SYSTEM/360 DOS CONSOLE: A SELF-INSTRUCTIONAL GUIDE

SYSTEM/360 DOS CONSOLE: A SELF-INSTRUCTIONAL GUIDE

ULTIMACC Systems, Inc.

George D. Weinstock
Robert A. Weber
Albert G. Franco
Fred David

PRENTICE-HALL, INC.

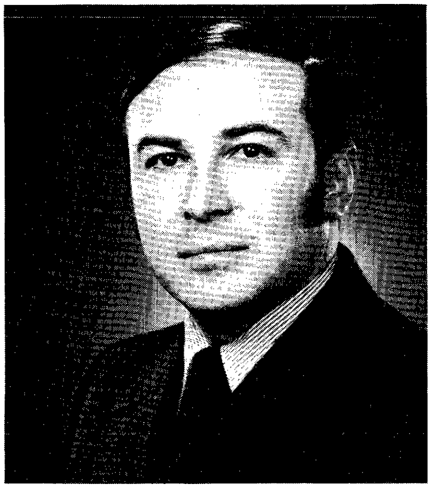
Englewood Cliffs, N.J.

Prentice-Hall International, Inc., *London*
Prentice-Hall of Australia, Pty. Ltd., *Sydney*
Prentice-Hall of Canada, Ltd., *Toronto*
Prentice-Hall of India Private Ltd., *New Delhi*
Prentice-Hall of Japan, Inc., *Tokyo*

Library of Congress
Catalog Card Number: 74-161678

PRINTED IN THE UNITED STATES OF AMERICA
ISBN–0–13–881821–5
B & P

ULTIMACC Systems, Inc.



GEORGE D. WEINSTOCK

Vice President—Operations and Marketing

George D. Weinstock's experience covers twelve years in the computer and communications system areas. His previous affiliations include staff positions at Computer Sciences, RCA and ITT. He is the author of eight articles in the information system sciences.

ROBERT A. WEBER

President

Robert A. Weber is an expert in the design and implementation of management information systems utilizing System/360's as well as other computers. He was formerly with Computer Sciences Corporation as Director of Information Systems Division, and his overall experience covers fifteen years in data processing and communications systems.

ALBERT G. FRANCO

Vice President—Engineering

Albert G. Franco has long been involved in the analysis and design of data processing communications. An expert in evaluating and implementing the integration of computers and communications systems, he has served as Adjunct Professor of Electrical Engineering at the Newark College of Engineering. His previous positions include staff and managerial posts at Computer Sciences and IBM.

FRED DAVID

Vice President—Product Development

Fred David has had seventeen years experience in the technical design and management of computer–communications systems. He is a specialist in computer-controlled switching systems for transfer of data and in the use of minicomputers in conjunction with System/360's and other large scale computers. He was previously in staff and managerial positions at Computer Sciences Corporation and Stromberg-Carlson.

ACKNOWLEDGMENT

The authors wish to express their appreciation to Messrs. Raymond Hill and Allen Biggs for their technical contributions, and Miss Fran Beckwermert for her patience and diligence in typing and proofreading the manuscript.

HOW THIS BOOK WILL HELP YOU MASTER THE SYSTEM/360 DOS CONSOLE

This book is written to enable the man who has a working knowledge of computers and their operation to master the operation of the IBM System/360 Disk Operation System (DOS). The man in question here is a corporate institutional Data Processing Manager, the Management Information Systems Director, an operator of second generation computer or unit record equipment, a programmer, an engineer, a comptroller, or even a corporate administrative vice president. Each of these men can benefit from this book in different ways. The corporate executive will learn how his computer system works and thus be in better position to more efficiently plan, schedule, and implement new company programs. The person who has had previous experience as an operator on second generation computers or unit record equipment will be able to upgrade himself and improve his understanding of modern complex computer systems.

The programmer will improve his working efficiency by being able to expand his software knowledge with the computer equipment operation and thus, for example, reduce his debugging time. The engineer by mastering computer operation can utilize his scientific and programming skills to efficiently solve his problems. In other words, this book is directed toward a variety of people to enable them to more efficiently utilize computer technology in their everyday business lives.

The operator of a modern computer is like a command pilot guiding a modern jet. The computer, like the aircraft, is only a machine and it requires a skilled operator to control it. To be in command the operator must learn certain basic procedures and statements. Sometimes, as in an aircraft, the operator will have to refer to operators' manuals for additional information. This book will guide the operator in finding additional information. But as in an aircraft, the operator, to efficiently run his ship (or computer system), must be knowledgeable in both the procedures and the reasons why the equipment or system responds to the procedures. Many a crash (both aircraft and computer) has been prevented by operators who understood both the procedures and the reasons behind them due to the system hardware and software operation. And this is the major benefit of this book as opposed to other documents or sources compiled with respect to System/360 DOS operation. The in-depth development of both the mechanical operating procedures and the reasons behind them as a result of the system structure provide a new basis for the reader to truly master the operation and control of such a complex system as the 360.

WHAT THIS BOOK WILL DO FOR YOU

This book will provide you with all the "tricks of the trade" associated with operating a System/360 under DOS. You will have a very good understanding of what the computer is doing when you give it commands from the operator's console. You will know what commands to give and what action to take when the computer requires information or cannot handle a problem. You will be able to recognize the symptoms of various computer system faults, and know what techniques and procedures to use in overcoming them. What are the functions of a System/360 DOS console operator? The following paragraphs provide the answer.

The computer operator has well defined duties to perform while he operates the system. His specialty is to set up the computer and its peripheral units accurately and in the minimum time possible. The operator is the interface between the programmer and the computer. In working directly with the equipment, the operator has an important impact on the efficient operation of the computer. The computer must not be allowed to be idle for significant periods between jobs, or when a job "hangs up," or when the programmer has not provided sufficient information to run the job to completion. The operator's role can be summarized with respect to the equipments he is responsible for during the processing of jobs as follows:

1) The console control panel—This panel allows the operator to visually determine the status of the system and permits him to control the processing operation (i.e. start, stop, etc.)

2) The console typewriter or keyboard printer—This device serves as the principal means for communication between the operator and the computer program, and is frequently used to print error messages and instructions to the operator.

3) The card read-punch—This device feeds input data to the computer and punches out any computer solution data required in card format. The operator checks card sequences and may add control cards.

4) The magnetic tape, drum, and disk file units—These are the auxiliary mass storage

units which significantly increase the volume of data that can be handled by the computer. They must be loaded and removed rapidly.

5) The line printer–This output device gives computer answers in the form of printed sheets, checks, ledger entries, parts lists, etc. This device requires adjustment and loading of proper forms.

In order to operate these equipments the operator must not only be familiar with them but must also possess mechanical dexterity to be able to quickly and accurately manipulate each unit as required. He must learn, for example, how to load a reel of tape on a tape unit, a disk pack into a disk unit, and blank forms into the line printer. If the operator has the normal mechanical skills these tasks can be learned without much difficulty. The operator must also learn certain basic commands on the console typewriter in order to "talk" to the computer. These mechanical skills must be supplemented by an understanding of how to prepare the system and run jobs. To this end the operator must learn how to read and interpret the instructions in the *run book*, keep logs of the events that occur during the running of a job, and most importantly, how to operate the console panel to control the computer system. Each of these operator tasks will be described in detail in the book. Under normal conditions the computer is capable of supervising itself with little or no manual intervention for routine tasks. The measure of an operator, however, and the prime purpose of this book is to enable the operator to identify and correct the unusual situation and to handle the more routine tasks with increased speed and accuracy.

HOW THIS BOOK IS ORGANIZED

Mastery of System/360 DOS console operation is organized into six chapers. Each chapter is designed as a building block to the subsequent chapter so that the reader will build his knowledge tree much in the same manner that the computer builds its knowledge tree. Where appropriate in each chapter, illustrations and step-by-step instructions are utilized to clarify operator procedures and explain system software structure.

The material presented in the first three chapters is designed to acquaint the reader with the working of System/360, the role of the console operator, and basic operational procedures and practices necessary for running jobs on a Model 30 installation. In particular, these chapters cover the following topics:

Chapter 1 – This chapter introduces the reader to System/360 and indicates why the Model 30 offers the best unit for DOS console operation mastery. Every component of a Model 30 installation is described in detail and its functions explained from the operator's viewpoint.

Chapter 2 – This chapter discusses and illustrates the mechanical aspects of operating System/360. The various controls and procedures for operating each component of the system are described on a step-by-step basis. Each component is characterized by its physical functions, controls, procedures, special cautions and "tricks of the trade."

Chapter 3 – This chapter reveals how the operator is to combine his knowledge of working with each component individually into the operating of the entire system in the running of jobs. The procedures and techniques for initializing, running, and stopping of systems jobs are described in detail. Two sample systems jobs are presented which permit the reader to test his knowledge in the planning and implementation of the runs.

At this point the reader has become knowledgeable with respect to the workings of System/360, the mechanical controls and procedures in setting up the various system components, and the techniques for running relatively simple jobs. System/360, however, is a large, complex computer system and as a result many different jobs are required to be run in short periods of time. These jobs utilize different Input/Output units and in order to efficiently run these jobs as rapidly as possible necessitates certain automation aids for the console operator. These aids are in the form of sets of programs and have been denoted by IBM as operating systems. The most popular Input/Output devices from the storage and speed of information access viewpoint is the Disk Storage Drive System. As a result the operating system used is known as the Disk Operating System (DOS). The final three chapters of the book therefore are devoted to the understanding of System/360 structure and its utilization of the Disk Operating System, the techniques for operator detection and correction of system errors using DOS, and general operator practices in running jobs under DOS. Specifically, these chapters include the following:

Chapter 4 – This chapter introduces the reader to the System/360 Disk Operating System. The structure of System/360 is described in detail to enable the reader to understand how and why various procedures, instructions and controls are required in operation. The organization and procedures used with DOS are presented with numerous examples to illustrate the meaning and reasoning behind them.

Chapter 5 – This chapter discusses the different types of system errors that can occur during operation. The methods for detecting these errors and taking corrective action are described in detail with examples of typical types of errors. Both component and system errors are covered.

Chapter 6 – This chapter concerns itself with the rules and procedures to be followed by the computer operator to insure that jobs he runs under DOS are processed efficiently and accurately. These good operator habits are important to establish as early as possible to promote immediate benefits in easing the operator's task and increasing the efficiency of the data processing activity.

George D. Weinstock
Robert A. Weber
Albert G. Franco
Fred David

CONTENTS

HOW THIS BOOK WILL HELP YOU MASTER THE SYSTEM/360 DOS CONSOLE 7

1. INTRODUCTION TO SYSTEM/360 17

1.1 APPEARANCE OF THE SYSTEM/360 MODEL 30 17
1.2 THE PROCESSOR 25
1.3 INPUT/OUTPUT DEVICES 26
1.4 STORAGE MEDIA 28
1.5 CONTROL PANEL AND OPERATOR'S CONSOLE 31
1.6 REGISTERS 34

2. MECHANICAL ASPECTS OF COMPUTER OPERATION 36

2.1 CONSOLE TYPEWRITER (KEYBOARD-PRINTER) 36
2.1.1 Switches 40
2.1.2 Keys 40
2.2 CARD READER-PUNCH (2540) 41
2.3 MAGNETIC TAPE UNIT 45
2.4 MAGNETIC DISK FILES 49
2.5 THE LINE PRINTER 53
2.6 CONSOLE CONTROL PANEL 66

3. RUNNING A JOB 72

3.1 PRELIMINARY ORGANIZATION 72
3.2 INITIALIZATION 73
3.3 OPERATING PROCEDURES 76
3.4 SUMMARY OF CONSOLE OPERATOR'S ROLE 79
3.5 SAMPLE JOBS FOR PRACTICE IN SETTING UP SYSTEM/360 JOB RUNS 80

4. INTRODUCTION TO SYSTEM/360 DISK OPERATING SYSTEM 91

4.1 NUMBERING SYSTEMS 92
4.2 CPU ORGANIZATION 95
4.3 PROGRAM EXECUTION 100
4.4 CHANNELS 104
4.5 MULTIPROGRAMMING 105
4.6 STRUCTURE OF THE DISK SYSTEM 112
4.7 JOB ENTRY 114
4.8 COMMUNICATING WITH DOS 115
4.9 ALLOCATING I/O DEVICES 116
4.10 OPERATING PROCEDURES 119

5. ERROR DETECTION AND RECOVERY PROCEDURES. 125

5.1 DEVICE ERROR RECOVERY 125
5.1.1 Card Reader Errors 126
5.1.2 Card Punch Errors 128
5.1.3 Line Printer Errors 129
5.1.4 Tape Errors 132
5.1.5 Disk Errors 133

5.2 ERROR RECOVERY USING INTERNAL SYSTEM DOS HARDWARE AND SOFTWARE AIDS 134
5.2.1 System-to-Operator Messages 135
5.2.2 Job Control Errors 139

5.3 CONSOLE BYPASS 140
5.3.1 IPL Error Messages 141
5.3.2 Device Error Recovery 141

5.4 RESTART AND SAVE PROCEDURES 142

6. GENERAL OPERATOR PRACTICE 144

6.1 LOGS 144
6.2 PRIORITIES 145
6.3 JOB RUNNING 146
6.4 OTHER GOOD OPERATOR PRACTICES 148
6.5 DOS OPERATOR TRAINING SYSTEM EXERCISE 153

INDEX. 167

ILLUSTRATIONS

Figure		Page
1-1	BASIC SYSTEM/360 MODEL 30 STRUCTURE AND DATA FLOW	18
1-2	DATA FLOW WITHIN IBM 2030 PROCESSING UNIT	26
1-3	IBM 2540 CARD READ-PUNCH	27
1-4	IBM 1403 LINE PRINTER	28
1-5	IBM 2401 MAGNETIC TAPE UNIT	29
1-6	IBM 2311 DISK STORAGE DRIVE	30
1-7	IBM 1052 MODEL 8 KEYBOARD	31
1-8	IBM SYSTEM/360 MODEL 30 CONTROL PANEL AND OPERATOR'S CONSOLE	32
1-9	IBM SYSTEM/360 MODEL 30 CONTROL PANEL	33
2-1	IBM 1052 PRINTER-KEYBOARD (CONSOLE TYPEWRITER)	37
2-2a	IBM 1052 MODEL 8 KEYBOARD AND SWITCH PANEL	38
2-2b	IBM 1052 MODEL 6 KEYBOARD AND SWITCH PANEL	39
2-3	IBM 2540 CARD READ-PUNCH: FILE FEED, CARD READ HOPPER, JOGGLER GATE, AND CARD WEIGHT	43
2-4	SCHEMATIC DIAGRAM OF IBM 2401 TAPE FEED	45
2-5	STRIPPING OFF TAPE LEADER	47
2-6	THREADING TAPE	48
2-7	PLACING COVER OVER DISK PACK	50

2-8 COVER OFF DISK PACK 51
2-9 REMOVING BOTTOM PLATE 52
2-10 LOWERING DISK PACK OVER SPINDLE 53
2-11 LINE PRINTER (WITH COVER RAISED) SHOWING MANUAL CONTROLS 54
2-12 IBM 1403 PRINTER MODEL N1 INDICATORS AND PUSHBUTTONS 55
2-13 FEED CLUTCH CONTROL 57
2-14 PRINT UNIT OPEN, SHOWING PAPER DRIVE 58
2-15 1403 TRACTORS, SHOWING LEFT-HAND UPPER TRACTOR OPEN 59
2-16 USE OF RIBBON SHIELD FOR ALIGNMENT 61
2-17 LATERAL ALIGNMENT OF PRINT MECHANISM 62
2-18 VERTICAL ALIGNMENT DETAIL 63
2-19 TESTING TAPE TENSION 64
2-20 MODEL 30 CONTROL PANEL 67
3-1 SAMPLE SALES REPORT PRINTOUT 80
3-2 SCHEDULE FOR CARD PROGRAM 81
3-3 RUN SHEET: CARD TO PRINT SALES REPORT 82
3-4 DECK OF 5 TIME CARDS 84
3-5 SCHEDULE SHOWING PAYROLL JOB 85
3-6 RUN SHEET: PAYROLL CHECK PROGRAM 86
3-7 TYPICAL DEVICE ADDRESSES 87
3-8 RUN SHEET: REGISTER PROGRAM 89
4-1 PROGRAM STATUS 101
4-2 CORE STORAGE ORGANIZATION IN A DOS MULTIPROGRAMMING SYSTEM 113
4-3 A JOB STREAM 115
6-1 SAMPLE JOB STREAMS 149
6-2 OS DUMP RESTORE 152

SYSTEM/360 DOS CONSOLE:
A SELF-INSTRUCTIONAL GUIDE

1

INTRODUCTION TO SYSTEM/360

In this chapter you are going to learn about a specific computer system, the IBM System/360, and a particular model, namely Model 30. You may ask why we have written this text to teach you to operate the IBM System/360 Model 30. The answer is very simple and practical. IBM accounts for 75% of all computers sold throughout the United States. The current generation (third generation) of IBM computers is the System 360 and of this series 80% of the sales are for Models 20 and 30.

The Model 20 installation is generally a minimum one, and will not use the range of peripheral equipments or operator aids found in a typical Model 30 installation. The Model 30 is one of the most common computers in use today and from a practical viewpoint if you learn how to operate it efficiently you will be better able to help yourself and your company.

1.1 APPEARANCE OF THE SYSTEM/360 MODEL 30

In a manner similar to any other computer system, the IBM System/360 Model 30 comprises a processing unit with its associated internal core memory, a console unit for the operator, and one or more Input and Output units. The computer operator works in close coordination with the System/360-30. He must be thoroughly familiar with all the equipments in the installation.

The System/360 Model 30 is a high-speed computer capable of solving problems of a scientific nature as well as complex commercial tasks. Figure 1-1 depicts the basic

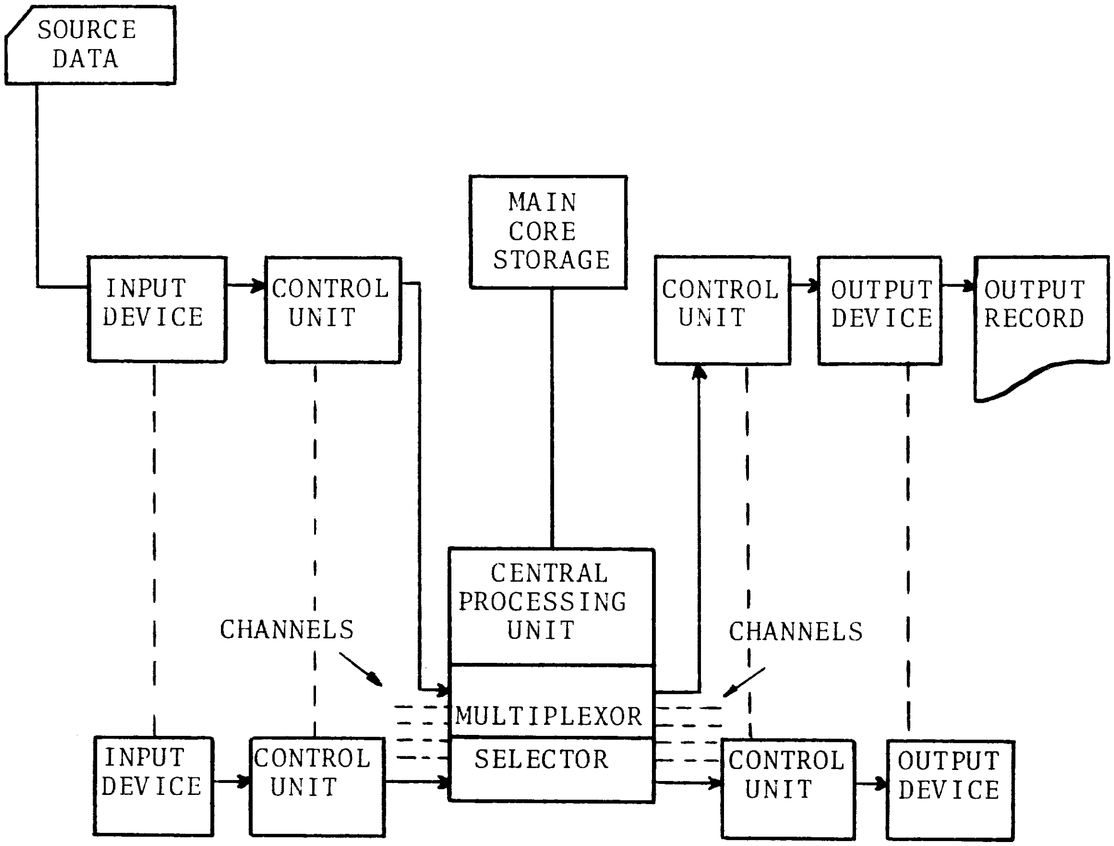


Figure 1-1. Basic System/360 Model 30 Structure and Data Flow

system structure and data flow for the System/360 Model 30. The source data, which can be in the form of punched cards, magnetic tapes, etc., is fed into the input device which, in conjunction with its communication control unit, transmits the information directly into the computer (i.e., central processing unit). The *multiplexer and/or selector* operate on the incoming data, if from more than one input source, to establish the sequence in which the processor will handle the information. The role of these channel devices will be more fully described later in this chapter. After the processing function has been completed the resultant data is transmitted under the direction of the *communication control unit* to the appropriate output device and presented as an output record in a printed page format, as punched cards, visual display, etc. Again the multiplexer and selector units establish the sequence and direct the data traffic to the appropriate output devices.

Table 1-1 lists all the equipments that are compatible for use with the Model 30 and their chief operational characteristics. The remaining sections of this chapter describe in detail the most commonly used equipments in Model 30 installation and give references for descriptions of other equipments.

TABLE 1-1

SYSTEM/360 MODEL 30 EQUIPMENT NOMENCLATURE AND CHARACTERISTICS

FUNCTION	DEVICE	IBM #	MODEL #(s)	SPECIFICATIONS
**Computation and Control*	CPU	2030	-	8-65K capacity 1.5–2 usec storage access time 1 usec machine cycle
Input/Output Card Equipment and Printers	Card Read-Punch	1442	N1	Reads 400 cpm, punches 160 columns per second
	Card Punch	1442	N2	Punches at 160 column/second
	Card Reader	2501	B1, B2	Reads at 600 cpm for B1, at 1000 cpm for B2
	Card Read-Punch	2520	B1	Reads or punches at 500 cpm
	Card Punch	2520	B2, B3	Punches at 500 cpm for B2, 300 cpm for B3
	**Card Read-Punch*	2540	1	Reads at 1000 cpm, punches at 300 cpm (buffered)
	**Control Unit*	2821	-	Works with 2540, 1403, and 1404
	**Printer*	1403	2, 3, N1, 7	Prints at 600 lpm for model 2 or 7 (132 and 120 positions), 1100 lpm for model 3 or N1 (132 and 120 positions)
	Printer	1404	2	Same as 1403
	Printer	1443	N1	Prints at 240 lpm, 120 position
	Printer (MICR)	1445	N1	Prints at 190 lpm
Paper Tape Equipment	Paper Tape Reader	2671	-	Reads up to 1000 cpm
	Paper Tape Reader Control	2822	-	Controls 2671

**Commonly used devices for System/360 Model 30 Installation*

TABLE 1-1 (Continued)

FUNCTION	DEVICE	IBM #	MODEL #(s)	SPECIFICATIONS
Magnetic Tape Equipment	**Magnetic Tape Unit*	2401	1 thru 6	Transfer rate is 30,000 bytes/sec., density is 800 bytes/inch, 37.5 inches/sec. speed, single drive
	Magnetic Tape Unit	2415	1 thru 6	Transfer rate is 15,000 bytes/sec., density is 800 bytes/wd., speed is 18.75 inches/sec.
	MTU Control	2415	1, 2	Control unit for 2415
	Magnetic Tape Control Unit	2804	1, 2	Controls 2401 Tape Unit. Special control feature allows 2 read-type operations to take place simultaneously
	Magnetic Tape Switching Unit	2816	1	Permits the addressing and control of up to 16 tape drives on 2401 under program control
Hyper Tape Equipment	Hyper Tape Unit	7340	3	Transfer rate is 170,000 bytes/sec., density is 1,511 bytes/inch, speed is 112.5 inches/sec.
	Hyper Tape Control Unit	2802	-	Control unit for 7340 Models 1, 2, 3
	Hyper Tape Switching Unit	2816	2	Controls 7340 Model 3 in conjunction with 2802 and permits up to 16 tape drives to be addressed under program control
Storage Devices and Control Units	Disk Storage Drive	2302	3, 4	Model 3 has one access area and 112-million byte capacity, model 4 has two arms and 224-million byte capacity
	**Disk Storage Drive*	2311	-	Interchangeable disk packs, 7.25 million bytes per disk pack

**Commonly used devices for System/360 Model 30 Installation*

TABLE 1-1 (Continued)

FUNCTION	DEVICE	IBM #	MODEL #(s)	SPECIFICATIONS
	Direct Access Storage Facility	2314	-	Capacity of 201 million bytes—has own control unit
	Data Cell Drive	2321	1	Capacity of 400 million bytes
	Storage Control Unit	2841	-	Controls 2302, 2311, 2321
	Auxiliary Storage Control	2844	-	Secondary Control Unit for 2314
Magnetic Character Readers	Magnetic Character Reader	1412	-	Reads MICR coding
	Magnetic Character Reader	1419	-	Reads MICR coding
Optical Readers	Optical Mark Page	1231	N1	Reads marks on score sheet
	Optical Reader	1285	-	Reads journal tapes
	Optical Reader	1287		Models 1 and 2 read alphanumeric documents. Models 3 and 4 can also read specialized handwriting.
	Optical Character Reader	1418	1, 2, 3	Reads documents
	Alphanumeric Optical Reader	1428	1, 2, 3	Reads documents
Display Equipment	Display Unit	2250	1, 2, 3, 4	Alphanumeric or graphic display
	Display Station	2260	1, 2	Alphameric display of up to 960 characters
	Display Station	2265	1, 2	Alphameric display of up to 960 characters

TABLE 1-1 (Continued)

FUNCTION	DEVICE	IBM #	MODEL #(s)	SPECIFICATIONS
	Display Copier	2285		Produces paper copy of images displayed on 2250 Models 1, 3, 4
	Multi-purpose Control Unit	2772		Controls 2265 Model 2 display station
	Display Control	2840	1, 2	Model 2 controls 2250 Model 3
	Display Control	2845		Controls 2265 Model 1
	Display Control	2848	1, 2, 3	Model 1 controls 2260 Model 2 and displays 240 characters, Model 2 controls 2260 Model 2 and displays 480 characters, Model 3 controls 2260 Model 1 and displays 960 characters
Documentary Console	Control Unit	1051	N1, 1	Control unit for interfacing 1052 with CPU
	**Keyboard Printer*	1052	3, 5, 6, 8	Prints at 14.8 characters/second
	Printer	1053		Auxiliary printing unit
	Paper Tape Reader	1054		Additional device for entering of punched paper tape data into CPU
	Paper Tape Punch	1055		Additional output device, punches paper tape in response to data received from CPU, local keyboard, card reader, and/or paper tape reader
	Card Reader	1056		Accepts punched card data for transmission to remote locations
	Card Punch	1057		Takes input data and punches cards

**Commonly used devices for System/360 Model 30 Installation*

TABLE 1-1 (Continued)

FUNCTION	DEVICE	IBM #	MODEL #(s)	SPECIFICATIONS
Communication Terminals and Control Units	Data Collection System	1030		Remote data collection devices can communicate with System/360 at 60 characters per second via 2701 Data Adapter Unit or 2702 Transmission Control Unit
	Data Communication System	1050		All the units from the 1051 thru the 1057 comprise what is called the 1050 system
	Data Communication System	1060		Attaches to System/360 via the 2701 unit with selective speeds of 134.4- or 180-bps for common-carrier lease private telephone service
	Process Communication System	1070		Attaches to System/360 via 2701 unit at selective speeds of 134.4- or 180-bps for private leased telephone service or 600- or 1200-bps for 4-wire full duplex telephone service
	Data Adapter Unit	2701		Provides direct correction of a variety of remote and local devices to System/360
	Transmission Control Unit	2702		Permits the users of System/360 to combine data processing and data communications with same system configuration
	Transmission Control Unit	2703		Permits the users of System/360 to combine data processing and data communications with same system configuration

TABLE 1-1 (Continued)

FUNCTION	DEVICE	IBM #	MODEL #(s)	SPECIFICATIONS
	Communication Terminal	2740		Data Terminal which permits the communication of typed messages over narrowband or voice lines to System/360
	Communication Terminal	2741		Data Terminal which permits the communication of typed messages over narrowband or voice lines to System/360
	Audio Response Unit	7770	3	Audio response to digital inquiry
	Audio Response Unit	7772		Audio response to digital inquiry
	Data Acquisition and Control System	1800		A high-performance system for process-control and high-speed data acquisition application. Available with either 2, 2.25, or 4 micro-second core storage cycle. 18-bit words (16-bit plus a parity bit and a storage protection bit) are moved in parallel between processor and data channels. Handles both analog and digital data
	Processor-Controller	1801	-	Control unit which permits connection of 1800 system to the System/360 Model 30
	Processor-Controller	1802	-	Control unit which permits connection of 1800 system to the System/360 Model 30. Is required if one or more 2401 Magnetic Tape Units are to be attached

TABLE 1-1 (Continued)

FUNCTION	DEVICE	IBM #	MODEL #(s)	SPECIFICATIONS
	Processor-Controller Adapter	1803	-	Control unit which permits connection of 1800 system to the System/360 Model 30. Is required if one or more 2401 Magnetic Tape Units are to be attached. Is required if core size is greater than 32K bits

1.2 THE PROCESSOR

The processing unit for the System/360 Model 30 is the IBM 2030. This unit provides the required arithmetic, logic, and control functions and contains a main internal core storage. The core storage is available in capacities of 8192, 16384, 24576, 32768, or 65536 bytes. A *byte* is composed of eight (8) bits, and in System/360 a word is thirty-two (32) bits. The time to access the main core storage for operations is 1.5 microseconds. Controlling functions are implemented in 0.75 microseconds by use of Read Only Storage (ROS). This means that the basic machine cycle time for performing operations is less than 1 millionth of a second. The 2030 processing unit has a multiplexer data channel which in turn has from 32 to 224* subchannels for input/output devices that can be attached to it depending on the core storage size. For the 8K (K = 1028), i.e. 8192-byte core size system, a maximum of 32 input/output units (I/O's) can be handled; for 16K or more, up to 96 I/O's can be attached to the multiplexor. *Channels* provide a data path and control for the input/output devices (e.g. card read punch, printer, etc.) as they transfer information to the computer or central processing unit (CPU).

When a multiplicity of input/output devices seek to communicate with the CPU either in a *synchronous* manner (i.e. the time sequence of events is controlled) or *asynchronously*, a *multiplexer channel* is required to distribute these device subchannels over the single data path or channel. A multiplexer channel is used primarily for simultaneously handling groups of subchannels from low-speed I/O devices. When the I/O device is a high-speed unit, the 2030 has another type of channel called a *selector channel* to provide the appropriate data path to the CPU. In either case, the incoming data to the channels is in the form of a byte (8 bits) and all channels and subchannels operate the same and respond to the same set of input/output instructions and commands.

The input/output units are connected to the communication channels (as shown in

*Special features of Models E30 and F30 permit expansion from 96 to 224 subchannels.

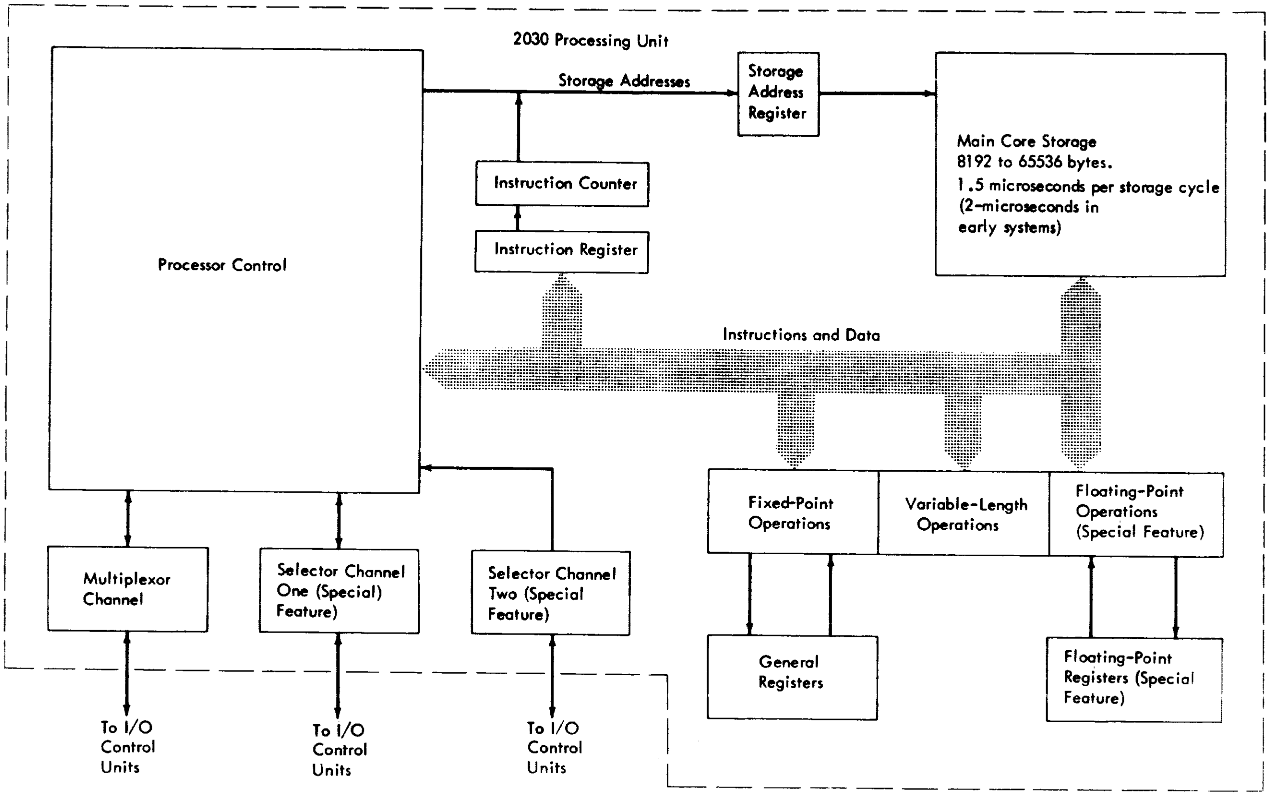


Figure 1-2. Data Flow Within IBM 2030 Processing Unit (Courtesy of IBM)

Figure 1-1) by I/O interfaces on control units. The input/output channels of the Model 30 are fully integrated with the CPU and up to three channels are provided: two (2) selector and one (1) multiplexer. Figure 1-2 shows in detail the components and data flow within the 2030 processing unit. The different type registers depicted in this figure and their roles in the system operation are described in detail in a later section of this chapter.

