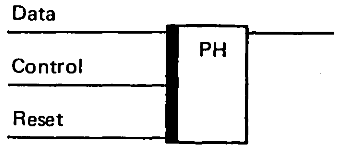
Parity Check



Flip Flop
The capacitor indicates that two edges of a pulse switch this circuit



Flip Latch



Polarity Hold

Thick Line at Input Side of Block
A thick line at the input of logic blocks represents that this block has the ability to store information.

Pullovers
Some circuits have *pullovers*. A signal on those lines directly switches the circuitry from one status to another, independent of the normal set and reset conditions. Because these pullover set and reset conditions are considered as normal set and reset conditions, they are ORed with all other set and reset conditions.

Single Output
Because potential independent logic is used, it is sufficient to show only one exit per logic block. This exit becomes active as soon as all entry conditions are satisfied. This single output is shown leaving the top of a logic block.

Active signal: represented by the line without additional marks.

Inactive signal: represented either by a wedge at the input to a logic block or by a Boolean "not" bar over a character within a bottle logic block.

Refer also to page 5-010.

Parity Checking

- Bits on buses are represented by defined voltage levels.
- For the internal bus system (excluding the standard interface) the parity bit is defined as shown in the example below. Note that significant bits are active at the down level and that the parity bit is active at the up level.

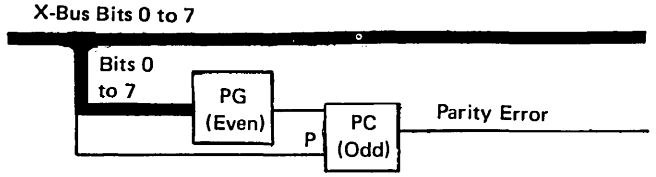
Example

Decimal	Active Bits (down level) 8	4	2	1	Number of Active Bits	Parity Bit Required	Parity Bit Level
1				X	Odd	No	Down
3			X	X	Even	Yes	Up

This means that any correct bit combination is represented by an odd number of up levels.

- If a bus connector is loose, all lines are at the down level. Because the bus parity is now active, an even number of bits is active at the parity generator, thus generating a parity bit. The single parity bit at the "odd" parity check circuit results in a parity error.

- The following circuit is used for parity checking:



The parity generator generates a parity bit on any even number of incoming bits; the generated parity bit is connected to an "odd" parity check circuit.

If there is only one parity bit at the "odd" parity check circuit, the bit combination is incorrect and results in a parity error (see table).

Decimal	Active Bits 8	4	2	1	Active Bit Line Level	Parity Bit Active on Bus	Active Parity Line Level	PG Out	Number of Parity Bits	Parity Error
1				X	Down	No	Down	No	None	No
3			X	X	Down	Yes	Up	Yes	Two	No

Incorrect bit combinations are usually caused by omitted or additional bits (that is, single-bit errors)

Assume that bit 2 is omitted.

3				X	Down	Yes	Up	No	One	Yes

Assume that bit 4 is additional

3			X	X	Down	Yes	Up	No	One	Yes

If the parity bit is erroneously omitted or present, this will also cause a parity error.

Example:

Bus Lines	Data	Parity	
Even	Odd – + + + + – – + + +	Odd ±	} Odd parity
Odd	Even – – – – – – – – + – – – – – – –	Odd ±	} Even parity

Odd number of + levels including data and P bits = correct parity

Timing Chart Principles

- Two versions are used to show the relationship between signals:
  1. Numbering version.
  2. Arrowhead version.

Numbering Version

- Uses Boolean symbology. A *dot*(●) represents an AND, a *plus* (+) represents an OR, and the *bar* indicates that an inactive signal is required.
- The signals are numbered. These numbers are used in combination with the Boolean symbols to exactly define the activation and inactivation of the signals.

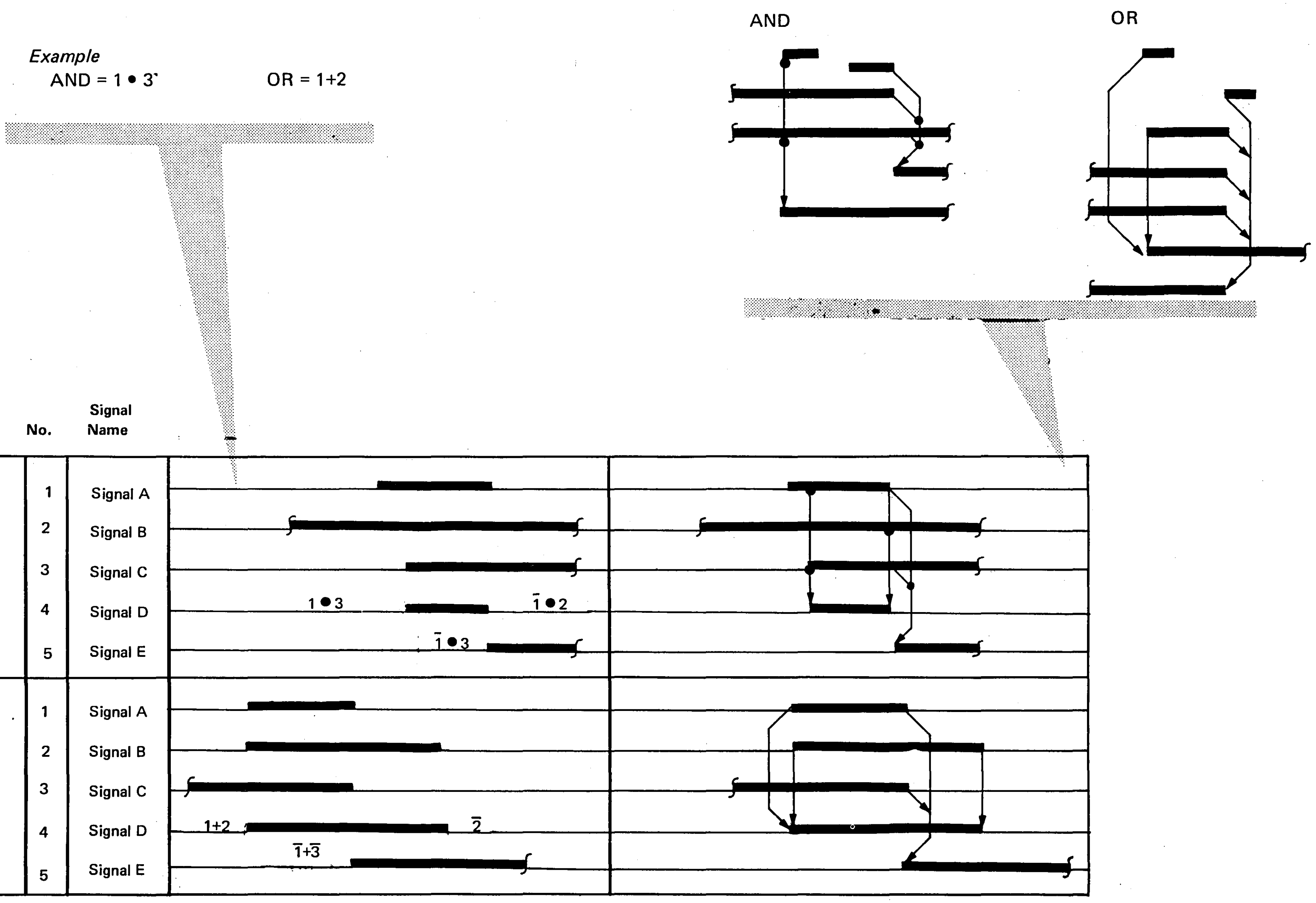
Example

AND = 1 • 3 OR = 1+2

Arrowhead Version

- Uses thin lines to show relationship between signals.
- A *dot* shows the AND function; an arrowhead shows the OR function.

Example



AND Function

This example shows the activation and inactivation of signal D and shows the activation of signal E.

Activation of signal D = 1 • 3
Inactivation of signal D = $\bar{1}$ • 2
Activation of signal E = $\bar{1}$ • 3

OR Function

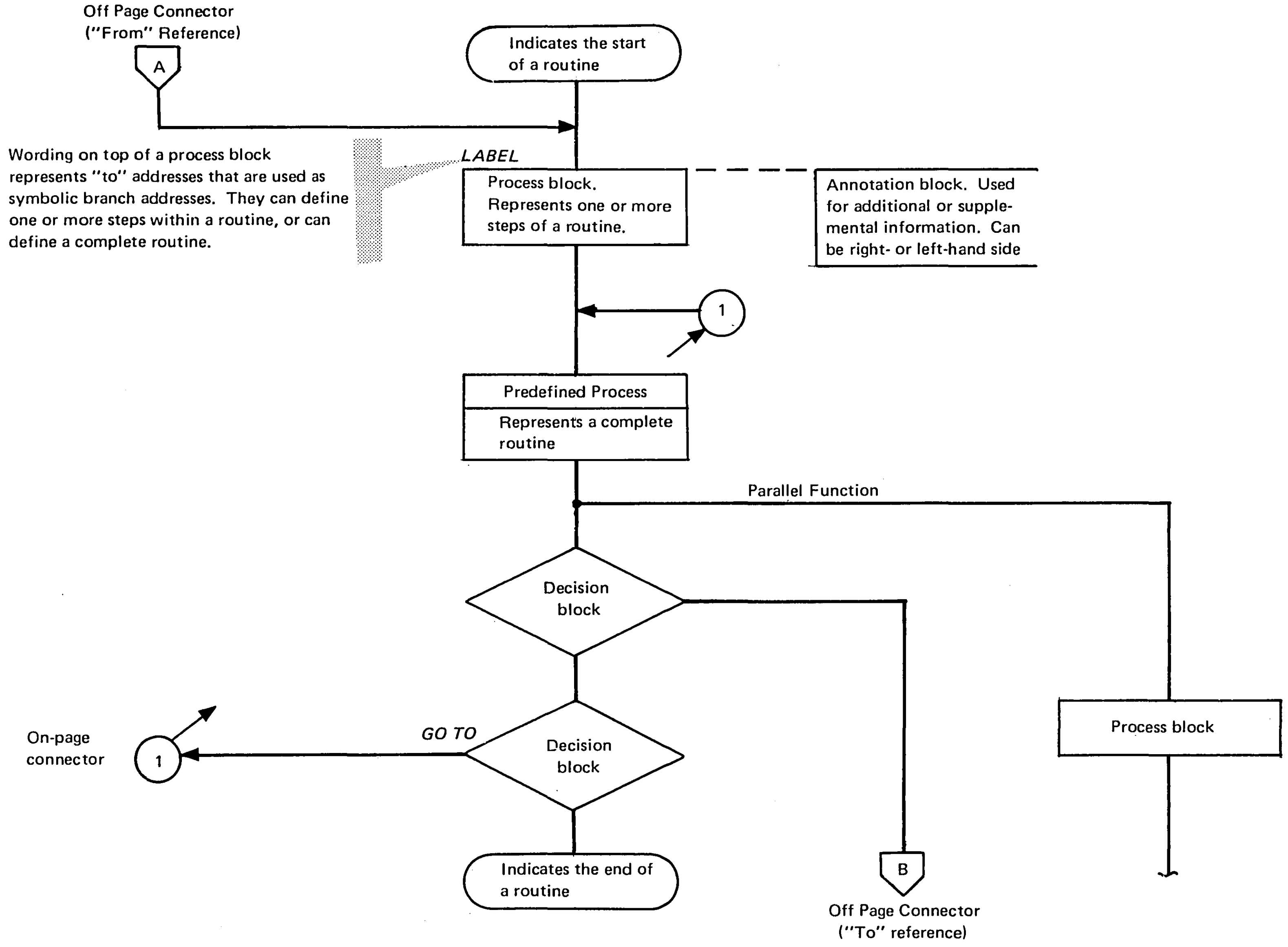
This example shows the activation and inactivation of signal D and shows the activation of signal E.

Activation of signal D = 1+2
Inactivation of signal D = $\bar{2}$
Activation of signal E = $\bar{1}$+$\bar{3}$

Flowchart Symbology

Standard Symbology

- This page defines the standard flowchart symbology that is used throughout the maintenance library manuals.


Off Page Connector
("From" Reference)
A
Indicates the start of a routine
Wording on top of a process block represents "to" addresses that are used as symbolic branch addresses. They can define one or more steps within a routine, or can define a complete routine.
LABEL
Process block. Represents one or more steps of a routine.
Annotation block. Used for additional or supplemental information. Can be right- or left-hand side
1
Predefined Process
Represents a complete routine
Parallel Function
Decision block
Process block
On-page connector
1
GO TO
Decision block
Indicates the end of a routine
B
Off Page Connector
("To" reference)

Flowchart Symbology (continued)

Four-Level Format

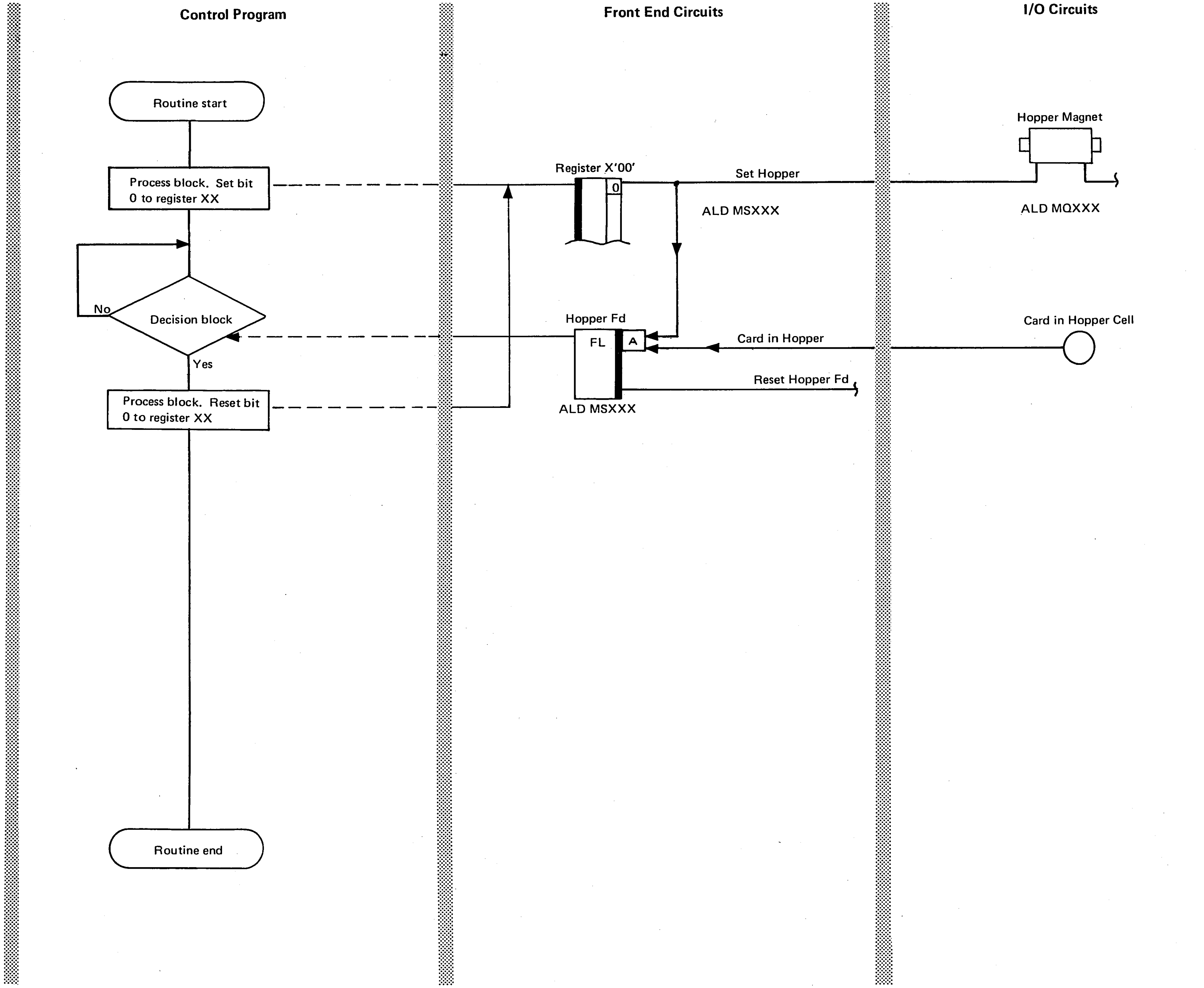
This page shows an example of a 'four-column format' flowchart, which is used throughout the maintenance library manuals describing the I/O devices.