1.3 INPUT/OUTPUT DEVICES

Table 1-1 indicates that there are a multiplicity of input/output (I/O) devices that can be connected to the IBM 2030 processing unit in a System/360 Model 30 installation. The most commonly used I/O devices in a Model 30 computer center are the IBM 2540 Model 1 card read-punch and the IBM 1401 Model N1 printer. The IBM 2540 is a high-speed card read-punch unit which is fully buffered and reads cards at the rate of 1000 per minute. Figure 1-3 depicts the 2540 Model 1 card read-punch. In order for this device to operate with the IBM 2030 CPU a separate control unit is

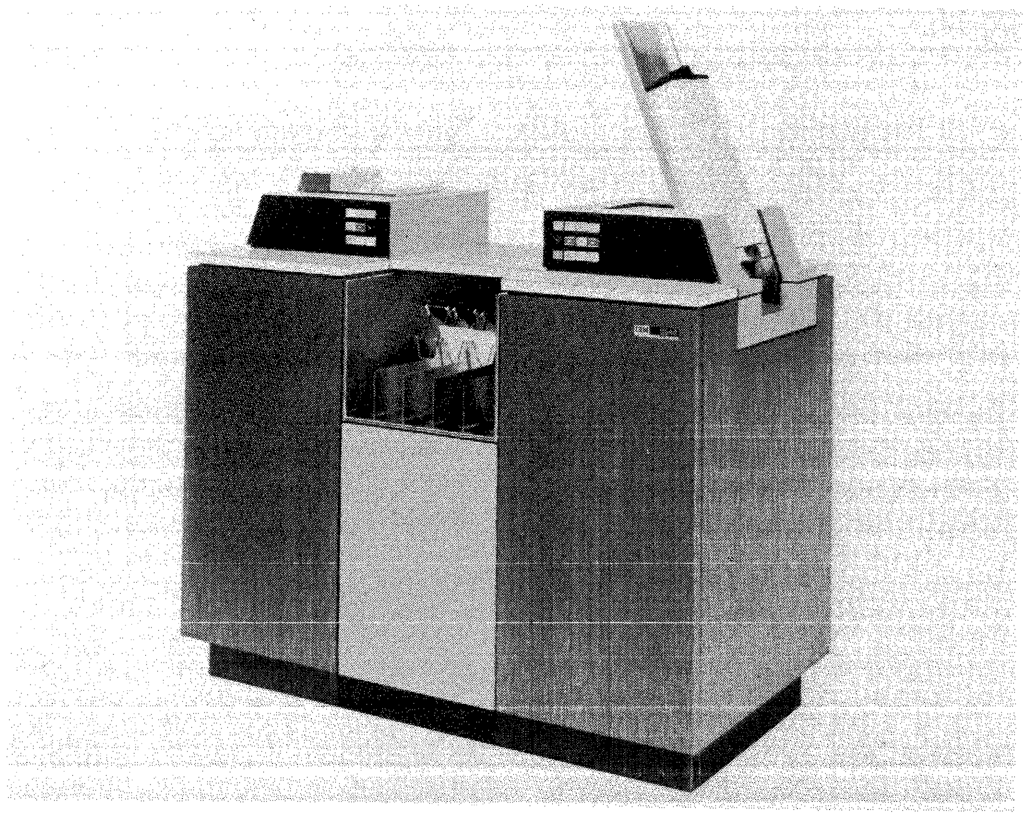


Figure 1-3. IBM 2540 Card Read-Punch (Courtesy of IBM)

required. This control device is the IBM 2821 and, referring to Figure 1-1, electrically connects the 2540 to the 2030 via the I/O communication channels. By a *buffered* device we mean a device which has its own buffer which in turn can hold data until the CPU can operate on it. In the case where there are many buffered I/O devices with their buffers simultaneously loaded, the buffer closest electrically to the CPU polling cycle will be operated on first and the others in sequence thereafter. The 2821 control unit controls the size of the information segments that can be transferred from the buffer to the CPU during each cycle or pass.

In the 360 we have learned that the basic information unit is the *byte* (8 bits) and with the 2821 it is possible to send a multi-byte during each cycle.

The IBM 1403 Model N1 is the line printer generally used with the System/360 Model 30. Figure 1-4 is a picture of a 1403 Model N1 printer. This unit prints the output data a full line at a time at speeds of either 600 or 1100 lines per minute. The 1403 printer uses the same 2821 control unit as the 2540 card read-punch. Once again, the 1403 is a buffered device so that the print cycle does not impact on the transfer of information from the CPU to 1403 buffer. The I/O channel views the data transfer as being completed when the 1403 buffer is fully loaded. When an installation has a 2540

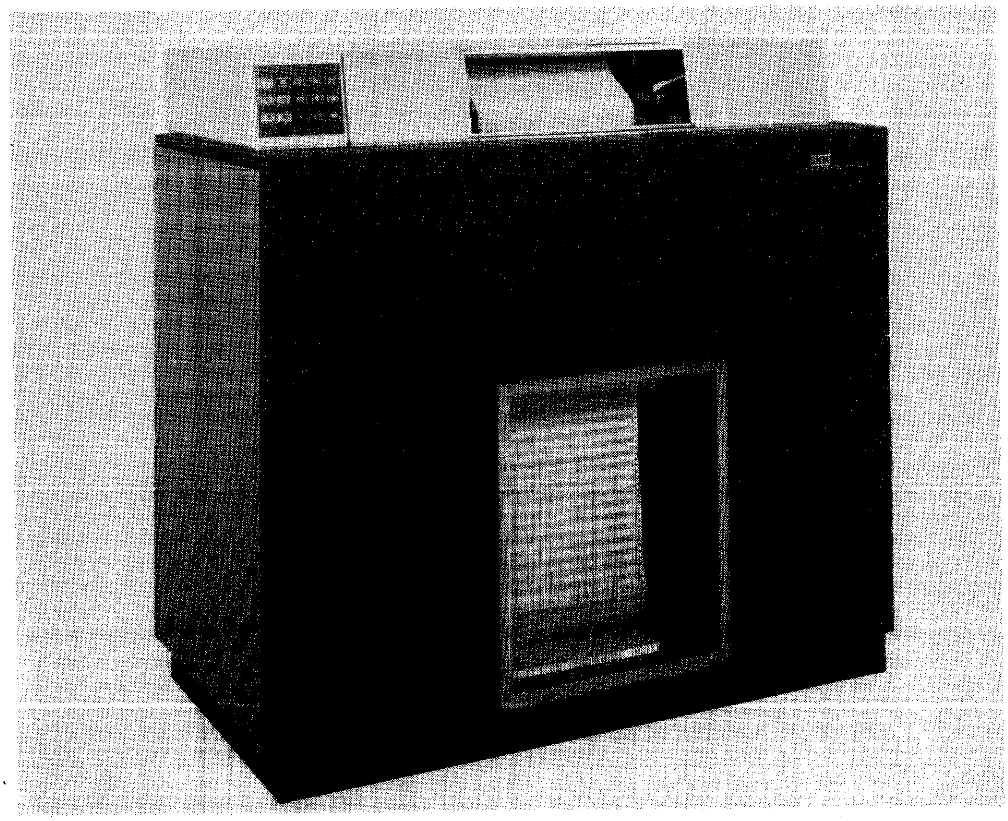


Figure 1-4. IBM 1403 Line Printer (Courtesy of IBM)

and a 1403 connected to the CPU, the 1403, the 2540 reader, and the 2540 punch each has its own separate control unit (exclusively) within the 2821.

1.4 STORAGE MEDIA

The major storage devices used in most installations are magnetic tape and disk storage. The range of different models of these storage devices used in the System/360 Model 30 is large. The most commonly used of these auxiliary storage media are (1) the IBM 2401 (Figure 1-5) Models 1 thru 6 Magnetic Tape Unit with its associated 2803, 2804 Tape Control and 2816 Switching Units and (2) the 2311 (Figure 1-6) Models 3 and 4 Disk Storage Drive with its 2841 Disk Storage Control Unit.

The magnetic tape used on these units is much like that used in home tape recorders. Data is recorded on the tape as magnetized spots, and as it is recorded causes the erasure of any previous data existing in the recording area. The data on tape is organized in units called *records*. The size of these records varies, and can be as small as a single character or as large as the total tape. Between each record is the *inter-record gap*, which is approximately three quarters of an inch.

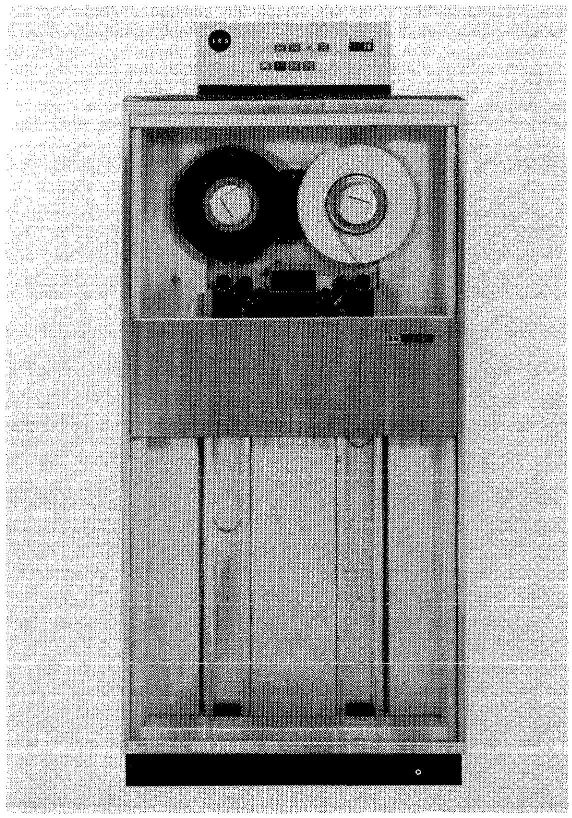


Figure 1-5. IBM 2401 Magnetic Tape Unit (Courtesy of IBM)

There are a number of characteristics of magnetic tape units which are of interest to the operator. One important characteristic is the *number of tracks*. There are *seven-track* tape drives and *nine-track* drives, the nine-track being the more modern. A seven-track tape *cannot* be run on a nine-track drive, nor can a nine-track tape be run on a seven-track drive.

Another important characteristic is *density*, that is, the number of characters per inch on the tape. The drive must be able to accommodate the density specified; several models have a variable density capability, i.e. they can accommodate 800 or 1600 density. Always check the density of the tape before you mount it on a drive, since this is a common reason for a malfunction.

Other characteristics of the 2401 are rewind times of 3, 1.4, and 1 minute for Models 1, 2, and 3 respectively.

A newer and, for many applications, faster storage device is the disk storage drive. A disk drive is composed of magnetically coated disks, which are stacked on a rotating spindle similar to a stack of records on a phonograph turntable. A movable access arm passes between the physical disks. The arm has two heads, each of which can be used for either writing or reading information. One head services the bottom of the disk directly above it; the other handles the upper portion of the disk directly beneath it.

The disk surfaces are divided into concentric tracks with the tracks on each disk surface physically located above the other disk surface. The tracks are numbered sequentially and thereby permit the development of an address coding scheme for accessing the disk devices. The arm mechanisms employed are arranged like teeth on a comb and move horizontally between the disks. The heads are aligned vertically, and are all moved together so that for each position of the access mechanism one entire cylinder surface is accessible to the read/write heads.

The most prevalent disk storage drive used in a Model 30 installation is the IBM 2311. The 2311 has a removable disk storage unit called a *disk pack*. This disk storage device provides random access storage for 7.25 million bytes on a single disk pack. The 2311 disk pack contains 6 disks for a total of twelve disk surfaces. Of these the inside ten are used for recording data and the two outside surfaces are protective plates. The storage control unit for the 2311 is the 2841 which can be attached to up to eight disk storage drives for a total capacity of 58 million bytes per 2841. The 2311 has a data rate of 156,000 bytes/second and sequential track-to-track access time is 30 milliseconds. The average time to seek out the data is 85 milliseconds and the maximum time is 145 milliseconds. The average rotational delay is 12.5 milliseconds. The 2841 storage control unit provides the capability of attaching a number of random access 2311 disk

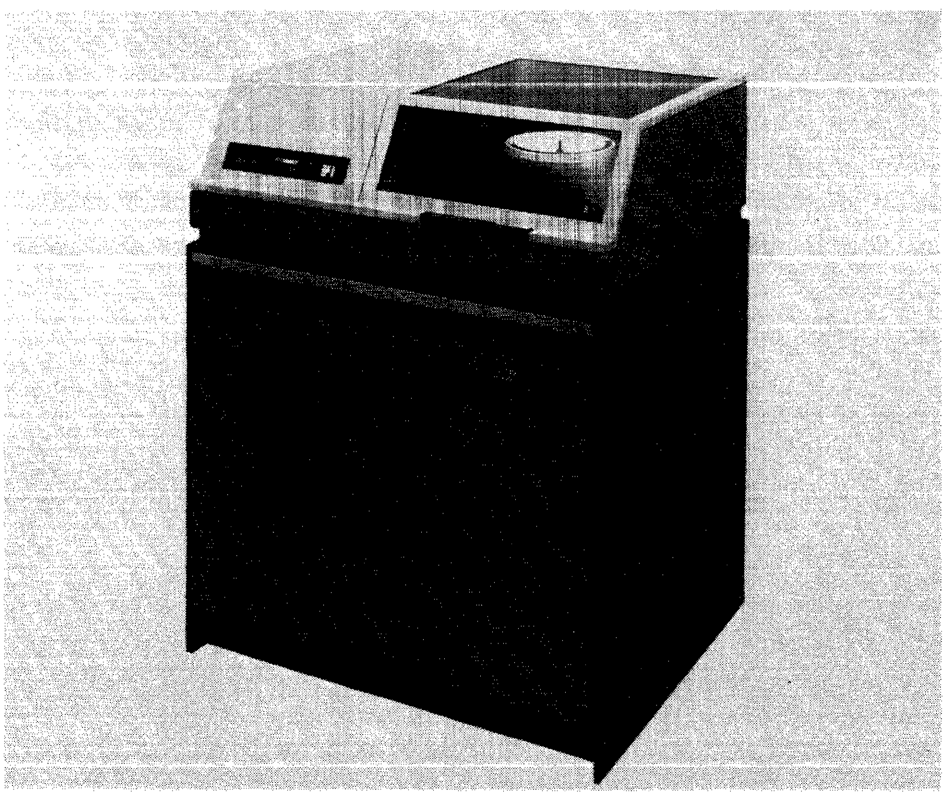


Figure 1-6. IBM 2311 Disk Storage Drive (Courtesy of IBM)

storage drives to the 2030 CPU. The storage drives are controlled by the 2841 through the use of adapters and for the 2311 there is one access mechanism per file for a maximum of 8 access mechanisms. This leads to the 8 × 7.25-million byte or 58-million byte total capacity of the 2841/2311 combination. Figure 1-6 is a picture of the 2311 Disk Storage Drive.

1.5 CONTROL PANEL AND OPERATOR'S CONSOLE

The basic communication device between the console operator and the computer is known as the console typewriter or printer-keyboard. This device prints a character at a time, whereas the line printer prints an entire line at one time. For the System/360 Model 30 this device is the IBM 1052 Printer-Keyboard. With this device the operator can "talk" to the computer and operate on programs. Other functions performed by the operator through this device are job logging, program checking, and program correction. The keyboard portion is used for inputting information to the 2030 processing unit and the printer portion accepts the computer output. From a coding point of view a single keyboard depression signifies an Input/Output interruption, while a double keyboard depression signals the completion of the entering of a set of

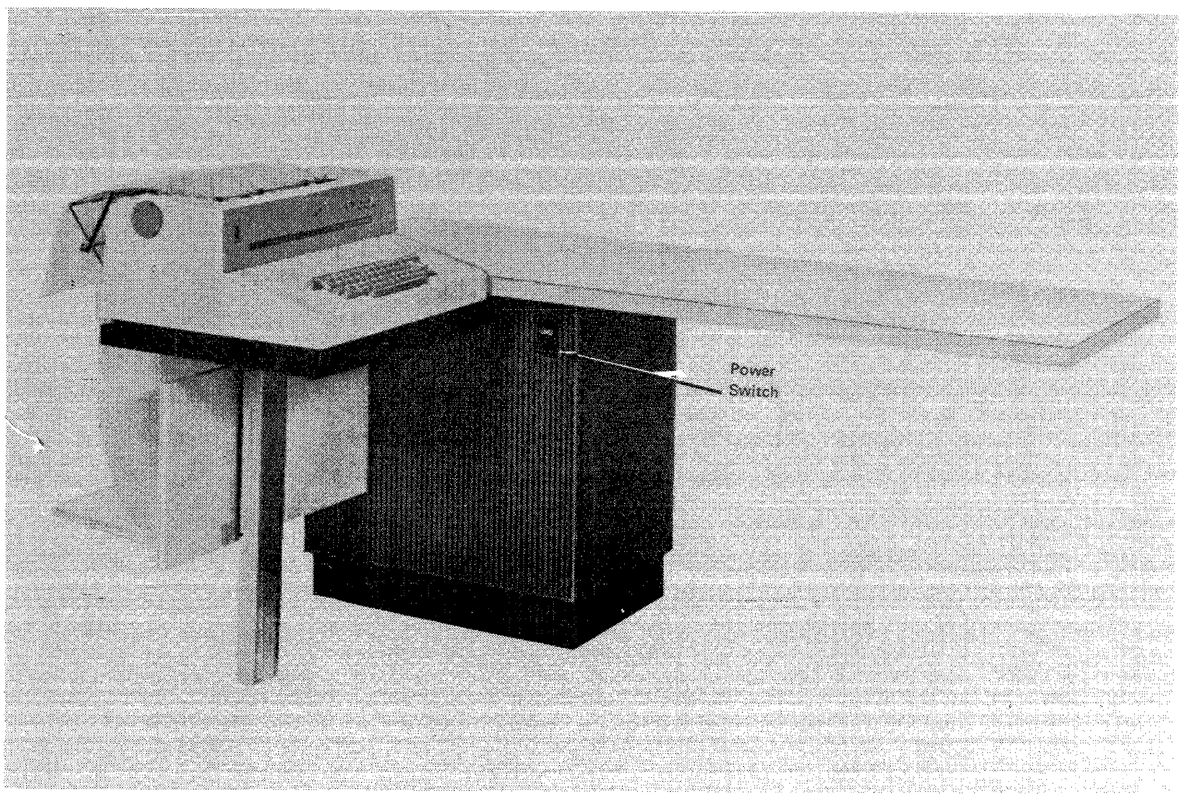


Figure 1-7. IBM 1052 Model 8 Keyboard (Courtesy of IBM)

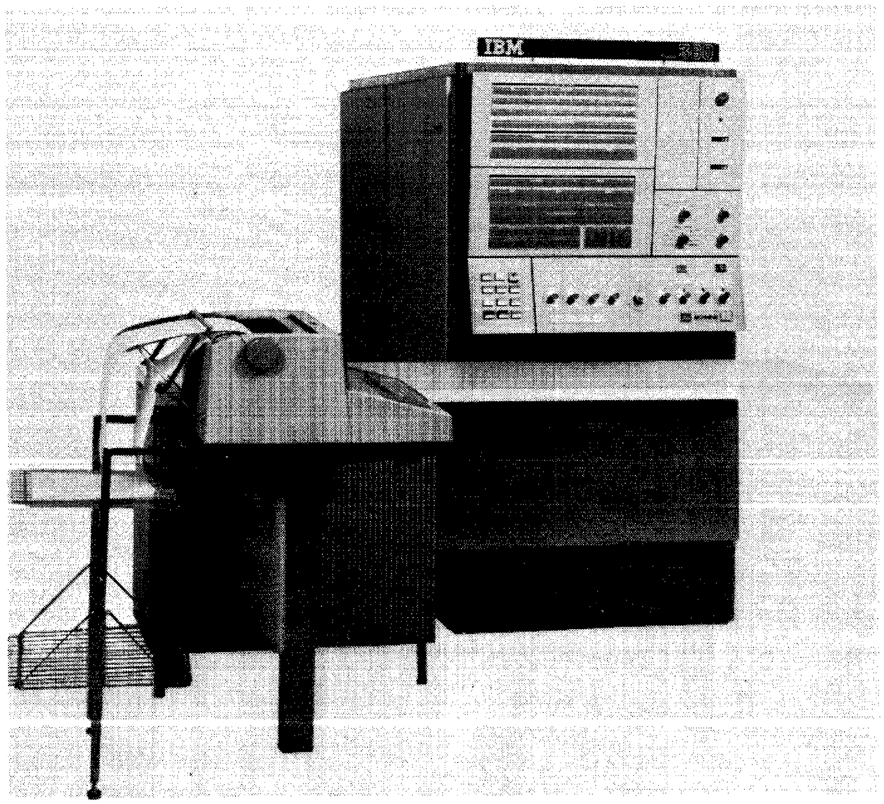


Figure 1-8. IBM System/360 Model 30 Control Panel and Operator's Console (Courtesy of IBM)

data. The 1052 prints at 14.8 characters a second and has a typewriter style keyboard. Figure 1-7 is a scaled picture of the 1052 keyboard. In the Model 30 the 1052 is connected to the 2030 via an IBM 1051 control unit. Referring to Figure 1-8, this control unit is depicted directly under the console table and adjacent to the 1052 Printer-Keyboard. The 1051 provides the required electronic circuitry necessary for controlling the communications between the 1052 and the 2030. In addition, other data communication devices that can be converted to the Model 30 are also attached to the 1051. When a unit or set of units is electrically and logically connected to the computer processor during its operation, it is said to be *on-line*. In the Model 30 the 1052 and 1051 are on line with the 2030.

The control panel which is located in the 2030 processing unit is used to maintain system status and alert the operator to existing or potential malfunctions. Figure 1-9 is a diagram of the Model 30 computer console control panel. The lights give the status of the different registers (fixed and floating) and indicators. The buttons and switches are for operator intervention. The most important items for the beginner are the main storage address register, the main storage address selectors, the local unit and data selectors, and the buttons and switches on the lower left and right portions of the control panel (see Figure 1-9).

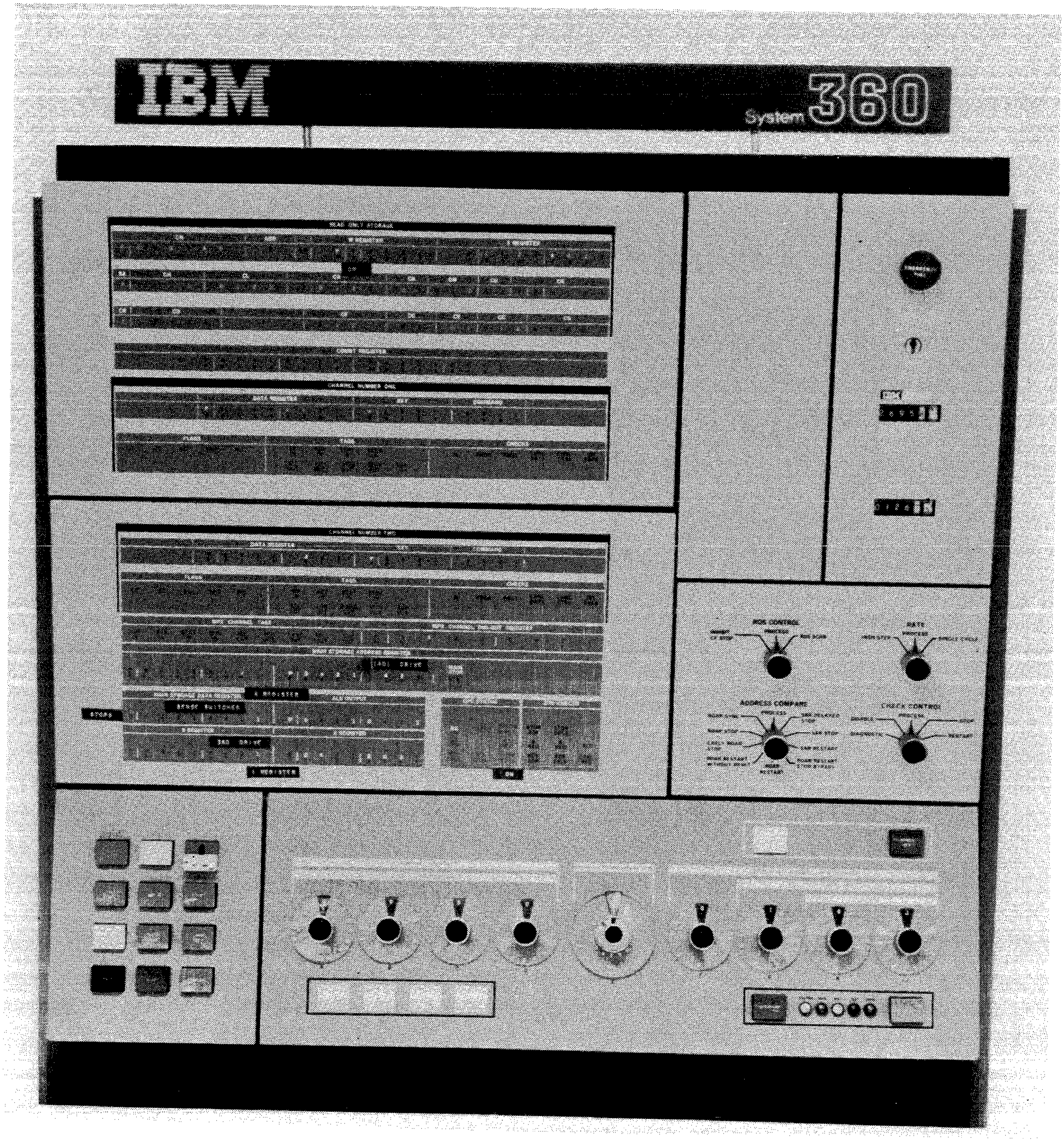


Figure 1-9. IBM System/360 Model 30 Control Panel (Courtesy of IBM)

1.6 REGISTERS

The 8K to 64K main core storage that is part of the 2030 processing unit is used primarily to contain input data, programs, and other operating instructions. For example, when the system is operating on a source or higher order language such as COBOL (Common Business Oriented Language) or FORTRAN (Formula Translation), a substantial part of the core storage, in the order of 8K, is allocated to the compiler program. A compiler program is a program which translates or converts the higher order language to the assembler or machine language. Except for certain instructions known as mnemonics, the machine and assembler languages have a one-to-one correspondence. By a *machine* language we mean the particular coded language that the computer "understands" and responds to. This machine language is generally more complicated to write a program in since every detailed operation must be specified. Therefore one often selects languages which use instruction sets and formats similar to those of the English language. The assembler language for System/360 is called BAL and you will learn more about it later to be able to properly operate the model 30 under DOS.

The System/360 Model 30 employs additional storage facilities called registers to accomplish its different storage functions. Sixteen (16) general, fixed point registers numbered 0 through 15, each with a 4 byte or 32 bit capacity are used for multiplicity of functions. These functions are:

a. Indexing Registers
b. Address Locations
c. Shifting and Logical Operations
d. Fixed Point Binary Accumulators

The primary uses of these 16 registers are addition, subtraction, multiplication, and division of integer numbers and for modifying addresses within the program. For certain types of computations such as multiplication and division two registers are coupled so as to preserve the precision of products and quotients.

In addition to the fixed point registers there are four (4) floating point registers available with the system. These are provided to accomplish mathematical computations involving non-integer numbers. These registers are each 8 bytes or 64 bits in length and permit operation on fractional numbers whose decimal values lie within the range of approximately 2.4×10^{-78} to about 7.2×10^{75}. In floating point operations the 16 general registers are used for indexing and address arithmetic. For those students who are not familiar with mathematical manipulations involving numbers or fractional numbers with exponents, this topic will be covered in Chapter 4 as part of the study of number systems.

System/360 Model 30 installations which are used for both commercial and scientific applications will contain what is referred to as a *universal* set of instructions. This set includes the *standard* set which is needed for basic system operation, the *decimal* set for fixed point arithmetic, the *floating point* set for floating point arithmetic, and storage protection special features instructions.

EXERCISES

1. List the equipment available in a typical System/360 Model 30 installation. Indicate type of equipment, model number, capacity, and speed.
2. What is the size of a word in System/360? What is a byte?
3. Define the following terms and indicate their application in System/360.
   a. General Register
   b. Floating Point Register
   c. Control Panel
   d. Printer-Keyboard
   e. Disk Storage Drive
   f. Magnetic Tape Unit
   g. Instruction Set
   h. Multiplexer Channel
   i. Selector Channel

ANSWERS

1. Reader to prepare table of equipment in an installation.
2. Thirty-two bits or four bytes. A byte is a segment of information used in System/360 and is composed of eight bits.
3. a. General Register is a wired memory, 32-bit storage unit used for performing various computations including fixed point arithmetic.
   b. Floating Point Register is a wired memory, 64-bit storage unit used for performing various computations involving non-integers.
   c. Control Panel–A panel of switches and indicators located on the 2030 CPU and used to maintain system status and alert the computer operator to existing or potential malfunctions.
   d. Printer-Keyboard–A console typewriter used by the operator in communicating with the computer to conduct various system functions.
   e. Disk Storage Device–A high-speed, large-capacity random access memory unit used for main auxiliary storage or as an I/O device.
   f. Magnetic Tape Unit –A high-speed, large-capacity memory unit used for mass auxiliary storage or as an I/O device.
   g. Instruction Set–A special set of instructions designed to enable the system to perform certain types of operation. System/360 Model 30 has a standard instruction set for basic operation, a decimal set for fixed point arithmetic, and a floating point set for non-integer arithmetic and storage protection. When an installation has all three instruction sets it is said to have a universal instruction set (i.e. it can handle both scientific and commercial application).
   h. Multiplexer Channel–Establishes the sequence in which the CPU will operate on data coming into the system from a multiplicity of low-speed I/O devices.
   i. Selector Channel–Same as multiplexer channel except it controls the data stream sequence of a number of high-speed I/O devices.

2

MECHANICAL ASPECTS OF COMPUTER OPERATION

This chapter describes in detail each of the major system equipment components with emphasis on the specific controls and procedures for operating each device. When you are familiar with the operation of each device, and feel comfortable performing the basic steps required to set up and control the equipment, you will be able to master the procedures for processing computer jobs.

Each equipment is described in terms of its physical characteristics, controls, procedures, special cautions, and "tricks-of-the-trade." Although there seems to be a lot to learn, you will find that all of the procedures will become *habit*.

2.1 CONSOLE TYPEWRITER (KEYBOARD-PRINTER)

The basic means of communication between the operator and the computer is the console typewriter or keyboard-printer (Figure 2-1). Although this device looks very much like a standard typewriter, there are major differences. For one, the keyboard is completely independent of the printer position, hence the name *keyboard-printer*. On the IBM System/360 Model 30, the IBM 1052 keyboard-printer is generally used. The keyboard is used to input messages to the computer; the computer responds via the printer or other devices. When the 1052 is working in this mode with the central processor (IBM 2030), it is termed on-line operation. This machine can also be used

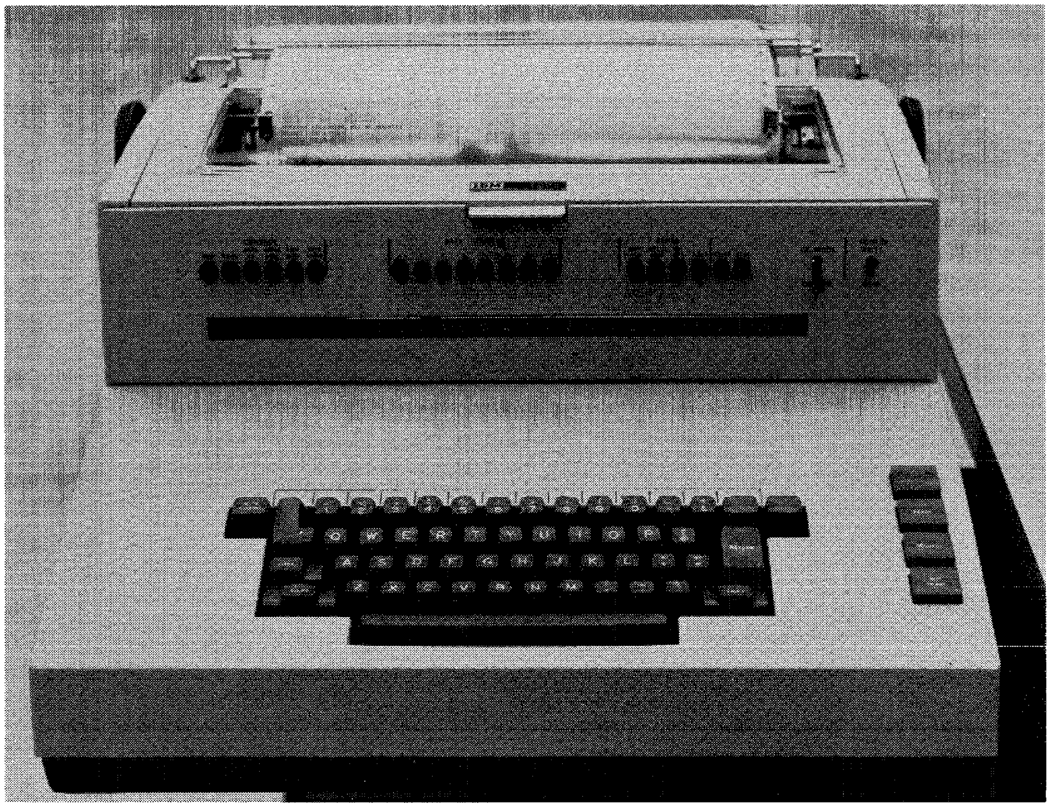


Figure 2-1. IBM 1052 Printer-Keyboard (Console Typewriter) (Courtesy of IBM)

off-line, i.e. not connected to the central processor, for such operations as listing cards.

The 1052 prints at 14.8 characters per second. The printing head is replaceable for special character sets or fonts. The keyboard is very similar to that of a standard typewriter. There are several models of the 1052 keyboard-printer. The most commonly used model with the System/360 Model 30 is the 1052 Model 8 depicted in Figure 2-2a. Additional buttons and switches are mounted on the 1052 depending on the model. In Figure 2-2a, each key indicates the symbols controlled by that key. Actual keyboards do not show lower-case alphabetic symbols; as with standard typewriters, a shift key is used to provide upper-case characters.

Prior to the Model 8, the 1052 Model 6 was the standard console keyboard for the System/360 Model 30. The Model 8 is compatible with all instructions for the Model 6, but it is important to note that since Model 8 has fewer controls, it ignores certain commands or spaces without printing. Comparing the Model 8 (Figure 2-2a) with the Model 6 (Figure 2-2b) keyboard, you will note that the TAB, BACKSPACE, and LINE-FEED keys are blank. These functions are not performed on the Model 8. Further, the Model 8 ignores ribbon shift and line-feed select codes.