This format is used to show connections between microprogram and hardware.

Objectives and Additional Text (Explanations)

This routine sets/resets the hopper magnet.

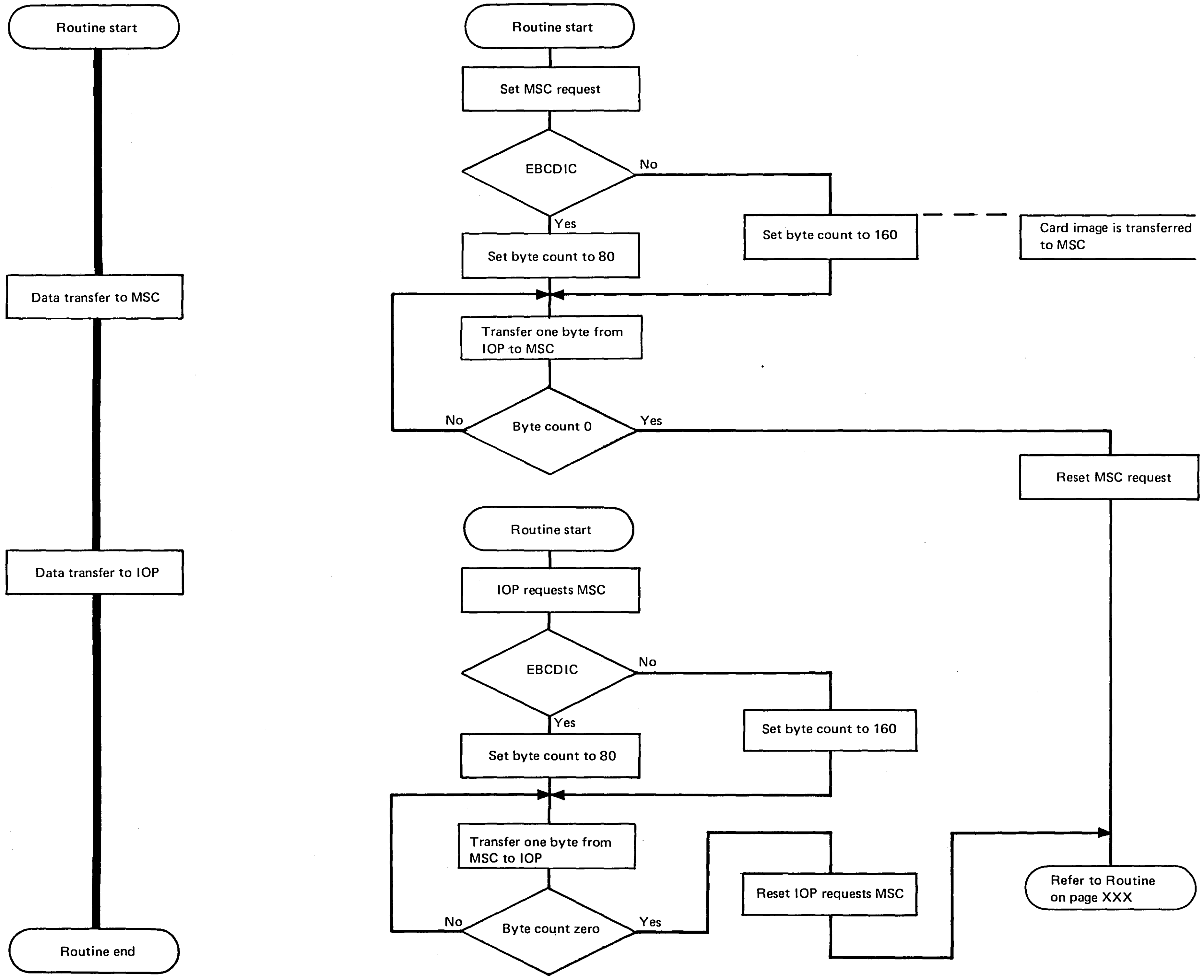
If a hopper check occurs refer to page X-XXX



Two-Level Format

- This page shows an example of a 'two-level format' flowchart which is used throughout the maintenance library manuals describing the subprocessors.

The first level of the flowchart gives brief information about the microprogram flow shown on the page.



This page is intentionally left blank.

Chapter 6. Maintenance Information

Maintenance Concept

- In the event of a machine malfunction, attempt to eliminate the cause by following the instructions given on the display unit.
- Whenever cards are exchanged, check the socket pins. This ensures proper connections between cards and boards and may eliminate an additional malfunction.
- Instructions to change a card (or cards) result from either the log analysis program or the microdiagnostic function tests.
- If changing cards does not remove the fault, the following facilities may be used in conjunction with the CE latch card or an oscilloscope:
  1. Manual log analysis
  2. Microdiagnostic function test run in the error stop mode
  3. Manual operations
  4. Use of one of the two matrix types.
- Full details of diagnostic techniques and test procedures are given in *IBM 3125 Central Test Manual.*

Tools and Test Equipment

CE Latch Card

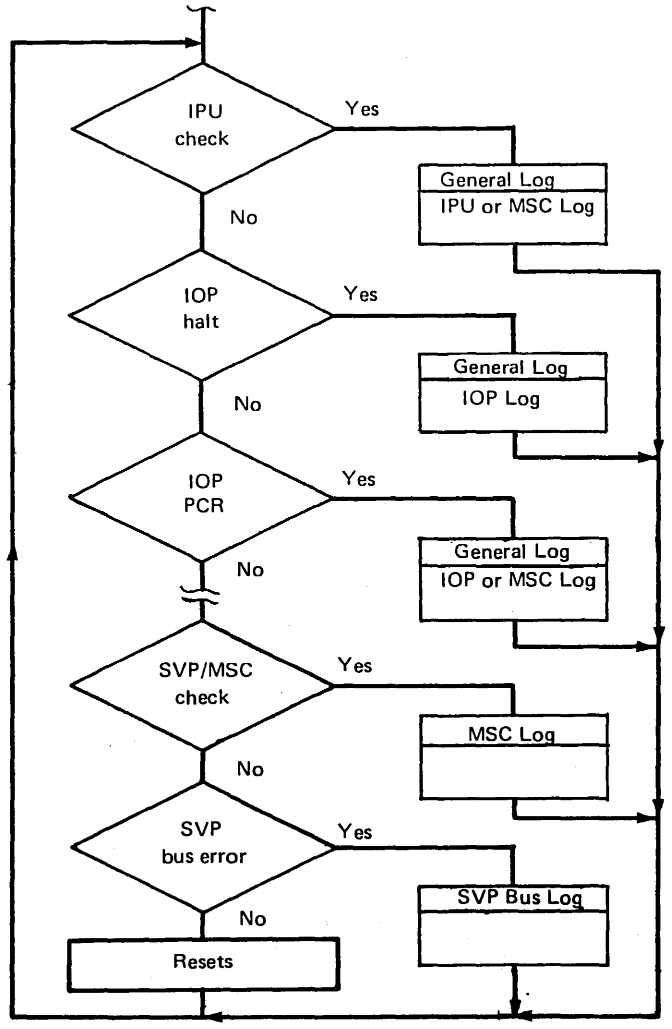
- The CE latch card is designed to assist in trouble shooting.
- Detailed information about the CE latch card is given in *IBM 3125 Processing Unit, Central Test Manual.*

Logging Principles

- To ensure error data retention for system malfunction analysis, all solid and intermittent errors are recorded.
- The recorded log data may be:
  1. Displayed.
  2. Evaluated by log analysis program.
  3. Erased (if no longer required).
- A separate log area is provided for each system component.
- Each log area consists of two portions: header and actual log data. The header is of a common format and contains the following:
  1. Name of log area.
  2. Log dates.
  3. Log count.
- Error recording is done as long as the SVP main sense loop is active, but is done neither during DFT applications nor during manual operations.
- If an error occurs while DFTs are running, or during manual operations, error conditions may be displayed by the use of LOG DISPLAY.
- While a log is being performed, the keyboard remains locked.
- For more detailed information about logging, refer to *IBM 3125 Central Test Manual,* and *IBM 3125 Processing Unit, Service Processor Subsystem,* SY33-1065.

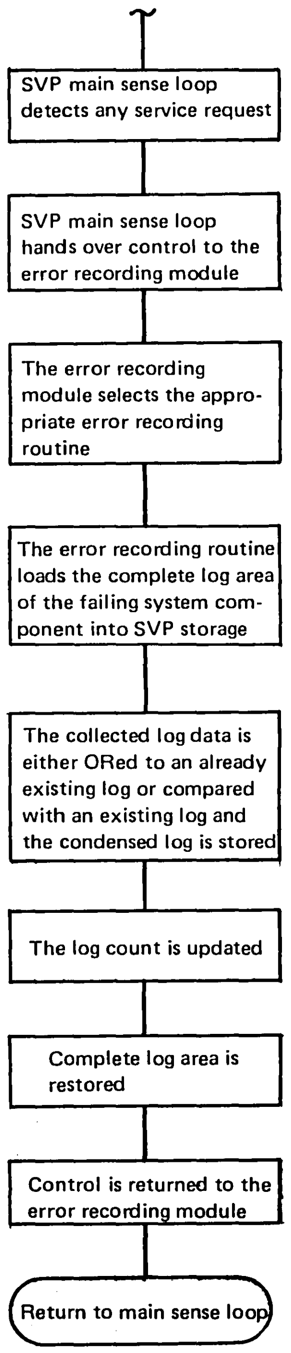
SVP Main Sense Loop

- The SVP main sense loop is an SVP microprogram loop, which periodically addresses all subprocessors and adapters of the system (except the MSC) and looks for SVP service requests.
- The addressed system component places its idle sense bit pattern onto the SVP data bus and the SVP analyzes the presented bit pattern.
- The idle sense is one-byte wide and contains the following:
  1. A four-bit wide address of the subprocessor or adapter.
  2. Four bits that signal different conditions as the cause for SVP action.
- The flowchart below shows the principle of the SVP main sense loop.



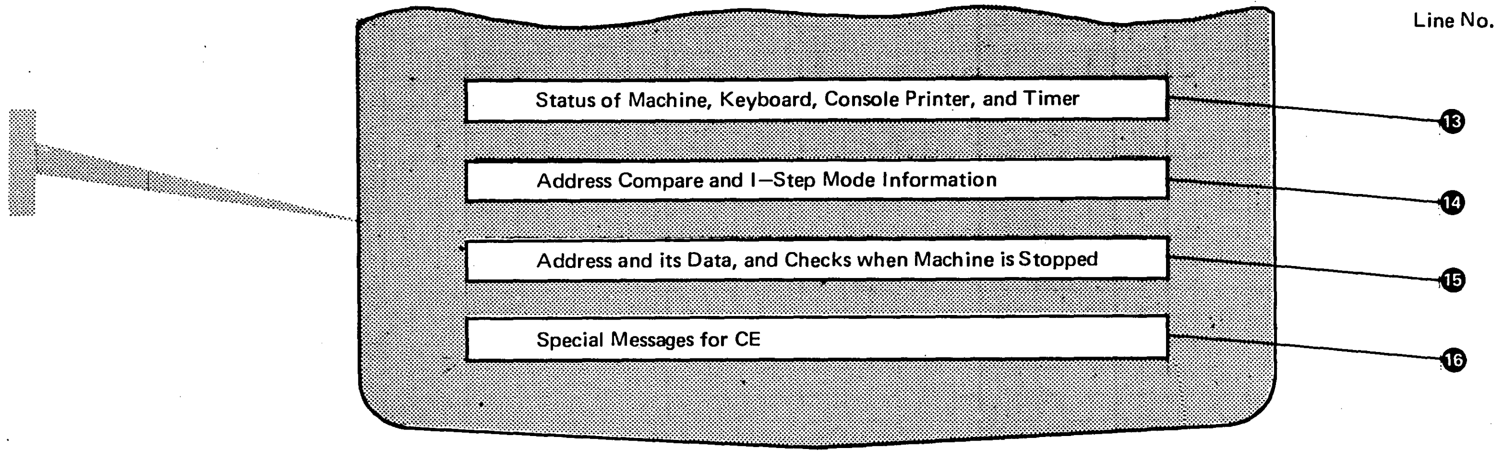
Log Recording Method

- The flowchart below shows the principle of log recording.

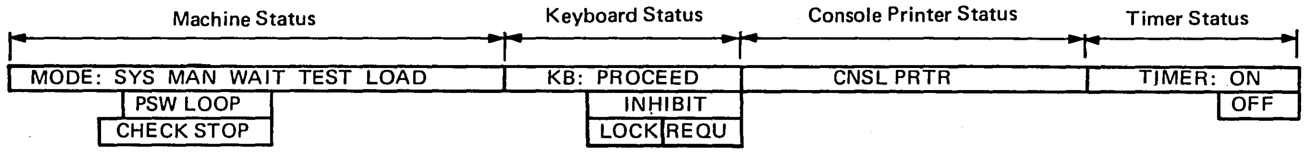

SVP main sense loop detects any service request
SVP main sense loop hands over control to the error recording module
The error recording module selects the appropriate error recording routine
The error recording routine loads the complete log area of the failing system component into SVP storage
The collected log data is either ORed to an already existing log or compared with an existing log and the condensed log is stored
The log count is updated
Complete log area is restored
Control is returned to the error recording module
Return to main sense loop

Machine Status Indication

- The lower four lines (numbered 13, 14, 15, and 16) of the display unit screen are called the status area and are used to indicate the machine status as long as the machine runs under control of the operating system.
- During maintenance no machine status is given. These lines are then used to display maintenance information.


Line No.
Status of Machine, Keyboard, Console Printer, and Timer
13
Address Compare and I—Step Mode Information
14
Address and its Data, and Checks when Machine is Stopped
15
Special Messages for CE
16

Line 13

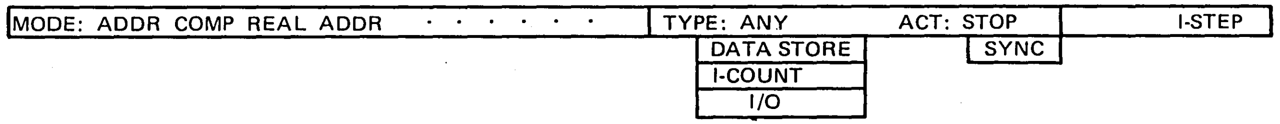


MODE
SYS: System running and meter on (blank if meter is off).
MAN: System stopped and in manual mode.
WAIT: When "wait" bit in current PSW is on.
TEST: Either I-STEP mode, or ADDR COMP mode, or CHK CTL mode, or inline tests running, or microdiagnostic tests running.
LOAD: On for load operation until valid PSW is loaded.
PSW LOOP: Program loops continuously on same PSW (hang up). This replaces MAN and WAIT.
CHECK STOP: Occurrence of errors. This replaces SYS, MAN, and WAIT (see Note).
KB
PROCEED: Keyboard unlocked and ready for keying-in.
INHIBIT: An attempt to key-in into protected screen areas.
LOCK: All keys that can interfere with a currently performed operation are locked.
REQU: A request via the REQUEST key is accepted (that is, stored).
CNSL PRTR: Console printer requires service (that is, it is not ready).
TIMER
ON: Timer clock is running.
OFF: Timer clock is not running.