Other controls and functions are also inoperative on the Model 8, namely:

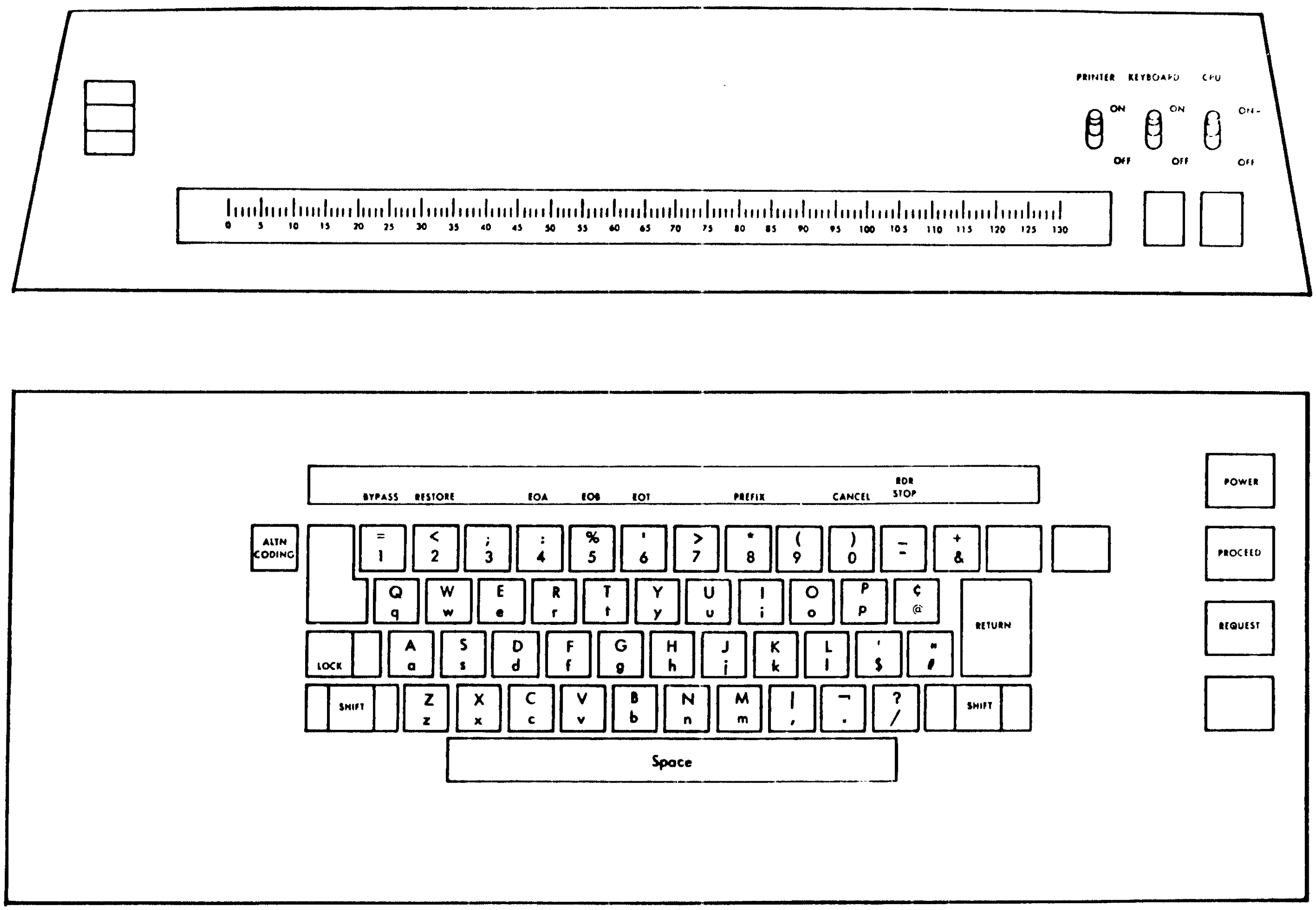


Figure 2-2a. IBM 1052 Model 8 Keyboard and Switch Panel (Courtesy of IBM)

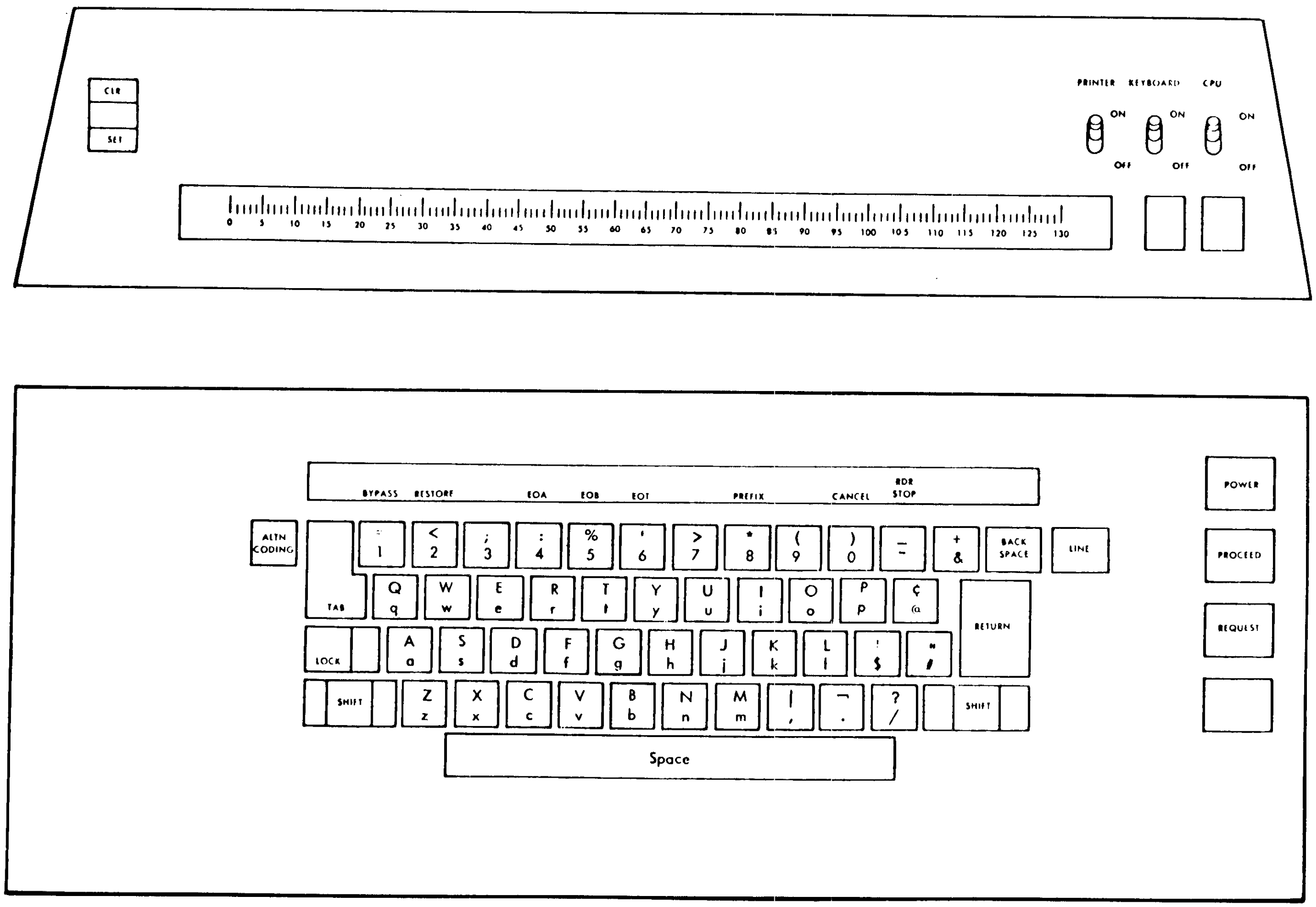


Figure 2-2b. IBM 1052 Model 6 Keyboard and Switch Panel (Courtesy of IBM)

a. Left and right margin set (Model 8 has fixed margins)
b. Single-double index lever
c. Paper release bar
d. Tab clear-set lever
e. Ribbon shift lever
f. End-of-line bell
g. Pressure feed rolls

2.1.1 Switches

Referring to Figure 2-2a we see that there are only three on-off switches on the switchboard used in the operation of the 1052.

The CPU switch in the *on* position connects the 1052 via the 1051 control unit to the 2030. If the 1051 power is on when the CPU switch is turned on, an operational signal lights up on the 2030 control panel. The CPU switch in the *off* position completely disconnects the 1052/1051 system from the 2030. Any commands for read or write are then rejected by the system.

The keyboard switch in the *off* position results in the keyboard being in a locked state. No data can be sent to the 2030 via the 1052 keyboard when the keyboard switch is in the *off* position. The keyboard switch in the *on* position releases the locking mechanism and puts the 1052 and the 1051 control unit in a ready state for keyboard transmissions to the 2030 CPU.

The printer switch in the *on* position puts it in a ready state to print data as supplied by the 2030. In the *off* position the printer is in a locked state and will not respond to print commands from the CPU.

The keyboard has three button indicators in addition to the various keys. The power indicator must be turned on before any system operation can take place. The proceed light indicates that the 1051 channel is ready for operator-initiated keyboard input to the 2030. When the proceed light is on, it also indicates that the 1052 keyboard is in the unlocked position. The request pushbutton and associated light are used for establishing an attention status in the 1051. When the 1051 is ready for receiving data from the 1052 it initiates an attention interrupt which is depicted on the 2030 control panel.

2.1.2 Keys

Eight of the keys at the top of the keyboard are assigned function codes in addition to their normal shift and upper-shift character assignments. These function codes (bypass, restore, EOA, EOB, prefix, cancel, and reader stop) are keyed by pressing down the associated key for the desired code, while holding down the ALTN coding key shown on the left side of the key set.* The purpose of the 8 keys is to permit the generation of control characters for error correction.

Keying the bypass code places the printer in a non-print status until a restore code is keyed. In the bypass state all subsequent characters (including another bypass) keyed

*Note: The ALTN key must be depressed first, then the Numeric key is operated. The ALTN is released last.

or read are punched but not printed. The restore code key is punched but not printed and returns the printer to print status.

The Cancel code is used to notify the receiving terminal that the message in progress is in error and should be ignored. Depressing the ALTN key and the zero key results in the cancel code being generated.

The End of Block (EOB) or equivalently End of Transmission (EOT) are keyed directly following the cancel code and generate an end to keyboard entry. The ALTN key and the number 5 key result in an EOB code, while the ALTN key and the number 6 key result in an EOT code.

The End of Address (EOA) code transfers the 1051 to a text mode.

The Reader (RDR) Stop code causes the reader to stop operating because there is an error in the data. The RDR Stop code also unlocks the keyboard.

The Prefix code is used in conjunction with the ALTN key and a numeric key for component-recognition control.

Other keys of interest are the Space Bar, the Carrier-Return, and the Shift and Shift Lock keys. Depressing the Space Bar moves the print element carrier one space to the right. The pressing of the Carrier-Return key returns the print element carrier to the left margin and automatically line spaces to the next printing line. Pressing either shift key puts the print element in the state for upper-case printing. To hold the print element in the upper-case mode the shift-lock key is depressed. Similarly to shift a print element from the existing upper-case mode to the lower-case mode requires pressing down on the locked shift key.

Having briefly described the basic switches, indicators, and keys on the 1052 Model 8 keyboard it is useful at this point to tie it all together by going through the setup procedures prior to system operation.

1. The first step is to turn the 1051 main-line switch on. This results in an automatic systems reset and is indicated on the 2030 Control Panel. After the main power has been turned on, the power button on the 1052 should be depressed and the indicator lighted.

2. The next step is to verify that the 1052 keyboard and printer are set correctly for shift status (upshift or downshift), line spacing (single or double), print element, printing position (print element at the left margin), and carriage return.

3. Check must be made of the positioning of all switches located on the 1052 switch panel. The Printer, Keyboard, and CPU switches should all be in the *on* position.

4. Finally a check must be made of the 1052 Printer for proper forms, sufficient paper quantity, and proper paper positioning.

Now let us practice operating the 1052 keyboard-printer to familiarize us with the actual equipment involved and to gain experience in communicating with the computer.

2.2 CARD READER-PUNCH (2540)

The basic I/O device in a Model 30 installation is the IBM 2540 Card Reader-Punch. The 2540 is a high-speed unit, fully buffered, that reads cards at the rate of 1000 per minute. The 2540 consists of two separate portions. The card reader reads in data from a deck of punched cards that has been placed in the card read hopper and transfers the data into the 2030 CPU storage section. After the cards have been read and checked they are passed into three of the five available card stackers. After the program has been run the operator manually removes the cards from the stacker and places them in the proper storage facility.

The card punch is used to record the resultant data after the computer has performed the required arithmetic operations. The card punch takes the data transferred to it from the CPU storage and punches the output data into a deck of blank cards. Prior to the punching operation the operator manually loads the card punch hopper with a set of blank cards. As the computer transfers the answers to the 2540, the appropriate holes are punched on this set of cards. After punching, these output cards fall into three of the five stackers in the same manner as in the card reader operation.

In order for the 2540 to operate with the 2030, a separate control unit is required. The 2821 is the control unit that electrically connects the 2540 with the 2030 CPU via the I/O communication channels. The 2821 also controls the size of the information segments that can be transferred from the 2540 buffer to the computer and vice versa during each cycle or pass. The 2821 permits the sending of multi-bytes of information during each cycle.

In order to have a better understanding of the card reader-punch operation it is important to have a pictorial representation of the unit and the various functions required of the computer operator. Figure 1-3 depicts the 2540 unit and the card hopper and stackers. In addition it reveals that the card reader portion is on the right-hand side of the machine and the card punch portion on the left. Both the card reader and the card punch may use up to three stackers. Looking at Figure 1-3 we see that the two stackers on the right (R1 and R2) are only allowed to be used by the card reader while the two left-most stackers are only usable by the card punch. The middle stacker may be used by either the reader or punch but not simultaneously. It should also be noted that stacker selection is controlled by the program and not the operator.

The card reader-punch is a relatively simple unit to operate and the following procedural description will best explain the operator's role. First, however, some general notes on the 2540. There is a panel of 12 square pushbuttons and indicators mounted on the card reader on the right-hand side of the 2540. It is necessary to clear the hopper (Figure 2-3) of any cards left there from a previous program run. It is possible to place a deck of considerable height in the card reader file-feed which is tilted at a slight angle to prevent the deck from toppling over. The cards are drawn into the hopper, and the joggler gate holds the cards in the appropriate position so that they may be evenly drawn into the read areas. A *card weight* is used to maintain just enough pressure on the deck to insure that the last cards will make firm contact with the feed mechanism. Where many card decks are being used, it is necessary to depress the END OF FILE pushbutton when the last deck is placed in the file-feed to prevent the computer from being falsely signalled that the end of the deck has been reached. This action is required because the card reader automatically stops before reading the last card unless the END OF FILE button is keyed. The result of depressing the END OF

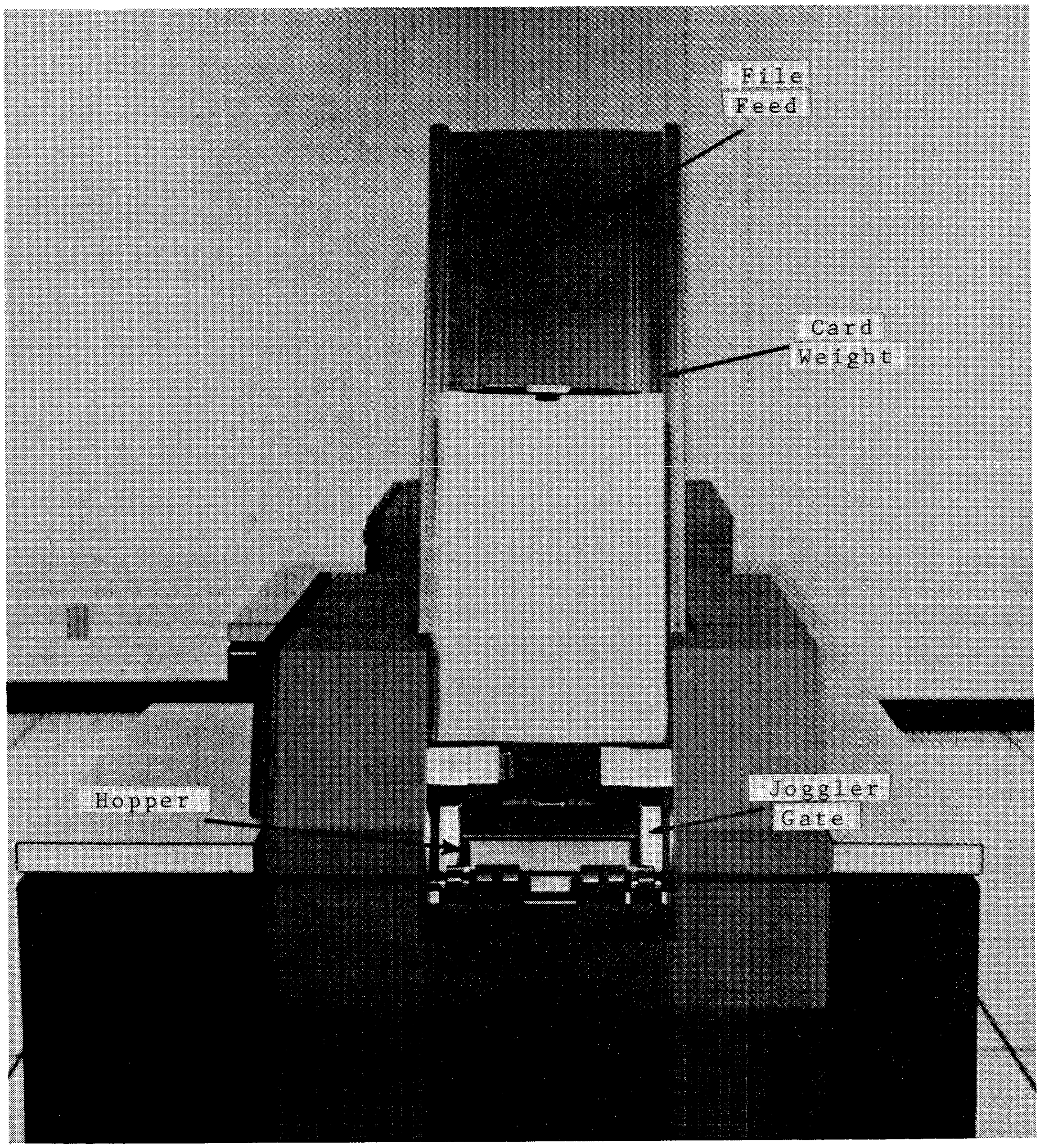


Figure 2-3. IBM 2540 Card Read-Punch: File Feed, Card Read Hopper, Joggler Gate, and Card Weight (Courtesy of IBM)

FILE button is the continued card feeding and reading process until the read hopper is empty and the last card is read, processed, and stacked.

The procedure for preparing the card reader for operation is as follows:

1. Depress the STOP pushbutton on the card reader (Figure 1-3).
2. Remove any cards from the card reader file-feed (Figure 2-3).
3. Open the joggler gate (Figure 2-3).
4. Remove any cards from the hopper (Figure 2-3).
5. Depress and hold the START pushbutton on the card reader.
6. Close the joggler gate.
7. Place the decks in the card reader file-feed (face down, 9 edge first) in the order specified in the run book.*
8. Place the card weight on top of the deck in the position shown in Figure 2-3.
9. Depress START pushbutton again.
10. If all cards to be read fit into file-feed, depress END OF FILE pushbutton and wait for END OF FILE indicator light to come on. If deck is too large to fit in file-feed do not depress END OF FILE pushbutton. Continue operation until the last deck fits into file-feed and then depress END OF FILE pushbutton.

Step 5 is designed to cause the passing of any cards remaining within the reader into the stacker without reading or processing. It is referred to as a non-process run out. Step 9 causes the first few cards to drop from the card reader file-feed into the hopper and to be fed into the read area stations. The completion of this action causes the READY (Figure 1-3) indicator light to come on which indicates that operation may be continued.

After the program has been run and the output data gathered, only a simple clean up function is required of the operator. He must take all the card decks that have fallen into the stackers, remove them and place the decks in the master file library or some other appropriate storage facility.

The procedure for preparing the card punch for operation is as follows:

1. Depress the STOP pushbutton on the card punch (Figure 1-3).
2. Remove the card weight and blank cards from the punch hopper (Figure 2-3).
3. Press and hold the START pushbutton for a few seconds. The cards that remained in the machine will feed into stacker Pl (the left-most stacker).
4. The last two cards fed into the stacker should be blank. Place these blank cards with the cards removed from the hopper.
5. Place the blank cards into the hopper with the card weight on top.
6. Depress the START key. The punch is now ready for operation.

Just as there are indicators on the read side of the 2540, there are indicators on the punch side. These indicators are lit when an error occurs.

At this point it is time to practice using the 2540 card reader-punch. We will use a prepunched set of cards to see how the reader works and a blank set that will be punched by a preset computer output solution.

*The run book is the instruction procedure set for performing various system operations and will be more fully described in Chapter 3.

2.3 MAGNETIC TAPE UNIT

Magnetic tape units are auxiliary storage devices which require a significant amount of operator attention in preparing a run. Where tape is used in an installation, the operator generally will mount tape for each new job being run and will unload the tape before starting the next job.

Tapes may be considered as input, output, or external working storage depending upon the program. In working with tapes, you must be extremely careful in following the programmer's instruction sheet (run book), since valuable data may be easily destroyed or lost. Tapes containing data will be referred to as volumes. The computer can write on a tape or read from a tape. To insure that the computer does not write over (and effectively erase) permanent data, the tape reel must be file-protected. File-protection is accomplished when a special plastic ring is *not* fitted into a groove on the tape-unit side of the reel. That is, unless this ring is in place, the computer cannot write on the tape. The rule to remember is

NO RING . . . . . . NO WRITE.

Figure 2-4 schematically depicts the essential features of an IBM 2401 tape unit. The tape reels are on spindles, with the file reel on the left and the take-up reel on the right.

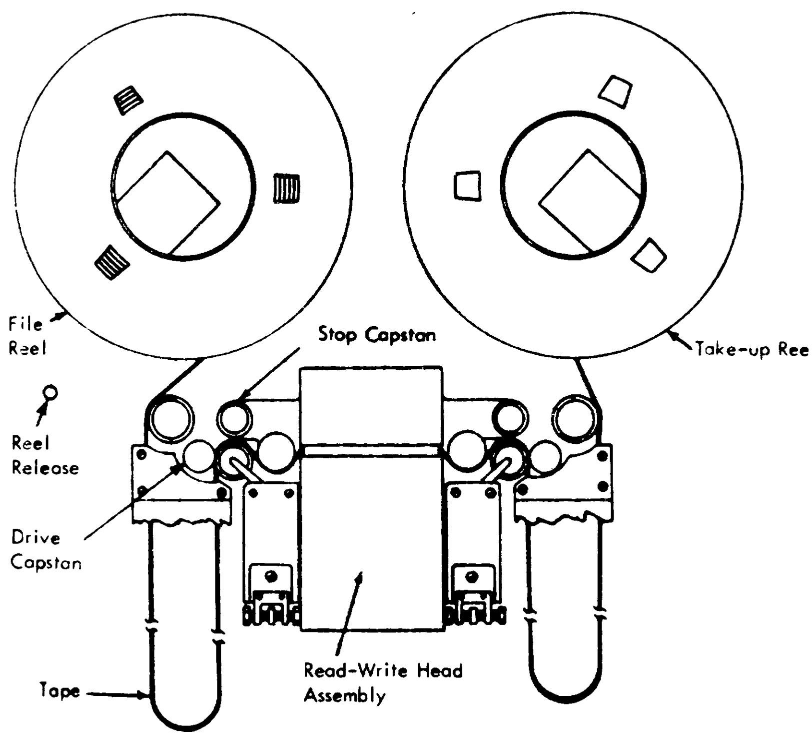


Figure 2-4. Schematic Diagram of IBM 2401 Tape Feed (Courtesy of IBM)

You will note that the tape is looped into vacuum columns and during operation, the reels spin behind closed glass doors. Tape moves clockwise off the file reel, down through the left vacuum column, across the read head, through the right vacuum column and clockwise onto the takeup reel. The loop provides sufficient *slack* for the two reels to move independently at different speeds. The vacuum columns allow the tape to move rapidly and suddenly without tearing. You will not have to measure or form this loop in the vacuum columns. As you will see, the tape is fed between the capstans and when the tape is readied, it is automatically pulled into a loop within the vacuum column.

The glass access doors have interlocks that do not allow the unit to operate unless the door is fully closed. The reels will stop moving immediately if the door is opened during processing. Do not open the access door while the tapes are moving since this results in a loss of vacuum and the sudden tension could snap or damage the tape.

Examine a tape reel. Note that one side has one or more finger holes and the other side has no finger holes but has a circular depression around the center hole. The side with the finger holes is the front and faces out, away from the machine, toward you. The file-protect ring, of circular shape with a small lip to ease removal, fits into the depression on the back or machine side. In handling tape, be extremely careful that you do not touch the tape surface, except on the leader. The slightest amount of dirt or grease on the face of the tape can cause incorrect readings of the magnetized areas. Also these pollutants on the tape may be transferred to the tape heads causing them to malfunction. The tape should not be handled by the edges and should be kept in protective cases when not in use.

Assuming for a moment that there is no tape mounted on the tape unit, the following procedure should be used to mount a tape:

Step 1. Review the programmer's instructions in the run book to insure that you have the correct tape. Check the tape label carefully, because you can easily destroy valuable data by mounting the wrong tape, or mounting it on the wrong unit. If the tape is a permanent file or is simply marked *input* be sure that the file-protect ring is not in place.

Step 2. Reference Figure 2-4. Flip open the hub latch on the left hub and place the reel on the hub or spindle. The file-protect ring (or depression if there is no ring) faces away from you; the side with the finger holes should be facing you. With your hand as close to the center of the reel as possible (to protect the outer edges of the tape) press the reel back onto the hub until it is seated against the stop. Snap the hub lock closed.

Step 3. With your left index finger depress the Reel Release button. This is found below the left-hand reel at about *seven o'clock* (Figure 2-5). This pushbutton must be held depressed any time the reels are turned manually. The reels cannot turn unless this button is held down.

Step 4. Take hold of the tape leader with your right hand and (holding the Reel Release button down from step 3) pull gently and steadily to unwind four or five feet of tape. You can now release the Reel Release button.

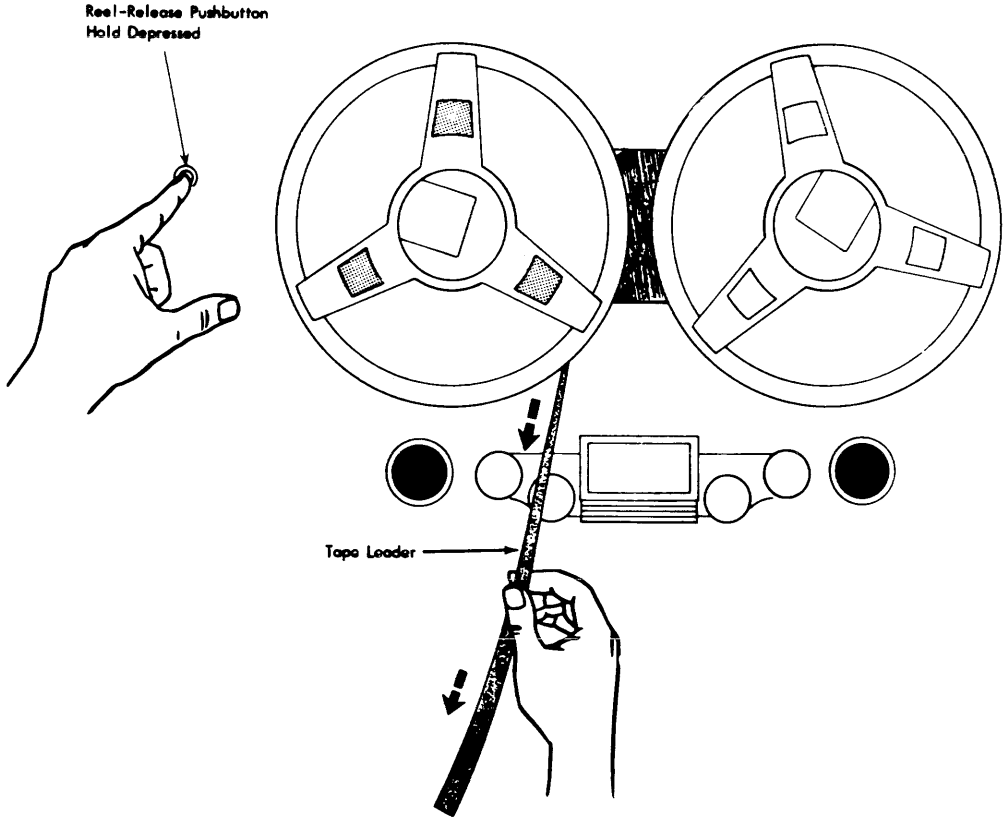


Figure 2-5. Stripping Off Tape Leader (Courtesy of IBM)

Step 5. Reference Figure 2-6. The leader comes off the file tape clockwise. Thread the leader around the left capstan counterclockwise as in Figure 2-6, and across to the other (right) capstan, passing over the read/write head assembly. Continue to thread the leader counterclockwise around the right capstan and clockwise onto the right take-up reel.

Step 6. Reference Figure 2-6. Put a finger through a cutout in the take-up reel so that you can hold the tape leader against the reel hub. Turn the take-up reel slowly in a clockwise direction, holding the Reel Release button depressed with the left forefinger. Continue to turn until the tape leader wraps around onto itself and holds firm. When the leader is firmly held onto the take-up reel, continue to turn the take-up reel using the finger hole as shown in Figure 2-6, carefully watching for the load-point marker to pass over the read/write heads. The load-point (or reflector strip) marker is a shiny aluminum strip about one-inch long on the tape.

Step 7. When the load-point marker passes the read/write heads, continue to wind the tape onto the take-up reel for two or three more turns, using the finger hold. Be careful that you do not touch the tape surface after the load-point marker has been passed. Release the Reel Release button.

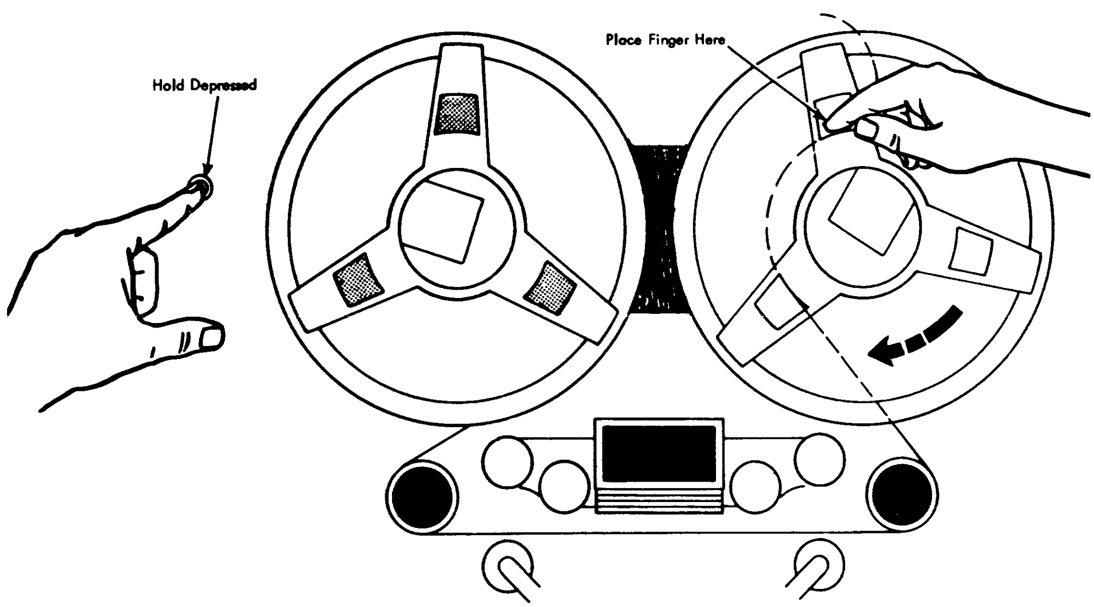


Figure 2-6. Threading Tape (Courtesy of IBM)

Step 8. Hit the Load Rewind button which initializes the tape station. The power window rises automatically, and the tape is pulled into a loop in the vacuum columns. The read/write head lowers and the tape rewinds back to the load-point marker leaving only the leader on the take-up reel. If you do not wind the tape beyond the load-point marker the tape will rewind completely onto the left-hand reel (searching for the load-point marker) and you will have to begin again.

Step 9. Press the Start button. The Ready indicator light should glow. If a file-protect ring was not placed on the tape reel, the File Protect indicator light will glow. Check the programmer's run sheet to determine whether or not the tape file is to be protected and if so, that the File Protect indicator is lighted. When the Ready and File Protect indicators are lighted, the tape station is initialized and ready for use.

Now, let's review these steps quickly:

STEPS	*CAUTIONS*
1. Check run book for instructions.	Any files to be protected?
2. Flip open hub latch and place reel on hub; close hub latch.	Hold reel near center.

3. Press Reel Release button.	
4. Unwind four or five feet of tape from file reel.	
5. Thread leader around left capstan and across read/write head assembly.	
6. Hook tape onto take-up reel, holding and turning with finger.	Tape leader must be firmly wrapped around itself.
7. Manually wind tape onto take-up reel past load-point.	Watch for load-point marker and wind it on to take-up reel.
8. Close access door. Hit Load Rewind button.	
9. Press Start button.	Check for Ready and File Protect lights.

Now that you have learned to mount a tape on a tape station, we will describe the steps necessary to remove or unload a tape.

When you approach the tape station, you may find that the Ready indicator light is on. This means that the unit is still under control of the computer and must be returned to manual control before opening the access door.

Step 1. Assuming that you found the Ready indicator ON, press the Reset button, which removes the unit from computer control. Press the Load Rewind button which rewinds the tape onto the file reel back to the load-point marker.

Step 2. The tape will stop when the read head encounters the load-point marker. Press the unload button which initiates an automatic unloading sequence. The tape is raised from the vacuum column, the read/write head lifts and the access window lowers.

Step 3. You must now manually rewind the tape leader onto the left-hand file reel. Press the Reel Release button down with your left forefinger and rewind the leader completely back onto the file reel. Release the Reel Release button.

Step 4. Open the hub latch and pull the reel from the hub. As in loading tape, hold the reel as close to the center or hub as possible, because you can easily pinch the tape edges by pressing the reel near the rim. If the reel sticks to the hub or is otherwise difficult to remove, place your fingers behind the reel, near the hub, and apply pressure to force the reel towards you.

2.4 MAGNETIC DISK FILES

This unit is a random access auxiliary storage device which stores very large amounts of data on magnetically coated disks. The disks are removable. One of the skills you will learn is how to ready a disk unit for operation. The *Disk Storage Drive* is the unit in which the *Disk Pack* is mounted so that it can be used for reading or writing data.

The Disk Storage Drive most commonly used with the IBM System/360 Model 30 is the IBM 2311 (Figure 1-6).

The disk pack is protected by a clear plastic container and is always to be kept in such container except when mounted on the drive shaft of the disk storage drive. Do not touch the magnetically coated surfaces of the disk; the disk pack may be loaded and unloaded by loading the cover to the disk, as described below.

There is a locking knob on the bottom of the plastic container which, when turned, allows the bottom plate to be removed. The disk pack remains locked to the top cover and may be carried by the *handle* on the top of the cover. The disk pack, held by the cover, is placed onto the Disk Storage spindle. The handle rotates the cover to lock the disk pack to the spindle and to free the top cover for removal.

The programmer's run sheet will specify one or more disk pack files for mounting on particular disk units. The disk drive must be fully stopped before a disk pack can be loaded or unloaded. Thus, if there is a disk pack on the drive, press the Stop switch *before* you go to the disk library to locate the file. Then the drive, which takes a while to stop, will be fully stopped when you return with the proper file(s). Do not attempt to replace a disk pack until the unit is *completely* stopped.

To remove a disk pack from the disk storage drive, perform the following steps:

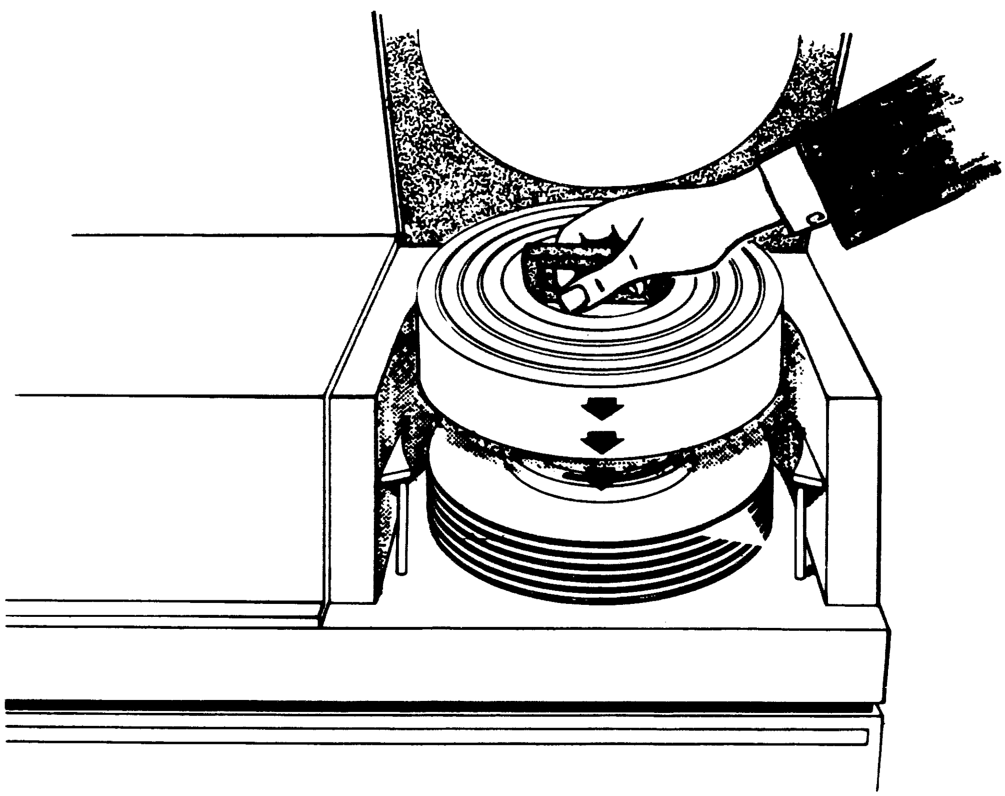


Figure 2-7. Placing Cover Over Disk Pack (Courtesy of IBM)

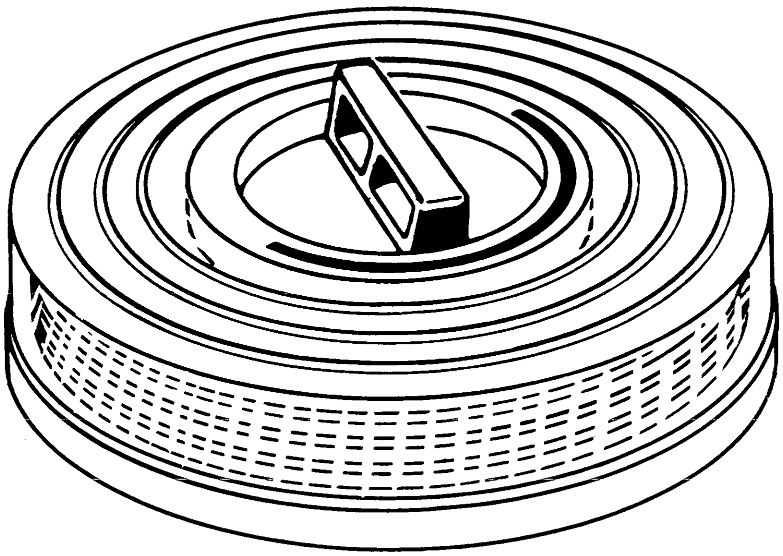


Figure 2-8. Cover Off Disk Pack (Courtesy of IBM)

Step 1. Push the Start/Stop switch on the disk unit (Figure 2-7) to Stop. Wait until the disk pack completely stops.

Step 2. Remove the bottom plate from an empty disk pack container.

Step 3. Raise the lid on the disk storage drive gently. Holding the handle on top of the disk pack container, lower the container carefully over the disk pack. Do not allow the cover to hit against the sides of the pack (Figure 2-7).

Step 4. When the cover is placed over the disk pack on the spindle, rotate the handle (and hence the top of the container) *two full turns* in the direction of the PACK OFF arrow on the cover (Figure 2-8).