Note: The following messages can occur with CHECK STOP:
IPL ERROR: This indicates the inability to load a valid PSW (that is, load device not ready, wrong address, wrong interrupt, etc.)
INV EC PSW: This indicates that the load process ended with the attempt to load an invalid PSW.
INV PSW ADDR: This indicates that the instruction address in the current PSW points to a position outside the actual storage.

Line 14

- This line is normally blank. Only when ADDR COMP Mode or I-STEP are selected is pertinent information displayed.



ADDR COMP, REAL ADDR: Shows the actual address that is to be compared.
TYPE:
ANY: ADDR COMP is active for any address.
DATA STORE: ADDR COMP is active for data store to main storage operations only. (Fetch addresses are ignored)
I-COUNT ADDR COMP is active for instruction — fetch operations only.
I/O ADDR COMP is active for I/O transfer operation only.
ACT:
STOP The machine stops when ADDR COMP EQUAL is signaled before that cycle is executed.
SYNC: The machine generates a signal when ADDR COMP EQUAL is signaled. This signal may be used for synchronization purposes.
I-STEP: Indicates that I-STEP mode is selected.

Line 15

BC	DAT	REAL ADDR	DATA	CHK CTL = I/O STOP
EC		VIRT		HARD STOP
				COMPATIBLE

BC: Basic control mode
EC: Extended control mode } according to bit 12 in the current PSW.
DAT: Dynamic address translation active.
REAL ADDR: Shows the actual address at which the machine will continue when re-started.
VIRT ADDR: Shows the virtual address at which the machine will continue when re-started.
DATA: Data associated with real or virtual addresses.
CHK CTL: Check control, representing a stop condition.
I/O STOP: I/O error (soft error only).
HARD STOP: Machine error (hard or soft error).
This stop occurs at the earliest time in an instruction or I/O command execution cycle at which a defined stop is possible. (see Note)
COMPATIBLE: Stop occurs after a machine check interruption or after a limited channel logout.

Note: If hard stop occurs none of the following is performed:

1. Machine Check Handling.
2. Log Handling.
3. Reload of Microprogram.

Line 16

- This line is reserved to display special messages for the Customer Engineer:

ID =	C=

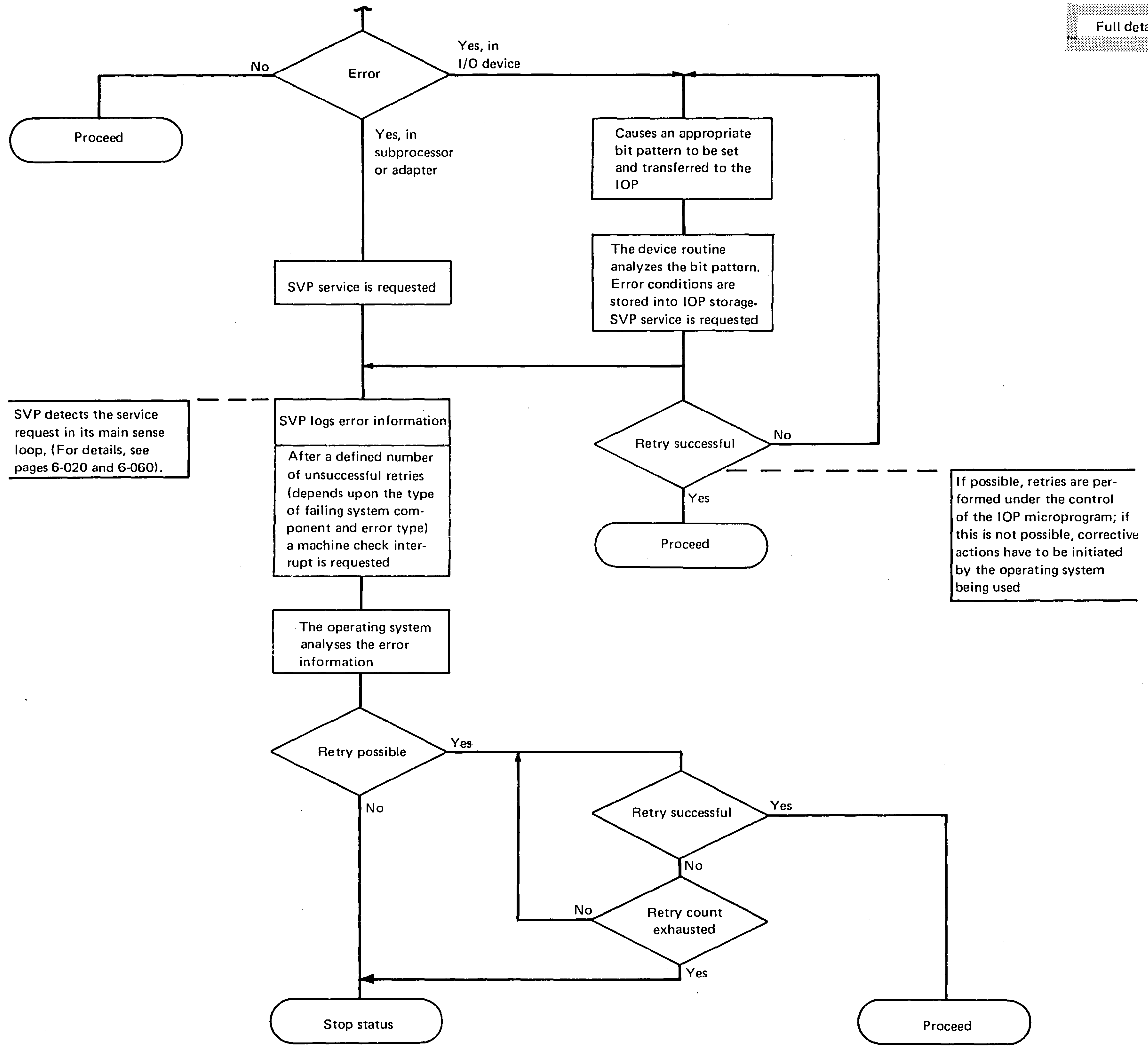
ID: Identifier. The name of the SVP phase that is currently being executed by the SVP is indicated.
C: Checks. The name of the phase in which a discrete read or write error occurred is indicated.

Notes:

1. The remainder of line 16 is used to display messages such as LOG IN PROCESS, or results of inline tests.
2. Information shown in line 16 is subprocessor- or adapter-dependent.
   Refer to *IBM 3125 Processing Unit, Central Test Manual.*

Error Types and Error Handling

Full details of error handling are given in *IBM 3125 Processing Unit, Central Test Manual.*



Error
No
Proceed
Yes, in I/O device
Yes, in subprocessor or adapter
Causes an appropriate bit pattern to be set and transferred to the IOP
The device routine analyzes the bit pattern. Error conditions are stored into IOP storage. SVP service is requested
SVP service is requested
SVP detects the service request in its main sense loop, (For details, see pages 6-020 and 6-060).
SVP logs error information
After a defined number of unsuccessful retries (depends upon the type of failing system component and error type) a machine check interrupt is requested
Retry successful
No
Yes
Proceed
If possible, retries are performed under the control of the IOP microprogram; if this is not possible, corrective actions have to be initiated by the operating system being used
The operating system analyses the error information
Retry possible
Yes
No
Retry successful
Yes
No
Retry count exhausted
No
Yes
Stop status
Proceed

Machine Malfunction Handling

- During execution of job programs, all subprocessors and adapters of the system are supervised by the SVP. This is done by the main sense loop of the SVP.
- The main sense loop of the SVP periodically addresses all subprocessors and adapters of the system.
- The addressed subprocessor or adapter transmits its *IDLE sense bit pattern.*
- The IDLE sense bit pattern consists of a four-bit wide subprocessor or adapter address, together with four bits of information that indicate any unusual condition in the addressed system component.
- The SVP initiates certain activities depending upon the information in the four information bits. The flowchart on the facing page illustrates the SVP activity.
- For information about the conditions under which the information bits are generated, refer to the appropriate subprocessor or adapter documentation.

The following symbols refer to the facing page.

1 Any IPU Check

This bit indicates whether any unusual condition exists in the IPU, irrespective of whether the condition is detected by the IPU circuitry or by the IPU microprogram.

2 IPU E-Stop

This is a result of IOP halt. After the SVP has reloaded the IOP, the SVP stops the IPU, transfers information to the IPU, and then restarts the IPU. The IPU subsequently switches to the machine check interrupt routine.

3 IOP Halt

This bit indicates that a hard error occurred in the IOP circuitry.

4 IOP PCR

This bit indicates that an error was detected by the microprogram.

5 SVP/MSC Check

During SVP/MSC communication, an error may be detected. This error is indicated by the MSC check bits, which cause the SVP/MSC check bit to be set (see "SVP/MSC Routine" on this page).

6 SVP Bus 0 Error

This bit indicates that a parity error exists on the SVP data bus 0.

A IOP Lock and SVP/MSC Lock

These bits ensure that, after an error has been logged, the complete main sense loop is executed. If the lock bit is not set to 'on', the supervising of all system components by the main sense loop is interrupted and the main sense loop idles in one of the check routines.

B MSC Successful

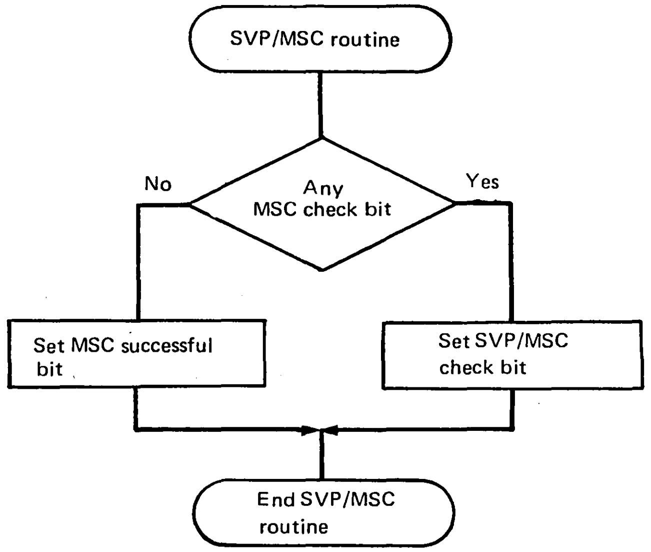
This bit prevents the logging of the same error condition more than once. When this bit is set to 'on', it ensures that at least one SVP/MSC communication between two errors was successful.

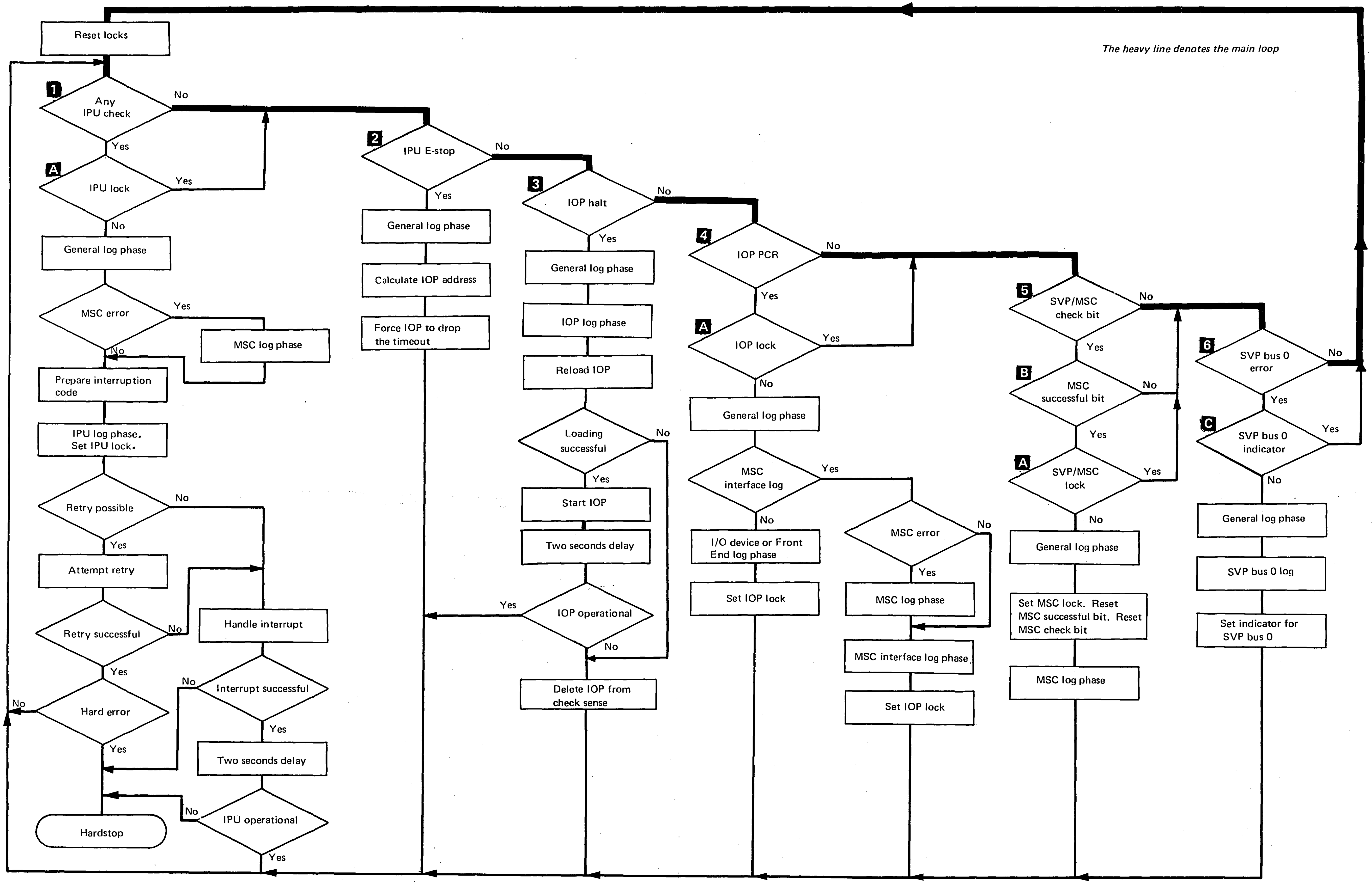
C SVP Bus 0 Indicator

This bit prevents the logging of consecutive SVP data bus 0 errors. It has the same basic principle as for B (above).

SVP/MSC Routine

- The SVP/MSC routine (shown below) is executed after each SVP/MSC communication.
- The bits set to "on" at the end of the routine indicate whether this communication was successful.
- If the communication was unsuccessful, the indication is shown by the SVP/MSC check (step 5) of the main sense loop (see flowchart on facing page).