Turning the cover locks the cover to the disk pack and frees the entire unit from the spindle of the disk storage drive.

Step 5. Lift the entire integral unit, cover and attached disk pack, from the spindle. Holding the unit by the handle on top, replace the bottom plate and lock firmly with the knob.

Step 6. Return the disk pack to its proper storage library location.

To load a disk pack onto the disk storage drive, perform the following steps:

Step 1. Check the programmer's run sheets to determine the disk pack to load and prepare a label if required.

Step 2. Holding the disk pack by the handle on the top cover, remove the *bottom* plate by turning the knob on the bottom. Set the bottom plate aside (Figure 2-9).

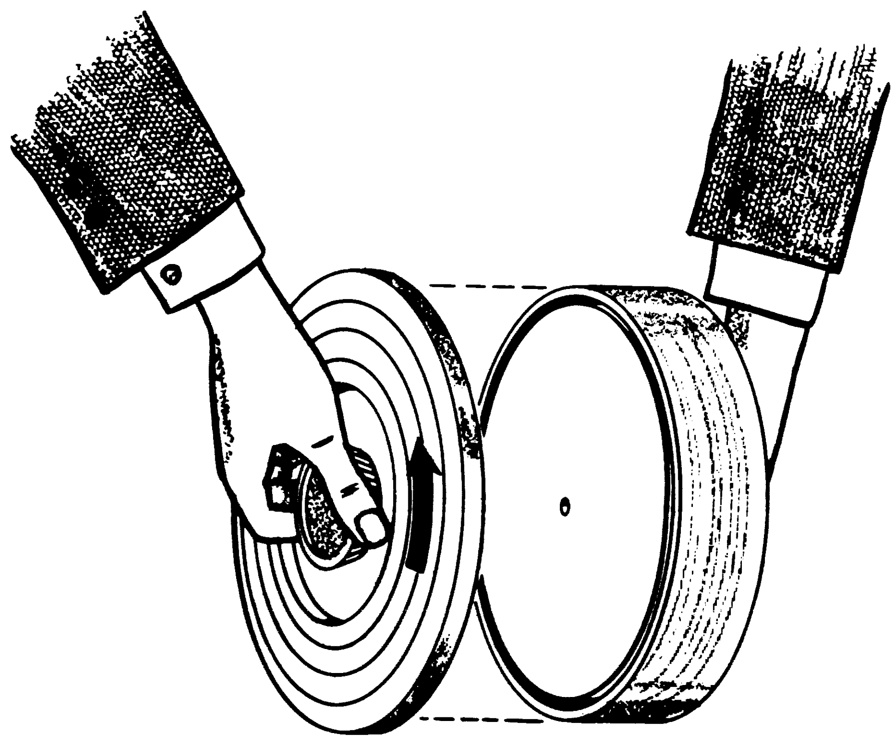


Figure 2-9. Removing Bottom Plate (Courtesy of IBM)

Step 3. Check that the disk unit is stopped, i.e. the spindle is not rotating. Open the drive lid and place the integral disk pack and top cover over the spindle. Lower carefully until it rests on the base (Figure 2-10).

Step 4. Turn the *handle* (and hence the top cover) in the direction PACK ON, COVER OFF shown on top of the cover. Continue to rotate until the cover encounters a firm stop, and then lift the top cover off. The disk pack will remain on the spindle. Do not hit the cover against the sides of the disk pack as the cover is raised.

Step 5. Connect the bottom and top plates of the container and store.

Step 6. Gently close the lid on the disk storage drive. Raise the Start/Stop switch on the panel to Start. The disk pack will begin to rotate. An automatic sequence begins that checks for proper speed, cleans the disk surface and moves the read/write heads into Start position. The unit comes up to speed in about one minute. The green disk unit number, which also serves as an indicator, will light when the disk is ready for use. The unit will not be Ready until the disk is up to speed.

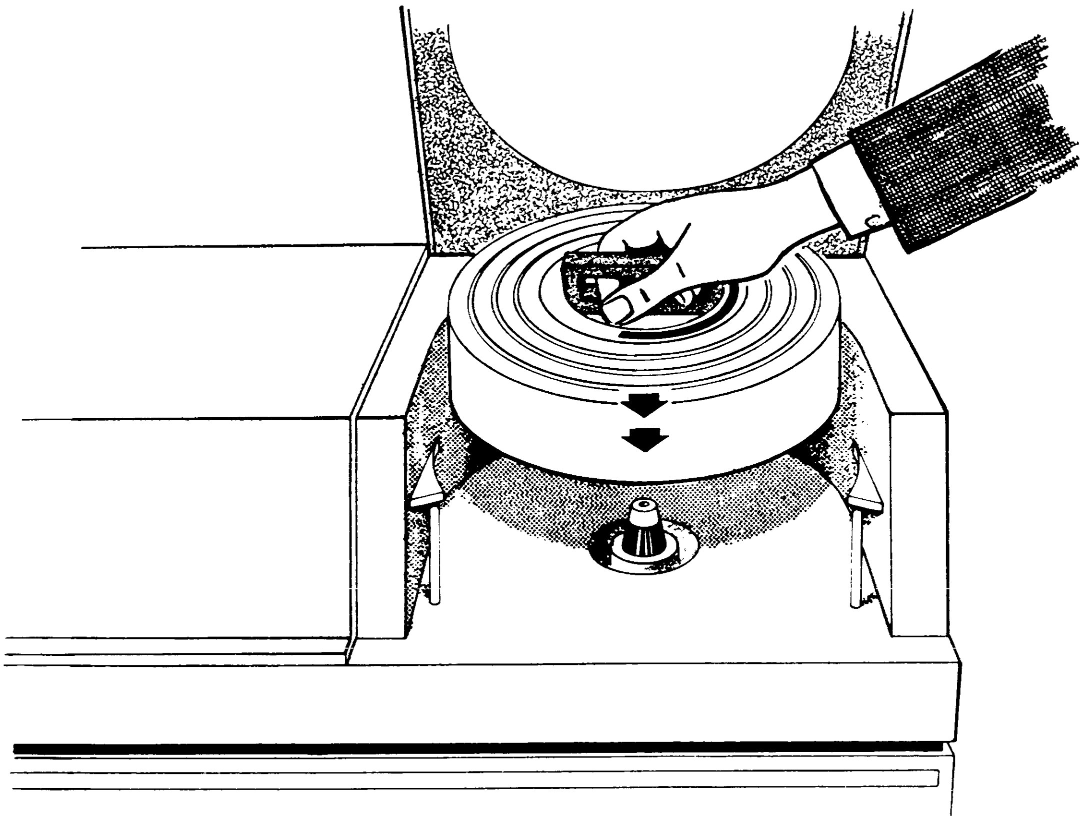


Figure 2-10. Lowering Disk Pack Over Spindle (Courtesy of IBM)

Step 7. Set the Enable/Disable switch to Enable, which connects the disk storage drive to the processor and places it under computer control.

The light labeled Select Lock indicates a malfunction in the unit which requires maintenance correction. This condition disables the drive. Contact the Customer Engineer.

2.5 THE LINE PRINTER

This device provides the bulk of printed material arising as the output of a run. The console typewriter, or keyboard printer, is used primarily for short messages or special information such as audit messages, counts, etc.; it is too slow for any volume of printing. The line printer, so named because it prints an entire 132 character line at a time, operates at speeds of 1100 lines per minute. As you might imagine, there are many crucial adjustments and tolerances required in a device in which the paper and printing mechanism is moving so fast. Your functions include selecting the proper form as specified by the run sheet, adjusting the various controls, checking to see whether the print quality and positioning is correct, and monitoring the feeding and stacking of forms.

The forms are one continuous sheet of paper, often in multiple copies, which are separated by perforated lines into separate sheets. The particular spacing, printing, and carriage movement are under program control and are not monitored by the operator.

There are six major operations required to make the line printer perform accurately and without malfunction. Each of these operations is more complicated to explain than to perform, but as with all of these hardware descriptions, the instructions are detailed so as to be as clear as possible. The six major operations are:

a. Loading the paper forms.
b. Aligning the forms.
c. Adjusting the printing controls.
d. Mounting the carriage control tape.
e. Adjusting the forms stacker mechanism.
f. Final checking and running.

It is first necessary to become familiar with some of the major controls, indicators, and pushbuttons on the printer. Each of these will be identified again as it is used in the operation.

Reference Figure 2-11. This is an overview of the entire printer unit. Note that the

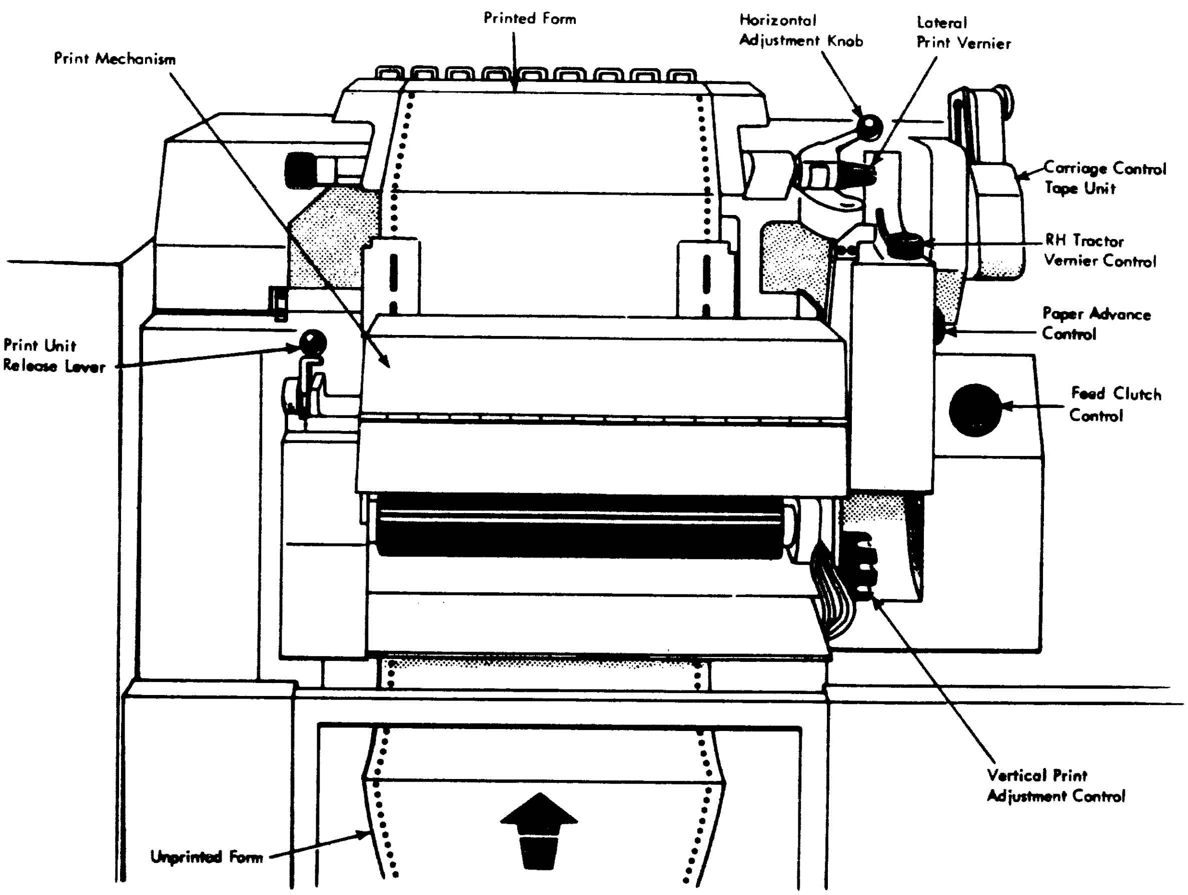


Figure 2-11. Line Printer (with Cover Raised) Showing Manual Controls (Courtesy of IBM)

form is pulled up and across the print mechanism and then is stacked. Sprocket holes in the paper form maintain alignment. The vertical spacing is controlled by the carriage control tape on the right-hand side.

The indicators and pushbuttons are shown and described in Figure 2-12.

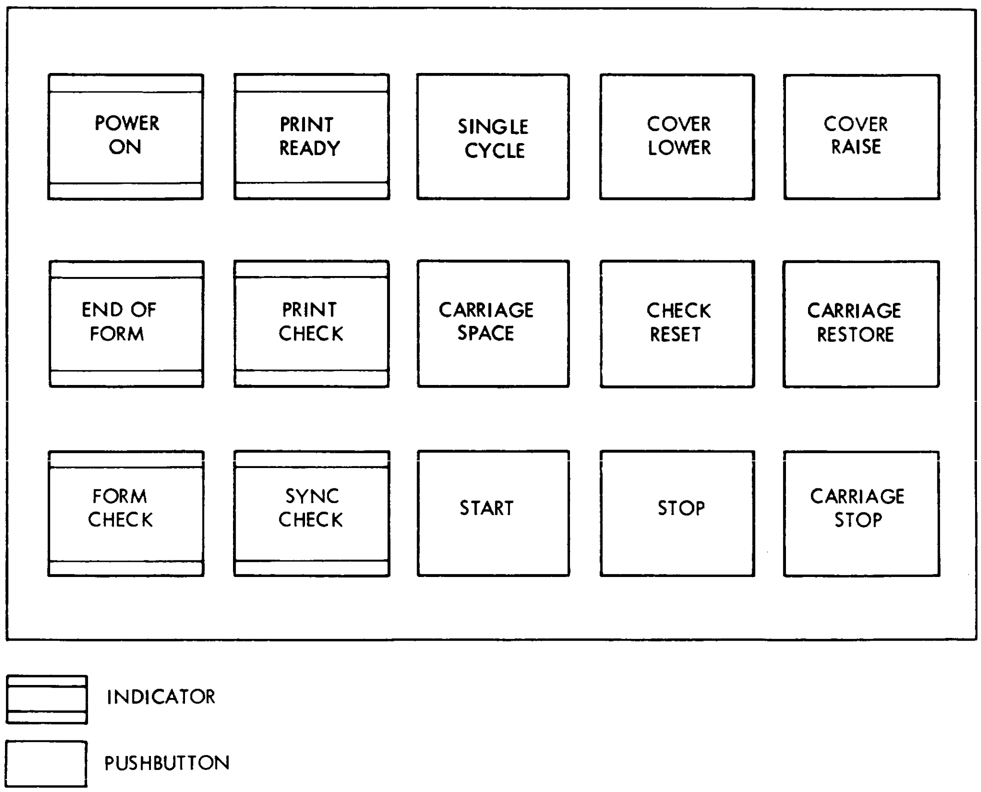


Figure 2-12. IBM 1403 Printer Model N1 Indicators and Pushbuttons (Courtesy of IBM)

START BUTTON—starts the printer unit. For your convenience there is also another START button on the back of the printer.

STOP BUTTON—stops the printer at the completion of the instruction in progress. For your convenience there is also another STOP button on the back of the machine.

COVER RAISE and *COVER LOWER*—these respectively raise or lower the top cover. The cover rises automatically on certain halts or if it encounters an obstruction while lowering.

SINGLE CYCLE BUTTON—this operates the printer for one cycle, i.e. one cycle of the carriage tape.

CHECK RESET BUTTON—resets printer error indications.

CARRIAGE RESTORE BUTTON—positions the carriage at its start (or home or channel 1) position. *Never press this button while the printer is printing.*

CARRIAGE STOP BUTTON—stops the operation of the carriage control, generally for realignment of the forms.

CARRIAGE SPACE BUTTON–moves the carriage one space.

END-OF-FORM, POWER ON, and *PRINT READY*–indicators are obvious.

FORMS CHECK INDICATOR–indicates a form feed problem.

PRINT CHECK INDICATOR–indicates a print error.

SYNC-CHECK INDICATOR–loss of synchronization between the printer drive mechanism and other elements.

The six major operations are now described in detail. Since the description is lengthy, try to keep in mind the purpose of each step in the context of the entire operation.

A. *Loading Forms*

Unless blank paper is used for output, you will generally have to change forms for each job. Check the run sheet and make certain you have the correct form *in the correct number of copies.*

Step 1. Check whether the top of the printer is clear. (It is good practice to keep forms, tools, and coffee cups off the printer cover, as the cover can automatically rise in response to certain malfunctions.) Raise the top by pressing the COVER RAISE button to expose the manual controls. Review again the control positions and names shown in Figure 2-11.

Step 2. The FEED CLUTCH control is the large knob facing you on the right-hand side of the machine. Note (Figure 2-13) that there are two positions marked NEUTRAL. Turn the knob so that the arrow points to either of the NEUTRAL positions. The form feeding mechanism is now disengaged from control of the printer.

Step 3. The PRINT UNIT RELEASE LEVER is the lever on the left-hand side of the print mechanism case. Pull this lever forward to the stop which unlocks the entire print mechanism and allows it to swing forward (on right-hand hinges) (Figure 2-14). This exposes the paper drive, forms tractors (which draw the paper through the printer), and portions of the print mechanism.

Step 4. Open all four TRACTORS (Figure 2-15). The forms now mounted may be removed by tearing off the form above the tractor area and letting it fall. It will fall over the top of the printer and into the stacker area at the rear of the machine. It may now be removed. The unprinted forms will fall back into the box at the front of the machine and may now be removed. Place the new box of forms in the front. Be sure the forms are facing in the proper direction.

Step 5. Set the LEFT-HAND TRACTORS slightly to the left of the first print position to be used for this job. The programmer's instructions will specify what the first print position is to be. The left-hand tractors are moved by lifting the latches (Figure 2-15) and sliding the tractor along the guide bar. Both tractors must be in the same vertical line and locked in position on one of the guide bar notches. The third notch from the left is the standard position. If notch 9 is used, the first available position for printing is position 38. Pull the forms up from the box, insert them on the tractor teeth, and close the tractors.

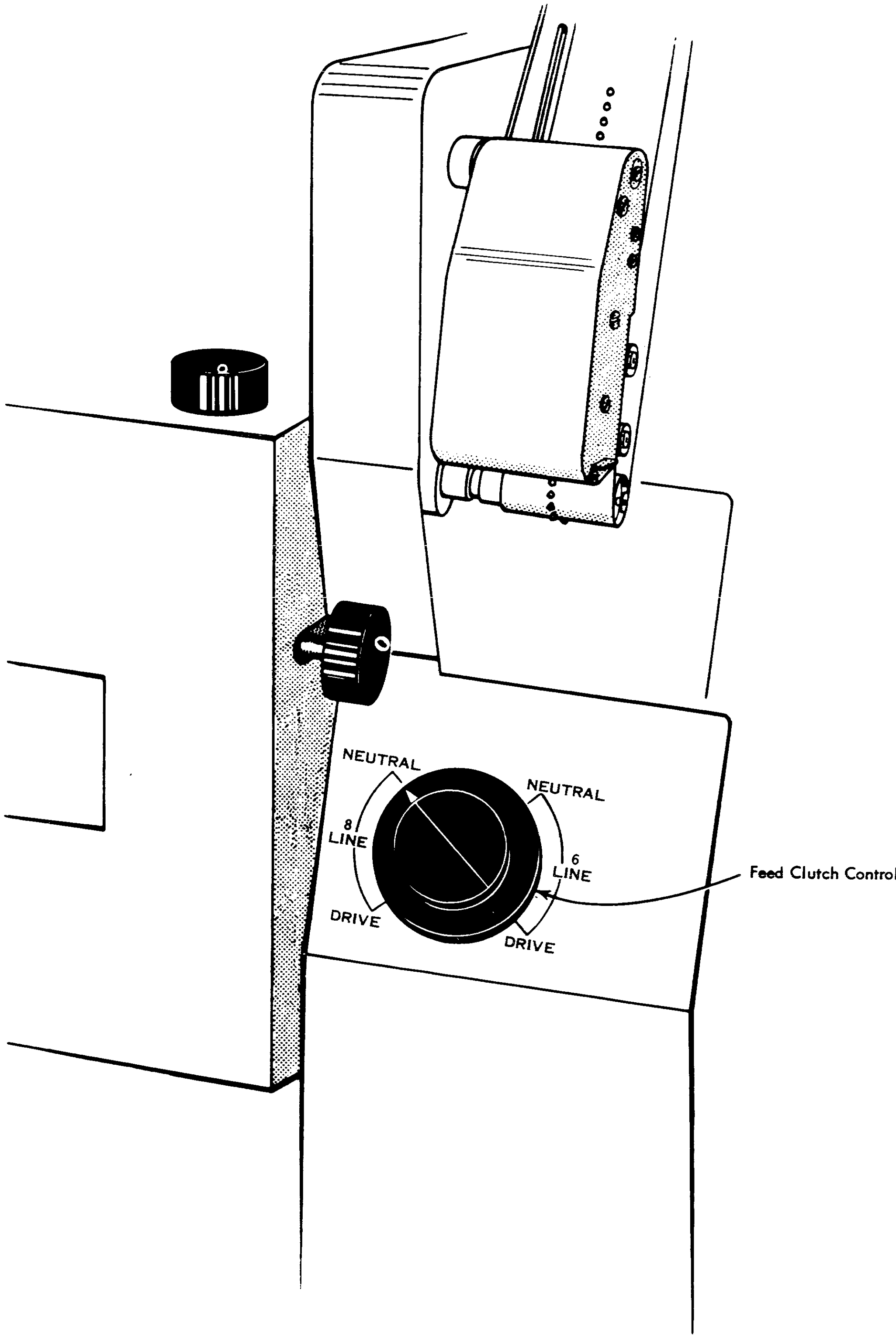


Figure 2-13. Feed Clutch Control (Courtesy of IBM)

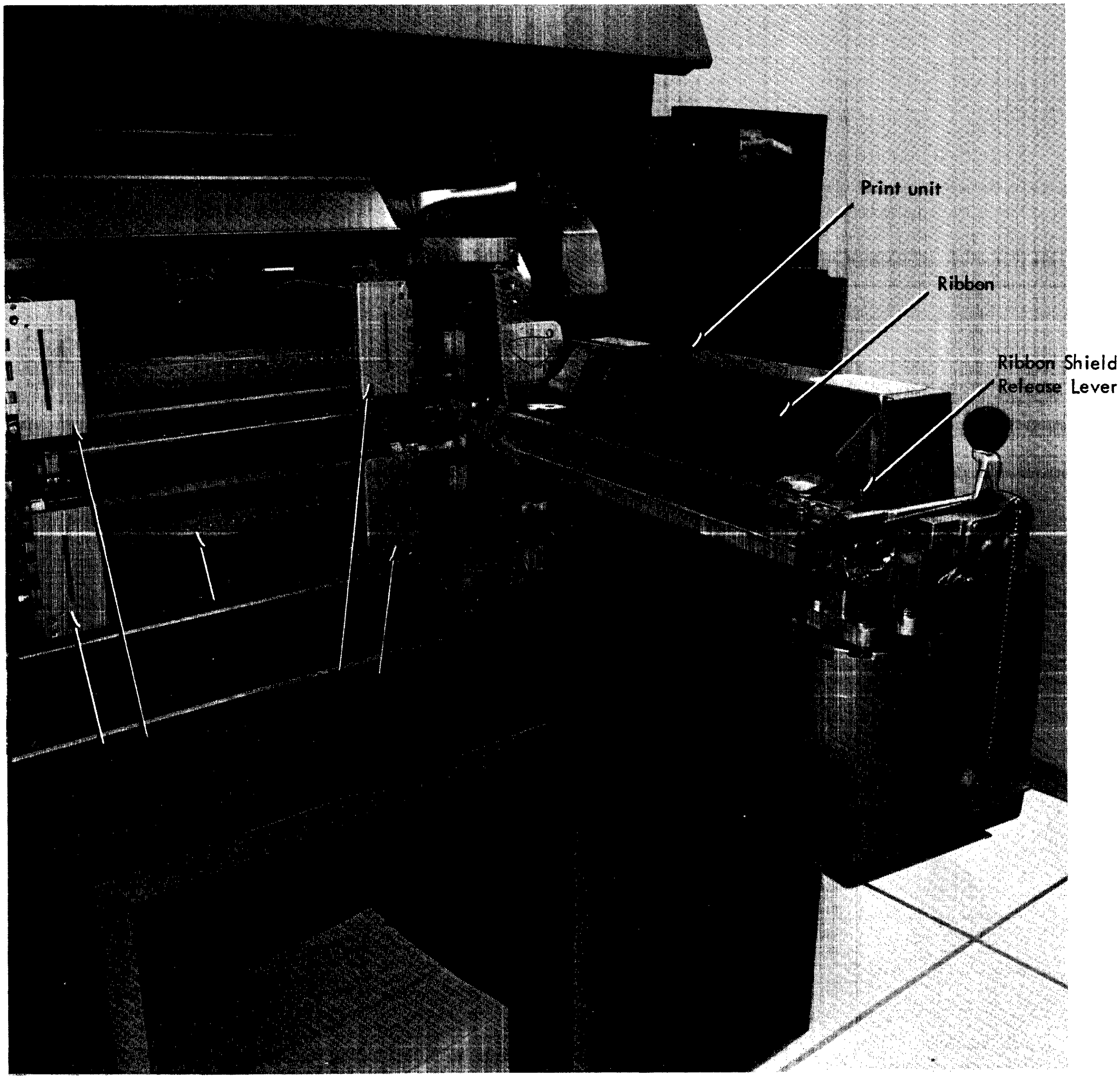


Figure 2-14. Print Unit Open, Showing Paper Drive (Courtesy of IBM)

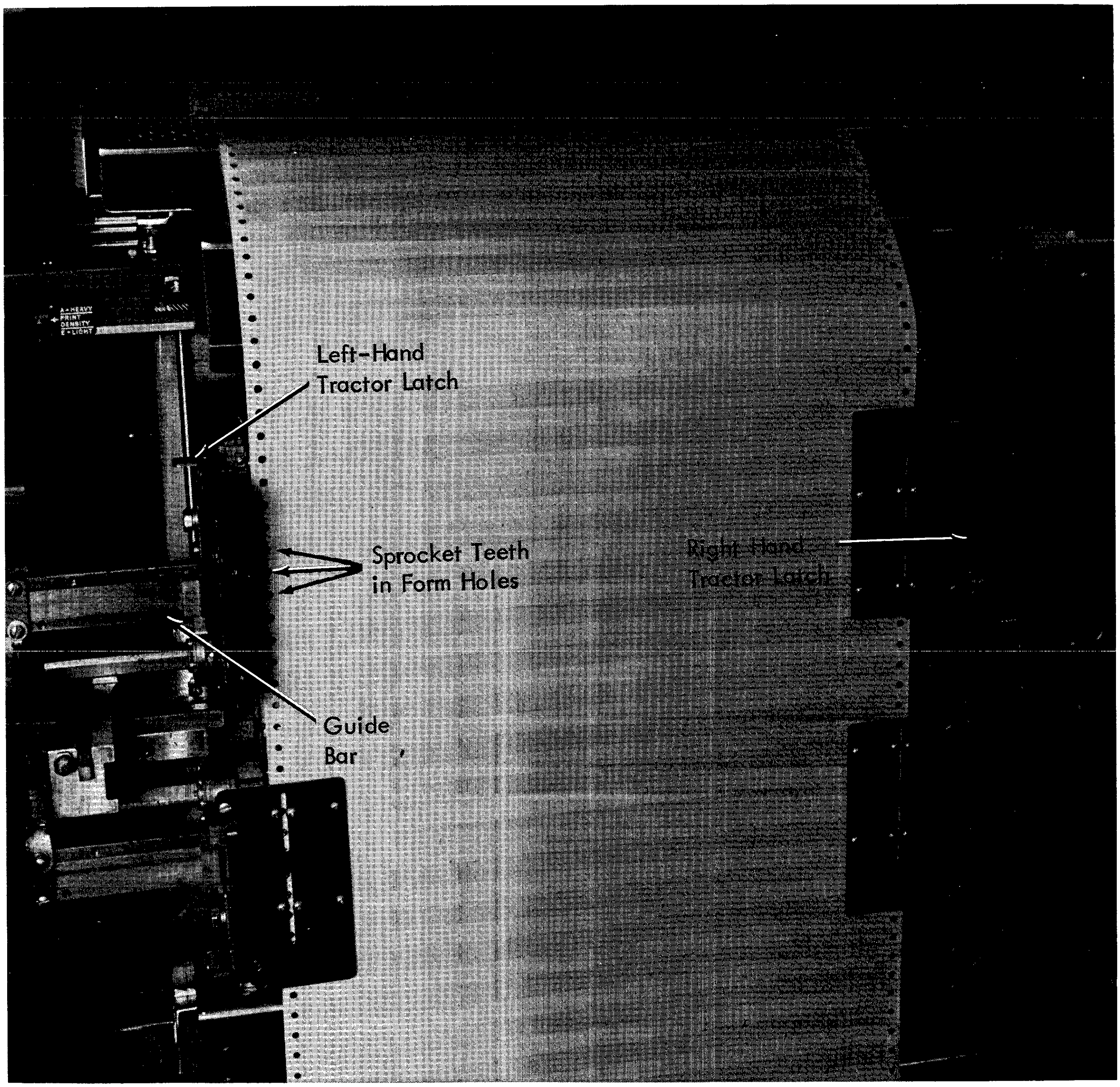


Figure 2-15. 1403 Tractors, Showing Left-hand Upper Tractor Open (Courtesy of IBM)

Step 6. Position the RIGHT-HAND TRACTORS so that their teeth engage the right-hand side of the form. The tractors are moved by pulling out the latch pins and sliding the tractor on the guide bars. The tractors are locked into position when the lock pin fits in one of the notches in the face of the guide bar. The form should now be slightly slack. Any fine adjustment necessary to accomplish this should be made with the vernier control (Figure 2-11).

Step 7. The tractor positioning can be done before leading the forms once the operator is familiar with the positioning. If this is done, the only portion of loading necessary is the placing of the forms in the tractors and the fine vernier adjustment. This is usually left to the preference of the individual.

The forms are not ready at this time for printing. They must be aligned properly to ensure that the printing will appear in the proper place on the forms. This is accomplished by a vertical and lateral alignment.

B. *Aligning Paper Forms*

By following the previous steps the forms have been positioned in the printer. Now it is necessary to align the forms for printing in the proper horizontal positions on the forms.

The lateral alignment may be done before vertical alignment. This is a three-step operation:

Step 1. Turn the *lateral print vernier control* left to the lock position, then right to the lock. By doing this a few times, it is possible to gauge the midpoint.

Once the vernier is set at the approximate midpoint, it is possible to proceed with the rough alignment.

Push the ribbon shield release lever (Figure 2-16) toward the forms. Push the shield against the forms.

Step 2. Raise the *horizontal adjustment knob* (Figure 2-17) and slide the entire print mechanism to the left or right as required. When the left margin of the form is even with the first ruled line on the ribbon-shield print line indicator, pull the *horizontal adjustment knob* down and lock the print mechanism. This will now be accurate within one print position.

Step 3. Turn the *lateral print vernier control* in either direction until the form is properly aligned with the print line indicator. Positioning of the print line horizontally is now complete.

The vertical adjustment of the print line is relatively simple. There are but two steps.

Step 1. The position of the bottom edge of the print line is indicated by the *ribbon shield opaque band* (Figure 2-18). The bottom edge of the print line will be even with the raised edge or an eighth of an inch above the band.

Turn the *paper advance knob* (Figure 2-11) until the first line is within a fourth of an inch of the band. If desired printing is to start at the top of the form, the form perforation should be just above the print line indicator.

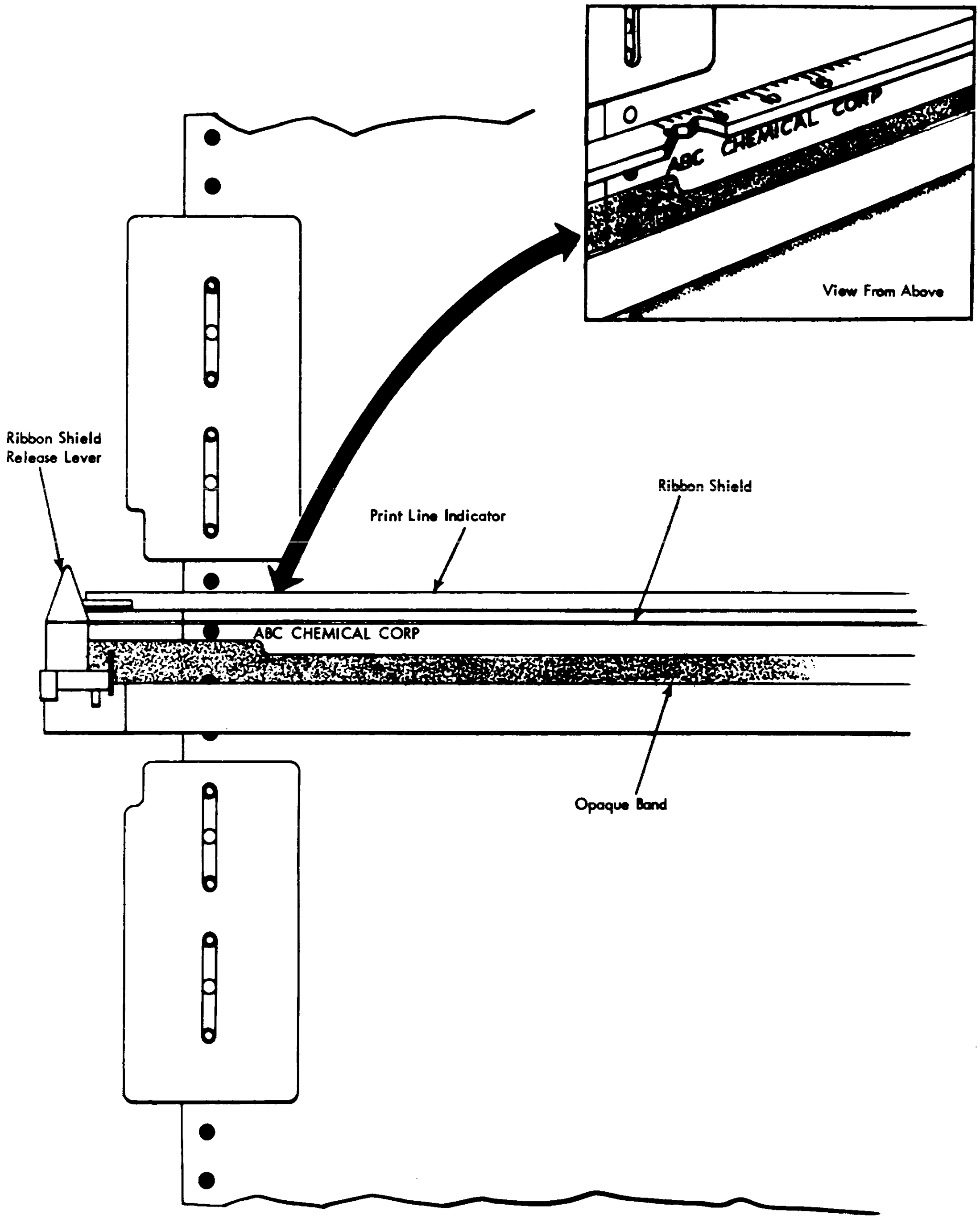


Figure 2-16. Use of Ribbon Shield for Alignment (Courtesy of IBM)

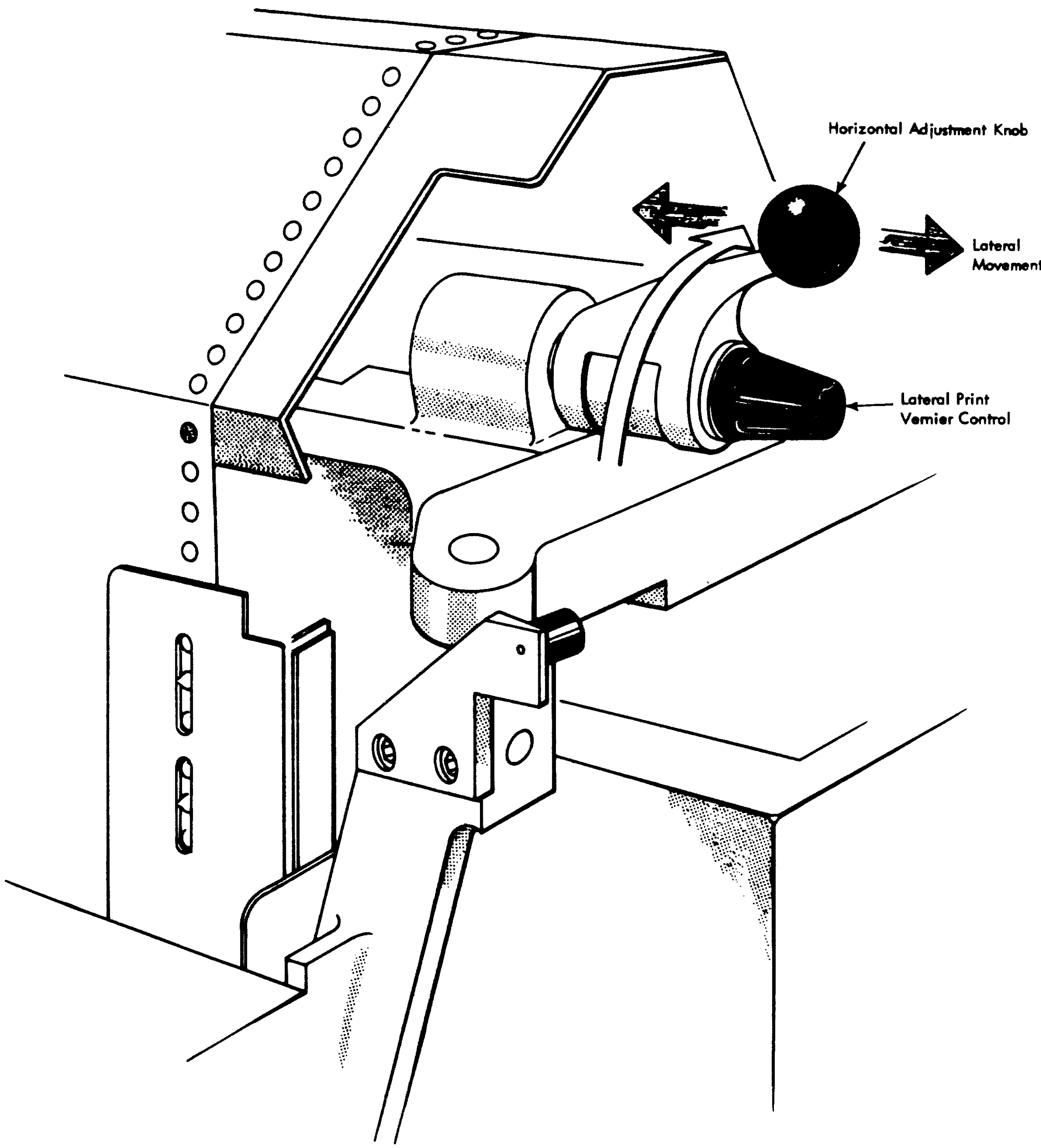


Figure 2-17. Lateral Alignment of Print Mechanism (Courtesy of IBM)

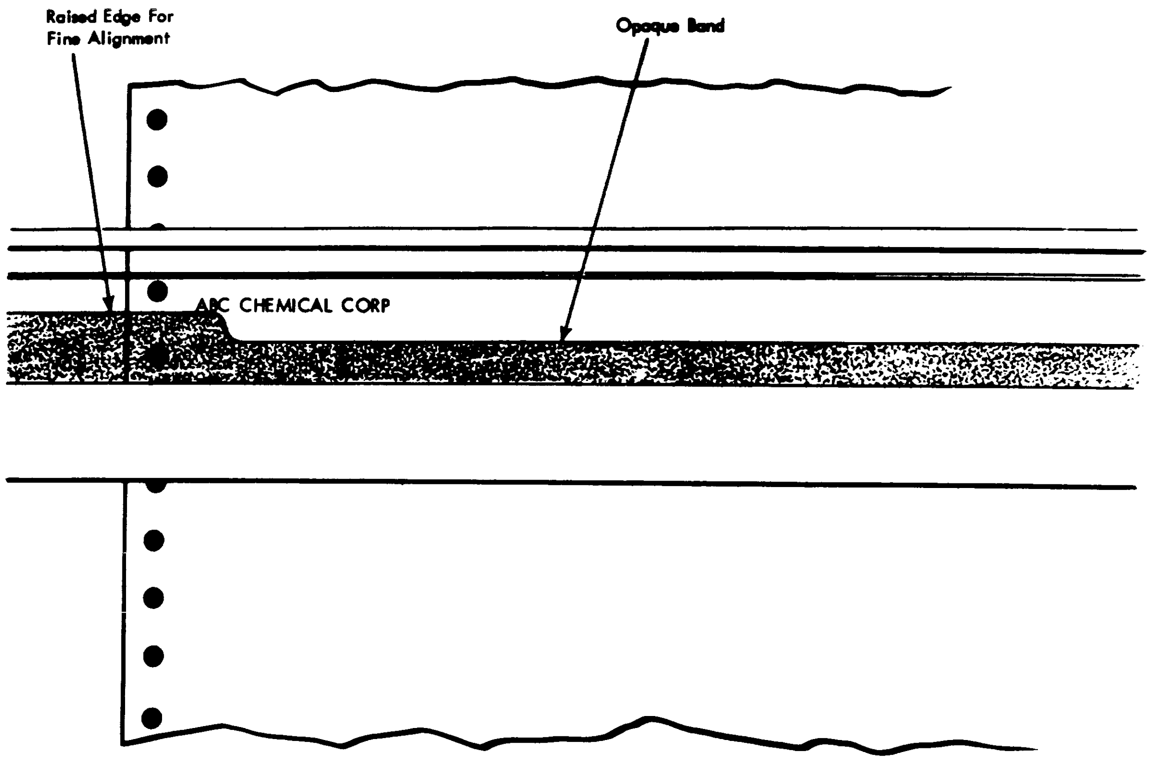


Figure 2-18. Vertical Alignment Detail (Courtesy of IBM)

Step 2. Using the *vertical print adjustment control* in either direction, align the print line to its proper position. Close the print mechanism and lock it into place with the print unit release lever.

C. *Adjusting the Printing Controls*

Before any printing is done, the print quality must be adjusted so that the print is neither too light nor too dark. This is important if a multi-part carbon form is being used. Wear of the print ribbon may also dictate some adjustment. This is commonly called print density adjustment.

The print density adjustment is accomplished by use of the *form thickness lever* and the *print density control.*

The *form thickness lever* should be set to a value equivalent to the approximate thickness of the form. The setting 0.003 is commonly used for single-part plain stock forms.

Use a preliminary print density setting near the middle of the range.

These settings may have to be repositioned after examining the first page of printing.

D. *Mounting the Carriage Control Tape*

The instructions supplied by the programmer will also indicate by name or number the proper carriage control tape to be used.

Mounting the carriage control tape is a simple operation, but it must be done correctly to ensure proper print control. There are three steps involved in mounting a carriage control tape.

Step 1. Before mounting the new control tape, remove the old tape and thus position the brush mechanism for loading.

Pull the latch on the right side of the brush holder forward and raise the brush holder to expose the tape drive and idler pulleys. There is a lock knob on the right end of the *idler pulley*. Turn the knob clockwise, and slide the idler down its slot.

Step 2. Remove the old carriage tape and place the new tape around the *pin-feed drive wheel* and *idler*, making sure the numbered edge is to the right and facing out.

Tighten the tape by sliding the *idler* up and turning the lock knob. No visible slack should appear, but if the tension is correct, the tape should give about one-fourth inch when pushed gently (Figure 2-19). Make sure the tape is positioned on the pin-feeds correctly or this will cause damage to the tape.

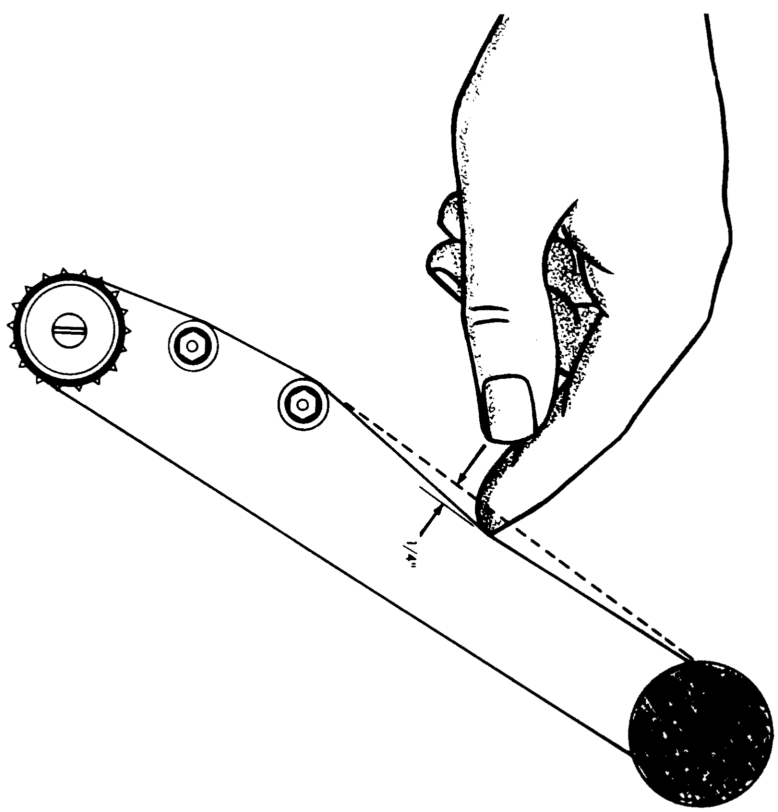


Figure 2-19. Testing Tape Tension (Courtesy of IBM)

Step 3. To position the control brush, lower the brush holder into place and the latch is engaged. The tape is mounted but it is not synchronized with the forms.

The synchronizing is simple. Press the CARRIAGE RESTORE button. It might be necessary to press CHECK RESET first, if the FORM CHECK light is on. Make sure at this time that the *feed clutch* is in one of the two neutral positions or it will be necessary to go through the entire operation again.

After the CARRIAGE RESTORE button has been pressed at least twice, the tape should be synchronized.

Turn the *feed clutch* to either the 6 line or 8 line drive position. Press the CHECK RESET button as the FORM CHECK light is on again.

Press the START button; the printer is now ready, but the paper stacking area may have to be adjusted to accept the forms just loaded.

E. *Adjusting the Forms Stacker Mechanism*

Once the job is started and the printer is printing, the paper forms that are printed will have to be initially guided over the printer form path and down into the stacker. The stacker may have to be adjusted to accept and stack the forms properly. This could also be accomplished by several carriage restores to completely set up the printer before the job is run.

The outer gate in the stacker area (at the rear of the machine) is adjustable. It may be necessary to guide the paper through the roller into the stacker by opening the access door at the rear of the machine and hand feeding the forms down through the rollers.

It is possible to release the tension of the rollers by grasping the tension bar and turning it out of the way. Now the forms may slip into the stacker area tension free. This allows for easier adjustment.

The stacker tension rollers slide up and down on their track. The initial setting should be about 1/2 to 2/3 the distance from the bottom of the stacker. The tension rollers should be adjusted periodically as the forms fill the stacker.

Any additional carriage restores needed to complete readying the stacker may be done by using the three keys in the upper right corner of the stacker area. The START, STOP, and CARRIAGE RESTORE keys here perform the same function as the keys on the front panel.

All should now be ready after the tension rollers are returned to position. It is always good to do a final check to ensure that all the mechanical operations were done correctly.

F. *Final Checking and Running*

It is not necessary to repeat any of the previous operations, but simply to check that they were done:

1. Is the feed clutch in either 6 or 8 line drive position?
2. Is the carriage tape mounted correctly so that you can see the numbers on the right-hand side of the tape?
3. Is the print mechanism locked so that the handle is pointing up instead of out?

4. Is the print density and form thickness set correctly? (This may have to wait until the job starts printing.)

5. Are any indicator lights on other than POWER and/or FORM CHECK? (Reset any check lights by pressing CHECK RESET.)

6. Are the tension rollers in position?

7. Are the forms loaded properly to print on the proper side and in the proper position?

8. Is the printer ready? (Press the START button.) Now lower the cover by pressing the COVER LOWER button. If multi-part forms are being used it might be necessary to leave the cover raised until the print density is adjusted.

2.6 CONSOLE CONTROL PANEL

The control panel which is located on the 2030 processing unit is used to maintain system status and to alert the operator to existing or potential malfunctions. Figure 2-20 is a diagram of the Model 30 Control Panel.

Each of the keys and switches located on the console serves a singular function. This discussion will cover them in alphabetic order as labeled on Figure 2-20, giving their names and functions.

KEY A (POWER ON): This key supplies power to the entire system. If power cannot be brought up on any unit attached to the system, the power-on sequence cannot be completed and the back-light behind this key will not come on.

Each control unit has a LOCAL/REMOTE switch. If the power-on sequence is executed and one of the units is in a local state, that unit will be left out of the power-on sequence.

Ask your customer engineer to identify the LOCAL/REMOTE switch on the control units for you. This will enable you to determine failures during a power-on of the system.

If any problems occur during the power-on sequence that prevent bringing power up on all devices, the customer engineer should be notified.

KEY B (POWER OFF): This key performs the reverse function of the POWER ON key. It removes power from all devices attached to the system.

If the power-off sequence fails to remove power from one of the devices, the control unit associated with that device must be switched to LOCAL.

If your customer engineer has identified the LOCAL/REMOTE switch for you, you may drop power on the unit by switching it from a LOCAL to a REMOTE state.

If problems occur during a power-off sequence, notify your customer engineer.

SWITCH C (EMERGENCY PULL SWITCH): The EMERGENCY POWER OFF switch removes power from the system immediately. It does not go through a power-off sequence.

This switch is to be used only in case of disaster. Do not pull this switch unless it is absolutely necessary.

This switch must be reset by a customer engineer. It has a lock on it which will not allow it to be reset by pushing it back in. The customer engineer will have to be called to reset this switch.

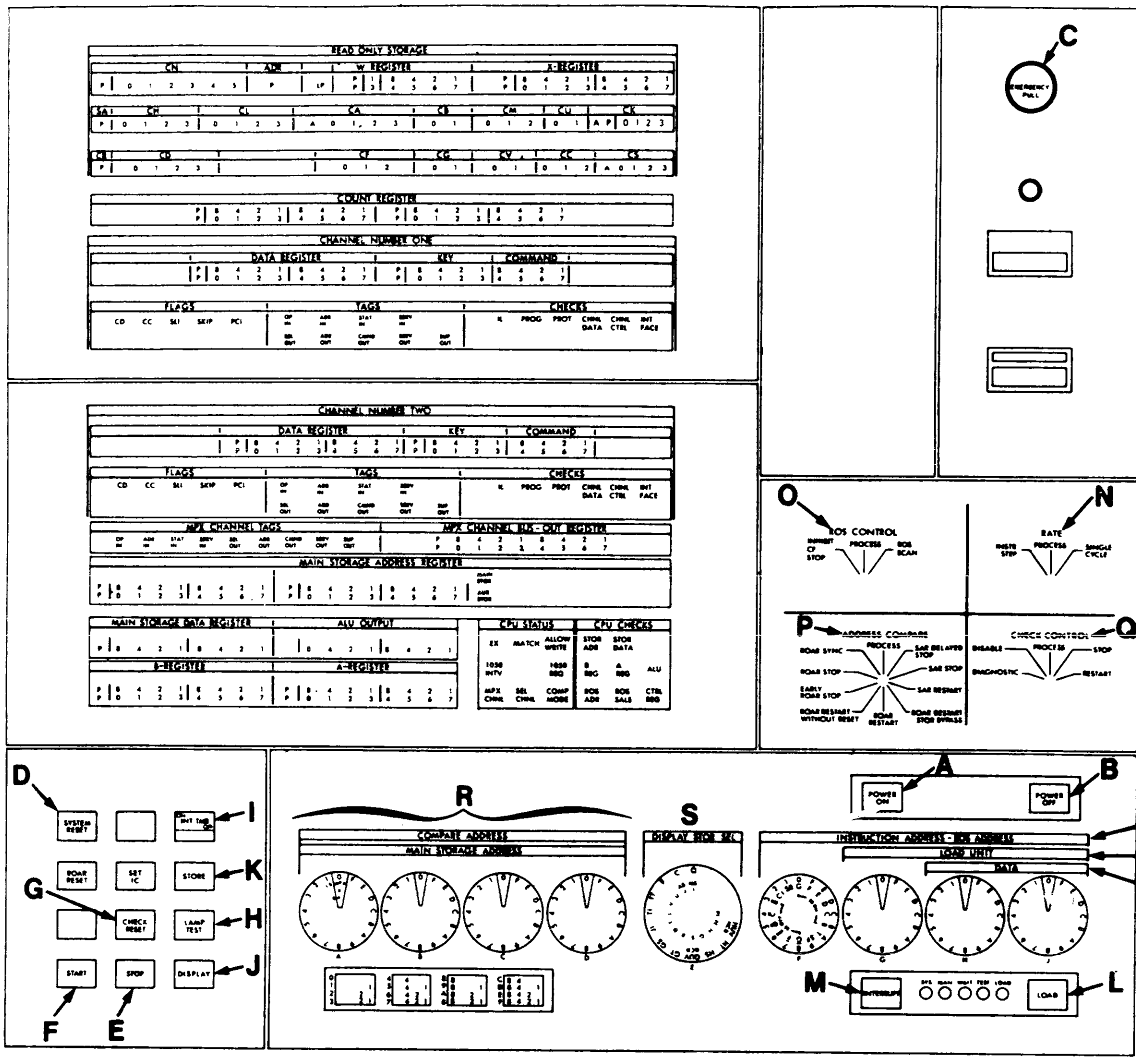


Figure 2-20. Model 30 Control Panel (Showing Keys and Switches) (Courtesy of IBM)

If the emergency power off is pulled, the wall circuit breaker for the machine room should also be pulled, if possible.

KEY D (SYSTEM RESET): The SYSTEM RESET key resets all the units on the system. It also stops all processing by placing the machine in a manual state. Always press the STOP key before pressing the SYSTEM RESET key.

The only time this key should be pressed is in the event of a hardware malfunction. This will be indicated on the CPU console (these indicators will be discussed).

KEY E (STOP): The STOP key places the system in a manual state and stops any further execution. The system will finish its last operation before entering this state.

KEY F (START): The START key allows us to restart after the STOP key has been pressed. The machine will continue processing from the instruction on which it stopped.

When the START key is pressed, the system will begin processing in the manner specified by the RATE switch (discussed later).

KEY G (CHECK RESET): The CHECK RESET key resets all error indicators on the system. There may be a few exceptions to this depending on the devices attached to the system.

KEY H (LAMP TEST): The LAMP TEST key enables you to determine if any lights are burned out on the console. It may be used at any time, and it does not affect processing. Burned-out bulbs should be noted and reported to the customer engineer for replacement.

SWITCH I (INTERVAL TIMER): The timer may or may not be present on your machine. It keeps track of the time of day and the duration of jobs. Some programs require that it be off. If your machine is equipped with a timer, it must be on when running under the Disk Operating System.

KEY J (DISPLAY): The DISPLAY key allows us to find out what is stored at a particular location in the machine. Information fed into the machine in the form of a program may be examined by means of this key.

The location we wish to display is specified by the use of other switches.

KEY K (STORE): The STORE key enables us to change a particular location in the machine. Information in the form of a program may be changed by the use of this key. The information being entered and the location into which it is being entered are specified by the use of other switches.

KEY L (LOAD): The LOAD key is responsible for the initial program loaded into the system. The device address is set up by other switches.

When the LOAD key is pressed, the system goes to the device that is specified in the load unit switches (usually a reader, disk, or tape drive) and attempts to read a program into the CPU.

When running under the Disk Operating System this is not done for each program, but this will be discussed later.

KEY M (INTERRUPT): the INTERRUPT key generates an interrupt in the system. The INTERRUPT key is used mostly in conjunction with jobs that do not run under an operating system.

SWITCH N (RATE): The RATE switch determines the rate at which a program will be executed. The normal position of this switch is PROCESS.

There is no "OFF" position for this switch. It will always be set to one of three positions–PROCESS, INSTRUCTION STEP, or SINGLE CYCLE.

If the RATE switch is set to PROCESS, the job will run at normal speed.

If the RATE switch is set to INSTRUCTION STEP, one instruction will be executed each time you press the START key, and the machine is thus executing the job at the rate you press the START key.

If the RATE switch is set to SINGLE CYCLE, the machine will execute your program at a rate of one micro instruction for each depression of the start key. There are many micro instructions executed for each instruction of a program, so this setting is seldom used.

Do not change the RATE switch setting while the job is running. Press the STOP key first before changing the rate.

Do not set the RATE switch to anything other than PROCESS unless you are told to do so.

SWITCH O (ROS CONTROL): The ROS CONTROL switch is set to PROCESS and has no value from an operator standpoint. Later we will discuss the only occasion you would have to use it.

SWITCH P (ADDRESS COMPARE): The ADDRESS COMPARE switch is used to stop the CPU at a predetermined address.

With the ADDRESS COMPARE switch set to anything but PROCESS, the CPU will stop when it reaches the address designated by the switch settings (to be discussed).

Only the settings which will be discussed later are important to an operator.

SWITCH Q (CHECK OR CHECK CONTROL): The CHECK CONTROL switch should be set to STOP when you are doing normal processing unless your shift supervisor is taking the responsibility for recording all error information.

The setting of this switch is of major importance if a machine CPU error occurs. We will use this switch in our normal operation procedures to detect and report errors.

If the CHECK CONTROL switch is set to STOP, all CPU errors will bring the system to a stop. These electronic errors produce diagnostic aids but the aids are only valid if the machine is run with CHECK CONTROL set to STOP.

SWITCH R (COMPARE ADDRESS/MAIN STORAGE ADDRESS): These four switches labeled MAIN STORAGE ADDRESS are used in conjunction with the DISPLAY, STORE, and ADDRESS COMPARE switches.

These four switches will contain a predetermined address at which the system will stop.

As an operator, the only time you will use these switches is when you are displaying information in the case of an error, or when the programmer's instructions tell you to use them.

The numbers in these switches are meaningless unless the associated switches are set.

SWITCH S (DISPLAY STORAGE SELECT): The SELECT STORAGE switch is a dual-level switch with three colors on the central switch and three rows of characters.

Each item on the outermost dial represents a location in the CPU in which we may find information. The only two that are of importance to an operator are the MS setting and the AS setting. MS is the area in which the program you are executing is located. This is commonly called MAIN STORAGE. AS is the area in which the aids for the execution of the program are operating.

The color-coded switch must be set with the pink area up to allow us to access information in the program area. MS must also be set to the UP position on the outer switch.

SWITCH T (INSTRUCTION ADDRESS–ROS ADDRESS): These four switches will be of no use to you as an operator unless you are performing some non-System/360 operation. These switches will not be discussed in this text.

SWITCH U (LOAD UNIT): The LOAD UNIT switches are used in the initial loading of a program. The switches should be dialed to the address of the unit from which you intend to load the program.

If you are running your job under the Disk Operating System, the switches will be initially set up to indicate the disk upon which your system is resident (the operations controller will tell you this).

An example of a common load unit is "OOC" which is the most common address of the card reader.

SWITCH V (DATA): The DATA switches are used in conjunction with the STORE and MAIN STORAGE ADDRESS switches.

If you intend to alter some area in the program, these switches must all be set up properly to guarantee that the information that is being altered is in the correct position and that the information you are inserting is correct.

As an example let us see what operations are required to display the contents of a memory location (i.e. a single byte of information). Referring to Figure 2-20 the first step is to press the STOP button (lower left). The second step is to set the Display Storage Selector to Main Storage (lower outer). Step 3 is to set the Main Storage Address Selectors to the hexadecimal byte address desired (lower center left of Display Storage Selector). Step 4 is to press the Display button (lower left). The address then appears in the Main Storage Address Register Display (center left) and the byte is shown in the Main Storage Data Register (center left). For diagnostic reasons the display of memory is a useful control feature.

EXERCISES

1. Define each of the following terms:

   a. File-Feed
   b. Hopper
   c. Joggler Gate
   d. Print Density
   e. Carriage Control Tape
   f. Off-Line, On-Line
   g. Feed Clutch
   h. Non-Process Runout

2. Which of the following is only an I/O device?

   a. Magnetic Tape Unit
   b. Disk Storage Drive
   c. Card Read-Punch
   d. Line Printer

3. What is the first step in readying the keyboard-printer for operation?

4. How many stackers are there on the 2540? How many of these are used for reader or punch operations?

ANSWERS

1. a. File-Feed–Vertical tray that holds the punched cards until entry into the hopper.
   b. Hopper–The hopper is the area where the cards are held in place for card reading of the punched holes.
   c. Joggler Gate–This is a mechanical device which opens and closes to allow proper alignment before cards are to be read.
   d. Print Density–This control adjusts the darkness of the characters for proper printing and the setting used is based on the number of carbon copies to be printed and the wear aspect of the printer ribbon.
   e. Carriage Control Tape–This tape controls the feeding of forms into the printer. The punches in the carriage control tape determine when the form is to skip and stop to allow printing in desired area.
   f. Off-Line, On-Line–These expressions refer to whether a device is connected electrically to the system (on-line) or is part of a configuration for performing a given function but not physically attached to the system (off-line).
   g. Feed Clutch–This knob is used to engage or disengage the form feeding mechanism from printer control to manual (operator) control and vice versa.
   h. Non-Process Runout–This is a process which causes any cards that remain in the card reader to fall into the stacker without being read or computer processed.
2. The line printer (d) can only be used as an I/O device.
3. The first step in readying the 1052 keyboard-printer for operation is to turn the 1051 control unit main-line power switch on.
4. There are five stackers on the 2540. Three of the five stackers are available for either reading or punching operations.

3

RUNNING A JOB

In Chapter 2 you have learned about the various components that constitute a System/360 Model 30 installation and how to operate each unit individually. In order to be a computer operator, however, it is necessary to learn how to operate the entire system. This Chapter, therefore, describes the procedures and techniques for setting up and running a job which involves all of the basic Model 30 units.

3.1 PRELIMINARY ORGANIZATION

Prior to the running of a job it is necessary to have available two sets of information. One set is referred to as the program and the other set is called input data. The program is a set of instructions which dictate to the computer as to what functions it is to perform. The input data can be viewed as the necessary information that must be given to the computer in order for it to solve the problem or utilize the program to obtain the required output information.

In general, the programmer writes his program in a special language that has been developed for speed and ease of use. This type of programming language is referred to as a *higher-order or human-oriented language* and the two most common examples of such a language are COBOL and FORTRAN. The computer or machine used to process the data and operate via the program can only understand and operate in its own language (i.e. machine language). It is necessary, therefore, to find a way of converting from the *higher order* language to machine language to enable the computer to execute the program. The higher order language program is referred to as the *source* program. The source program is converted to a program in machine language by a computer run through the use of a *translator or compiler*. The resultant machine language program is called the *object program*. The object program is computer executable and hence can operate on the input data to give the resultant output data. It should be noted that the object program can be stored on cards as a program deck, or on magnetic tape, or on a disk.

In addition to the program and input data the programmer also supplies the computer operator with a set of instructions to follow in the processing of his particular program. This instruction set is called the *run book*. The type of information contained in the run includes program error messages, error correction procedures, which auxiliary storage and I/O devices are to be used, and how many of each, which forms to use in the line printer, and what procedures to use in order to continue processing when deviations from normal operations are required or what procedure should be used in the event a restart is necessary. It is evident, therefore, that the computer operator must work closely with the programmer in order to achieve maximum results in the processing of a job. The run book, similar to the program and input data, is a tool of the trade through which the operator interfaces with the machine to effect the processing of the data. Each program has associated with it a specific set of instructions and thus the name run book results as a collection of different sets of program instructions.

It is important to note that the computer operator need not understand all the reasons behind the run book instructions or the program structure. He need only understand what functions he is to perform in the course of running the job. The operator also is not responsible for the daily scheduling of jobs or the assignment of priorities. These functions are the responsiblilty of the computer installation manager. The daily schedule identifies the jobs to be run (the operator's work load) and whether or not any special functions must be undertaken by the operator. For example, various units are scheduled for preventive maintenance at preassigned time intervals and these items must be posted on the daily schedule as they come up.

In general, the schedule is a straightforward document with column headings describing the job number and name and the type of job being run. The first column is used to indicate the sequence of jobs to be run and any changes can be effected by a simple transposition of numbers in this column. The last column is used for operator signature after a job has been completed.

3.2 INITIALIZATION

Before a job can be run it is first necessary to initialize or ready all of the components of the system that are required to be operational. In Chapter 2 we learned how to ready and operate individual units. Now we are going to learn how to ready the entire system. For the purposes of this description we will assume that the job to be run requires the use of the basic Model 30 components, namely, the 2540 card reader-punch, the 1403 line printer, and the 1052 keyboard. Later on, we will indicate the initialization procedures for jobs which use magnetic tape units and disk storage drives in addition to the card reader-punch.

The step-by-step procedure for readying the Model 30 to process any job is as follows:

1. Tape Unit Caution
2. Turning on Power
3. Readying the Card Reader
4. Readying the Line Printer

The Tape Unit Caution is the first step in setting up the system because it prevents the accidental erasure by the operator of information on a tape reel which has not been removed from the tape unit following a prior job run. The procedure is to check whether any tape reel is mounted on any of the tape units on the system. If there are tape reels mounted, the next step is to see if the tape heads are in the *up* or *down* position. If the tape heads are in the *up* position then no damage can occur by turning on system power. If, on the other hand, the tape heads are in the *down* position then the power cannot be turned on since it can cause the loss of information stored on the tape. In this case, the operator or his supervisor should remove the tape reel(s) in question prior to turning on system power.

The procedure for turning on system power requires nine steps. The controls used are found on the 2030 control panel except for certain indicator lights which are located on particular units. Referring to Figure 2-20 (2030 Control Panel) the procedure is as follows:

1. Set ROS CONTROL switch to PROCESS.
2. Set RATE switch to PROCESS.
3. Set ADDRESS COMPARE switch to PROCESS.
4. Set CHECK CONTROL switch to PROCESS.
5. Depress POWER ON pushbutton indicator. POWER ON button lights up in a few seconds indicating that power has been applied to entire system except for 1052 Printer Keyboard. At the same time the MAN (manual indicator on lower right) light should also come on indicating that the system is in the *stopped* or *manual* state.
6. Assure that full system power is on by checking that the card reader POWER, the line printer POWER ON, and the tape units FILE PROTECT lights are all on.
7. Check the LP (low pressure) display on the control panel to see if it is lighted (found to left of W Register on top row of lights). If the display is lit up, notify your supervisor immediately and continue the procedure.
8. Conduct display lamp test by depressing LAMP TEST pushbutton (lower left) and seeing whether all lights turn on. If any lights are burned out notify your supervisor and continue the procedure.
9. Finally, turn on 1052 POWER switch and wait for POWER indicator light to come on. Set the Printer, Keyboard, and CPU control switches (see Figure 2-2a) to the *on* position. The entire system power is now on and we can proceed with the next part of the initialization process.

The procedure for readying the card reader has been described in Chapter 2 and is therefore only relisted here for system procedure continuity. It should be noted that since we are describing a card program to line printer output report system the card program deck and the data deck are both fed into the file-feed. The sequence is program deck first, to permit compilation, and input data deck directly behind the program deck. The procedure is:

1. Depress STOP pushbutton on card reader.
2. Remove any cards from card reader file-feed.
3. Open joggler gate.
4. Remove any cards from hopper.
5. Depress and hold START pushbutton on card reader.
6. Close joggler gate.
7. Place card decks in card reader file-feed (face down, 9 or bottom edge first) in order specified by run book.
8. Place card weight on top of card deck(s).
9. Depress card reader START pushbutton. Wait for card reader READY light to come on to proceed.
10. If the card deck(s) fit into file-feed, depress END OF FILE pushbutton on card reader. If deck is too large, wait until sufficient cards have been processed for last group of cards to fit in file-feed and then depress END OF FILE pushbutton.

The line printer readying procedure has also been described in Chapter 2 and is included below for continuity, as with the card reader. The procedure is:

1. Load appropriate paper forms.
2. Align forms to insure correct printing area.
3. Set printing controls for best quality printing.
4. Mount the carriage control tape.
5. Set up paper stacker.
6. Conduct final readying and checking.

The above set of procedures has readied the Model 30 to run a card program and produce a printed output report. The items covered in the initializing procedure were prevention of tape reel information damage, turning on system power, and readying of the console keyboard-printer, card reader, and line printer. The system is now ready to run a job, but the commands for starting the processing have not yet been issued. These commands to begin the readying and processing of cards in the card reader are thus the next step for the operator in preparing the system to run a job.

Before proceeding to describe the procedures in the starting of system processing it is necessary for the sake of completeness to indicate what additional steps are required if the I/O devices consist of magnetic tape units and disk units in addition to the card reader. The details for readying each unit have been given in Chapter 2 and our intention here is to list the sequence of initializing operations. Let us assume that the system involves the processing of accounts payable. The master file is kept on a disk, the program for updating the file is maintained on tape, the new monthly payable entries are punched on cards, and the checks are printed on predesigned check forms. The procedure for readying the system to process this program is as follows:

1. Unload any tape reels on tape unit.
2. Mount the accounts payable program tape.
3. Mount the master accounts payable file disk pack.
4. Insert the control cards and the monthly accounts payable card deck in the card reader.
5. Ready the printer to print the accounts payable checks.

The only step which has not been described previously is the readying of the line printer with the check forms. The setup procedure is the same as that described in Chapter 2. The alignment procedure requires greater care since the printing process must assure that the printing will occur in the designated areas on the checks. The run sheet will dictate the print positions for the different items such as account name and address, and dollar amount.

3.3 OPERATING PROCEDURES

Going back to the simple card program and input data to line-printer output report system we shall now describe the procedures necessary to activate system start and data processing. These procedures are:

1. Go to 2030 control panel and set switches labeled G, H, and J (each switch is labeled with a letter) to the address of the I/O unit from which instructions are to be read (in this case the 2540 card reader).

2. Depress the LOAD pushbutton on the 2030 control panel. All control program instructions are immediately read by the system. After this action is completed the system goes into a wait state which is indicated by the wait light coming on.

3. Depress the INTERRUPT pushbutton on the 2030 control panel to continue data processing.

4. When the last of the data cards fit into the card reader file-feed, depress the END OF FILE pushbutton on card reader. This action causes the END OF FILE indicator light to come on and the processing continues to completion.

The run sheet for this program will indicate the address of the I/O unit from which the instructions are to be read. Under the entry item regarding LOAD instructions the sheet will read "OOC LOAD." In the case of the card reader, "OOC" is the character code identification address.

The INTERRUPT button directs the computer to read further instructions from the I/O unit which in this case is the card reader. Since the program is also on cards and follows the control cards in sequence the INTERRUPT signal causes the program to be read into the 2030 core storage and executed. The program execution is the instruction set that tells the computer what operations to perform on the input data and what to print on the output reports. This process continues until all the input data has been handled.

Upon the culmination of the processing a number of mop up operations are required. The first item is to remove the program and data cards from the card reader stacker and place them in the master card-storage facility. The paper forms should then be removed from the line printer either by the procedure described in Chapter 2 or by tearing off the paper along the perforation at the end of the last form used in the printing process. If other forms are to be used in a subsequent job run, the blank paper forms should be removed according to the procedure detailed in Chapter 2.

Although we have discussed two types of program that can be run on the System/360 Model 30 it should be emphasized that this computer system has been designed to be able to process a large number of different types of programs automatically. These programs are handled sequentially and executed immediately, one after the other. As a result the operator will not have to operate the 2030 control panel LOAD pushbutton to load in each program in the sequence. A control program which is loaded into the computer via the card reader prior to the series of job runs will automatically instruct the system where to go to find and load each of the jobs in succession.

It is important at this point to describe the operating procedures used in running the accounts payable system. In order to insure that the alignment of check forms is correct the program has in it an instruction which causes the printing of a dummy check each time the 2030 control panel INTERRUPT pushbutton is depressed. This allows the operator to print as many dummy checks as necessary to achieve proper check form alignment. Several other new functions are required of the operator in processing this particular job. The first function is the specifying of the process by which the operator can communicate with the program. Three basic means are available for this purpose:

a. Via job control cards.
b. Via the console typewriter (1052 keyboard-printer).
c. Via the 2030 control panel.

The reason for the operator communicating with the program is that he can as a result alter the computer's course of action. Such action may be necessary in the course of normal system operation as well as for the correction of potential errors or malfunctions. Data can be fed into the card reader or entered into the 2030 control panel via rotary switches, A, B, C, D, H, and J (see Figure 2-20). For the accounts payable job the purpose behind the operator communicating with the program is to instruct the program to stop printing dummy checks and start printing valid ones.

Another function required of the operator is keeping track of both the *location* of the data to be stored and the data itself. The reason for this is that the data can come in different lengths and very large quantities of data can be stored in the main core memory. The data locations are referred to as *data addresses* and are assigned by a code. The result is that the operator in order to enter data on instructions into the computer must specify what is to be stored and the address where it is to be stored. Operator–program communications for entering data via the 2030 control panel is a manual data storage technique. The data address is indicated by the use of rotary switches A, B, C, and D and the data itself is specified by rotary switches H and J.