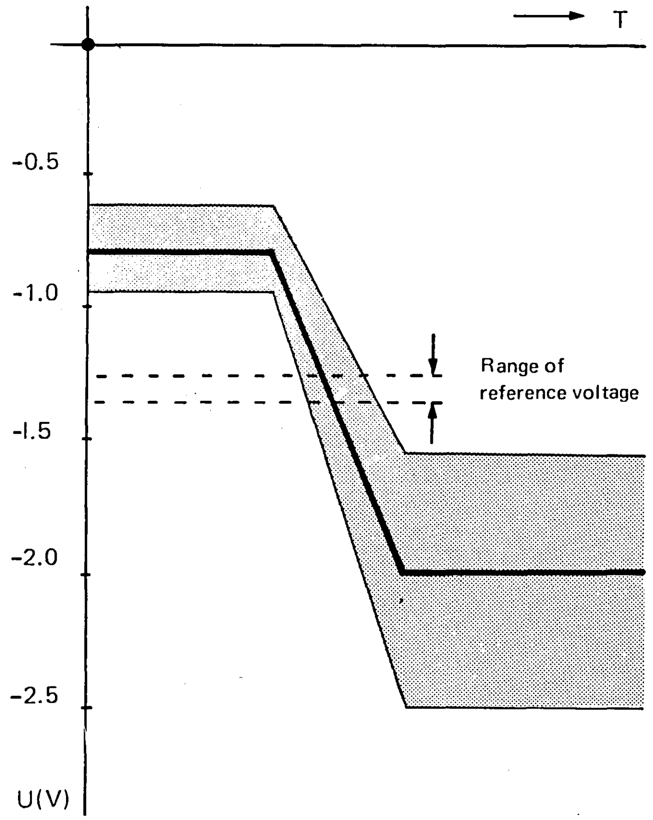


The heavy line denotes the main loop

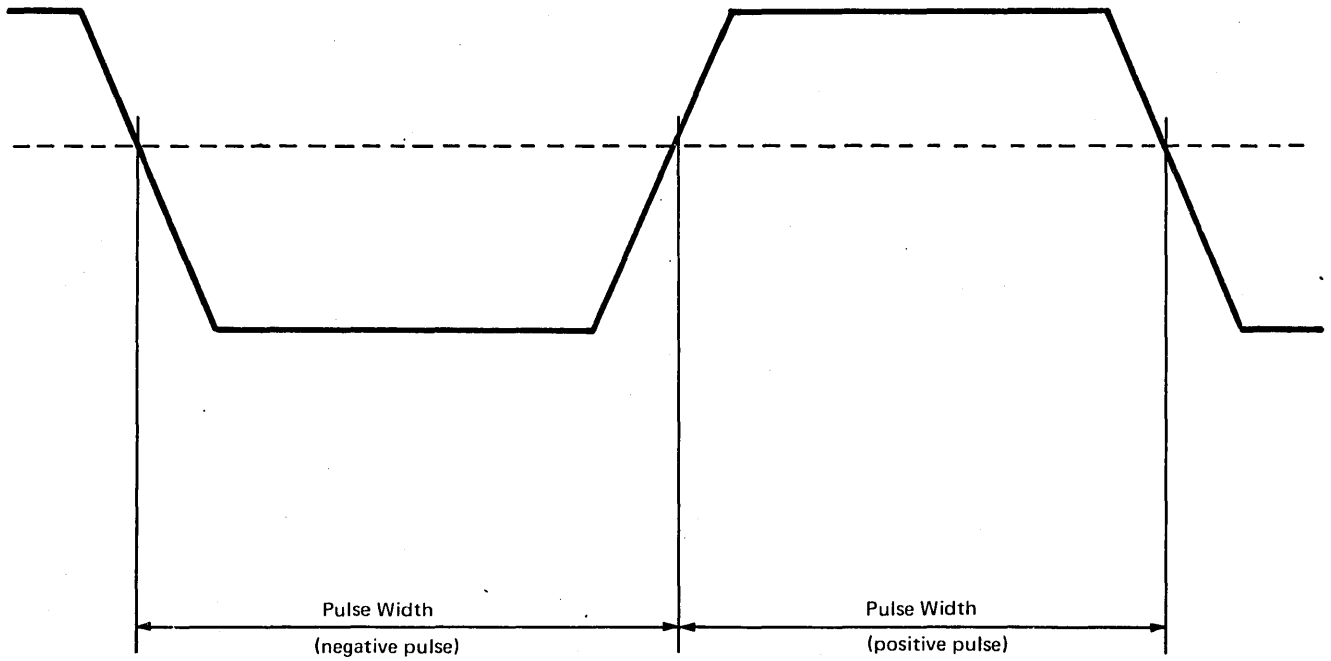
MST 1 Technique

Signal Levels

- MST 1 technique operates at the voltage levels shown below. For proper functioning, signal levels have to be within the limits of the shaded range.
- The center line (shown as a thick line) may be considered as an average voltage level. The line represents the ideal pulse shape.
- This ideal pulse shape is used to define pulse width or pulse duration.



Ideal Pulse Shape

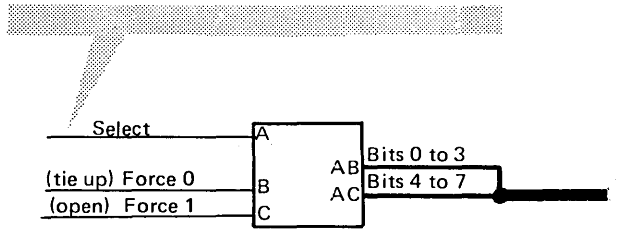


Note:
Pulse width and pulse duration are defined as the distance between the points of intersection of:
average reference voltage (denoted by broken line)
and
leading and trailing edges of ideal pulses.

Bias Module

- The bias module represents a small power supply that is normally located on each card.
- This power supply generates the reference voltage of −1.32V.
- The reference voltage depends upon the load and the temperature, and can vary between −1.27V and −1.32V. Use −1.3V as an average value for the reference voltage.
- A *tie up* voltage is also generated. This tie up voltage is used to satisfy logic blocks when necessary (see example below).
- Tie down voltage is not required because a feature of MST 1 technique is that all open input lines are at down level (see example below).

With the select signal active a bit pattern of 00001111 is forced.



Chapter 7. Reference Information

Abbreviations and Glossary

A

A	symbol for logical AND function
ABI	adapter bus in
ABO	adapter bus out
access cycle	IOP cycle, during which instructions are fetched
ACT	action
addr	address
adpt	adapter
ALD	automated logic diagram
ALU	arithmetic and logic unit
ASCP	automatic system checkout program
ATI	adapter tag in
ATO	adapter tag out
att	attachment
attach	attachment
attachment program	part of the IOP '8' microprogram, which is I/O specific and is therefore documented in the appropriate Front End MLMs

B

BAR	bus address register
base	one-byte wide value that defines a storage block
bc	bus check
BC	basic control
BCS	binary synchronous communication
BCU	bus control unit
BDR	bus data register
bias module	circuitry that generates reference voltages
bit	(from binary digit). Combinations of these smallest components are used to represent numerical, alphabetical, and special characters
bpi	bytes per inch
branch condition	a result or a condition used for logical decisions
BSB	backward space block
buffer	interim storage
burst mode	operating mode of the multiplexer channel when a device is not disconnected from the standard interface for the complete operation
bus	number of lines that interconnect system components (for example, subprocessors and adapters)
bus lines	lines transferring information that is identified by 'tag lines'
bwd	backward
byte	eight-bit combination that is used to represent numerical, alphabetical, and special characters by different bit combinations
byte count	the number of bytes to be transferred

C

CAW	channel address word. Points to the first CCW
CC	condition code
CC	command chaining
CCL	chain control line. CCLs define transfer conditions between the MSC and other subprocessors and adapters
CCW	channel command word. Defines conditions for I/O instructions
CD	chain data
CDF	console disk file
CDR	complement data register
CE	channel end
CE	control unit end
CE	customer engineer
chaining	allows a string of CCWs to be used
channel status	indicates conditions of a channel
ChE	channel end
CHK	check
clock	timing equipment to define distinct periods (cycles) of a subprocessor or adapter
cmd	command (code). Defines device operation
CMP	console matrix printer
CNSL	console
COMP	compare
complement	ALU operation as a result of sign analysis of the operands
control instruction	this instruction type transfers microprogram information into hardware
control program	is executed by the IOP in time slice mode. Performs housekeeping functions
control register	these are not real registers. The indicated circuits are set directly by the respective data bit when addressed
CP	console printer
crt	cathode ray tube
CSAR	control storage address register
CSW	channel status word. Defines conditions of an I/O operation
CTL	control
CTL	control instruction (machine language)
CTM	central test manual
CTRL	control instruction (microprogram language)
ctrl	control
ctrl stg	control storage
CU	control unit. Necessary to run I/O devices
CUB	control unit busy
CUE	control unit end

D

DAT	dynamic address translation
data address	address to/from which first data transfer is made
data fetch	storage read (data taken from storage)
data store	storage write (data set into storage)
DDA	direct disk attachment
DE	device end
dec	decoder
decr	decrement
decrement	subtract a predetermined value from an amount
delta cycle	shifted access or process cycle
DFT	diagnostic function test
digit	four-bit combination used to represent numerical values 0 to 15
disp	display
displacement	one-byte wide value that defines an individual storage position. (Has to be added to base)
doubleword	combination of eight bytes
DSE	data security erase
dummy module	a module without address, length, or relocation factor. Generated by the link program to provide forward references

E

E-phase	execution of an instruction
EC	engineering change
EC	extended control
ECSW	extended CSW
EOJ	end of job
EOT	end of tape
ERP	error recovery procedure
ESTV	error statistics by tape volume
ESV	error statistics by volume
EVA	error volume analysis
even	the exit of this circuit block becomes active with an even number of entries

F

FEALD	field engineering automated logic diagram
fcn	function
F'End	*see* "Front End"
FF	flip flop
FL	flip latch
"from" address	address where data is taken from
Front End	part of an I/O attachment
FRU	field replaceable unit
FSB	forward space block
fwd	forward

Abbreviations and Glossary (continued)

G

GSI	general system information

H

hard error	a hardware error that stops the system
HW	halfword (combination of two bytes)

I

I-count	instruction count
I-phase	instruction fetch cycle
I-step	instruction step
IBG	interblock gap
ICA	integrated communications adapter
ID	identifier
IDA	indirect data address(ing)
identifier	a symbol that identifies, indicates, or names a body of data
idle sense	information delivered by subprocessors and adapters when addressed by SVP main sense loop
IF-register	serves as communication register between attachment and control program
ILT	inline test
IMPL	initial microprogram load
inbound	main storage write operation (data is set into storage) or store operation
incr	increment
increment	add a predetermined value to an amount
INDAW	indirect data address word
index word	controls execution of IOP microprograms
instr	instruction
instruction	defines: actions to be performed; immediate data to be processed; "to/from" addresses of data
interrupt	interruption of the job program. A separate routine completes any I/O operation
interruption code	address of device from which interrupt is accepted (stored in PSW)
INV	invalid
inv	invert
I/O	input/output
IOP	input/output processor
IPL	initial program load
IPU	instruction processing unit
IRG	interrecord gap
irpt	interrupt

J

job program	customer program which handles customer problems and processes customer data

K

K	1024 bytes; used in referring to storage capacity
KB	keyboard
K/B	keyboard
key	codes of storage areas (storage blocks) that are used to allow (if equal) or to prevent (if unequal) access to storage locations

L

label	symbolic address
LCL	limited channel logout
LDR	left data register
LED	light emitting diodes
length count	*see* "byte count"
link	part of index word pointing to next index word
LL	line level
log	stored error information
log area	storage area where error information is stored
log record	string of bytes stored on the SVP service diskette when log is required
LRC	longitudinal redundancy check
LS	local storage
LWR	loop write to read

M

macroinstruction	defines a string of instructions
mag	magnetic
main sense loop	SVP microprogram loop that periodically senses all subprocessors and adapters
MALD	maintenance automated logic diagram
MAN	manual
MFT	microdiagnostic function test
microprogram	machine or system internal program that controls execution of job program instructions
MLM	maintenance library manual
modem	modulator-demodulator necessary to transfer data via communication lines
modifier (+1, –1)	circuitry that enables alteration of an amount
modify	alter an amount with a predetermined value
module	a program unit that is discrete and identifiable with respect to compiling, loading, and combining with other program units
MPX	multiplexer channel
MS	main storage
MSC	main storage controller
MSC data bus	exchanges information between MSC and other subprocessors and adapters
MST	monolithic storage technology
MTA	magnetic tape adapter
MTE	multiple track error
multiple byte mode	*see* "Multiplex Mode"
multiplex mode	operating mode of multiplexer channel when devices are disconnected from standard interface between each data byte transfer or between a predetermined number of data byte transfers

N

NRZI	non-return-to-zero change on ones recording. A modified form of non-return-to-zero recording in which a binary 1 is represented by a flux reversal on tape, with a binary 0 represented by the absence of such a change
ns	nanosecond
NSA	next sequential address

O

OBR	outboard recorder
OCL	octopus control line. Defines transfer conditions
odd	the exit of this circuitry block becomes active with an odd number of entries
OE	exclusive OR circuitry
OLT	online test
OLTEP	online test executive program
OLTSEP	online test standalone executive program
Op	operation
Op code	operation code. Defines action to be performed
op-reg	instruction from storage is set into operation register where the instruction is decoded and necessary signals for execution are activated
operating system	program which controls execution of one or more "job" programs
OR	logical OR function
OS	operating system
osc	oscillator
outbound	main storage read operation (data is read from storage) or fetch operation

P

paging	subdividing storage in storage blocks or "pages"
PC	parity check
pch	punch
PCR	program controlled request
PE	phase encoding
PG	parity generator
PH	polarity hold
phase encoding	a recording method where a binary 1 is represented by a magnetic flux change in a defined direction

PIL	potential independent logic
pointer	part of index word pointing to microinstruction to be executed next
POR	power-on reset
pr	printer
print option	the moment a type slug is positioned to a hammer for print
printscan	up to 50 printscans form one print line predetermined sequence in which requests are accepted and handled
process cycle	IOP cycle during which instructions are executed
prtr	printer
PS	power supply
PSC	print scan counter
PSW	program status word. Controls execution of job programs. It contains information of the currently executed instruction, and the address of the next instruction to be executed
pullover	entry to a circuitry. It directly changes the status independent of the levels of the regular set and reset lines

R

rd	read or reader
read operation	information is fetched
recomplement(ation)	ALU operation according to carry from high-order position to correct a result
register	circuitry that is able to store information
relocation factor	a value or constant used for address modification
requ	request
routine	group of instructions of a program either for common or multiple use

S

sens	sense instruction (microprogram language)
sense	sense instruction (microprogram)
sense instruction	this instruction type transfers circuitry conditions into microprogram
sense register	registers in which hardware conditions can be latched. This information can be fetched with a SENS operation
shared UCW	see "UCW"
sns	sense instruction, machine language
soft error	a software error. It does not necessarily stop the system
SS	singleshot
standard interface	standardized exit of I/O channels to which all types of I/O devices may be connected
subchannel	*see* "UCW"
subscan	three subscans form one printscan
supervisor	*see* "Operating System"
SVP	service processor
SVP addr bus	SVP address bus. Addresses the subprocessors and adapters of the system
SVP console units	are the display unit, keyboard, and console matrix printer, which are part of the SVP subsystem
SVP data bus	exchanges information between SVP and the subprocessors and adapters of the system
SYS	system
system mask	part of PSWs. Allows or prevents interrupts

T

TA	tape adapter
tag lines	lines that identify the type of data on 'bus lines'
TCU	tape control unit
TD	time delay
TDR	true data register (ALU of IPU)
TI	tape indicator
TIC	transfer in channel. Allows a string of CCWs not located on adjacent doubleword boundaries (ascending order of addresses) to be used
time slicing	allows two or more I/O devices (connected to one IOP) to be run simultaneously
timing	*see* "clock"
"to" address	address to which data is transferred
true	ALU operation as a result of sign analysis of the operands

U

UC	unit check
UCW	unit control word. Controls the execution of I/O operations. It contains information pertinent to the device. One UCW is required per device if these devices are to be operated simultaneously. If I/O devices cannot be operated simultaneously (for example, tape- or file-drive units) several I/O devices may use the same UCW. In this case this UCW is called a shared UCW
UE	unit exception
unit status	indicates conditions of an I/O device

V

V	volt
VID	volume identifier
VIRT	virtual

W

word	combination of four bytes
wr	write
write operation	information is stored

X

xing	paging
XOR	exclusive OR circuitry

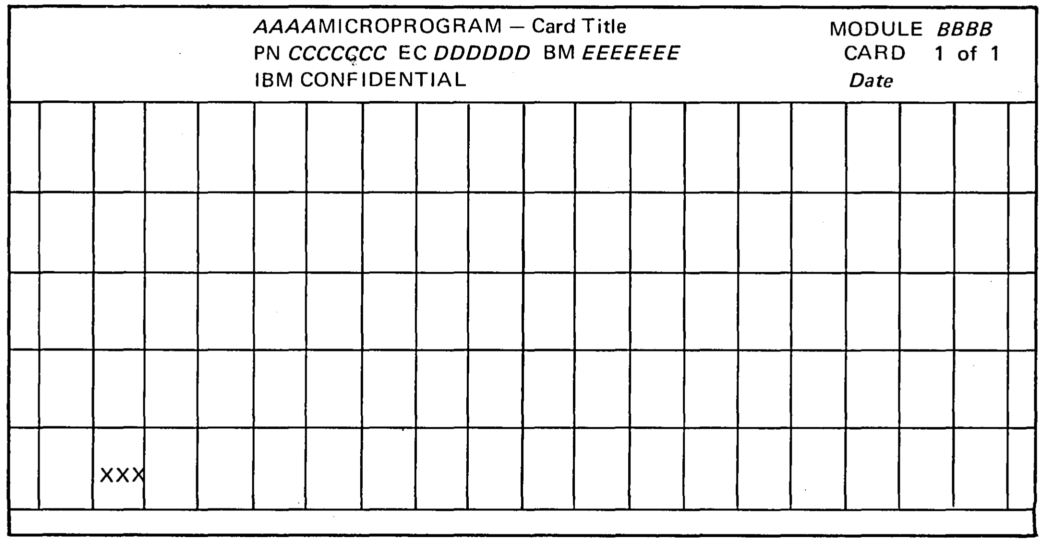
Appendix A. Programming Information

Microfiche Cards

- Microprogram listings are held on microfiche cards. These cards contain a *header area* and an *information area,* as shown on this page.
- A special viewer is needed so that the microfiche cards can be read.