In order to enter data into the computer the operator must recall from Chapters 1 and 2 that the smallest information unit in System/360 is the byte and therefore he must know how to load a byte of data into a particular core location. The byte, as explained previously, is composed of eight (8) bits. In addition, a ninth bit is included for error checking purposes and is called the *parity* bit. It is not our intention at this time to go into various number systems including the System/360 hexadecimal number system. This topic will be covered in depth in Chapter 4 when we introduce the Disk Operating System. For the purposes of this program it will suffice to state that one byte is required to specify the data and two bytes for the address of the data. The procedure for manual data storage is as follows:

1. Depress the STOP pushbutton on the 2030 control panel to place the CPU in the manual mode, and set the DISPLAY STOR SEL rotary switch to MS (MS = main storage, AS = auxiliary storage, etc.).
2. Set the MAIN STORAGE ADDRESS rotary switches A thru D to the proper two-byte storage address.
3. Set DATA rotary switches H and J to the byte to be stored.
4. Depress the STORE pushbutton and check the MAIN STORAGE ADDRESS REGISTER and MAIN STORAGE DATA REGISTER displays to verify that the data and data address have been properly stored.

It should be noted that manual communication with programs can only be undertaken after the program itself has been loaded in the CPU main core. If we consider that this second job immediately follows the first job then it is not necessary to load this program since the control program has already been loaded. Having readied all the appropriate I/O units the operating procedure is as follows:

1. Depress 2030 control panel INTERRUPT pushbutton.
2. The control program orders control cards to be read from the card reader. The control cards indicate the I/O unit which contains the program (in this case the tape unit) and the program is automatically read into the CPU.
3. At this point we will assume that our program has been written so that the CPU goes into the wait state (WAIT light comes on).
4. Depress INTERRUPT button again to cause printing of dummy check. CPU then goes back into wait state.
5. If check form alignment is proper, manually store the designated data in the appropriate byte location using the above procedure to bypass the dummy check instruction and start the running of the accounts payable program. If check form alignment is not satisfactory, follow printer alignment procedure (Chapter 2) and keep printing dummy checks via INTERRUPT button until proper alignment is achieved. Dummy checks should be marked void, not just discarded.
6. Depress the START and INTERRUPT pushbuttons. The program starts printing accounts payable checks and runs to completion. After the final check has been printed, the 1052 keyboard-printer prints out the END OF JOB message, and the program tape rewinds to the load-point marker.

The computer operator also has a number of functions to perform during and after the completion of this program run. While the program is being processed the operator must inspect the printer from time to time to make certain that the stacker is operating properly. The print quality and alignment of the check forms must be monitored and readjusted if necessary. As indicated previously, the console typewriter or keyboard-printer is the main mode of communications between the operator and the program. The operator uses the keyboard to enter information into the CPU and the computer uses the printer to inform the operator of any mistakes, omissions, or other problems that exist with the program or procedure. Messages that are expected from the CPU during a particular program processing are listed in the message sheet portion of the run book and indicate what response is required of the operator. In the program just described two messages are in the run book. They are, NO DATE and END OF JOB. The NO DATE message indicates that the operator hasn't inserted the date card in front of the accounts payable cards and the program doesn't know what date to put in the date box on the checks. This message is printed as soon as the absence of the date card is detected by the card reader. The corrective procedure is to remove all cards from the hopper and effect a nonprocess runout. Then place the date card in front of the deck and reload the entire deck into the hopper. Next, depress the START pushbutton on the card reader and the INTERRUPT pushbutton on the 2030 control panel. The program will now run in the normal mode.

Finally, after the job has been run the operator removes the printed checks from the line printer, removes the card deck from the card reader stackers, and disconnects the tape reel and disk packs, and stores each item in its designated storage facility.

3.4 SUMMARY OF CONSOLE OPERATOR'S ROLE

The material presented in the first three chapters has been designed to acquaint you with the workings of the System/360, the role of the console operator, and the basic operational procedures and practices necessary for running jobs on a Model 30. You should at this point have a strong familiarity with the System/360 terminology and the functions of each of the system components, namely the CPU, I/O devices, main memory and auxiliary storage, and the console typewriter. Chapters 2 and 3 have stressed the mechanical and operational procedures required to work with the Model 30 in the running of program jobs. You have had the opportunity to practice the mechanical manipulations required in readying each of the I/O devices (i.e. card reader-punch, line printer, keyboard printer, tape unit, and disk unit) and the CPU control panel. You have also learned how to incorporate the various components in the running of specific jobs. In this respect you have learned what operator functions are necessary in readying the system to perform a job, what procedures to use to start and maintain data processing, what problems can be anticipated during the system run and what remedies are available, and finally what cleanup functions are to be undertaken after the program runs have been completed. These functions have become familiar to you through practice in setting up and running designated jobs. You have reached the point, therefore, where you understand and have some facility in performing the functions required of a System/360 console operator.

We have discussed the various control functions and instructions that exist in aiding the console operator to do his job. We have also described the running of two relatively basic programs using the control procedures available without resorting to any supplementary operator aids. In general, however, when an installation such as the Model 30 is used the complexity and quantity of the jobs and components involved is such that efficient system performance requires the operator to have at his disposal additional automated aids. Since the disk storage drive has become the most popular form of auxiliary storage and for use as an I/O device, an automated control system has been developed to assist the console operator in running jobs utilizing disks. This control system, part of which sits resident in the CPU main core storage, is referred to as the Disk Operating System (DOS). The remaining chapters of this text, therefore, are devoted to describing the principles of DOS and how the console operator utilizes it in the running of jobs and functions related thereto.

3.5 SAMPLE JOBS FOR PRACTICE IN SETTING UP SYSTEM/360 JOB RUNS

1. *Card Program to Print Sales Report**

The program deck, if it existed, would contain instructions that would tell the computer how to process the information. The program instructs the computer to compute various totals, and to use the line printer to print a *sales report*. The monthly sales report (see Figure 3-1) includes:

– Total sales (in dollars) for all districts and for all salesmen
– Total sales (in dollars) for each geographical district
– Total sales (in dollars) for each salesman
– Quantity of each product (by part number and name) sold in each district

DISTRICT NO NEW ENG	DIODE 1N3611	POST DF30RC	POST DF39BC	LAMP NE-51	TERM XAE-18	CLIP 6005-IN	TOTAL AMOUNT
18 Maine	7368	1686	2161	667	31040	5193	4814.09
1807 Kalen	2186	982	192	200	13624	831	1463.62
1812 Johns	3214	563	1123	165	9016	1962	2214.18
1819 Altra	1968	1841	846	302	8400	2400	1136.29
19 Vermont	4342	2836	2071	960	17631	3672	3791.00
1902 Willi	1479	862	951	240	6419	1800	1923.25
1903 Baine	2863	1974	1120	720	11211	1872	1867.75
20 N Hamps	6513	2046	1091	522	2443	2964	2888.95
2001 Mahon	3614	924	321	522	1029	1402	1899.62
2002 Coope	2899	1122	720	000	1414	1562	0989.33

Figure 3-1. Sample Sales Report Printout

*Reprinted by permission from *360 MOD 30 360 DOS Training Manual.* © 1969 by International Business Machines Corporation.

The data deck, if it existed, would contain information concerning salesmen who work for a particular company including the names of the salesmen, the geographical districts in which they work, how many sales of each product were made by each salesman in a given amount of time, the part numbers for each product sold, and the name and price of each product.

The report is printed on plain printer paper, one line at a time. The computer presents these figures in a form which business managers find quite useful in conferences, especially when it is necessary to determine which are the most valuable salesmen and which districts demand the largest quantity of a particular product. Comparisons with other salesmen and other districts become quite simple.

The operator is given the data deck which is described previously. He will also be given a schedule (Figure 3-2) and a run book. One page from the run book, which we shall refer to as a *run sheet* is shown in Figure 3-3.

SCHEDULE

Date 10 / 14 / 71
month day year

Sequence	Job No.	Job Name or Description	Return to	Operator Initials
1	D-8040	CARD PROGRAM TO PRINT SALES REPORT	W. SHAW	

Figure 3-2. Schedule for Card Program

The explanation of the run sheet for this sample program is quite simple. Examine Figure 3-3, line by line, to be sure that you understand each of the instructions provided for the operator. The run sheet entry contains the following information:

– Job No. D-8040. This entry is the number of the particular batch of work. The

JOB NO.	TITLE			PROGRAMMER
D-8040	CARD PROGRAM TO PRINT SALES REPORT			W. SHAW
TYPE OF RUN	RUN FREQUENCY	TIMER		LOAD INSTRUCTIONS
[X] PROD [ ] TEST	MONTHLY	[ ] ON [X] OFF		OOC, LOAD

IN	OUT	DEVICE TYPE	ASSIGN.	UNIT ADDR.	VOL. NO.	NAME OR DESCRIPTION
X		CARD READER		OOC		CONTROL DECK & PROGRAM NO. 18 - DECK NO. 2018
X		CARD READER		OOC		SALES RECORD DECK DECK - NO 667
	X	PRINTER		OOE		SALES REPORT (USE PLAIN STOCK)

SPECIAL INSTRUCTIONS:
1. Use Carriage Control Tape No. 3.
2. Print positions 1 thru 50 are used.
3. Print 6 lines per inch.
4. Load data deck after Control and Program decks.

Figure 3-3. Run Sheet: Card Program to Print Sales Report

job number is simply one of the means of keeping track of the work done at the installation.

– Card program to Print Sales Report. This title gives the purpose of the program, which was written and coded by programmer W. Shaw.
– Type of Run. Since this is not a test run, but an actual production run, the box marked PROD is checked.
– Run Frequency. This program is run each month to process the sales made during that month.
– Timer. The timer switch, marked INT TMR, is found in the lower left-hand corner of the control panel. (Some programs use the timer to keep track of the intervals

of time required to perform certain operations. This program, however, does not use the timer, so the run book indicates that the switch should be set to OFF.)

– In (Inputs).
  1. The program instructions that tell the computer how to process the data punched in cards. The proper deck of cards is kept on file at the computer installation and is identified as deck number 2018. The control deck consists of a special deck of cards that precedes the program deck.
  2. Sales Record Deck. The sales information is also punched in cards and, like the program deck, is fed into the computer via the card reader.
– Out (Output). The output produced by this program is a sales report printed on plain printer paper.
– Special Instructions. The carriage control tape is a loop of paper tape in which holes are punched to control vertical spacing. This tape is mounted in the printer and controls the area of the forms in which the printing is done.

2. *Tape Programs to Print Payroll Checks and Register Forms**

A payroll check and register operation is performed once a week by most companies to compute the payroll (including all the various deductions) and to print the payroll checks. In addition to printing checks, a second printout is frequently required to obtain a payroll register. Register sheets provide the company with a tabular record of the payroll for accounting purposes. In an installation with one printer, the register printing must be performed as a separate printout. In this example, the results of the payroll computations made when the checks are printed must be placed in some form of temporary storage until the printer is loaded with register forms.

Several sets of inputs to the computer are required. First, there is the payroll program itself. For this example, it is assumed that the program is already on tape. The hours worked by each employee must be entered. These amounts, which vary from week to week, are assumed to have been punched into the *time card* deck (Figure 3-4).

In addition to the number of hours worked, a master employee file is required, which is a permanent record of all employees, their payroll numbers, hourly rates, exemptions, total earnings, and amounts withheld. This file is updated by each payroll program run as personnel are hired or leave the company, when raises are awarded, and when exemptions change. At the end of the year, the totals for each employee are used to produce the W-2 withholding statements. The final item to be entered is the date of the payroll. The date may be punched on a card and loaded along with the time card deck.

Note: An alternate means of entering the date is to use the 1052 Printer-Keyboard (console typewriter). In some situations, the computer causes the console typewriter to type a message to the operator. One of these messages might read ENTER DATE. The operator responds by typing the date, and since the console typewriter is linked electronically to the CPU, the date is transmitted to the computer. (This procedure will

*Reprinted by permission from *360 MOD 30 360 DOS Training Manual.* © 1969 by International Business Machines Corporation.

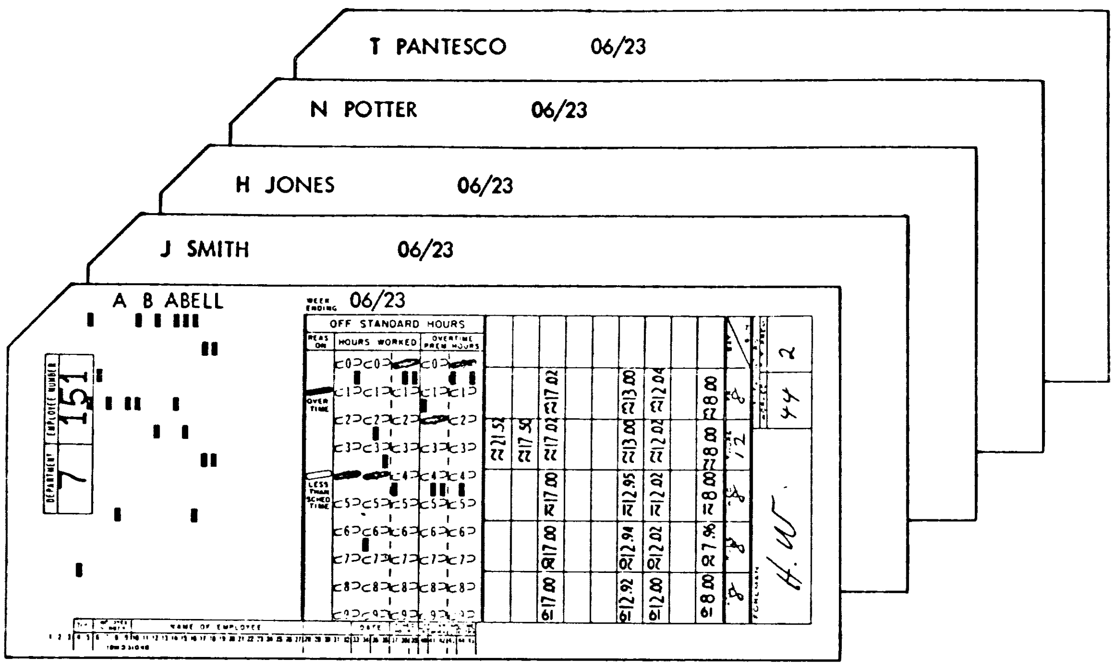


Figure 3-4. Deck of 5 Time Cards (Courtesy of IBM)

be discussed, together with other messages to the operator, in a later portion of this manual.)

Figure 3-5 is a schedule for the payroll check and payroll register programs. A run sheet for the payroll check program is shown in Figure 3-6. The items are a simple program description, a tabular listing of input/output devices and data packages used for this job, and special instructions. This program requires a tape unit and a disk unit, plus the card reader and the line printer. The payroll program is on tape to speed up the operation. Information on tape can be transferred to the CPU much faster than information on punched cards. It is a common practice, therefore, to transfer frequently used programs from cards to tape (or disk) to reduce the time required to load the program into the CPU. Similarly, master files (such as the employee file) are usually maintained on tape or disk. In this program, we have arbitrarily assumed that the master file is on disk and the payroll program is on tape.

The program will store the calculated payroll information on a work area on disk for use by the next program. A disk work area is an area on the disk that is available for temporary storage. As each check is calculated and printed, the results are also stored in this area. When the run is finished, the check forms must be removed from the printer and replaced with plain stock. Then the next program, which prints the register sheets, is run. Since the results of the payroll program are in the work area, no

SCHEDULE

Date 10 / 14 / 71
month day year

Sequence	Job No.	Job Name or Description	Return to	Operator Initials
1	MP 809	PRINT PAYROLL CHECKS	N. P.	S. F.
2	MP 810	PRINT PAYROLL REGISTER	"	S. F.

Figure 3-5. Schedule Showing Payroll Job

additional calculations are needed to print the register sheets. The register program merely transfers the data from the work area to the printer in the correct order for register printing.

Readying a Tape Unit

The run sheet specifies that the payroll program is on volume number 004002 and is to be mounted on tape unit 180. (Refer to the UNIT ADDR column in Figure 3-6.) Volume number 004002 is the identifying number of the program tape *reel* and appears on the label in the center of the reel. Similarly, the number 180 is the number by which the CPU identifies the tape unit. Either the last digit of each number (for example, 0) or the entire number appears on the tape unit itself.

These addresses are necessary because the CPU must be able to determine which device is being used to deliver data into main storage. Every I/O unit in the system is referenced by the CPU with a three-character code. The first character always refers to the channel, while the second and third characters identify the specific unit on that channel. Channels are small processors that respond to their own commands and direct the flow of data back and forth between the I/O devices and main storage. A system can be equipped with two types of channels: *selector* and *multiplexor.*

A *selector* channel is used when high-speed I/O devices are attached to the system.

JOB NO.	TITLE	PROGRAMMER
MP809	PRINT PAYROLL CHECKS	E. KRONG

TYPE OF RUN	RUN FREQUENCY	TIMER	LOAD INSTRUCTIONS
☒ PROD ☐ TEST	WEEKLY	☐ ON ☒ OFF	INTERRUPT

IN	OUT	DEVICE TYPE	ASSIGN.	UNIT ADDR.	VOL. NO.	NAME OR DESCRIPTION
✓		TAPE		180	004002	PRINT PAYROLL CHECKS PROGRAM
✓		DISK		190	004008	EMPLOYEE FILE
✓		RDR		00C		TIME CARD DECK – PREFIX WITH DATE & CONTROL CARDS
	✓	PTR		00E		PAYROLL CHECKS
	✓	DISK		190	004008	DISK WORK AREA FOR REGISTER DATA

SPECIAL INSTRUCTIONS:

1. PRINT CHECKS 8 LINES/INCH; USE CARRIAGE CONTROL TAPE #6
2. NAME AT PRINT POSITION 8.
3. STORE OF AT MAIN STORAGE BYTE 01FC TO STOP DUMMY CHECKS AND RUN PAYROLL.
4. PRESS START ONCE FOR EACH DUMMY CHECK NEEDED.
5. NOTE ERROR MESSAGES.

Figure 3-6. Run Sheet: Payroll Check Program

For this reason, tape and disk units are usually attached to selector channels. A *multiplexor* channel is usually used to permit simultaneous operation of several (as many as 248) low-speed I/O devices. I/O units specially suited for multiplexor use are card readers, card punches, and printers.

The System/360 Model 30 may have a maximum of two selector channels and one multiplexor channel. I/O device addresses specify which channel is to be used for a particular device by the first character in the code:

0 = multiplexor channel
1 = selector channel 1
2 = selector channel 2

The next character of the device address is conventionally assigned as follows:

0 = card reader, card punch, printer
1 = typewriter
2 = tape unit
3 = disk unit

The third character of the code specifies the individual device. Card readers, punches, and printers are identified with alphabetic characters. Specific tape and disk units are identified with numeric characters. A set of addresses for a typical installation is listed in Figure 3-7.

Device	Address
Tapes (2401), Selector Channel 1	180, 181, 182, etc.
Tapes (2401), Selector Channel 2	280, 281, 282, etc.
Card Read Punch (1442)	00A
Printer (1443)	00B
Card Reader (2540)	00C
Card Punch (2540)	00D
Printer (1403)	00E
Typewriter-Keyboard (1052)	01F
Disk Storage Drives (2311), Selector Channel 1	190, 191, 192, etc.
Disk Storage Drives (2311), Selector Channel 2	290, 291, 292, etc.

Figure 3-7. Typical Device Addresses

Running the Payroll Program

1. Load program into computer by loading 00C.
2. Depress the INTERRUPT pushbutton. This action starts the program, which causes the printer to print one dummy check. At this time the program halts and the CPU goes back into the wait state.
3. Check the alignment of the printing (which may consist of all X's or the word VOID) on the first dummy check. Use the lateral print vernier and the VERTICAL PRINT ADJUSTMENT control to improve the alignment, if necessary.
4. If you are still not sure of the alignment, go back to the console control panel and

depress the INTERRUPT pushbutton again to cause another dummy check to be printed. Readjust the printer alignment if it is not right. This procedure can be repeated until the alignment is satisfactory.

5. When the alignment process is completed, manually store 0F in byte location 01FC of main storage to bypass the dummy check instruction and go on with running the payroll program. To do this, first depress the STOP pushbutton and check that the MAN indicator lights and the ALLOW WRITE display is extinguished. Set the rotary switch labeled DISPLAY STOR SEL to MS (for main storage). Then, set rotary switches A, B, C, and D of 01FC (the data address) and rotary switches H and J to 0F (the data to be stored). Depress the STORE pushbutton and check the MAIN STORAGE ADDRESS REGISTER and MAIN STORAGE DATA REGISTER displays to ensure that the bytes were stored properly. Note the hexadecimal-to-binary conversion:

0 = 0000 1 = 0001 F = 1111 C = 1100	Address
0 = 0000 F = 1111	Data

6. Depress the START and INTERRUPT pushbuttons. The program begins printing payroll checks and runs to completion. After the last check is printed, the console typewriter prints out the message END OF JOB, and the program tape rewinds to the load-point marker.

While the program is running there are certain things to monitor. Go behind the printer periodically and make sure that the stacker is operating properly. Check the print quality and alignment on the checks as they enter the stacker.

Running the Next Program to Print Register Sheets

Termination of the previous job requires the completion of a few cleanup procedures. These procedures involve removing the finished product (checks) from the printer, removing any cards from the card read-punch, and removing tapes and disk packs that are no longer required. Each item must be stored in its proper place so that it can be found when needed. The checks and the time card deck (minus the date card) are sent to the accounting department or, in the case of an independent computer installation, back to the customer.

Before dismounting tapes and disk packs or removing stock from the printer, it is advisable to check the run sheet for the next job. If some of the same items are required for the next program, they can be left in place. The run sheet for the print payroll register job (Figure 3-8) shows that the employee file is also required for this program and that this program is on volume 004002, as was the previous program. The paper in the printer must be changed to register forms. From the run sheet, the setup

JOB NO.	TITLE	PROGRAMMER
MP810	PRINT PAYROLL REGISTER	G. SMITH

TYPE OF RUN	RUN FREQUENCY	TIMER	LOAD INSTRUCTIONS
☒ PROD ☐ TEST	WEEKLY	☐ ON ☒ OFF	INTERRUPT

IN	OUT	DEVICE TYPE	ASSIGN.	UNIT ADDR.	VOL. NO.	NAME OR DESCRIPTION
✓		CARD READER		00C		CONTROL CARDS FOR PAYROLL REGISTER PROGRAM
✓		TAPE		180	004002	PAYROLL REGISTER PROGRAM
✓		DISK		190	004008	EMPLOYEE FILE
	✓	PRINTER		00E		REGISTER FORMS
✓		DISK		190	004008	DISK WORK AREA CONTAINING REGISTER DATA

SPECIAL INSTRUCTIONS:

PRINT 6 LINES PER INCH, USE CARRIAGE CONTROL TAPE #3.

Figure 3-8. Run Sheet: Register Program

procedures should be clear since all the items have been covered in previous examples. A brief summary is given to bring out the salient points.

Running the Register Program

1. Remove the check forms from the printer and mount register forms. Check the run sheet for the number of lines per inch and proper carriage control tape. Ready the printer.
2. The payroll register program is also on tape volume 004002. Therefore that tape can remain mounted on tape unit 180.
3. Place the control cards for the register program in the card reader. Ready the card reader.
4. Depress the INTERRUPT pushbutton on the console control panel. The control cards are read and the payroll register program is automatically loaded from tape. The program runs to completion and prints the register sheets.

While the program is running, observe the same precautions as for the payroll program. Watch the printer to ensure that the forms are stacking properly and that the print quality remains good. Note that a message may be printed on the typewriter indicating that the proper payroll data is not on the disk. This could happen if another program were run between the payroll and register programs.

The control program is able to check whether the proper disk is mounted by examining predetermined locations on the disk to see what type of information is stored. If an error message is printed on the typewriter indicating that the data is not on the disk, the data must be restored by rerunning the payroll check program.

4

INTRODUCTION TO SYSTEM/360 DISK OPERATING SYSTEM

In the previous chapters we have learned the mechanical aspects of computer operation and the techniques and procedures in running simple jobs. We recognize, however, that in most installations using computers as large as System/360 many different jobs must be run in short periods of time. These jobs involve the use of different I/O units and in some cases more than one unit of the same type is required. In addition, these different jobs should be run one after the other in order to save computer time and eliminate certain operator functions between jobs (setup time). This method of loading many different jobs one after the other for automated running and processing is called a job stream. As a result of the increased demand of jobs and the system and operator efficiencies effected, IBM has developed a specialized set of programs known as operating systems. These operating systems contain both control programs and processing programs and are generally stored on tape or disk. The control programs are designed for the handling of I/O operations detection, and handling of special I/O conditions, loading of programs, operator communication procedures, and job control. The routines that permit usage of these control functions are, as an entity, referred to as the SUPERVISOR. Processing programs include compilers; librarian programs for cataloging and placing user programs into permanent storage; utility programs for updating of data files, etc. with regard to user programs; sort/merge programs for rearranging data files into a desired sequence; and a linkage editor program which processes the compiler output and produces executable program portions.

There are a number of different operation systems with varying capabilities for use with System/360. The System previously was designed to work with a Tape Operating System (TOS) which was capable of controlling configurations using punched card and magnetic tape I/O units. With the advent of disk storage drives as storage and I/O devices the Tape Operating System became inadequate and a new operating system was developed. This system is called the Disk Operating System (DOS) and is the main operating system now used on the System/360. The remainder of this chapter describes the software structure of System/360 and the fundamentals and applications of DOS to such an installation.

4.1 NUMBERING SYSTEMS

As explained earlier, the System/360 employs the *byte* as the basic information block and the *byte* consists of eight bits. A ninth bit is added for internal error control and is referred to as the parity bit. The computer is designed to be able only to respond to bits (binary digits), and therefore all instructions must be translated to the binary number system before the computer can execute them. It is impractical and error producing, however, for both programmers and computer operators to employ binary notation in communicating with the system. As a result, a different number system is used in System/360 which is easier to write and reduces the chance of errors. This number system is referred to as HEXADECIMAL and we will become very familiar with it in the next few paragraphs.

4.1a Bases and Exponents–Decimal

All number systems have a base or radix, an absolute value, and an exponent or a power. To illustrate this let us use the number system we are most familiar with, notably the decimal system. In this system the base is 10, the absolute values are the digits 0 to 9, and the exponent values range from $-\infty$ to $+\infty$. For example, the number 9 can be written as 9×10^0, 90 as 9×10^1, 900 as 9×10^2, 0.9 as 9×10^{-1}, 0.09 as 9×10^{-2}, etc. Negative exponents result in fractions and the absolute value of the exponent determines how many places from the decimal point the absolute value digit is to be located. A number with an exponent of zero (0) is always equal to 1. A final example will serve to show the entire structure of the decimal system. The number 101.59 can be rewritten as:

$$
\begin{aligned}
1 \times 10^2 &= 100.00 \\
0 \times 10^1 &= 0.00 \\
1 \times 10^0 &= 1.00 \\
5 \times 10^{-1} &= 0.50 \\
9 \times 10^{-2} &= \underline{0.09} \\
\text{SUM TOTAL} &= 101.59
\end{aligned}
$$

4.1b Binary

The computer, as we have stated before, responds only to binary numbers. The most important number system to learn, therefore, in computer operation is the binary system. In this system the base is 2, the absolute values are 0 and 1, and the exponent range as before is from $-\infty$ to $+\infty$. As an example let us choose the decimal number 90. In binary we have

$$
\begin{aligned}
1 \times 2^6 &= 64 \\
0 \times 2^5 &= 0 \\
1 \times 2^4 &= 16 \\
1 \times 2^3 &= 8 \\
0 \times 2^2 &= 0 \\
1 \times 2^1 &= 2 \\
0 \times 2^0 &= \underline{0} \\
\text{SUM TOTAL} &= 90
\end{aligned}
$$

and the binary number is 1011010. Negative exponents have the same effect as in the decimal system, producing fractions. Consider the number 65.25. In binary we have

$$
\begin{aligned}
1 \times 2^6 &= 64.00 \\
0 \times 2^5 &= 00.00 \\
0 \times 2^4 &= 00.00 \\
0 \times 2^3 &= 00.00 \\
0 \times 2^2 &= 00.00 \\
0 \times 2^1 &= 00.00 \\
1 \times 2^0 &= 01.00 \\
0 \times 2^{-1} &= 00.00 \\
1 \times 2^{-2} &= \underline{00.25} \\
\text{SUM TOTAL} &= 65.25
\end{aligned}
$$

which is written as 1000001.01. In converting from the decimal system to the binary system it must be remembered that certain decimal fractions cannot be exactly reproduced in binary and that the approximation is dependent on the degree of accuracy required. For example, the number 1.38 is approximated in binary as 1.011 which actually equals 1.375.

4.1c Hexadecimal

The hexadecimal system has a base or radix of 16 and an absolute value set composed of numbers 0 to 9 and letters A to F. This system uses numbers from 1 to 15 to represent absolute values and, because only one character is used to represent each value, a group of letters is added to the number set. Because the base of the hexadecimal system is 16 and that of the binary system only 2, notations of hexadecimal value in binary require the use of 4 binary numbers or a 4-bit code.

The following list compares all three number systems:

DECIMAL	HEXADECIMAL	BINARY
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Converting from one number system to the other employs the straightforward approach used in the binary to decimal conversion and vice versa. For instance, the decimal number 201 is converted to hexadecimal as

$$\begin{aligned} (C = 12) \times 16^1 &= 192 \\ 9 \times 16^0 &= \underline{9} \\ \text{SUM TOTAL} &= 201 \end{aligned}$$

and written as C9. In binary notation C9 is 11001001. Conversion of decimal numbers with fractions creates the same difficulty in hexadecimal as binary and the approximation technique must be used. The IBM System/360 Reference Data Card (green reference card), X20-1703, contains conversion tables for binary and hexadecimal.

Since the *byte* is composed of 8 bits (parity bit not counted) and 4 bits are required to define a hexadecimal character, then 2 hexadecimal characters represent a byte, 8 hexadecimal characters a *word* (4 bytes).

4.1d BCD

Another number system which is used in System/360 for data representation is the BINARY-CODED DECIMAL system. In this system a set of 4 binary digits is used to represent a decimal system character. For example, the decimal number 385 is written in binary-coded decimal as:

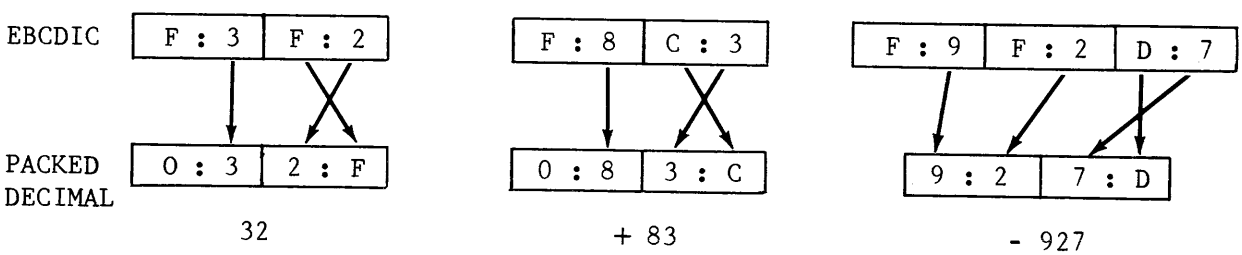
decimal	3	8	5
binary-coded decimal	0011	1000	0101

4.1e EBCDIC

Within core storage Extended Binary Coded Decimal Interchange Code (EBCDIC) is utilized. One character of data is represented on one byte. In this code, Hollerith zoned data is represented by the hexadecimal digits C through F, and digit data by BCD. C equals a zero zone, D an eleven zone and E a twelve zone. A hexadecimal F indicates no zoned data. When working with EBCDIC numerics, C in the low-order byte zone represents a positive number, and a D in the same position a negative number. Thus:

4.1f Packed Decimal

This form must be used for V.F.L. arithmetic and edit operations, and compresses the data by removing all non-essential F zones. This is achieved as follows:



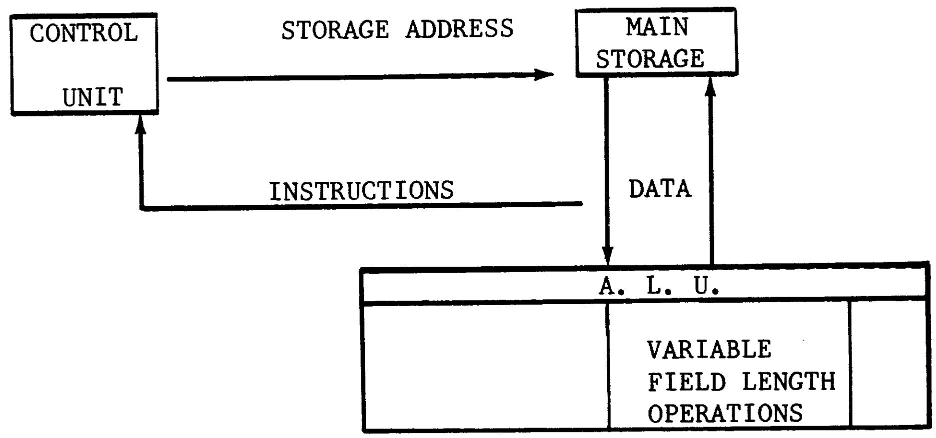
4.2 C.P.U. ORGANIZATION

Two main uses are made of computers—commercial and scientific. Historically, commercial applications were "I/O bound," and had a high editing requirement. Scientific applications on the other hand, were "process bound," and had little editing requirement. Today however, this picture is changing, and a scientific program might well require much I/O and editing, as well as high processing speeds. Similarly, commercial computing has developed to a state where it too requires high processing speed. System/360 was designed with these factors in mind. Three (3) instruction sets exist for System/360. These are the commercial set, the scientific set and the universal set. The commercial set is composed of the standard instructions and the decimal feature; the scientific set of the standard instructions and the floating point feature; and the universal set of all three classes of instructions.

The decimal feature supports variable field length (V.F.L.) arithmetic and edit operations, the standard instructions support fixed point operations and the balance of the V.F.L. operations, while the floating point feature allows floating point operations to proceed.

4.2a Variable Field Length Operations

System/360 uses the storage-to-storage concept for most of its V.F.L. operations.



In this concept both the data fields involved in the operation are found in main storage, and the instruction is said to have an SS format (storage-to-storage). In order for a field of variable length to be specified, both the address and the length of the field must be given. The field is referenced by the address of the first (high order) byte, and the length code (which is always in binary) is expressed by a number which is one less than the total number of bytes in the field. Thus a length code of 0101 would specify a variable field length of 6 bytes. Length codes may be of either 4 or 8 bits in length, dependent on the instruction. Therefore, a length code of 11111111 would represent a field 256 bytes long, the maximum obtainable.

Three forms of instruction exist:

a) *Actual (or machine language)*
This form is that which is the product of the language translator program, and as such is stored within the computer.

b) *Symbolic*
This form serves to specifically identify a field via a symbolic term. These symbols are converted to addresses and length codes by the language translator program.

c) *Explicit*
This form is explicitly encoded by the programmer in terms of the address and field length.

In System/360, addressing is done on a base-displacement principle. If we knew how far away a field was from a reference point, and we knew the location of this reference point, we would know the address of the field. The location of this reference point (base address), is in one of 16 general registers, now termed a base register. For example, if the contents of general register 14 is 2,600 and the displacement is 960, the address of the first byte of the field would be 2,600 + 960 = 3,560. The number of bits available for the displacement code is 12; therefore, the highest number of bytes displaced can only be 4,095, ($2^{12} - 1$).

The advantage of base-displacement addressing is that a very large address can be coded in a short form. For example, 16,777,216 bytes of main storage may be accessed by System/360. This is equivalent to 24 bits in straight binary, which compares most unfavorably with the 16 bits needed for base-displacement addressing. Four (4) bytes are used to specify the base register (numbered 0-15, $2^4 - 1$) and 12 bits for the displacement, ($2^{12} - 1$). However, general register number 0 may not be used as a base register, and it is inadvisable to use registers 1, 13, 14 and 15, as they are used for I/O operations under DOS.

The storage-to-storage instruction may have one additional form, termed the SI format. This is when one byte is used for initializing, testing switches, inserting code characters or special symbols in a field, etc. and is referred to as Immediate data. These bytes do not have to be addressed, always form the second operand of an instruction and are always explicitly expressed.

It is a feature of variable length fields that they operate on data from the second operand to the first operand. Thus in the following SS format instruction:

OPCODE	FIRST OPERAND			SECOND OPERAND		
AP	23 (	5 ,	9) ,	392 (	8 ,	6)
(Add Packed)	Displacement	Length	Base register number	Displacement	Length	Base register number

This instruction means:

> "Add in packed decimal format the information eight bytes in length found at an address which is the sum of the contents of general register 6 and the displacement 392, to the information 5 bytes long found at the address given by the sum of the contents of general register 9 and a displacement of 23, and store the results at this last address."