Header Area

AAAA:	Machine type for this microprogram
BBBB:	Designation of microprogram module
CCCCCCC:	Part number of microfiche card
DDDDDD:	Engineering change number: this may also be an REA number
EEEEEEE:	Bill of material number
Date:	Release date of EC


AAAAMICROPROGRAM – Card Title
PN CCCCCCC EC DDDDDD BM EEEEEEE
IBM CONFIDENTIAL
MODULE BBBB
CARD 1 of 1
Date
XXX

Information Area

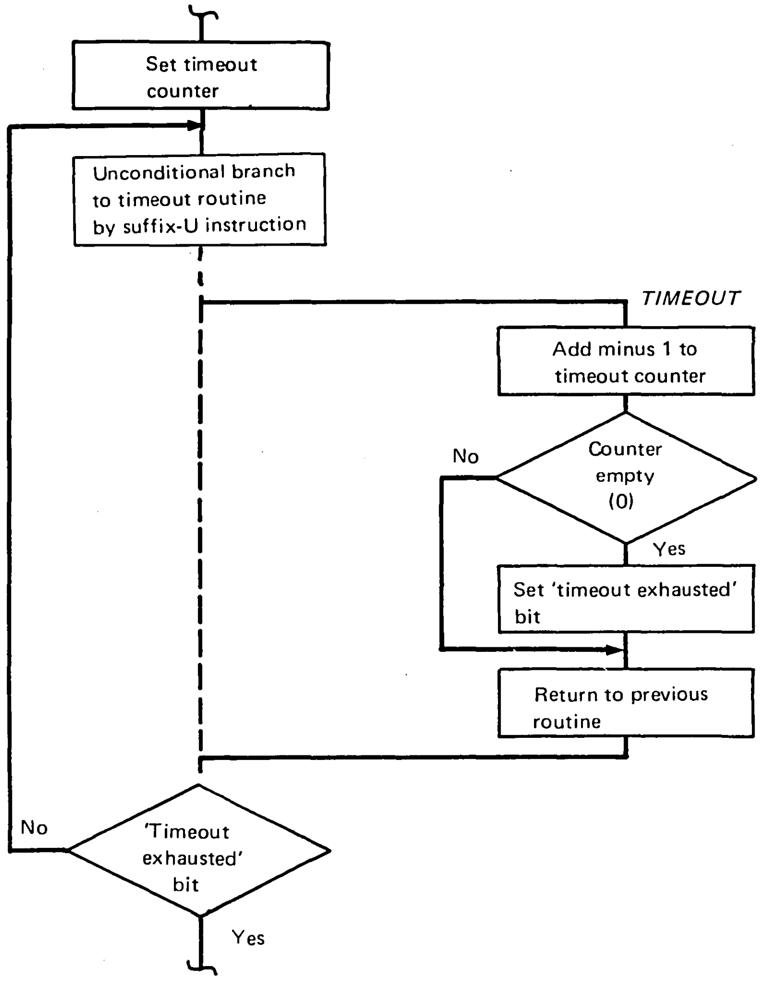
Consists of 5 x 18 subareas that allow 89 pages of a microprogram listing to be shown. The 90th subarea (XXX in lower left corner) is reserved for the header.

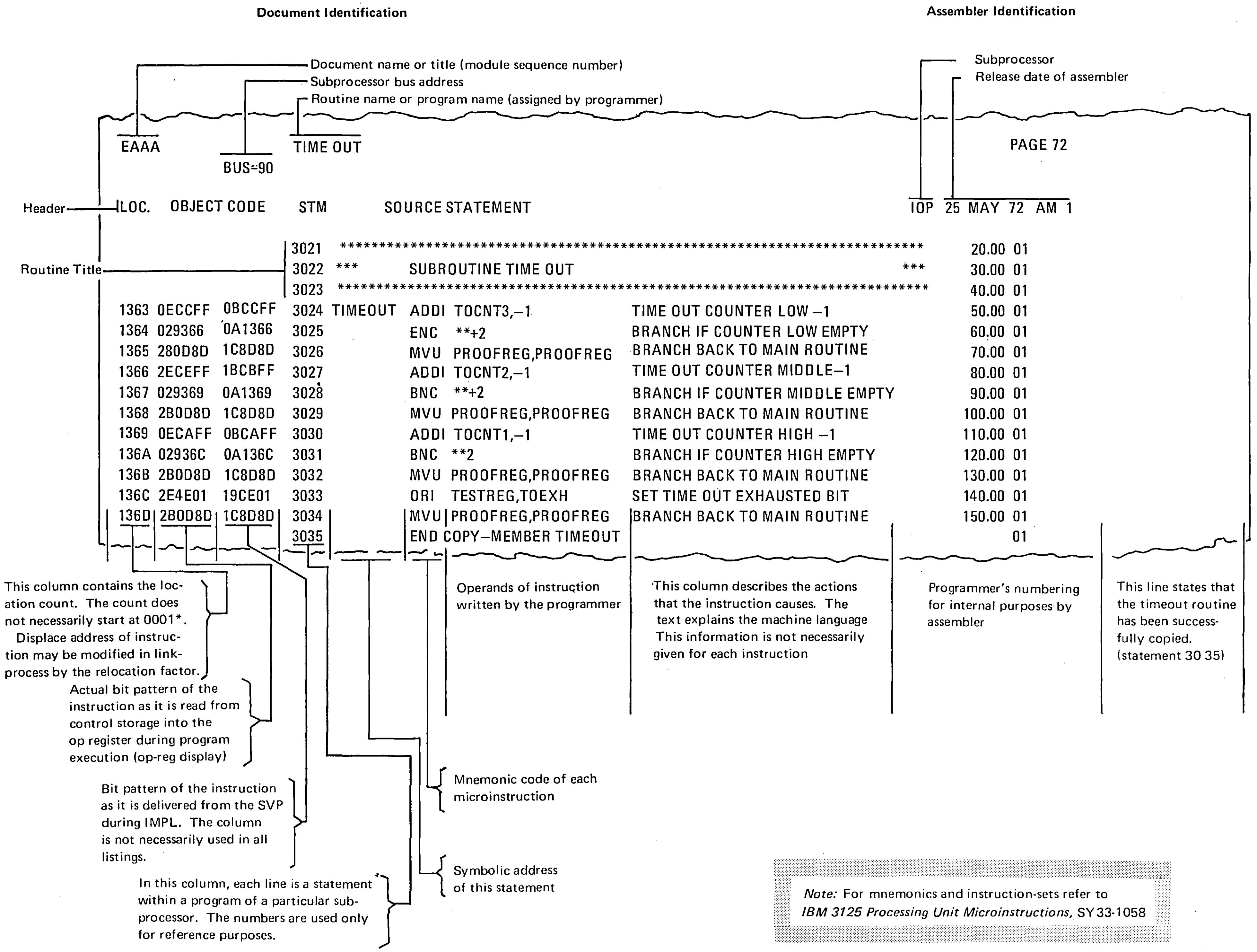
Explanation of Microprogram Listings

All control programs (IPU, IOP, and SVP) except ICA, are documented by flowcharts and microprogram listings.

Flowcharts show the principle of the microprograms and are a guide through the listings; the flowcharts are given in the appropriate maintenance library manuals. Listings show the details of each instruction and the sequence of execution; the listings are held on microfiche card (see page A-001). A portion of a typical microprogram listing, and the corresponding flowchart, is given on this page.

Information on microprograms is given on pages 3-010 and 3-020. The generation of microprograms and their listings by assembler programs is shown on page A-005.


Set timeout counter
Unconditional branch to timeout routine by suffix-U instruction
TIMEOUT
Add minus 1 to timeout counter
No
Counter empty (0)
Yes
Set 'timeout exhausted' bit
Return to previous routine
No
'Timeout exhausted' bit
Yes



Document Identification

Document name or title (module sequence number)

Subprocessor bus address

Routine name or program name (assigned by programmer)

Assembler Identification

Subprocessor

Release date of assembler

Header

Routine Title

```
EAAA                      TIME OUT                                                          PAGE 72
          BUS=90
LOC.  OBJECT CODE   STM      SOURCE STATEMENT                                       IOP 25 MAY 72 AM 1

                    3021 ************************************************************************  20.00 01
                    3022 ***        SUBROUTINE TIME OUT                                       ***  30.00 01
                    3023 ************************************************************************  40.00 01
1363 0ECCFF 0BCCFF  3024 TIMEOUT  ADDI TOCNT3,-1           TIME OUT COUNTER LOW -1             50.00 01
1364 029366 0A1366  3025          ENC  **+2                BRANCH IF COUNTER LOW EMPTY         60.00 01
1365 280D8D 1C8D8D  3026          MVU  PROOFREG,PROOFREG   BRANCH BACK TO MAIN ROUTINE         70.00 01
1366 2ECEFF 1BCBFF  3027          ADDI TOCNT2,-1           TIME OUT COUNTER MIDDLE-1           80.00 01
1367 029369 0A1369  3028          BNC  **+2                BRANCH IF COUNTER MIDDLE EMPTY      90.00 01
1368 2B0D8D 1C8D8D  3029          MVU  PROOFREG,PROOFREG   BRANCH BACK TO MAIN ROUTINE        100.00 01
1369 0ECAFF 0BCAFF  3030          ADDI TOCNT1,-1           TIME OUT COUNTER HIGH -1           110.00 01
136A 02936C 0A136C  3031          BNC  **2                 BRANCH IF COUNTER HIGH EMPTY       120.00 01
136B 2B0D8D 1C8D8D  3032          MVU  PROOFREG,PROOFREG   BRANCH BACK TO MAIN ROUTINE        130.00 01
136C 2E4E01 19CE01  3033          ORI  TESTREG,TOEXH       SET TIME OUT EXHAUSTED BIT         140.00 01
136D 2B0D8D 1C8D8D  3034          MVU  PROOFREG,PROOFREG   BRANCH BACK TO MAIN ROUTINE        150.00 01
                    3035          END COPY-MEMBER TIMEOUT                                            01
```

This column contains the location count. The count does not necessarily start at 0001*. Displace address of instruction may be modified in link-process by the relocation factor.

Actual bit pattern of the instruction as it is read from control storage into the op register during program execution (op-reg display)

Bit pattern of the instruction as it is delivered from the SVP during IMPL. The column is not necessarily used in all listings.

In this column, each line is a statement within a program of a particular subprocessor. The numbers are used only for reference purposes.

Mnemonic code of each microinstruction

Symbolic address of this statement

Operands of instruction written by the programmer

This column describes the actions that the instruction causes. The text explains the machine language This information is not necessarily given for each instruction

Programmer's numbering for internal purposes by assembler

This line states that the timeout routine has been successfully copied. (statement 30 35)

Note: For mnemonics and instruction-sets refer to *IBM 3125 Processing Unit Microinstructions*, SY33-1058

* With the relocation factor, the count starts at 0000. This location count specifies the actual storage address. For details, refer to "Link Program Listing", page A-003.

Link Program and Link Program Listing

- The output of assembler programs represents the actual machine language program. This output consists of program blocks (modules) that are not yet formatted, linked together, or relocatable.

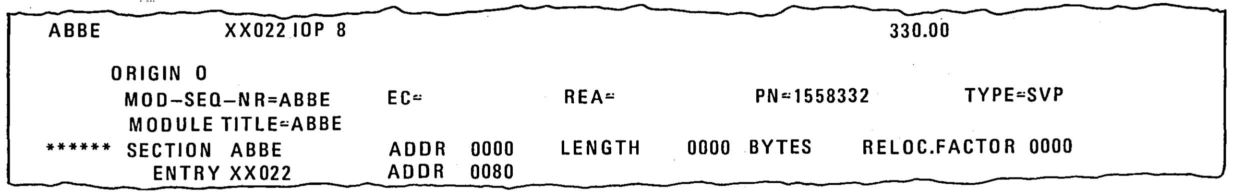
- The link program:
  - *Formats* the assembler output to loadable records, called "identifiers" (IDs). The length of the identifiers may vary and the number of identifiers generated from one module may also vary. In the link program listing, identifiers are separated by dotted lines.
  - *Links* the modules together. For proper linking, backward references as well as forward references must be possible. To allow forward references for programs that perform their job in one pass, dummy modules have to be generated first; therefore, the listing shown here starts with these dummy modules. The programmer defines the module that has to be generated as the dummy module.
  - *Relocates* the modules if necessary. This means that a relocation factor is assigned to the modules according to requirements. In different system configurations, the modules need to be placed in different storage areas; this can easily be done by changing the relocation factor.

  Each link program listing ends with information for the user, such as the comments:

  No error found during link
  Number of tracks written on diskette.

Link Program Listing

System Configuration



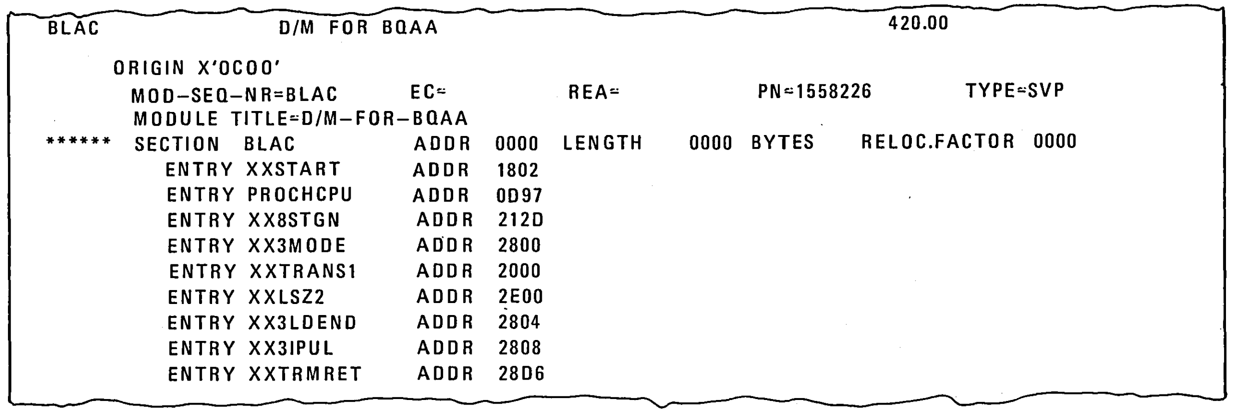
```
ABBE          XX022 IOP 8                                                    330.00

        ORIGIN 0
         MOD-SEQ-NR=ABBE      EC=              REA=              PN=1558332        TYPE=SVP
          MODULE TITLE=ABBE
****** SECTION  ABBE          ADDR  0000     LENGTH     0000  BYTES      RELOC.FACTOR  0000
          ENTRY XX022         ADDR  0080
```

Each link program listing starts with the system configuration, shown here by a number of typical dummy modules.

The first line of this module shows the MLC record. This record includes the module sequence number and title.

The indication of a dummy module is that address, length, and relocation factor are always zero.

Dummy Module (Allows Forward Linking)



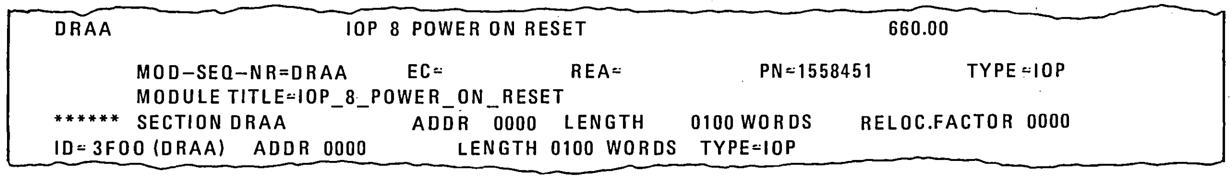
```
BLAC                     D/M  FOR  BQAA                                        420.00

        ORIGIN X'0C00'
         MOD-SEQ-NR=BLAC        EC=              REA=              PN=1558226        TYPE=SVP
          MODULE  TITLE=D/M-FOR-BQAA
******   SECTION   BLAC         ADDR   0000   LENGTH     0000  BYTES      RELOC.FACTOR  0000
            ENTRY  XXSTART      ADDR   1802
            ENTRY  PROCHCPU     ADDR   0D97
            ENTRY  XX8STGN      ADDR   212D
            ENTRY  XX3MODE      ADDR   2800
            ENTRY  XXTRANS1     ADDR   2000
            ENTRY  XXLSZ2       ADDR   2E00
            ENTRY  XX3LDEND     ADDR   2804
            ENTRY  XX3IPUL      ADDR   2808
            ENTRY  XXTRMRET     ADDR   28D6
```

The system configuration is followed by more dummy modules; these contain similar headers as the dummy modules above and listings of all their entry labels.

The first line shows the module sequence number of the dummy module. The line also shows the actual module sequence number of that module, to which forward references are made from other modules by this dummy module.

The indication of a dummy module is that address, length, and relocation factor are always zero.

Modules Stored on Diskette



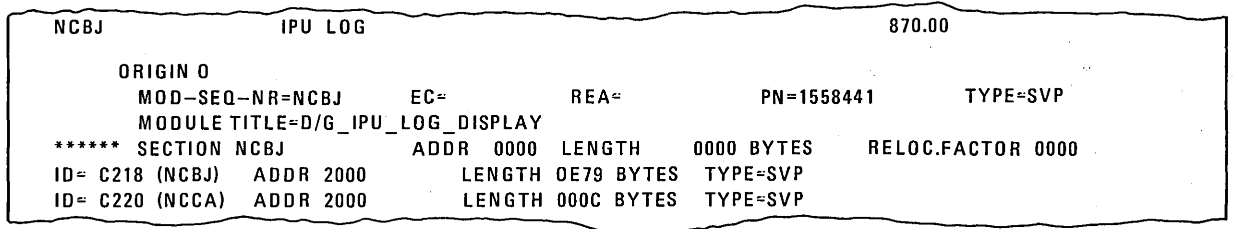
```
DRAA                          IOP  8  POWER  ON  RESET                          660.00

            MOD-SEQ-NR=DRAA       EC=              REA=              PN=1558451        TYPE=IOP
            MODULE TITLE=IOP_8_POWER_ON_RESET
******      SECTION DRAA              ADDR   0000   LENGTH     0100 WORDS      RELOC.FACTOR  0000
ID= 3F00 (DRAA)    ADDR  0000          LENGTH  0100  WORDS   TYPE=IOP
```

The dummy module portion is followed by the modules that are stored.

This example shows that one identifier was generated from one module, which also means that the section equals the identifier.

In addition to the module sequence number and module title, the following are shown here:

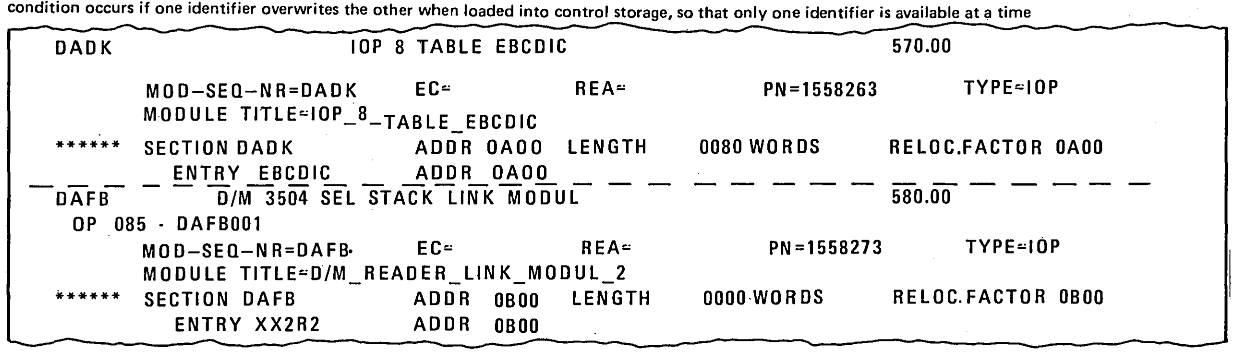
The start address of the identifier in control storage,
The length of the identifier,
A possible relocation factor



```
NCBJ                  IPU  LOG                                                 870.00

        ORIGIN 0
            MOD-SEQ-NR=NCBJ       EC=              REA=              PN=1558441        TYPE=SVP
            MODULE TITLE=D/G_IPU_LOG_DISPLAY
******      SECTION  NCBJ             ADDR   0000   LENGTH     0000  BYTES      RELOC.FACTOR  0000
ID=  C218  (NCBJ)    ADDR  2000          LENGTH  0E79  BYTES   TYPE=SVP
ID=  C220  (NCCA)    ADDR  2000          LENGTH  000C  BYTES   TYPE=SVP
```

This example shows that two identifiers were generated from one module. The section is of no significance.

Same comments as above regarding the identifier. In addition, the same start addresses of the two identifiers show an overlay condition. An overlay condition occurs if one identifier overwrites the other when loaded into control storage, so that only one identifier is available at a time



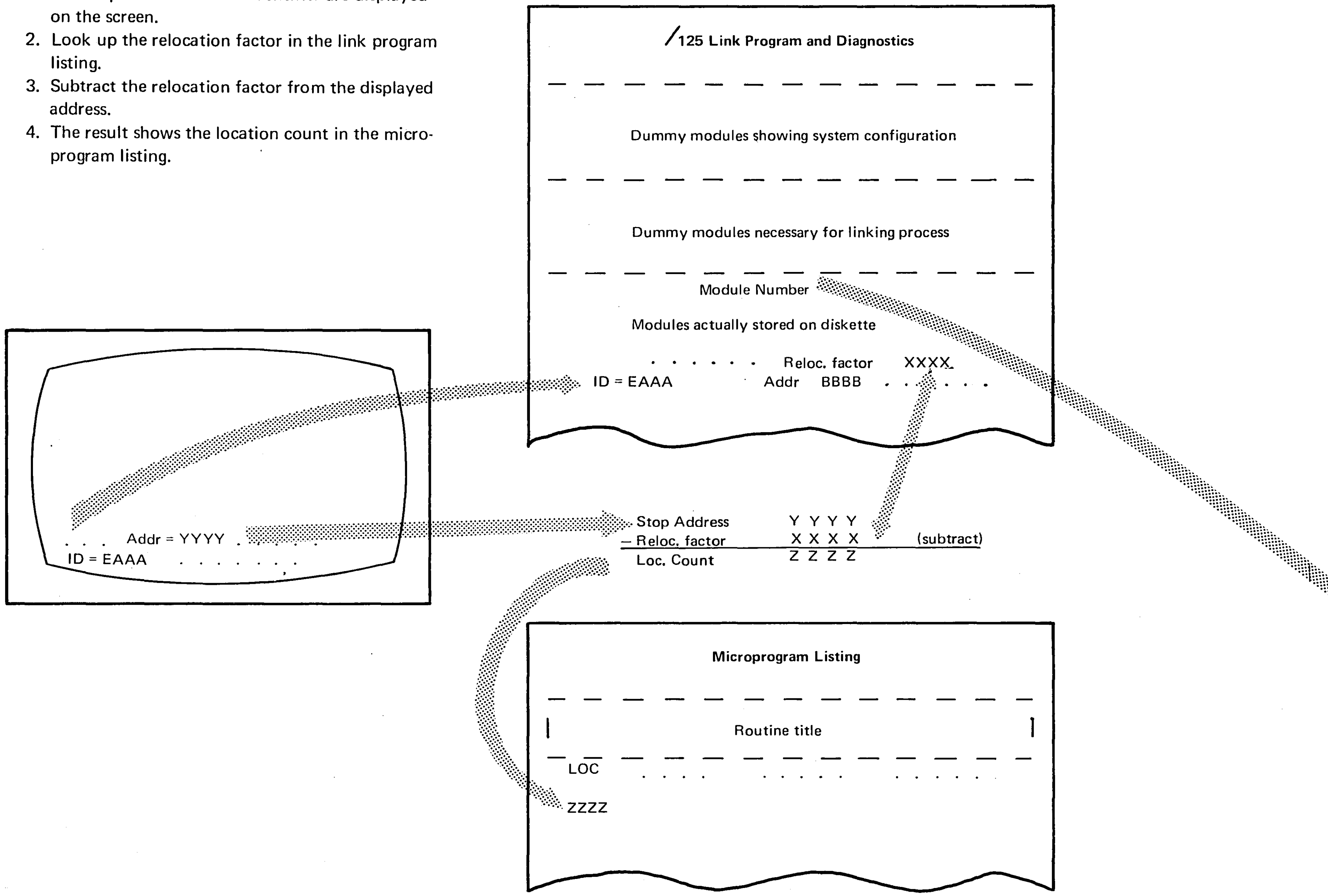
```
DADK                          IOP  8  TABLE  EBCDIC                             570.00

             MOD-SEQ-NR=DADK       EC=              REA=              PN=1558263        TYPE=IOP
             MODULE  TITLE=IOP_8_TABLE_EBCDIC
******       SECTION DADK              ADDR  0A00   LENGTH      0080 WORDS        RELOC.FACTOR  0A00
_ _ _ _ _ _ _ ENTRY  EBCDIC _ _ _ _ _  ADDR  0A00 _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
DAFB             D/M  3504  SEL  STACK  LINK  MODUL                             580.00
  OP  085 - DAFB001
             MOD-SEQ-NR=DAFB       EC=              REA=              PN=1558273        TYPE=IOP
             MODULE  TITLE=D/M_READER_LINK_MODUL_2
******       SECTION  DAFB            ADDR   0B00   LENGTH      0000 WORDS        RELOC.FACTOR  0B00
                ENTRY  XX2R2          ADDR   0B00
```

This example shows that several modules together generate an identifier. The sections define the modules within that identifier.

Same comments as above regarding the section. With the second section, a length of zero is specified, because reference is made to a dummy module

Link Program Listing (continued)

- The following information from the link program listing is important for the customer engineer:
  - *System configuration.* In this, the CE finds information about standard and optional features installed in a particular system.
  - *Start address and relocation factor.* Assume that the system has stopped. To determine which instruction either has been executed or will be executed next:
    1. The stop address and the identifier are displayed on the screen.
    2. Look up the relocation factor in the link program listing.
    3. Subtract the relocation factor from the displayed address.
    4. The result shows the location count in the microprogram listing.



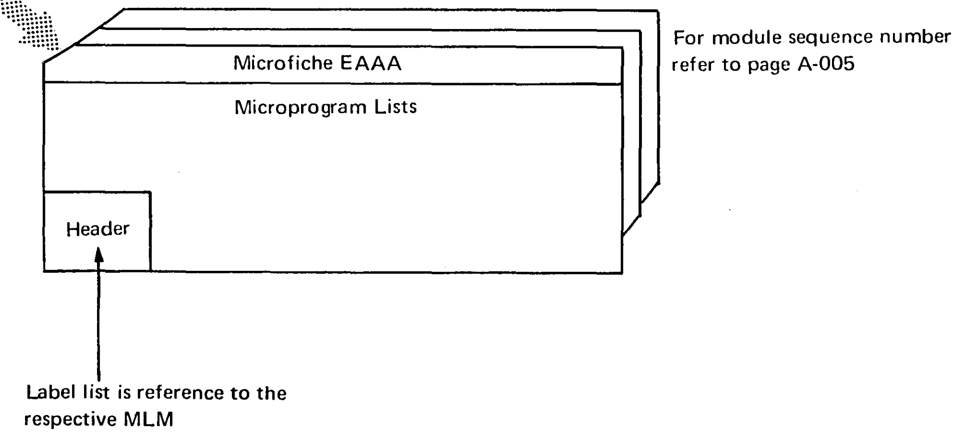
/125 Link Program and Diagnostics
Dummy modules showing system configuration
Dummy modules necessary for linking process
Module Number
Modules actually stored on diskette
Reloc. factor XXXX
ID = EAAA
Addr BBBB
. . . Addr = YYYY
ID = EAAA
Stop Address Y Y Y Y
– Reloc. factor X X X X (subtract)
Loc. Count Z Z Z Z
Microprogram Listing
Routine title
LOC
ZZZZ

Diskette Label

From data in the header record, the select and link program prepares the following 64-byte field as a label in the first track of the diskette.

The diskette label can be displayed on the CRT by pressing M I ENTER ENTER.

Byte Position	No. of Bytes	Contents		
1 to 4	4	Machine type and model		
5	1	(Blank)		
6 to 12	7	Machine serial number		
13	1	(Blank)		
14 to 16	3	Diskette sequence number, assigned by MLC for history		
17	1	Diskette type C: Control D: Diagnostic		
18	1	Shipment reason N: New built C: Change installation F: Feature installation E: Emergency For the letters N, C, and F, bytes 19 to 32 have the following meaning:		
		N	C	F
19 to 25	7	(Blank)	EC number	FFBM number
26	1	(Blank)	(Blank)	(Blank)
27 to 32	6	(Blank)	FCSI number	MES order number
33	1	(Blank)		
34 to 38	5	MLC run date		
39 to 64	26	(Blank)		



For module sequence number refer to page A-005

Label list is reference to the respective MLM

Module Sequence List

This is a list of the module sequence for System/370 model 125.

The module (program Block) is identified by a four-column alphabetic numbering system with a sequence from A–R.