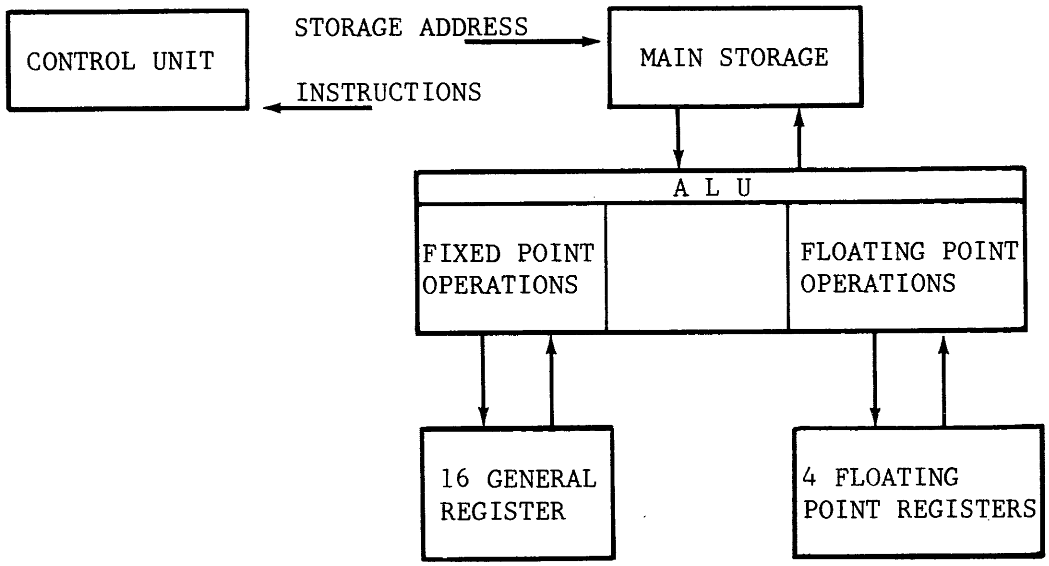
4.2b Fixed Field Length Operations

These operations work on a storage-to-register (accumulator) or register-to-register concept and are performed on binary data. This form of data is typified by high speed of operation, and therefore, high-speed arithmetic is its largest application.

As you can see from the illustration on the following page, fixed field length operations are subdivided into fixed point operations and floating point operations.

Fixed Point Operations:

These operations are performed using fixed length fields, which may be 2, 4, or



8 bytes long (half, full or doubleword), depending upon the particular instruction. This data normally consists of signed whole numbers with the high order bit representing the sign (0 = +, 1 = −)

1000 0000 1000 1001

Above is a typical halfword representing minus 137.

As fixed length data may reside in either main storage or general registers, both RS and RR operations may be performed. As the general registers are high-speed storage devices each of one-word storage capacity, a register-to-register operation is the fastest that may be performed. A half-word operand is placed in the low order 16 bits and a doubleword operand (as used in fixed length divide) would fill two adjacent registers. The first register used is always even, and the operand would be addressed by this register.

Below are examples of the RR and RS formats:

RR AR 2,7

The binary data in register 7 is added to that in register 2, and stored in register 2.

RS A 3, TOTAL

The binary data at symbolic address TOTAL in core storage is added to the data at register 3, and stored in register 3.

A further format, that of RX (index storage to register), might be used. In this format, the address of the second operand is given by the sum of the contents of a base register, a displacement, and an INDEX REGISTER. This once again is a general register, and contains the index which applied only to the instruction that references the index. The main use of base-index addressing is where successive fields of a record have to be operated upon. To continually update the base register is not practical, as this data references a large area (4,096 bytes). Therefore, the index is incremented by the required amount, and the sum of the index, base address and displacement gives the required address of the new data. An RX Instruction might read:

A	9	362	(8	4)
I	I	I	I	I
ADD, STORAGE TO REGISTER	REGISTER 9	DISPLACEMENT	INDEX REGISTER	BASE REGISTER

This means "ADD THE DATA FOUND AT THE ADDRESS IN MAIN STORAGE GIVEN BY THE SUM OF THE CONTENTS OF BASE REGISTER 4, INDEX REGISTER 8, AND THE DISPLACEMENT, TO THE CONTENTS OF GENERAL REGISTER 9, AND STORE THE RESULT AT GENERAL REGISTER 9."

In fixed point instructions, data is always operated upon from the SECOND operand to the FIRST, except when

1) Converting a binary number in a register to packed decimal form in storage.
2) Storing data from a register to main storage.

4.2c Floating Point Operations

These operations are most useful for working with very large numbers, where considerable accuracy is required. When "short precision" is required, a full word of 32 bits is used, but when greater precision is required, a "long precision" form of a 64 bit doubleword is used. Four (4) floating point registers, each of a 64 bit (doubleword), are available and are numbered 0, 2, 4 and 6. These are *NOT* the same as general registers 0, 2, 4 and 6.

In binary, the largest number expressible as a full word is approximately 2×10^9. In floating point we can express as high as 7×10^{75}, a massive increase. Basically we derive a floating point number from decimal by converting it to hexadecimal, taking the sign and value of the exponent, and the sign and value of the fraction.

Thus, if a hexadecimal number were −7E3F.21, the sign and value of the exponent would be +4, and the sign and value of the fraction would be −.7E3F21, ($-.7E3F21 \times 16^4 = -7E3F.21$). In both long and short precision floating point, the high order bit is used to represent the sign of the fraction, the next 7 bits to represent the sign and the value of the exponent, and the remaining bits (24 in short precision, 56 in long precision) the value of the fraction. The sign of the fraction is represented by a 0 for minus, and a 1 for plus. To represent the signed exponent, "Excess 64 arithmetic" is used. Under this technique, 64 is added to the value of the exponent. The range in signed exponents available was −64 to +63, but the addition of 64 makes this range from 0 to 127. Thus, no sign is needed, as the value is never negative. The highest number available (127) can be fitted into 7 bits ($2^7 - 1 = 127$). This addition of 64 to the exponent makes the exponent the CHARACTERISTIC.

We see that the 7 bit characteristic is 75; thus we know that the signed exponent was +11, (75 − 64), and our fraction is the hexadecimal +.290E84. Therefore, our number is $+.290E84 \times 16^{11}$.

A SHORT PRECISION FLOATING POINT NUMBER MIGHT LOOK LIKE THIS IN CORE.

+	75	2 9 0 E 8 4
0	100 1011	0010 1001 0000 1110 1000 0100
SIGN OF FRACTION	CHARACTER-ISTIC	VALUE OF FRACTION

However, a computer printout of the above number that we found in core storage would be 4B290E84, as the computer would read only a series of 8 hexadecimal digits.

4.3 PROGRAM EXECUTION

To recapitulate and expand upon our knowledge of the System/360 control program, we note that this program must be in the lower order addresses of main storage at all times while the system is operating. Because of the nature of the functions it performs for the problem program, it is referred to as the supervisor program. The input to the supervisor is in the form of job cards, which tells it how and when to schedule the execution of problem programs. Problem programs involve input, output and processing, and interact with the supervisor during their execution. The supervisor also utilizes a series of "Physical Input/Output Control System" (PIOCS) routines to control and coordinate I/O for each problem program. The supervisor program also provides "Timer" services, by decrementing on a specific time interval basis a full word at storage address 80. When this word equals zero, an external interrupt occurs. We can summarize the supervisor's role by saying that it schedules the execution of problem programs, uses PIOCS routines to control and coordinate I/O for the problem program, handles all "interrupts" and provides time service. (See 4.3a–Program Status.)

4.3a Program Status

The CPU is termed to be in various "Program States" at a given time, and these states indicate the current status of the machine. There are four classes of program state that a CPU may be in, and each class has two alternatives. These are the *STOPPED* or *OPERATING STATE*, *RUNNING* or *WAITING STATE*, *SUPERVISOR* or *PROBLEM STATE*, and *MASKED* or *INTERRUPTIBLE STATE*. Figure 4-1 shows these states, the effect they have on CPU function, an indication of the status, and how this status is arrived at. The illustration on page 102 shows the possible relationship between these program states.

More, however, needs to be said with regard to the masked and interruptible states. When an exceptional condition comes about within the system, an "interrupt" is generated which signals a request for a CPU status change. We know that an I/O

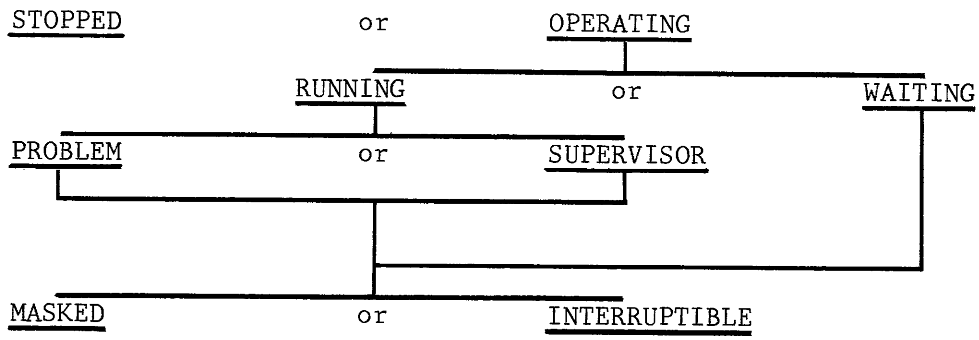


Figure 4-1. Program Status

operation can only be started or serviced when the CPU is in the supervisor state, but when processing data the CPU is in the problem state. Status switching is needed in order to let the I/O operation proceed, and this is brought about by an interrupt which occurs at the end of the processing. Similarly when an I/O operation is finished, an exceptional condition occurs which generates an interrupt to transfer the program status from the supervisor to the problem state. These interrupts are known as "I/O interrupts." An "external interrupt" may be brought about by either an operator action (pressing "interrupt" key), or by an internal timer signal. A "program check interrupt" results from an attempt to overfill a storage or register location, or a result which departs significantly from anticipated bounds. A "machine check interrupt" results from a machine malfunction, such as detected in a parity error. A "supervisor call interrupt" occurs when a supervisor call instruction is executed by the problem program. The CPU, however, is not automatically subject to a status change when an interrupt occurs. If this were the case, an I/O interrupt from another I/O device would be interfering with the supervisor program's execution of a previous I/O interrupt. However, through the machine's use of the program status words (PSW's), we are able to mask interrupts. In nearly all cases, these interrupts are kept pending and not just ignored.

4.3b Program Status Word

A Program Status Word (PSW) is a doubleword identified with a permanent storage location, and contains coded information on the status of the program being executed. The "current PSW" is the one which currently influences the computer, but an interrupt will fetch the appropriate "new PSW" from its permanent storage address and place what is now the "old PSW" in its permanent storage address.

The presence of a "zero" in the mask field indicates a masked condition, a "one" in these fields an unmasked condition. These fields are as follows:

1. The system mask field from bits 0 to 7, contains individual bits to mask the multiplexor channel, selector channels 1 through 6, and the external mask, masking operator and timer interrupts.

PROGRAM STATE	CPU FUNCTIONING	STATUS INDICATION	HOW SWITCHED
STOPPED	Incapable of any function	"Manual" light on console.	"Stop"key on console
OPERATING	Capable of executing instructions and being interrupted	"System" or "Wait" on console	"Start" key on console
RUNNING	Instruction fetching and execution proceed normally.	A zero bit in position of the program status word (PSW)	I/O interrupt. External interrupt.
WAITING	No instruction processing. I/O and external interrupts accepted unless masked. Timer is updated.	A one bit in position 14 of the program status word (PSW) "Wait" light on console	Load PSW Instruction Any interrupt.
SUPERVISOR	All instructions are valid.	A zero bit in position 15 of the PSW	Any interrupt
PROBLEM	All I/O instruction and a group of control instructions are valid.	A one bit in position 15 of the PSW	Load PSW instruction
MASKED	I/O, External and Machine-check interrupts (individually masked) remain pending. Program interrupts are ignored.	Zero bits in the system mask, program mask, and machine-check mask fields of the PSW	Set Program Mask instruction Set System Mask instruction Load PSW instruction Any interrupt.
INTERRUPTIBLE	Interrupts of all unmasked classes accepted.	One bits in the system mask, program mask, and machine-check mask fields of the PSW.	Same as "Masked" above

The Format of the PSW is as follows:

------SYSTEM MASK------

	Mr	S1	S2	S3	S4	S5	S6	X1	PROTECTION KEY	A	M	W	P
BIT #	0	1	2	3	4	5	6	7	8 11	12	13	14	15

	INTERRUPTION CODE	
BIT #	16	31

------PROGRAM MASK------

	ILC	CC	P1	P2	P3	P4	INSTRUCTION ADDRESS
BIT #	32 33	34 35	36	37	38	39	40 47

	INSTRUCTION ADDRESS	
BIT #	48	63

2. The machine mask bit at position 13 in the PSW, masks machine check interrupts.
3. The program mask field in bits 36 through 39 masks program check interrupts. This masking can be applied individually to (sequentially from 36 to 39) fixed point overflow, decimal overflow, exponent underflow and significance.

Bits 8 through 11 contain the "storage protection key" which locks out partitions of core to ensure the instructions and data of programs are not destroyed by overwriting. Protection is provided in increments of sections containing 2K bytes. Bit 12 contains the ASCII (American Standard Code for Information Interchange) mode bit.

Bits 14 and 15 contain the "Wait" and "Problem" state code respectively; "one" in each of these bits indicates "Wait" or "Problem" state, a zero the "Running" or "Supervisor" state.

Bits 16 through 31 contain the interruption code which indicates the reason for changing program sequence.

Bits 32 and 33 contain the instruction length code which specifies (in half words) the length of the data that was found at the last instruction address that was executed. If the length of this instruction is incremented to the instruction address, the address of the next instruction will result, as the instructions are stored sequentially from lower to higher storage addresses.

Bits 34 and 35 contain the condition code, which stores the conditions existing after the execution of the last instruction.

Bits 40 through 63 specify the address of the next instruction to be executed. This instruction will be found somewhere within the supervisor, as it handles all interrupts. Immediately after an interrupt has occurred, the supervisor will take control of the computer.

4.4 CHANNELS

A channel is a device whose functions are to direct and control the flow of data between I/O devices and main storage as directed by instructions from the supervisor program.

There are two forms of channels employed with System/360, "Multiplexer" and "Selector." There may be only one multiplexer channel per CPU, but a variable number of selector channels depending on the CPU model. Multiplexer channels are intended for the relatively slow I/O devices, and selectors for the fast ones. Each type of channel provides a data path between the I/O control units and storage, along which one byte may travel at one time. As the multiplexer caters mainly to slow devices, there is a significant amount of time between the transmission of bytes from such an I/O device. In System/360, this time is utilized by interleaving bytes from other I/O devices. This is known as "Byte Interleaf" mode, or byte mode. If faster devices are used on a multiplexer channel there is insufficient time available for byte mode operation and a "burst mode" is utilized whereby data flows in a contiguous stream from the I/O device. A selector channel operates in burst mode only. From the above it follows that any I/O device can use either a multiplexer or a selector channel, but a multiplexer when operating in burst mode can only service one I/O device.

4.4a Overlapping

Overlapped processing can only be performed by "buffering" data. By overlapping we mean the simultaneous performance of at least two of the three actions of input, processing and output. If all three actions occur together, we have *completely* overlapped processing. The most widely used form of buffering is the so-called "flip-flop" technique, whereby data is read into a finite I/O storage area and processed there. When this input is complete, more data is read into a second area and this is processed when the processing of data contained in the first area is completed. It follows that no two input, processing or output operations can proceed at any one time. Under a different technique, a "work area" may be set aside for processing. This work area must be big enough for the largest logical record within the data stream to be processed. Only one storage area need be set aside, and when processing of the previous data in the work area is completed fresh data is transferred to the work area. This is possible only when a single logical record is processed in an equal or greater time than it takes to read the entire record block into the storage area. These techniques are made possible by the use of channels. The storage cycle (time taken to transfer a byte to or from storage) of the CPU is always less than the storage cycle of I/O devices. Nevertheless, System/360 is organized to use this disparity. I/O operations have priority over CPU operations so that a byte of data is transmitted via the channel to the CPU in preference to a CPU operation, such as the transfer of data from main storage to the ALU. However, as the I/O storage cycle is relatively large, while the I/O device is "gathering" itself for its next data transfer instructions fetching and data transfer within the CPU may proceed. I/O operations can therefore be "overlapped" with processing.

4.4b Channel Program

Each I/O operation is started by the supervisor program placing the address of the first command of a list of commands in the "Channel Address Word," and issuing a "Start I/O" (SIO) instruction that activates a channel and tells it which I/O device to use. The CPU may now be used for other processing. The channel executes the first command (the command found in the first "Channel Command Word" (CCW). The location in storage of the next CCW is found in the first CCW, and the processing of CCW's is continued until the entire channel program is executed. The final CCW contains a "zero" bit in position 33 that tells the channel that this CCW is the last within the channel program. This channel program causes data to be transferred to or from the I/O device either from or to main storage. When the channel program is finished, the channel causes an I/O interrupt which stores the current PSW and loads the new I/O PSW. Information about the completed program is stored in the "Channel Status Word" (CSW) in storage. This new I/O PSW interrogates the old PSW to determine which channel program caused the interrupt, and the CSW to determine whether or not the I/O operation was successful.

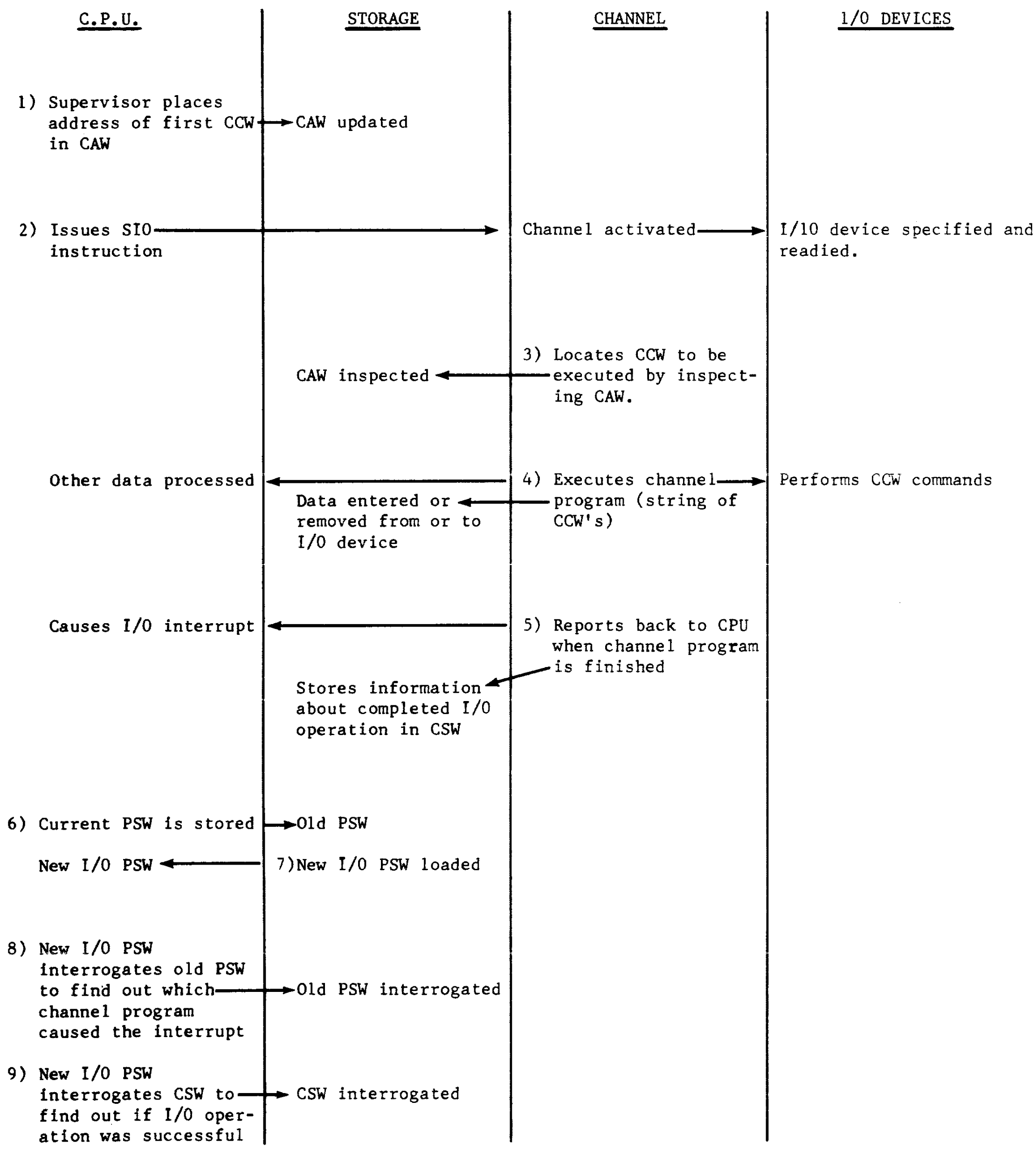
It is important to note that different CCW's are not required to bring about the required operation of differing I/O devices. A card reader will respond to a particular CCW as well as a magnetic tape unit. This is because System/360 uses standard "interfaces" between channels and I/O control units. Channel commands are of several different types. One of these, the "Control Command," allows the channel to direct the I/O units to perform their special functions (such as seeking disk addresses, or selecting a stacker). Control units interpret the control commands and make the I/O device perform its particular order.

4.5 MULTIPROGRAMMING

For systems with a main storage capacity equal to or greater than 24K, the Disk Operating System offers multiprogramming support. This support is referred to as Fixed Partitioned Multiprogramming and the number and size of the partitions is fixed during system generation. The operator, however, may redefine partition sizes to run jobs requiring more storage area.

In a multiprogramming situation, two types of problem programs exist: background and foreground. Foreground programs may operate in either the batched-job mode or

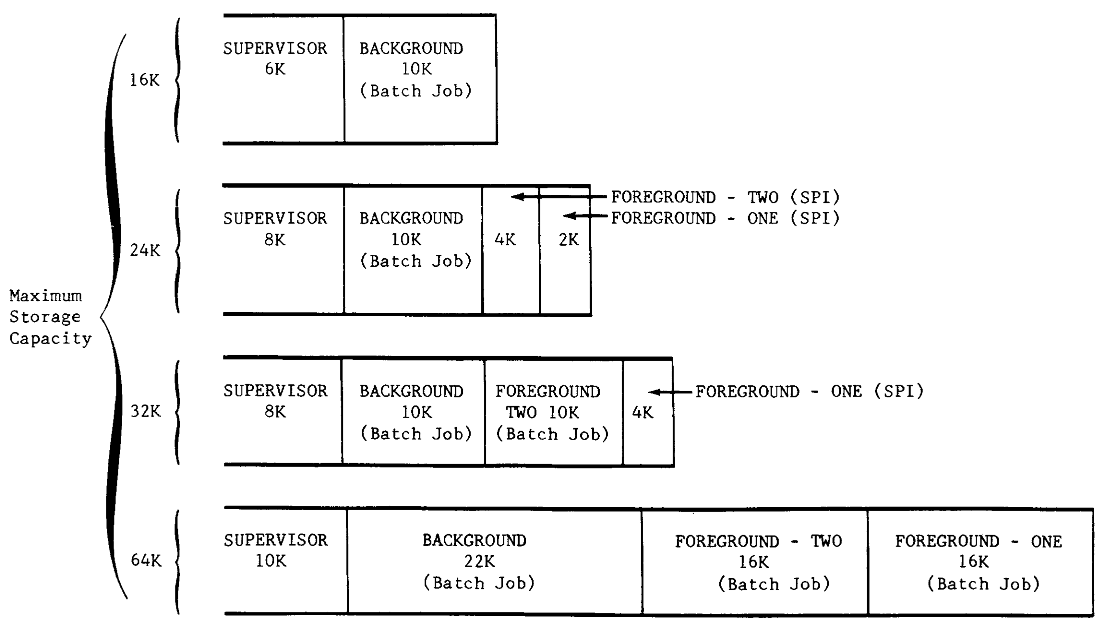
PROCESSING OF CHANNEL PROGRAM



in the single program mode (SPI). Background programs and batched-job foreground programs are initiated by Job Control from the batched-job input streams. Single-program-foreground programs are initiated by the operator from the 1052 printer-keyboard. When one program completes, the operator must explicitly initiate the next program.

A multiprogramming system is capable of concurrently operating one background program and one or two foreground programs. Priority for CPU processing is controlled by the Supervisor, with the foreground programs having priority over background programs. However, control is taken away from a high priority program when that program encounters a condition that prevents continuation of processing until a specified event has occurred. For example, this condition would occur when a WRITE operation is issued to a tape unit. Control is taken away from a low priority program when an event on which a higher priority program was waiting has occurred. In the previous example, control would return to the high priority program when the WRITE operation has been executed. When all programs in the system are simultaneously waiting, the system is placed in the wait state. When an interruption to the Supervisor satisfies a program's wait condition, that program becomes active and competes with other programs for CPU processing time.

The illustration below shows how core storage is organized for various size compu-



Possible Storage Allocation for System/360 with Various Storage Capacities ters. This figure shows that multiprogramming requires at least 24K of storage. The background can never be less than 10K; a computer with 24K could therefore have one foreground of 6K or two foregrounds of 4K and 2K respectively. Batch-jobs could be run in the background partition and SPI programs could be processed in the foreground. A computer with 32K storage could process batch-jobs in two partitions, one in the background and one in the foreground; the remaining foreground area could be used for SPI programs. A 32K computer could also be set up with 14K in the background to enable it to process assembler jobs, which require at least 14K; in doing this however, the computer loses its capability of running two batch-jobs, as the supervisor now requires 10K. The remaining 8K could be used for SPI programs in one or two foreground partitions. Computers with at least 48K can process batch-jobs in all three partitions.

4.5a Initiating Batch Processing in a Foreground Area

As is noted in the preceding section, in order to process batch-jobs in a foreground area the computer must possess core storage of sufficient size to support batch processing in the foreground area(s)—at least 10K in each foreground which is to handle batch-jobs. There are other factors that must be present in order to process batch-jobs in a foreground area. These are:

1. The foreground batch processing option must have been specified when the system was generated.
2. There must be separate input/output devices for the foreground area(s).
3. No job being run may require the use of SYSLNK or the maintenance function of the Librarian.

4.5b Initiating Multiprogramming Operation—Batch Processing in One Foreground

1. Set switches for Load Unit (IPL)
2. Console typewriter prints:

   010A GIVE IPL CONTROL COMMANDS

At this point, the operator should follow the normal IPL procedure, i.e. making changes to the PUB table by using the ADD or DEL commands and then setting the date and time.

3. After setting the date, two messages will be printed:

   01201 DOS IPL COMPLETE

   1100A READY FOR COMMUNICATIONS.

4.5c Initiating Multiprogramming Operation—Batch Processing in Two Foregrounds

At this point, the operator should use the ALLOC command to allocate main storage for the foreground program. Assuming that a 32K machine is being utilized, the Supervisor taking up 8K, the Background 10K, and we are to set up a Foreground containing 10K, the command would be as follows:

ALLOC F2 = 10K, F1 = OK

This command will allocate 10K to Foreground 2 and the remaining 4K will automatically be allocated to the Background area.

In order to begin batch processing in the foreground area, it is necessary to stop the processing in the background first; this is accomplished by the STOP command:

STOP BG

After typing this command the computer will enter a wait state and it will be necessary to press the REQUEST key on the console typewriter in order to input the next command, which will be:

BATCH F2

Processing will now begin in the Foreground 2 partition, by then pressing the REQUEST key and typing:

START BG

The background program will restart processing, the system is now processing batch-jobs in two partitions. The same procedure, with some slight variations, would be followed if three partitions were to process batch-jobs.

4.5d Single Program Initiation

The execution of background programs is handled by the job control program which only operates in the background area. Foreground programs are initiated by the Foreground Initiator, which provides job-control-type functions by reading and interpreting foreground initiation commands. The Foreground Initiator handles such functions as assigning symbolic units and loading the program for execution.

The Foreground Initiator is called into the partition in which it is to be used as a result of issuing a START command after pressing the REQUEST key on the console typewriter. It is possible to enter commands for the initiator from either the console typewriter or a card reader. Control is normally at the console typewriter, but by issuing a READ command control can be given to a card reader. Many installations use a card reader to issue initiator commands to save so much typing into the console typewriter and thus save a great deal of time when running programs that require many commands.

Here follow several examples of initiating SPI programs using various system configurations.

EXAMPLE I

The procedure should be followed to initiate an SPI program at IPL time when one card reader is available and assigned to SYSIN.

1. Place the Job Control cards for the foreground program in the card reader, followed by any batch-job cards.
2. Ready the reader.
3. Perform the IPL procedure with a 1052 as described under STARTING THE SYSTEM (IPL PROCEDURE).
4. Type: ASSGN SYSIN, UAⒷ
   STOP Ⓑ
5. Press the request key and wait for the message:
   AR II60A READY FOR COMMUNICATIONS
6. Type: START F1 Ⓑ
   F2
   READ X 'cuu' Ⓑ
7. Wait for the foreground program to begin processing. This will occur as soon as the EXEC control statement is processed.
8. Press the request key and enter commands:
   START BG Ⓑ
   ASSGN SYSIN, X 'cuu' Ⓑ
   Ⓑ

EXAMPLE II

As in the case in *Example I*, one card reader is assumed to be assigned to SYSIN.

1. Press the request key on the 1052 printer-keyboard and enter the following commands:
   PAUSE Ⓑ
   Ⓑ
2. WAIT for the message:
   BG 1160A READY FOR COMMUNICATIONS
   This message will appear at the completion of the current job step.
3. Run out the cards in the reader and separate the ones that have been processed from those that have not been processed.
4. Place foreground control cards in the reader, followed by the batch-job cards that have not been processed.
5. Perform the steps shown in *Example I*.

EXAMPLE III

This example is similar to *Example II*.

1. Press the request key on the 1052 printer-keyboard and enter:
   START F1 Ⓑ
   F2

2. Type: LISTIO UA Ⓑ
3. Determine which of the card readers is unassigned, and place the foreground control cards in that reader.
4. Type: READ X'cuu' Ⓑ

EXAMPLE IV

This example is for systems that do not have any card readers. All initiation is accomplished by using the 1052 printer-keyboard. If there are a great number of commands necessary, such as several DLBL and EXTENT statements for multiple-file processing, this method of initiation can be very time-consuming. The system throughput may be greatly affected, because system processing can be continued only while the logical transient area is not being used by an active program. From the standpoint of system throughput, foreground initiation using two or more card readers is the most efficient method. Somewhat less desirable is initiation using a single card reader or a 1052 printer-keyboard.

1. Press the request key on the 1052 printer-keyboard and enter the following commands:

   START F1 Ⓑ
   F2

2. Type in programmer request control statements.

4.5e Single Program Termination

An SPI program is terminated under its own control by issuing an EOJ, DUMP, or CANCEL macro instruction, or through operator action, program error, or certain input/output failures. When an SPI program is terminated, the following action is taken:

1. All I/O operations that the program has requested are completed. If telecommunication-device I/O requests are outstanding, they are terminated by the Halt I/O.
2. Tape error statistics (if specified when the system was generated) are typed on the printer-keyboard for tapes used by the program.
3. DASD extents in use by the program for purpose of DASD file protection are dequeued. (DASD file protection is an option that may be selected when the system is generated.)
4. The operator is notified that the program is completed and of the cause of termination, if abnormal. The main storage used by the program remains allocated for the appropriate foreground program area.
5. The program is detached from the system's task selection mechanism.
6. All I/O assignments are reset unless a previous HOLD command was issued for the area(s) terminated.

Following the completion of an SPI program, the operator may initiate another program for the specific area.

Foreground programs operating in batch mode terminate in the same manner as background jobs.

4.6 STRUCTURE OF THE DISK SYSTEM

The Disk Operating System maintains its operating procedures on either the 2311 Disk Storage Drive or on one of the nine disk drives of the 2314 Direct Access Storage Facility. DOS is designed to accommodate punched card equipment, magnetic tape units, three types of direct access storage devices, and terminal devices for data transfer with remote stations over communication networks. The use of terminal devices enables the System/360 to serve as a central processing facility for remote locations which require computer capability with regard to their business operations.

As indicated earlier in this chapter the DOS system consists of a control program and processing programs. The control program has the same three segments, namely the Supervisor, Job Control, and the Initial Program Loader (IPL). The processing programs available with DOS are Language Translators, Service Programs, and User-Written Problem Programs. The Language Translator programs that can be used under DOS are Basic Assembler Language (BAL), COBOL, FORTRAN, PL/1, and RPG. The Service Programs available are linkage editor, librarian, sort/merge, utilities, and auto test.

Most DOS installations will not require all the available routines and as a result each installation will use its System/360 to generate its own DOS System. Each installation, therefore, will have its own DOS system which incorporates the routines necessary to handle the projected workload. The processing programs and control program routines are kept in three separate libraries: the source statement library, the relocatable library, and the core image library. The source statement library houses the compiler information required for compiling source programs. The relocatable library houses the object programs that result from the compilation of the source programs. The core image library is the storage facility for the various system executable programs such as the Librarian, Sort/Merge, Utility, Language Translators, and the Linkage Editor processed user-written program portions. In general, all the library programs are stored on disk packs. The core image library, however, must always be stored on the system disk pack.

DOS is also capable of handling multiprogramming situations. Up to three different programs can be executed simultaneously, although each program must be executed from different core storage areas. In the multiprogramming mode, the DOS system can still operate batch-processing jobs through the use of Job Control. In a DOS multiprogramming system, the organization of core storage is set up in the manner depicted by Figure 4-2. This organization of core storage permits a number of different combinations of the three programs that can be simultaneously executed. Although multiprogramming is a way in which the system can increase its work output per given time period, most Model 30 installations do not incorporate it in their DOS system generation. As a result, additional descriptions of DOS multiprogramming system operation are not included in this text. If the particular installation being used does provide for multiprogramming there are a number of IBM documents that describe the capabilities and operating procedures in this environment.

The minimum DOS system configuration for a Model 30 installation includes the following devices:

Supervisor
Background Program Area can contain such programs as: Job Control Linkage Editor User-written Problem Programs
Foreground-two Program Area can contain such programs as: Job Control Single Program Initiator User-written Problem Programs
Foreground-one Program Area can contain such programs as: Job Control Single Program Initiator User-written Problem Programs

Figure 4-2. Core Storage Organization in a DOS Multiprogramming System

Three	2401	Magnetic Tape Units
One	2311	Disk Storage Drive
One	2540	Card Read-Punch
One	1403	Printer Model N1
One	1052	Printer-Keyboard Model 8
One	2030	Central Processing Unit

In general, however, at least three 2311 Disk Drives are used in a Model 30 System Configuration.

4.7 JOB ENTRY

We have previously discussed the control program as one portion of the DOS system. We have also described the three routines, namely the *Supervisor*, *Job Control*, and *Initial Program Loading* (IPL) that make up the control program. These three routines are used by the operator in entering jobs into the computer. The first function performed by the operator at the start of each day of system operation is to initiate the IPL procedure. This procedure loads the IPL program and part of the Supervisor into core storage. This portion of the Supervisor program remains in core for the entire day's operation. Other portions of the Supervisor are loaded into core if and when required for job execution. At the culmination of the IPL procedure the Job Control Program routine is loaded into core. Job Control is a special program which is designed to provide for proper sequencing of job step to job step and job-to-job transitions in a job stream. "Job step" is a term referring to an individual program in a job which is composed of a number of programs.

Job Control is capable of reading and analyzing the special control cards that accompany the job stream. After the first job has been started Job Control is subsequently loaded into core at the culmination of each job step and each compile job. When Job Control is loaded into core during a job it analyzes the control cards in the job stream to determine the next job step to be executed in the job. It then provides for the transition to the next job step by accessing the system tape or disk pack and reading the appropriate program into core. When the job is terminated, Job Control again is loaded into core, analyzes the job stream control cards, and provides for the transition to the next job. The sequence of the individual programs (job steps) in a job, and the sequence of jobs in a job stream are specified by the control cards. The programmer is responsible for preparing and submitting these specifications to the operator prior to system running. The job specification statements are keypunched on the control cards and instruct the system as to start of job, end of job, and job step descriptions. The basic job control statements required in every job stream are:

```
// JOB   jobname  –Defines start of a job
// EXEC xxxxxxxx–Defines a job step and specifies the program to be executed
/&                –Defines end of a job
```

The job step definition instruction indicates that the operand may be blank (which states the Linkage Editor processed program is to be executed), or may name the program to be executed (e.g. // EXEC DSORT, // EXEC FORTRAN, etc.)

Figure 4-3 illustrates the basic statements in a typical job stream requiring the execution of two jobs. The first job (JOB1) contains two job steps which in sequence call for the execution of PROGRAMA followed by the execution of PROGRAMB. The second job (JOB2) has only one job step which involves the execution of the Disk Sort/Merge program DSORT. If input data for the various job steps were available in card form it would follow the //EXEC xxxxxxxx cards.