ABAA ADRR	Dummy modules with XX labels
BBAA BLRR	External label definition without code
BMAA BRRR	SVP control program
CAAA CRRR	IPU control program
DAAA DRRR	IOP ‘8’; control programs 2560, 5424, 3504, 3525, 1403, 3203
EAAA ERRR	MPX control program; IOP ‘9’
FAAA	3330 Disk Storage; 3340 Direct Access Storage Facility
FRRR	IOP ‘A’ control programs
GAAA GRRR	IOP ‘B’; control programs ICA
HJAA HRRR	For manufacturing use only
NAAA NRRR	RAS programs on diskette 1 and/or 2
PAAA PRRR	RAS programs only on diskette 2
QAAA QRRR	Control programs of SVP
RAAA RRRR	RAS log areas on diskette 1

Model 125 Microprogram Assembler

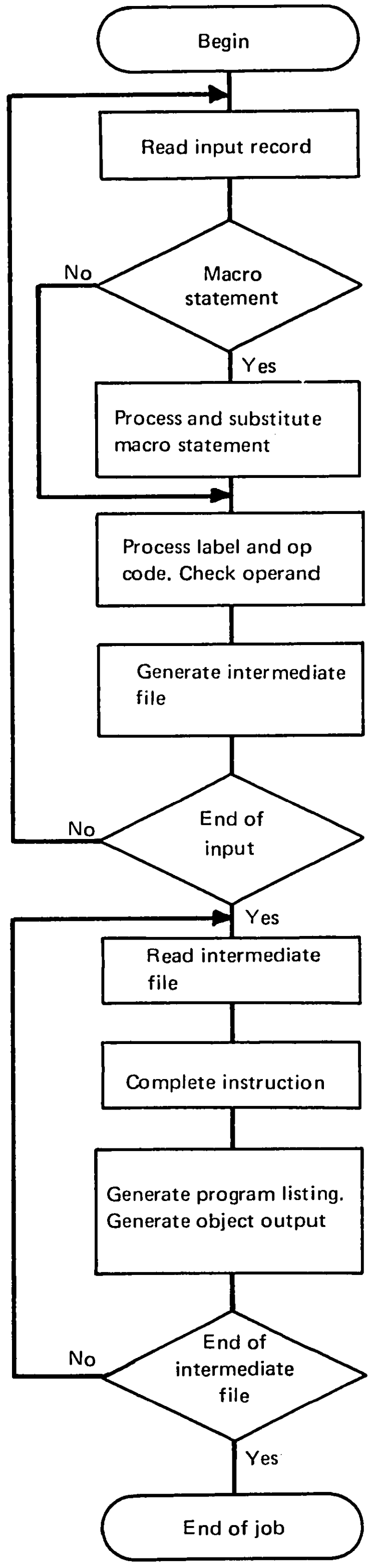
Programs are not easily generated in machine language, programmers use assemblers for this work. An assembler allows a symbolic language program to be used for program generation.

The assembler generates the machine-language program from the symbolic language program by substituting absolute op codes for symbolic op codes and absolute addresses for symbolic addresses.

The assembler uses its own language (called source language) that includes symbolic machine language statements directly corresponding to the instruction formats and data formats of the subprocessor.

It is not possible in this manual to give a detailed description of the System/370 Model 125 microprogram assembler and its language. The flowchart outlines the assembler operation; the instructions and statements are those that may appear in the IOP, IPU, and SVP microprogram listings.

Principle of Assembler Program


Begin
Read input record
No
Macro statement
Yes
Process and substitute macro statement
Process label and op code. Check operand
Generate intermediate file
No
End of input
Yes
Read intermediate file
Complete instruction
Generate program listing. Generate object output
No
End of intermediate file
Yes
End of job

Instructions

DC Defines data constants or address constants. Eleven different operands are available:

DC C'	for a character constant
DC X' ...	for a hexadecimal constant
DC Y ( )	for a two-byte address constant
DC H'	for a two-byte value constant
DC D ( )	for a D-type address constant
DC B ( )	for a B-type address constant
DC B'	for a binary constant
DC L ( )	for a link constant for determining whether a feature is installed
DC K ( )	for a K-constant used for inserting a value into a storage location if the expected bit pattern is not received
DC E ( )	for an exclusive address constant. Advantageous when branching to feature microprograms
DC D'	for a display constant used in connection with the data definition for a display unit in the System/370 Model 125

DS Reserves storage. Four different operands are permitted:

DS C DS X	Both cause one byte of storage to be reserved
DS H	Two bytes of storage are reserved
DS B	256 bytes of storage are reserved

ORG For switching to the next microprogram block, as specified by the operand of the ORG

ERU Associates meanings to distinct bit patterns or single bits, such as in a register

Statements

COPY Allows a common group of statements to be "copied".

RELIN Specifies release information such as:
Part number
EC number
Sequence number
REA number
RAS information

START Allows an initial value to be set for the location counter.

TITLE Enables the user to specify a page heading for a group of pages.

This page is intentionally left blank.

Appendix B. Sense Byte Information

Sense Bytes

Sense bytes represent detailed data which is additional to the unit status information. If information is required, in addition to unit status (as part of the CSW), an SIO instruction with a sense command in the CCW has to be initiated. This sense command causes the transfer of sense bytes into main storage in an ascending order of address, starting with the address specified in the CCW. Here the sense bytes are handled in the same way as data bytes during a data transfer. Sense bytes are generated individually by the I/O devices and their control units. Therefore the number of sense bytes varies depending upon device type. The first six bits of sense byte 0 are common to all devices that are generating sense information and have the following meaning:

Bit 0 — Command Reject

The device has detected a programming error. A command has been received which the device is unable to execute, such as 'read backward' issued to a direct-access storage device, or which the device cannot execute because of its present state, such as 'write' issued to a file-protected tape unit.

'Command reject' is also indicated when the program issues an invalid sequence of commands, such as 'write' to a direct-access storage device, without previously designating the data block.

Bit 1 — Intervention Required

The last operation could not be executed because of a condition requiring some type of intervention at the device. This bit indicates conditions such as an empty hopper in a card punch or the printer being out of paper. It is also turned on when the addressed device is in the not-ready state, is in test mode, or is not provided on the control unit.

Bit 2 — Bus-Out Check

The device or the control unit has received a data byte or a command code with an invalid parity over the I/O interface. During writing, bus-out check indicates that incorrect data have been recorded at the device, but the condition does not cause the operation to be terminated prematurely. Parity errors on command codes and control information cause the operation to be immediately terminated and suppress checking for 'command-reject' and 'intervention-required' conditions.

Bit 3 — Equipment Check

During the last operation, the device or the control unit has detected equipment malfunctioning, such as an invalid card hole count or printer buffer parity error.

Bit 4 — Data Check

The device or the control unit has detected a data error other than those included in bus-out check. Data check identifies errors associated with the recording medium and includes conditions such as reading an invalid card code or detecting invalid parity on data recorded on magnetic tape.

On an input operation, data check indicates that incorrect data may have been placed in main storage. The control unit forces correct parity on data sent to the channel. On writing, this condition indicates that incorrect data may have been recorded at the device. Unless the operation is of a type where the error precludes meaningful continuation, data errors on reading and writing do not cause the operation to be terminated prematurely.

Bit 5 — Overrun

The channel has failed to respond on time to a request for service from the device. Overrun can occur when data are transferred to or from a non-buffered control unit operating with a synchronous medium, and the total activity initiated by the program exceeds the capability of the channel. When the channel fails to accept a byte on an input operation, the following data in main storage are shifted to fill the gap. On an output operation, overrun indicates that data recorded at the device may be invalid. The overrun bit is also turned on when the device receives the new command too late during command chaining.

All information significant to the use of the device is normally provided in the first two bytes. Any bit positions following those contain diagnostic information. The amount and the meaning of the sense information are peculiar to the type of I/O device and are specified in the publication for the device.

I/O Sense Information

Console Printer

Sense Byte 0

The bits in sense byte 0 have the following assignments:

Bit	*Designation*
0	Command reject
1	Intervention required
2	(Not used)
3	Equipment check
4	(Not used)
5	(Not used)
6	(Not used)
7	(Not used)

Video Display

Sense Byte 0

The bits in sense byte 0 have the following assignments:

Bit	*Designation*
0	Command reject
1	Intervention required
2	(Not used)
3	(Not used)
4	(Not used)
5	(Not used)
6	(Not used)
7	Operation check

1403

Sense Byte 0

The bits in sense byte 0 have the following meanings assigned:

Bit	*Designation*
0	Command reject
1	Intervention required
2	Bus out check (not used)
3	Equipment check
4	Data check
5	Chain buffer parity check
6	(Not used)
7	Channel 9

Sense Byte 1

Sense byte 1 is not used, and contains all zeros.

Sense Byte 2

The bits in sense byte 2 represent six malfunctions, any one of which can cause the 1403 to lose its ready state, and the intervention required bit 0 to be set in sense byte 0.

Bit	*Designation*
0	Chain interlock
1	Forms check
2	Coil protect check
3	Subscan ring check
4	Chain buffer address register check
5	(Not used)
6	Any hammer on check
7	(Not used)

Sense Byte 3

Sense byte 3 is not used, and contains all zeros.

Sense Byte 4

The bits in sense byte 4 represent eight error conditions, any one of which can cause the equipment check bit to be set in sense byte 0.

Bit	*Designation*
0	Hammer reset failure check
1	No fire check
2	Misfire check
3	Print data buffer parity check
4	Check bit buffer parity check
5	Chain buffer parity check
6	Buffer address register parity check
7	Clock check

Sense Byte 5

Bit 0 in sense byte 5 represents one further error condition (in addition to those in sense byte 4) which, when set, causes the equipment check bit to be set in sense byte 0.

Bit	*Designation*
0	Open hammer coil check
1 to 7	(Not used)

3203

Sense Byte 0

The bits in sense byte 0 have the following meanings assigned:

Bit	*Designation*
0	Command reject
1	Intervention required
2	Bus out check (not used)
3	Equipment check
4	Data check
5	Chain buffer parity check
6	No channel found
7	Channel 9

Sense Byte 1

Sense byte 1 is not used.

Sense Byte 2

The bits in sense byte 2 represent eight conditions, any of which can cause the 3203 to lose its ready state. The setting of one of these bits causes the intervention required bit to be set in sense byte 0. The bits in sense byte 2 have the following meanings assigned:

Bit	*Designation*
0	Interlock (chain gate open)
1	Forms check (jam)
2	Coil protect check
3	Subscan ring check
4	Chain buffer address register check
5	Hammer unit shift check (applies to 3203 Model 1 only)
6	Any-hammer-on check
7	Device ready check

Sense Byte 3

The bits in sense byte 3 represent three types of error associated with carriage control. The setting of one of these bits causes the intervention required bit to be set in sense byte 0. The bits in sense byte 3 have the following meanings assigned:

Bit	*Designation*
0	(Not used)
1	(Not used)
2	(Not used)
3	Carriage inhibit check
4	(Not used)
5	(Not used)
6	Step check
7	Move check

Sense Byte 4

The bits in sense byte 4 represent eight error conditions, any one of which can cause the equipment check bit to be set in sense byte 0. An equipment check caused by a condition in sense byte 4 is a program-correctable error.

Bit	*Designation*
0	Hammer reset failure check
1	No fire check
2	Misfire check
3	Print data buffer parity check
4	Check bit buffer parity check
5	Chain buffer parity check
6	Buffer address register check
7	Clock check

Sense Byte 5

Bit 0 in sense byte 5 represents one further error condition (in addition to those in sense byte 4) which, when set, causes the equipment check bit to be set in sense byte 0. This equipment check is a program-correctable error.

Bit	*Designation*
0	Open coil check
1	(Not used)
2	(Not used)
3	(Not used)
4	(Not used)
5	(Not used)
6	(Not used)
7	(Not used)

2560

Sense Byte 0

The bits in sense byte 0 have the following meanings assigned:

Bit	*Designation*
0	Command reject
1	Intervention required
2	Bus-out check (not used)
3	Equipment check
4	Data check
5	Feed check/machine check
6	No card available
7	(Not used)

Sense Byte 1

Sense byte 1 contains detailed information about obstructions (jams, misfeeds, and so on) that caused a feed check. The individual bits are set to show where the card path was obstructed and have the following assignments:

Bit	*Designation*
0	Cover interlock
1	Jam bar check
2	Corner station check
3	Cell 8 to 9 feed check
4	Print station feed check
5	Punch station feed check
6	Read station feed check
7	Input station feed check

Sense Byte 2

Sense byte 2 contains information on the location of individual cards in the 2560 at the beginning of the cycle in which a feed check or machine check was detected. The bits in sense byte 1 have the following meanings assigned:

Bit	*Designation*
0	Secondary select
1	Card in punch station
2	Preprint SC 7 exposed
3	Prepunch SC 5 exposed
4	Prepunch SC 4 exposed
5	Preread SC 3 exposed
6	Preread SC 2 exposed
7	Input station SC 1 exposed

Sense Byte 3

Sense byte 3 contains the stacker selections assigned to cards at the primary and secondary prepunch stations, as shown in the following table.

Bit	*Meaning*	*Card Location*
0	primary card 0	Primary prepunch station
1	binary value 4	
2	binary value 2	
3	binary value 1	
4	secondary card 1	Secondary prepunch station
5	binary value 4	
6	binary value 2	
7	binary value 1	

Sense Byte 4

Sense byte 4 contains the stacker select numbers for two successive card stations. Bits 0 to 3 represent the stacker designation and the origin of the card in the punch or preprint station, bits 4 to 7 show the designation and origin of the card at the print station, as shown in the following table.

Bit	*Meaning*	*Card Location*
0	primary (0)/secondary (1)	Punch or preprint station
1	binary value 4	
2	binary value 2	
3	binary value 1	
4	primary (0)/secondary (1)	After print station
5	binary value 4	
6	binary value 2	
7	binary value 1	

Sense Byte 5

Sense byte 5 contains, in bits 0 to 3, the final stacker selection that is placed into the select magnets. Bits 4 to 7 hold the stacker number of the card that was just stacked. The origin of each card is also given, as shown in the following table.

Bit	*Meaning*	*Card Location*
0	primary (0)/secondary (1)	Corner station (to be stacked)
1	4	
2	2	
3	1	
4	primary (0)/secondary (1)	Stacker pocket (was just stacked)
5	4	
6	2	
7	1	

Sense Byte 6

Bit	*Designation*
0	Multi data check
1	Binary value 64
2	Binary value 32
3	Binary value 16
4	Binary value 8
5	Binary value 4
6	Binary value 2
7	Binary value 1

3340

Sense Byte 0

The bits in sense byte 0 have the following meanings assigned:

Bit	*Designation*
0	Command reject
1	Intervention required
2	(Not used)
3	Equipment check
4	Data check
5	Overrun
6	Track condition check
7	Seek check

Sense Byte 1

The bits in sense byte 1 have the following meanings assigned:

Bit	*Designation*
0	(Not used)
1	Invalid track format
2	End of cylinder
3	(Not used)
4	No record found
5	File protected
6	Write inhibited
7	Operation incomplete

Sense Byte 2

The bits in sense byte 2 have the following meanings assigned:

Bit	*Designation*
0	RPS feature present
1	Correctable
2	(Not used)
3	Environmental data present
4	(Not used)
5	(Not used)
6	Data module size
7	Data module size

Sense Byte 3

Sense byte 3 contains the restart command which is generated when the operation incomplete bit (sense byte 1, bit 7) is set. The restart command assists in identifying the operation which was in progress when the interruption, caused by the incomplete operation, occurred.

Sense Byte 4

Sense byte 4 identifies, in the following way, the physical drive and storage control that were addressed by a 'sense I/O' command.

Bit	*Identity*
0	Drive A
1	Drive B
2	Drive C
3	Drive D
4	Drive E
5	Drive F
6	Drive G
7	Drive H

Each of these physical drives may have been given any of the eight 12-bit logical device addresses, so sense byte 4 allows the SVP to relate a disk module to its assigned address during logging. The sense information is thus identified with a physical drive.

Sense Byte 5

Sense byte 5 identifies the eight low-order bits of the cylinder address in the most recent seek argument, as follows:

Bit		
0	128	Cylinder Number (low)
1	64	
2	32	
3	16	
4	8	
5	4	
6	2	
7	1	

Sense Byte 6

Sense byte 6 identifies the three high-order bits of the cylinder address and the four read/write head address bits in the most recent seek address, as follows:

Bit		
0	1024 Cylinder	Cylinder number (high)
1	512 Cylinder	
2	256 Cylinder	
3	0 (not used)	
4	8	Head number
5	4	
6	2	
7	1	

Sense Byte 7

Sense byte 7 has two functions. Firstly, it specifies the format of sense bytes 8 to 23. Secondly, it provides message tables which give additional information on errors. The table below shows how bits 0 to 3 of sense byte 7 specify the format of sense bytes 8 to 23.

Bits				Format of Sense Bytes 8 to 23
0	1	2	3	
0	0	0	0	Format 0: Programming or system check
0	0	0	1	Format 1: Device and control unit equipment check
0	0	1	0	Format 2: Disk attachment equipment check
0	0	1	1	Format 3: (Not used)
0	1	0	0	Format 4: Data checks not providing displacement information
0	1	0	1	Format 5: Data checks providing displacement information
0	1	1	0	Format 6: Usage/error statistics

For each of the formats shown in the table a unique message table is provided which defines the error condition more specifically. The error tables are formed by bits 4 to 7 of sense byte 7.

Sense Bytes 8 to 23

Sense bytes 8 to 23 define the various kinds of checks that can affect the 3340 disk subsystem. These bytes also provide usage/error statistics. The 16 bytes do not have unique assignments but their information content varies according to the format specified by bits 0 to 3 of sense byte 7. The seven formats available are listed in the description of "Sense Byte 7" on this page.

The error definitions given in sense bytes 8 to 23 may be of programming checks, system checks, equipment checks, or data checks, depending on the format. The usage/error statistics provide accumulated counts of significant events during subsystem operation, such as the number of bytes read and searched, and the number of access motions initiated by the channel.

I/O Sense Information (continued)

2311-1/3330 Feature

Sense Byte 0

The bits in sense byte 0 have the following meanings assigned:

Bit	*Meaning*
0	Command reject
1	Intervention required
2	(Not used [0])
3	(Not used)
4	Data check
5	(Not used)
6	(Not used)
7	Seek check

Sense Byte 1

The bits of sense byte 1 have the following meanings assigned:

Bit	*Meaning*
0	(Not used)
1	Track overrun
2	End-of-cylinder
3	Invalid sequence
4	No record found
5	File protected
6	(Not used)
7	Overflow incomplete

Sense Byte 2

Sense byte 2 is not used by the 2311-1/3330 feature and is always zero.

Sense Byte 3

The bits in sense byte 3 have the following meanings assigned:

Bit	*Meaning*
0	Ready
1	Online
2	(Not used)
3	(Not used)
4	Online
5	End-of-cylinder
6	(Not used)
7	(Not used)

Sense Byte 4

Sense byte 4 is not used.

Sense Byte 5

Sense byte 5 contains a hexadecimal number which identifies the command in progress when an overflow incomplete condition is detected. It also gives extra information for commands which involve a comparison.

Hexadecimal value of sense byte 5	*Meaning*
04	Read command was in progress.
05	Write command was in progress.
25	'Search key and data equal' was in progress, and comparison is equal to this point.
45	'Search key and data high' was in progress, and comparison is equal to this point.
65	'Search key and data high or equal' was in progress and comparison is equal to this point.
55	'Search key and data' was in progress and comparison is low, or a 'search key and data equal' was in progress and comparison is high.
75	'Search key and data high' or 'search key and data high or equal' was in progress and comparison is high.

3504

The sense information for the 3504 is similar to that for the 3525 Card Punch. The setting of unit check indicates that significant sense information is available in the two sense bytes provided. The following paragraphs describe the contents of sense bytes 0 and 1. (Sense bytes 2 and 3 contain all zeros for a natively-attached 3504.)

Sense Byte 0

The bits in sense byte 0 have the following meanings assigned:

Bit	*Designation*
0	Command reject
1	Intervention required
2	Bus out check (not used)
3	Equipment check
4	Data check
5	Overrun (not used)
6	Abnormal format reset
7	Permanent error key

Sense Byte 1

Bits 0 to 3 in sense byte 1 are set for situations which can require operator or operating system action. The bit assignments are shown in the following table:

Bit	*Designation*
0	Permanent error
1	Automatic retry
2	Motion malfunction
3	Retry after intervention complete
4 to 7	(Not used)

3525

The sense information for the 3525 is similar to that for the 3504 Card Reader. The setting of unit check indicates that significant sense information is available in the two sense bytes provided. The following paragraphs describe the contents of sense bytes 0 and 1. (Sense bytes 2 and 3 contain all zeros for a 3525 attached via the integrated card I/O attachment.)

Sense Byte 0

The bits in sense byte 0 have the following meanings assigned:

Bit	*Designation*
0	Command reject
1	Intervention required
2	Bus out check (not used)
3	Equipment check
4	Data check
5	Overrun (not used)
6	Abnormal format reset
7	Permanent error key

Sense Byte 1

Bits 0 to 3 in sense byte 1 are set for situations which can require operator or operating system action. The bit assignments are shown in the following table:

Bit	*Designation*
0	Permanent error
1	Automatic retry
2	Motion malfunction
3	Retry after intervention complete
4 to 7	(Not used)

5425

The following paragraphs describe the contents of the eleven bytes of sense information provided for the 5425.

Sense Byte 0

The bits in sense byte 0 have the following meanings assigned:

Bit	*Designation*
0	Command reject
1	Intervention required
2	Bus-out check
3	Equipment check
4	Data check
5	Overrun (not used)
6	No card available
7	(Not used)

Sense Byte 1

The bits in sense byte 1 act as check condition indicators which are used for error recovery procedures. Some of the bits in sense byte 1 are also summary indicators of more detailed conditions represented by the bits in sense bytes 3, 4, and 5. The bits in sense byte 1 have the following meanings assigned:

Bit	*Designation*
0	Read check
1	Punch check
2	(Not used)
3	Print data check
4	Print clutch check
5	Hopper check
6	Feed check
7	(Not used)

Sense Byte 2

Sense byte 2 contains information about the location of cards in the 5425 transport at the time the 'sense' command is executed. The bits of sense byte 2, when set, do not cause status indications. The bits have the following meanings assigned:

Bit	*Designation*
0	(Not used)
1	(Not used)
2	Card in primary wait station
3	Card in secondary wait station
4	(Not used)
5	Hopper cycle not complete
6	Card in transport counter bit 2
7	Card in transport counter bit 1

Sense Byte 3

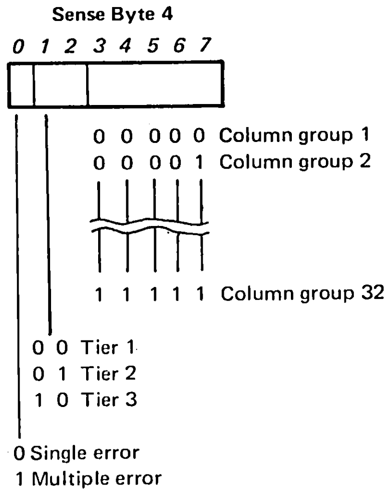
Sense byte 3 contains a hexadecimal number whose value can represent any one of 21 feed checks and emitter checks in the 5425. It is used for error logging and analysis. All checks represented in this byte are 5425 hardstops. All the check conditions (except stacker jam, gear emitter check, and fire CB check) are activated, via a fiber optic bundle, by photo-electric cells in the card path. The state of these cells, in conjunction with timing circuits, enables checks to be recognized when cards are not in their correct positions in the card path. Checks are reset by depression of the NPRO key. Any one of these checks will set the feed check bit in sense byte 1. The checks also light a numbered feed check light on the 5425 operator panel. Values of sense byte 3, check names, and operator panel light numbers are shown in the following table.

Hexadecimal Value of Sense Byte 3	Check Name	Operator Panel Light
01	Hopper eject check	1
02	Read inject check	2
03	Read station check	3
04	Read eject check	4
05	Early wait eject check	5
06	Wait eject check	6
07	Punch inject check	7
08	Punch registration check 1	8
09	Punch station check	9
0A	Punch registration check 2	10
0B	Punch transport check	11
0C	Punch eject check	12
0D	Corner station check	13
0E	Corner eject check	14
0F	Print inject check	15
10	Print station check	16
11	Print eject check	17
12	Stacker transport check	18
13	Stacker jam	19
14	Gear emitter check	20
15	Fire CB check	20

Sense Byte 4

Sense byte 4 defines the card column group and tier where the error was detected which caused the first read check or punch check of a card cycle. If more than one read check or more than one punch check occurs during a card cycle, a multiple check indicator in sense byte 4 will be set. The read check (bit 0) and punch check (bit 1) bits in sense byte 1 show whether sense byte 4 contains read check or punch check information. In the unusual case where a read check and a punch check occur during the same card cycle, bits 0 and 1 will both be set in sense byte 1, which means that the contents of sense byte 4 will be undefined.

The illustration shows how sense byte 4 defines whether a single error or multiple errors occurred and, if a single error, the tier and column group in which the error was detected.



Sense Byte 5

Sense byte 5 specifies the row or rows for the tier and column identified in sense byte 4 in which a read check or punch check error occurred. Thus, sense bytes 4 and 5 together define the position of the error which caused the first read check or punch check during a card cycle down to one or more of the 24 read cells or punches or their associated circuits. The bits in sense byte 5 have the following meanings assigned:

Bit	*Designation*
0	D row miscompare
1	C row miscompare
2	B row miscompare
3	A row miscompare
4	8 row miscompare
5	4 row miscompare
6	2 row miscompare
7	1 row miscompare

Sense Bytes 6, 7, 8, 9, and 10

Sense bytes 6, 7, 8, 9, and 10 form a table of the five most recent command strings. When a new last command string appears in sense byte 6, the previous contents of sense byte 6 are shifted down to sense byte 7, the previous contents of sense byte 7 are shifted down to sense byte 8, and so on. (A command string starts with the first command following any command causing a feed and ends with the next command causing a feed.)

The bits in sense byte 6 refer to the most recent command string. This command string may not have been carried out completely because of an error condition occurring after acceptance of a command causing one feed but before acceptance of a command causing the next feed. Sense byte 6 contains information about the last command accepted. The bit assignment is the same in each of the bytes (6 to 10).

Bit	*Designation*
0	Secondary
1	Print four lines
2	Stacker select M2
3	Stacker select M3
4	Punch
5	Feed command sample
6	Print
7	Read

3410/3411

Sense Byte 0

The bits in sense byte 0 have the following meanings assigned:

Bit	*Designation*
0	Command reject
1	Intervention required
2	Bus out check (not used)
3	Equipment check
4	Data check
5	Overrun
6	Word count zero (not used)
7	Data converter check (not used)

Sense Byte 1

The bits in sense byte 1 have the following meanings assigned:

Bit	*Designation*
0	Noise
1	Tape unit status A
2	Tape unit status B
3	7-track TU (not used)
4	At load point
5	Write status
6	File protected
7	Not capable

Sense Byte 2

The bits in sense byte 2 contain information that is used only by the control unit itself, not the program. This is because the bits represent the track-in-error information. For PE read operations, each bit that is set represents a track that has a phase error or is dead. (A dead track is one that is damaged, or has never been written on.) For PE write operations, each bit that is set represents a track that has an envelope check (see "Sense Byte 3") or phase error.

During NRZI read operations, the bits in sense byte 2 represent the cyclic redundancy check (CRC) information. During NRZI write operations, sense byte 2 is not used and contains the code 03 (hex).

Sense byte 2 is made available to the tape control unit when the 'request track in error' command is given.

Sense Byte 3

The bits in sense byte 3 have the following meanings assigned:

Bit	*Designation*
0	Vertical redundancy check
1	Multiple track error (PE) or longitudinal redundancy (NRZI)
2	Skew
3	End data check (PE) or cyclic redundancy check (NRZI)
4	Envelope check (PE only)
5	1600 bpi
6	Backward
7	C-compare (not used)

Sense Byte 4

The bits in sense byte 4 have the following meanings assigned:

Bit	*Designation*
0	Tape unit positioning check
1	Tape unit reject
2	End of tape
3	(Not used)
4	(Not used)
5	Diagnostic track check
6	Tape unit check
7	Illegal command

Sense Byte 5

The bits in sense byte 5 have the following assignments:

Bit	*Designation*
0	New subsystem
1	New subsystem
2	Write tape mark check
3	PE identification burst
4	PE compare (not used)
5	Tachometer check
6	False end mark
7	Reserved for RPQ

Sense Byte 6

The bits in sense byte 6 have the following meanings assigned:

Bit	*Designation*
0	7-track unit (not used)
1	Short gap mode
2	Dual density
3	Alternate density
4 to 7	Tape unit model

I/O Sense Information (continued)

Sense Byte 7

The bits in sense byte 7 have the following meanings assigned:

Bit	*Designation*
0	Lamp check
1	Left column check
2	Right column check
3	Ready reset
4	Data security erase
5	(Not used)
6	(Not used)
7	(Not used)

Sense Byte 8

The bits in sense byte 8 have the following meanings assigned:

Bit	*Designation*
0	(Not used)
1	Feedthrough
2	(Not used)
3	End velocity check
4	No read-back data
5	Start velocity check
6	(Not used)
7	(Not used)

Integrated Communications Adapter

One byte of sense information is available. The bits in sense byte 0 have the following assignments:

Bit	*Designation*
0	Command reject
1	Intervention required
2	Bus out check (not used)
3	Equipment check
4	Data check
5	Overrun
6	Lost data
7	Timeout complete

Note: All conditions indicated in sense byte 0 set unit check in the CSW.

Index

3125 Processing Unit,
General System Information

Order No. SY33-1059-1

READER'S COMMENT FORM

Your views about this publication may help improve its usefulness; this form will be sent to the author's department for appropriate action. Using this form to request system assistance or additional publications will delay response, however. *For more direct handling of such requests, please contact your IBM representative or the IBM Branch Office serving your locality.*

Possible topics for comment are:

Clarity Accuracy Completeness Organization Index Figures Examples Legibility

What is your occupation? ________________________________

Number of latest Technical Newsletter (if any) concerning this publication: ____________

Please indicate 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.)

Cut Along Line

3125 Processing Unit,
General System Information

Order No. SY33-1059-1

READER'S COMMENT FORM

Your views about this publication may help improve its usefulness; this form will be sent to the author's department for appropriate action. Using this form to request system assistance or additional publications will delay response, however. *For more direct handling of such requests, please contact your IBM representative or the IBM Branch Office serving your locality.*

Possible topics for comment are:

Clarity Accuracy Completeness Organization Index Figures Examples Legibility

What is your occupation? ________________________________

Number of latest Technical Newsletter (if any) concerning this publication: ____________

Please indicate 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.)

SY33-1059-1

Your comments, please . . .

This manual is part of a library that serves as a reference source for customer engineers. 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

First Class
Permit 40
Armonk
New York

Business Reply Mail
No postage stamp necessary if mailed in the U.S.A.

Postage will be paid by:

International Business Machines Corporation
Department 813B
1133 Westchester Avenue
White Plains, New York 10604

Fold Fold

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)

Cut Along Line

SY33-1059-1

Your comments, please . . .

This manual is part of a library that serves as a reference source for customer engineers. 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

First Class
Permit 40
Armonk
New York

Business Reply Mail
No postage stamp necessary if mailed in the U.S.A.

Postage will be paid by:

International Business Machines Corporation
Department 813B
1133 Westchester Avenue
White Plains, New York 10604

Fold Fold

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)

System
Maintenance
Library

System

— — — cut here — —

3125 Processing Unit, General System Information Printed in U.S.A. SY33-1059-1

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)