The symbolic language that is used in the job control statements is commonly referred to as *job control language* (JCL). In reality, it is no more than a set of special mnemonics that are compatible with the assembler language. The Job Control program

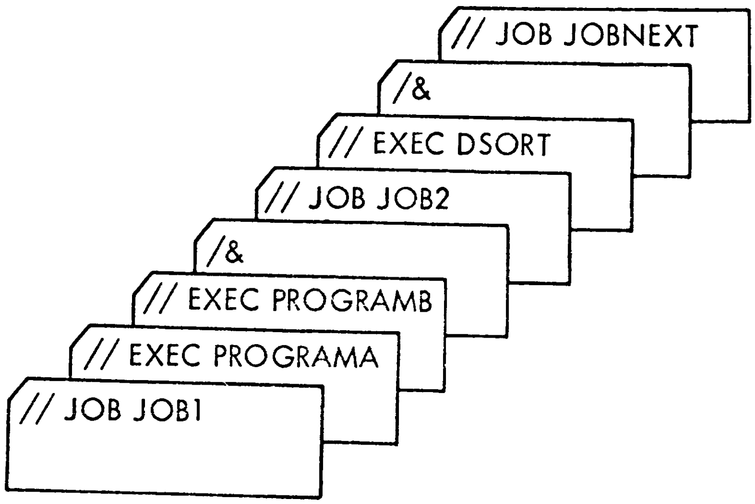


Figure 4-3. A Job Stream (Courtesy of IBM)

is thus a special program which is capable of interpreting the job control statement language and following the procedures called for by these control cards. A more detailed description of the use of JCL in DOS operation is given in the latter part of this chapter and in Chapter 6.

4.8 COMMUNICATING WITH DOS

A major feature of the DOS system is that the control program permits communication between the operator and the system. In this manner modifications can be introduced to the system during operation to insure proper job runs. There are two means of communicating with DOS; the first is between the system and the operator, and the second is between the system and the programmer. In the first mode the system can *talk* to the operator via the sending of messages which are printed on the 1052 printer-keyboard. The operator can issue commands to the system by typing instructions on the 1052 keyboard.

The second type of communication results from the programmer issuing instructions to the system via the use of job control statements and the system *talking* to the programmer mainly by the issuing of messages on the line printer. Sometimes the 1052 keyboard-printer is used for producing system messages to provide information to the programmer.

Although the good operator should be aware of the type of information transfer between the system and the programmer he is mainly concerned with the functions required of him in communicating with the DOS system. It is important therefore to describe some of the more significant reasons for communicating with the system.

a. The computer determines that a particular I/O device is not operating properly and wishes to call this fact to the attention of the operator so that he can correct the situation. The computer message to the operator in this case appears on the 1052 printer. For example, to indicate that the card reader is not in the ready state the computer message is

0P08A INTERV REQ SYSRDR = 00C

The computer operator can then go to the card reader and determine what the cause of the problem is and take the necessary action (add more cards, alleviate a card jam, depress the START button, etc.) to bring the device to the ready condition.

b. The computer has determined that a program error has caused the job to be terminated. The message to the operator is

0S00I JOB XXXXXXXX CANCELLED

and the name of the job cancelled is given by the X's.

c. The operator wants to effect a pause in the processing operation. The body of the command issued to the system is PAUSE. When whatever other functions he wanted accomplished are completed the operator then issues the appropriate command to restart the processing.

d. The operator has encountered some condition in the system that causes him to want to terminate a job before the run is completed. The body of the command typed on the 1052 keyboard is

CANCEL.

e. The operator has discovered the unavailability of a given device in the system. The body of the command issued to the computer informing it of this fact is DVCDN X 'cuu' where cuu is the address of the device in question.

There are numerous other messages and commands in the system-operator communications mode. In the case of messages from the system the operator becomes familiar with the basic ones and has an IBM manual for looking up the meaning and required response for other messages. The IBM document in question is *IBM System/360 Disk Operating System: Operating Guide (C24-5022).* Each message statement has an alphanumeric code preceding the message. This code is the basis for using the manual in determining the cause of the message and the corrective action required. The first and last positions of the code are of particular significance to the operator. The first position identifies the component responsible for the message (e.g., 0 indicates that the component is the Supervisor). The end position identifies the response required of the operator (e.g., A indicates that some specific manual action is necessary, D requires the operator to decide between alternative courses of action, and I states that the message is for information purposes only and that no operator action is required).

4.9 ALLOCATING I/O DEVICES

There are many different system configurations possible in a Model 30 installation. In order for the system to function properly it must know which devices are to be used as I/O devices, which ones are to be used for work purposes such as auxiliary storage, and which device will house the DOS system pack. Every device in a Model 30 configuration has an actual address associated with it and the computer operator must know or have a list of the addresses of every unit in the system. As a result the majority of installations either post a list of device addresses or place sticker tags on all units with the identification label.

It is generally impractical to use the physical device addresses in source programs since there are differences in these addresses from installation to installation and, in addition, it reduces the system's flexibility in configurations where there are a number of identical units (e.g., 4 disk drives or 5 tape units).

This latter flexibility problem can be better understood if one considers a Model 30 configuration which has 3 identical disk drives and 4 identical tape units. A particular program to be executed requires 2 disk drives and 2 tape units. In the running of this program it would be extremely efficient if the program could be written permitting the use of any 2 out of the 3 disk drives and any 2 out of the 4 tape units during the execution cycle. To overcome the difficulties associated with the use of actual physical addresses a set of symbolic names was developed to represent I/O device addresses. These symbolic names are used by the programmer in the writing of source programs. In actual operation the system makes sure that a symbolic name is assigned to each I/O unit prior to program execution.

The structure of the symbolic name set involves a five- or six-letter character code of which the first three letters are always SYS. Furthermore, the name set is partitioned into two types, *system logical units* and *programmer logical units*. The first category is used by the control program and certain of the processing programs. User programs, however, can also make use of system logical unit names. Programmer logical units are used by some of the processing programs and by user programs. The format for symbolic names using *system logical units* is SYSxxx, where the x's specify distinct alphabetic characters. Examples of symbolic names representing system logical units are SYSRDR, SYSLOG, and SYSPCH. SYSRDR is the name associated with the system logical unit that reads job control statements. SYSLOG is identified with the system logical unit that is used as the operator communication device. SYSPCH is the name of the system logical unit that is the main unit for punched output. In each case the device that serves the purpose can be chosen from more than one type of system unit (e.g., card reader, card punch, tape unit, disk pack, etc.). The format for *programming* logical *units* is SYSnnn where the n's represent numerical units starting with 000 and in consecutive order going to 221. These names can be assigned to any device in the system that can be supported by DOS and is not already identified by the *system logical units*.

The following list is a tabulation of symbolic names together with a brief description of the functions of the logical units and the types of devices to which they can be assigned. The last two names, however, can only be used in certain system commands and statements, and therefore are not permitted in the writing of problem programs.

SYSRDR Card reader, magnetic tape unit, or disk extent used for Job Control Statements.

SYSIPT	Card reader, magnetic tape unit, or disk extent used as the input unit for programs.
SYSPCH	Card punch, magnetic tape unit, or disk extent used as the main unit for punched output.
SYSLST	Printer, magnetic tape unit, or disk extent used as the main unit for printed output.
SYSLOG	Printer-keyboard used for operator messages and to log Job Control Statements. Can also be assigned to a printer.
SYSLNK	Disk extent used as input to the Linkage Editor.
SYSRES	System residence area on a disk drive.
SYSRLB	Disk extent used for a private relocatable library.
SYSSLB	Disk extent used for a private source statement library.
SYS000-SYS221	All other units in the system.
SYSIN	Name that can be used when SYSRDR and SYSIPT are assigned to the same card reader or magnetic tape unit. This name must be used when SYSRDR and SYSIPT are assigned to the same disk extent.
SYSOUT	Name that must be used when SYSPCH and SYSLST are assigned to the same magnetic tape unit.

Note that "disk extent" means an area on a disk pack. SYSRES must always be assigned to an area on disk. An installation specifies at system generation time whether it wishes to be able to assign other system logical units (SYSRDR, for instance) to an area on disk.

Standard assignments for both types of logical units are set up at the time the installation generates its own particular DOS system. In this manner, SYSRDR, SYSLOG, SYSPCH, etc. are assigned to particular devices such as the card reader, the 1052 printer-keyboard, the card punch, etc. Each device of course has a physical address identifying it as well. During the course of a job stream the standard assignments can be modified for the purpose of system efficiency. These modifications can be effected by programmer job control statements or operator-keyed commands implemented between job steps, or at the time of IPL loading.

The format of the assignment command statement differs slightly from that of the job control assignment statement. The formats for an assignment statement and a command statement are as follows:

```
// ASSGN SYSXXX, X 'cuu'     (Assignment statement)
ASSGN SYSXXX, X 'cuu'        (Command statement)
```

The only format variance between the two types of statement is that the command statement does not have the two slashes in the first two space positions. Operationally, the main difference is that an assignment statement is a *temporary statement* while a command statement can be either *temporary* or *permanent*. A *temporary* statement is replaced by the standard assignment at the end of a job. A *permanent* statement retains its status until it is modified between job steps or at IPL time as a result of an operator command on a job control statement. A *permanent* statement can be issued by using the command statement format shown above. A *temporary* statement can be issued by adding an additional field with the letters TEMP to the basic command format. Such a statement, therefore, would have the following format:

ASSGN SYSXXX, X 'cuu', TEMP

A source of potential errors in the use of assignment statements and command statements is the case where both statements are identical. The rule in such cases is that the system accepts as valid the last assignment it is given regarding any particular symbolic name.

A second type of conflict arises in the following situation:

```
// JOB EXAMPLE 5
// ASSGN SYS004, X '132'
// ASSGN SYS006, X '132'
// ASSGN SYS004, X '133'
```

The last assignment rule holds so that the name SYS004 is assigned to device 133 and not 132. In addition, we note, however, that one device can have two symbolic names (first and second assignment statement cards specify the same physical device). In contrast, one symbolic name cannot be used in conjunction with two different physical devices unless the second device is an alternate for the first. To illustrate this case consider a data file which is too large to fit on one disk drive and requires two drives. The logical unit to be assigned to the data file is SYS005. The first set of disk packs corresponding to the first portion of the data file is placed on the first disk drive whose physical address is 141. The second set of disk packs is placed on the second disk drive whose address is 142, while the first set is being processed. The assignment statements would then read as follows:

```
// ASSGN SYS005, X '141'
// ASSGN SYS005, X '142', ALT
```

The command statements would have the same format except for the first two space position slashes.

4.10 OPERATING PROCEDURES

Having learned about the software structure of System/360 and the structure and components of DOS we are now ready to summarize the functions and responsibilities of the operator in running jobs under DOS. The five basic functions that the operator is required to perform are:

a. Compilation and/or execution of a program.
b. Compilation of a program and placing of it into the appropriate library.
c. Execution of an existing library program.
d. Execution of an existing system utility program.
e. Execution of a sort/merge program.

As indicated earlier, the programmer is responsible for supplying the operator with the necessary data and instructions regarding his job via the run sheet, memoranda to the operations department, and job control cards. If a source program is required to be compiled and/or executed the program can be stored on cards, tape, or disk. A program which is in executable form, however, must be placed in the core image library prior to execution unless the job calls for both compilation and execution in the same run.

The operator is able to modify or correct an existing program through use of the 1052 keyboard-printer. This device provides the operator the communications medium for supplying information to the DOS control program. In addition, the 1052 serves as the information transfer medium by which the DOS system can send messages to the operator. These messages alert the operator to problems occurring during the processing and help him in effecting corrective actions. If the operator spots a problem in system running he can query the status of the system and request a listing of all I/O device assignments be printed, as they are read by the system, on either the device assigned to SYSLST (main unit used for printed output) or on SYSLOG (printer-keyboard) or on both. In a similar manner the operator can request that all job control statements be printed on these devices. A special feature for use by the programmer is the OPTION statement which, in the case of job termination prior to completion of the run, results in all the information in core storage and the register being dumped into SYSLST.

The types of information that the operator, depending on the situation, would impart to the system include the date, I/O device assignments, a request to cancel a job in response to a system message, a request to PAUSE between job steps, and restart information. A number of these operator-to-system communications are the result of system or program errors and these situations will be discussed at great length in Chapter 5.

The good operator understands enough about DOS system operation and programmer language to be able to remedy many problems that occur during processing. A minimal number of jobs should be terminated and referred to the programmer for modification. The means by which the operator can substitute for or replace the programmer in supplying information to the system is through the issuance of statements or commands. Statements are identified by the format of the first two positions (i.e. // . . .). Statements are generally entered in the SYSRDR (device used for job control statements) and can only be read by the system between job steps. Commands, on the other hand, don't have the slashes in the first two positions and are almost always communicated to the system via SYSLOG, the console typewriter (1052 keyboard-printer). Commands can be entered into the system at different times during the processing. Commands which can only be issued between jobs and job steps are referred to as job control commands. Commands issued at IPL time are called IPL commands. Attention (ATTN) Commands can be entered into the system at any time after the operator has depressed the 1052 REQUEST button. Job control and IPL commands can be entered into the system via the SYSRDR device as well as SYSLOG. Similarity between statements and commands must not confuse the operator into thinking that the same functions are being performed.

It is important to describe the operator procedure for entering commands to the system via the console typewriter. It is also important to discuss some of the more significant commands and the resultant actions in controlling and directing the system operation under DOS.

Operator commands to the system during processing, via the 1052 keyboard, are initiated by the depression of the REQUEST key. If the system is in a ready state for acceptance of a command the PROCEED indicator lights up on the keyboard and the following message is issued on the console typewriter:

1I60A READY FOR COMMUNICATIONS

This message causes the keyboard to become unlocked and ready for operator issuing of one of the ATTN commands. Four basic ATTN commands are LOG, NOLOG, CANCEL, and PAUSE. The LOG command dictates the printing of all job control statements on the console typewriter. The NOLOG command negates the LOG command. The CANCEL command causes the immediate termination of the program under execution. The PAUSE command puts the system into the wait state at the end of the current job step. Every ATTN command consists of a single word and is not read by the system unless followed by an end-of-block (EOB) character and until the EOB character is transmitted to the system. The procedure, therefore, is to issue an ATTN command and immediately follow with an EOB signal. The EOB character is generated, as explained in Chapter 2, by holding down the ALTN CODING key and depressing the 5 key on the 1052 keyboard. It should be noted that EOB character is represented by the sign Ⓑ in IBM usage, but this sign is not really printed out on the console typewriter. The culmination of a series of commands is followed by another Ⓑ in order to release the keyboard. In other words the last Ⓑ following a series of commands locks the keyboard and only the depressing of the REQUEST key will once again initiate the keyboard unlocking procedure.

The termination of the job being processed by the issuance of the CANCEL command results in the data in core storage and the registers being dumped onto the SYSLST device if DUMP was given as an option during DOS system generation or if the job control cards included the statement // OPTION DUMP. In this case the system bypasses all the remaining job cards and scans for the succeeding group of job control cards which are located in the SYSRDR unit.

The issuance of the PAUSE Ⓑ command dictates the system to go into the wait step and the conclusion of the current job step. At the end of the job step the system sends the message:

1I00A READY FOR COMMUNICATIONS

The system then stays in the wait condition until either another Ⓑ or additional commands are issued. During this period the operator can perform numerous manual functions such as changing tape reels, disk packs, etc. in preparation for the next job step. The PAUSE command can also be used to specify a pause at the end of a job. In this case the format is

PAUSE, EOJ Ⓑ

The programmer may want the operator to ready an I/O device or do some other function during the PAUSE interval. This programmer-to-operator information can be issued via a PAUSE statement in card form in combination with a message regarding the function to be performed. This message appears on the console typewriter in the following format:

// PAUSE MOUNT SCRATCH DISK PACK ON SYS201

Processing is resumed by the operator transmitting Ⓑ to the system.

The use of the console typewriter by the operator in directing jobs and the smooth running of the system under DOS will be further expanded in Chapters 5 and 6. Chapter 5 deals with system errors, how they are detected, and the means for recovery. Chapter 6 describes the running of jobs under DOS and presents good operator practices.

EXERCISES

1. Convert the following numbers:
   a. Decimal to Binary
      (1) 150
      (2) 3080
   b. Binary to Decimal
      (1) 10110101
      (2) 111101000111
   c. Decimal to Hexadecimal
      (1) 135
      (2) 61
   d. Hexadecimal to Decimal
      (1) 3D4
      (2) E1F
   e. Binary to Hexadecimal
      (1) 11011011
      (2) 11110001
   f. Hexadecimal to Binary
      (1) EE2BC
      (2) BCD2A
2. How many different codes are there for representing data in System/360? Name the codes.
3. How many different instruction formats are there in System/360? Identify the formats and define the operation involved.

4. Define the following terms:
   a. OpCode
   b. Program Status Word
   c. IPL
   d. Job Control
   e. System Logical Units
   f. SYSRDR
   g. Assignment Statement
   h. Command Statement
5. State the devices and their quantity that are used in a basic DOS system configuration for a Model 30 Installation.

ANSWERS

1. a. (1) 10010110
      (2) 110000001000
   b. (1) 181
      (2) 3911
   c. (1) 87
      (2) 3D
   d. (1) 980
      (2) 3615
   e. (1) DB
      (2) F1
   f. (1) 111011100010101111100
      (2) 101111000101001010 10
2. a. Four
   b. Fixed-Point Binary Number, Binary-Coded Decimal Number, Floating-Point Number, Logical Information
3. a. Five
   b. RR–Register to Register Operation
      RX–Register to Indexed Storage Operation
      RS–Register to Core Storage Operation
      SI–Core Storage to Immediate Operand Operation
      SS–Core Storage to Core Storage Operation
4. a. Specifies the operation to be performed and is 8 bits in length.
   b. A double word used to control sequencing of instructions and the relationship of it to the system to program execution status.
   c. Segment of the DOS control program is used to load part of the Supervisor and Job Control into core storage.

d. Segment of the DOS control program that is used to interpret the job control instructions and provide for proper sequencing of job step to job step and job-to-job transactions in a job stream.

e. One part of a symbolic name system used to represent I/O device addresses. The control program uses this category to identify devices that are associated with system control operation.

f. I/O device used for Job Control Statements.

g. A temporary statement that is replaced at the end of a job by a standard assignment and is used to assign physical devices to the symbolic names representing I/O device addresses.

h. A temporary or permanent statement. As a permanent statement it retains its status until modified by an operator action on a job control statement either between job steps or at IPL time.

5.

Three	2401	Magnetic Tape Units-
One	2311	Disk Storage Drives
One	2540	Card Read-Punch
One	1403	Printer Model N1
One	1052	Printer-Keyboard Model 8
One	2030	Central Processing Unit

5

ERROR DETECTION AND RECOVERY PROCEDURES

This chapter deals with the types of errors that occur in system operation and the means for detecting the errors and recovery procedures. The topics covered include device error recovery procedures, console bypass error detection and correction techniques, internal DOS hardware and software aids including diagnostic control programs, and restart and save procedures. In each case the types of errors that can occur and the actions required of the operator are described in detail. After facility has been gained with the various system devices the operator will find that error correction procedures can, in general, be accomplished rapidly and information losses kept to a minimum. The student should note that mastery of error detection and recovery actions is a primary factor in becoming a good and efficient operator.

5.1 DEVICE ERROR RECOVERY

The purpose of this section is to describe the corrective action that must be taken to recover from errors that might occur with input-output devices and other devices which are used in the DOS system operation. We will deal with each device individually and cover the types of errors that most readily occur.

5.1.1 Card Reader Errors

Mechanical errors which might occur during processing will be indicated by the lights on the reader side of the 2540 card reader-punch (e.g. feed stop, read check, validity check, stacker, transport). It is possible to recover from these errors using the following simple procedures.

a. *Feed Stop*

The feed stop is probably the simplest of all errors from which to recover but the steps involved in clearing a feed stop are basic to recovering from most read errors.

1. Remove cards from stacker since these cards have already been processed.
2. Open joggler gate and lift the cards in the hopper gently but firmly.
3. Push the *stop* button; this will turn off the *End of File* light and put the reader in a Non-Processing Mode.
4. While still holding the cards behind the joggler gate, push the *Start* button. This will enable the cards (if any) under the read brushes to drop into the R1 stacker.
5. Examine those cards which have fallen into the R1 stacker for nicked edges, the major cause of feed stops, or rips or tears. If a nicked edge is discovered it can be corrected some times by running your thumbnail along the edge until the card is again flat and thus able to be reprocessed. If the cards are ripped or torn, duplicate the cards, keeping them in sequence and continue processing. In the event that no cards are dropped into the R1 stacker the problem card is that card which is at the beginning of the deck behind the joggler gate. Examine this card for nicks, rips, or tears and duplicate it if necessary.
6. Now that the damaged cards have been replaced we are ready to restart processing. Place the cards, or their duplicates, back into the hopper at the beginning of the deck behind the joggler gate. Remember to keep the cards in sequence. Close the gate, press the *Start* button and the *End of File* button and continue system processing.

b. *Read Checks*

The Read Check light comes on when a card is not being read properly. This can result from either off-punched cards or incorrect registration of cards in the transport. The Read Check recovery procedure is very similar to that of the Feed Stop with only slight variations as follows:

Follow steps 1–4 under Feed Stop instructions. After doing the Non-Process Runout you will notice that three cards have dropped into stacker R1. It is the card that *preceded* these cards that caused the Read Check. Examine this card for off-punching (i.e. punched holes in wrong positions). At times off-punching is not easy to detect and you will need the aid of a Card Gauge.

It is possible that the Read Check was not caused by an off-punched card but by a card that is warped or bent. It is possible to flatten the card and continue processing, but if the entire deck is warped so badly that it causes constant Read Checks and Feed Stops it might be necessary to have the deck reproduced before proceeding with the run.

After detecting the error and replacing the card if necessary, place the four cards at the beginning of the deck in the hopper behind the joggler gate and continue processing.

c. *Validity Checks*

A Validity Check error is handled in the same manner as a Read Check. The only difference is in the way in which the error is caused.

Validity Checks are caused by over-punching in the card. The only area that is checked for over-punching by the machine is rows 1 through 7 of each column. An over-punch in rows 12–0 (zone punches) or 8 and 9 will be undetected. Detection of over-punching is accomplished by examining rows 1 through 7 of a card for more than one punch in one column. When the error is detected the card should be repunched correcting only that column in which the error is found.

After correcting the error, place the four cards, the three that dropped into the R1 stacker and the card that *preceded* them, at the beginning of the deck behind the joggler gate and continue processing. It should be noted that the Validity Check light does not go out until the card is read correctly.

d. *Stacker*

When processing a deck that consists of many input cards it is possible that the stacker might be filled many times before the completion of the run. When the stacker is filled a sensing device on the reader will cause the Stacker light to go on. By removing the cards from the stacker the light is turned off and the system is ready for processing again. This action is initiated by pressing the card reader *Start* button.

e. *Transport Checks*

The Transport light comes on for one of two reasons, either a card has gone astray on its path to the stacker or a jam has occurred at one of the read stations. The following steps should be taken to recover from a Transport error:

1. Take everything off the top of the reader and raise the cover. This is accomplished by grasping the middle or lower section just above the stackers and raising gently.

2. If you see that the cause of the error was merely a card that has deviated from its normal path, remove the card, lower the cover and press the *Start* button to continue processing. In this situation, the card that caused the error has already been read. More often than not the above occurrence is not the reason for the error and you will have to clear a jam. If after raising the cover you discover that a card jam is the cause of the error, you must take steps to clear the jam while remembering to keep the cards in sequence.

3. After discovering the jam, first remove those cards which are loosely jammed. These are the cards which are free of the transport rollers.

4. Then remove the read brushes to free the cards jammed at the read stations. This is accomplished by squeezing the finger clasps at the top of each block, allowing the locking pins to be released from the sides, and then pulling up on the block until it is free.

5. Once having removed the blocks, remove the loosely jammed cards that were at the read stations. Some cards will be beneath the transport rollers and will require hand feeding in order to permit removal of the cards.

6. In order to hand feed, it is necessary to open the front panel of the reader. This can be accomplished by inserting your hand into the cutout in the right-most panel and pulling it open. Inside the panel you will see a wheel which is the feed mechanism. To the left of the wheel is a toggle switch which must be pulled in order to release the clutch. After releasing the clutch, grasp the jammed card with your left hand and, pulling firmly but not ripping the card, turn the feed mechanism in a *clockwise* direction with your right hand. Pull the card up and out and the jam is cleared.

7. Make sure that no card fragments remain in the reader. Check the read brushes to ensure that they weren't damaged. Re-insert the read brushes by squeezing the finger clasps. Repunch the damaged cards and place them back in sequence at the beginning of the deck behind the joggler gate and continue processing.

Note: Before leaving the Card Reader it should be noted that most read difficulties can be avoided by proper and careful handling of the input decks and by making sure that foreign elements such as paper clips or rubber bands are removed from card decks before processing.

5.1.2 Card Punch Errors

The errors that occur on the card punch are quite similar to read errors and are handled in much the same way as read errors. The types of punch errors which might occur are indicated by the lights on the punch and are Chip Box, Punch Stop, and Punch Check.

a. *Chip Box*

The Chip Box light indicates that the chip box is full and must be emptied. The chip box is located directly under the punch area behind the left-most panel of the front of the machine. It is made accessible by inserting your hand into the cutout in the panel and pulling the panel open. After emptying the chip box, replace it, push the *Start* button and continue processing.

b. *Punch Checks*

Punch check recovery is a simple operation:

1. Remove cards from hopper. At this time, *do not press any buttons.*
2. The error card is the card that has dropped into stacker P1. If the normal path of the cards is stacker P1, the last card in the stacker is the error card. Discard the error card.
3. Remove the cards from the stacker and Non-Process Runout the punch. This is accomplished by pressing first the *Stop* and then the *Start* buttons. This will reset the punch check but the Punch Check light will remain lit.
4. Replace the cards in the stacker, place blank cards in the hopper and press the *Start* button. The card that was punched in error will automatically be repunched and the run will continue processing.

c. *Punch Stops*

It is possible for a card jam to occur in the transport area of the punch similar to the jam which occurs in the reader. The steps for clearing a card jam on the punch are as follows:

1. As with the jam on the read side we must first raise the cover to examine the jam. Remove the cards from the hopper and the stacker.
2. Remove the punch check station block by squeezing the finger clasps and raising. Remove any loosely jammed cards. All these cards must be repunched and placed behind those which had previously fallen into the stacker.
3. Next, tilt back the hopper in order to clear those cards at the blank station or prepunch area. This is accomplished by pushing the metal latch, which is found behind the hopper, toward the back of the machine. We can then tilt the hopper out of the way and gain access to the prepunch area and easily remove the loosely jammed cards found there. Hopefully the jam has now been alleviated, but if it is seen that there are more cards jammed beneath the punch die the job is not quite finished.
4. In order to remove the die-stripper it is necessary to gain access to the left front side of the machine in a manner similar to removing the chip box. Remove the chip funnel, as this will enable the stripper to be reached. The stripper handle must be turned to allow it to drop down toward the chip box. At the same time squeeze the small lock handle toward the stripper shaft so that the stripper can drop free of the die area.
5. Examine the area from which the stripper was removed. If there are any loosely jammed cards or card fragments present, remove them. It is now possible to hand feed and remove the cards in the transport. Before hand feeding, it is necessary to trip the clutch by using the clutch trip handle to the left of the stripper area. Move the handle to the right while turning the hand feed wheel. When the clutch makes a clicking sound, the clutch is engaged. At this point release the clutch trip handle and hand feed to remove the card. It may be necessary to repeat this operation to ensure that the punch area is clear. After the transport area is clear of all card fragments, put the punch back together in the following order:

   a. Replace the die-stripper.
   b. Replace the chip funnel.
   c. Replace the punch-check brush block.
   d. Reposition the hopper and lock it.
   e. Close the cover.
   f. Replace blank cards in hopper.
   g. Non-Process Runout the punch.
   h. Replace cards in the stacker with repunched cards.
   i. Press the *Start* button and continue processing.

5.1.3 Line Printer Errors

Mechanical errors which might occur during printing operations are of two types: those indicated by the error lights on the printer (i.e. form checks, print checks, end of forms, and sync checks) and those not indicated by the lights, such as runaway forms, improperly placed output and non-form movement. The following instructions will show how to recover from these errors.

a. *Form Checks*

The *Form Check* light indicates that a form feed error has occurred in the tractor area. The following steps should be followed to correct the error.

When a *Form Check* occurs the printer cover is automatically raised. Be sure never to leave anything on top of the printer cover. Pull the print mechanism release lever (Figure 2-11). This will allow viewing of the entire form feed. If the paper is still one continuous form from the stacker, the trouble lies in the tractor area. Open the tractors. Observe whether or not the form is ripped or rippled beneath the tractors. If this is the case, smooth the form. It might also be necessary to adjust the *RH Tractor Vernier* knob in order to avoid a recurrence of the error (Figure 2-11). If it is noticed that the sprocket teeth are too far to the left or right, causing a form check, make the adjustment as necessary. Close the tractors and the print mechanism, press the *Check Reset* button and then the *Start* button and continue processing (Figure 2-12).

b. *Print Checks*

A print check occurs when there is an error in the last line of output. This error might be either a hardware malfunction or a result of the UCS (Universal Character Set) feature of the machine.

If the error occurs and the machine is not equipped with UCS, note the line of print, press the *Check Reset* button, press *Start* and continue processing (Figure 2-12).

If the error occurs and the machine is equipped with UCS, try the above method of recovery. If the error persists, load the UCS buffer program and restart the run. The UCS buffer program will be further discussed in Chapter 6.

c. *End of Forms*

An *End of Forms* light indicates that no more paper is available to the print mechanism. The recovery method is to load more forms but there are a couple of operations which must be taken care of before loading the forms.

1. The printer cover rises when an *End of Form* occurs. Glance down behind the print mechanism just above the printer to see if the last form has been used entirely. If the bottom of the form can be seen above the ribbon, go on to step 3; if it cannot, follow the instructions in step 2.

2. Press the *Single Cycle* button. This will enable the printer to print only one line and then stop. Keep pressing the *Single Cycle* button until the bottom of the form can be seen. Do not get ahead of the printer while using the single cycle; wait until the line has printed and the printer stopped before pressing it again. One lost line can be cause for a rerun.

3. Now that the bottom of the form is in view, open the print mechanism. Open the tractors and, leaving the form in place, take the top edge of the form about to be loaded and line it up with the top edge of the form that is in the printer. Align the

form holes with the sprocket teeth. Close the tractors. Close the print mechanism. Press the *Start* button and continue processing. As the forms are being printed, ensure that they are being fed properly into the back of the printer.

d. *Sync Checks*

The *Sync Check*light indicates an internal timing problem. To recover from this error, press the *Check Reset* button, press the *Start* button and continue processing.

e. *Runaway Forms*

A runaway form is a condition that occurs when the form appears to be skipping or sliding from one form to the next without any intervening stop or print. If this error should occur press the *Carriage Stop* button immediately.

The causes of a runaway form are in the carriage control tape area. If one of the following errors is found, notify the programmer or the supervisor:

1. A missing one-punch or a missing twelve-punch. A missing one-punch will cause the printer to continue searching for a printable position. A missing twelve-punch will cause the printer to keep searching for an end of form.
2. Ensure that there is only one punch in each of the punched columns. More than one punch in a column will cause the printer to continue to reject a skip to that position.
3. Be sure that the carriage control tape is mounted properly. If the tape is mounted backwards or inside out the run will not be printed correctly.
4. Check the control brushes. They should be clean, not bent, and making proper contact.

f. *Improperly Placed Output*

A worn or torn carriage tape could cause the printer to make irregular or unscheduled skips. At times this condition will cause some pages to print properly but other pages will be incorrectly printed. Check the punched holes to ensure that they are not oversized. Check the feed holes to ensure that they are not worn and making lateral movements causing misreading of control punches.

g. *No Form Movement*

This error occurs when the printer is printing but the form is not moving. First check to ensure that the feed clutch is not in neutral. If the clutch is in the proper drive there may be a mechanical problem. Notify the supervisor.

Most printer problems can be avoided if the following steps are taken:

1. Before mounting a carriage control tape, check it to ensure that it is not torn or worn.
2. Be sure that the continuous forms can freely flow from the carton in which they are contained.
3. Periodically check the print alignment, the stacker, and the amount of paper left in the front base of the machine.

5.1.4 Tape Errors

Before entering a discussion of tape error recovery methods there are certain terms pertaining to tapes with which the operator should be familiar. Following are these terms and their definitions:

Scratch Tape—A scratch tape is a tape that is used in a job, but has no pertinent data on it prior to the job.

Work Tape—A work tape is a tape that is used in a job but has no pertinent data on it prior to the job. It will have no valuable data on it after the job is completed.

Output Tape—An output tape is a tape that will be used to receive data during a job and must be saved because the data on it is valuable. An output tape may be a scratch tape at the beginning of a job.

Scratch tapes, work tapes, and output tapes all must have a file-protect ring, as these tapes will be written upon.

Input Tape—An input tape is a tape which contains valuable data at the beginning of a job. The file-protect ring should be removed from this tape, as the information on it must not be destroyed.

BPI—BPI is the number of bits per inch that can be or were written on a tape. This is also referred to as the *density* of a tape.

Tape Mark—A tape mark is an ending indicator for the records on the tape. It is a series of electronic impulses on the surface of the tape that are pertinent to the operation of the machine.

It is important that the operator become familiar with these terms in order to communicate with the programmer and to better understand the instructions written by the programmer on the run sheet.

Now that some basic terms have been defined it will be easier to understand how to handle error problems that may arise.

Data Check—The data check error is indicated by a message that is discussed later. It is usually caused by a defective spot on the tape. An oxide deposit may have formed on the read/write surface causing a film on which data cannot be written.

The tape drive will attempt to clean the spot from the tape by making several backward and forward passes over the tape cleaning blade. If the attempt is unsuccessful the data-check error message will print out.

The operator's response to a data check error depends upon the operation the job was attempting. In the discussion of error messages this will become more clear.

The steps that should be followed if a data check occurs are:

1. Ensure that the correct tape is mounted.
2. Clean the tape drive and retry the job.
3. If the tape is a work tape or scratch tape, mount a different tape and retry.
4. If the right tape is mounted and the drive is clean, reassign the tape to a different drive. If the job runs successfully report the original drive to the customer engineer or supervisor as a down drive.

Load-Point Errors—Load-point errors are those errors which are associated with the front portion of the tape near the load point. Following are these types of errors and how to recover from them:

a. *Pinched tape*—The majority of data checks that occur at load point are caused by a pinched tape. Magnetic tape is inflexible and once it has been pinched it is unable to be read or written upon. If a tape has been pinched it is necessary to remove that portion of the tape which is bad by cutting it off past the load point.

After removing the bad portion of the tape it is necessary to put a new load point on the tape. This is accomplished by placing the reel on the tape drive and proceeding as if the tape was to be mounted and readied.

After the tape has been mounted, press the release button and spin the tape takeup reel at least 12 complete turns (Figure 2-6). This will put ten to fifteen feet of tape on the takeup reel which will be the leader of the tape. Affix a new load-point sticker to the shiny portion of tape about 1/8 of an inch from the side of the tape nearest the window. The tape is now ready to be put back into use.

b. *Runaway tape*—A runaway tape occurs when the job is trying to find a tape mark on the tape. The SELECT light comes on and stays on while the machine runs through the entire reel of tape.

If the runaway tape is an input tape the operator should call the shift supervisor or the programmer for assistance.

If the tape is an output tape, work tape, or scratch tape, stop the job, put a tape mark on the tape, and retry the job. If the problem still persists call for assistance.

c. *Tape that will not load*—When in mounting and readying, the tape comes off the takeup reel, this indicates either a missing or misplaced load point. If the load point is missing, the operator should put a new load point on the tape ten to fifteen feet from the end of the tape and 1/8 of an inch from the edge nearest the window.

d. *Attempting to read at wrong density or mode*—During a job run, many times an error will occur that will seem to indicate that something is wrong with the tape but upon examination the operator finds that both the tape and the tape drive are in excellent condition. The operator should first check to see if the program is trying to read the tape at the correct density. If a tape is written at 800 BPI, it can only be read at 800 BPI. If a tape is written on a 9 track drive, it cannot be read on a 7 track drive. If a tape has been written in BCD mode it cannot be read in EBCD mode. A quick check of this type can, many times, save valuable computer time.

Many of the tape errors that occur can be avoided by careful and conscientious operation habits. The operator should be sure the tape drives are always clean. The operator should always check the label on a tape to ensure that he is mounting the proper tape reel on the proper drive.

5.1.5 Disk Errors

Most disk errors that occur are closely associated with the console typewriter messages that print out while the job is processing. This discussion will deal with the errors that occur while processing on a 2311.

Select Lock—If a Select Lock indicator lights on a disk drive, it indicates an unsafe condition exists on the device. The drive should not be used until it can be checked by a customer engineer. Do not take the disk out of ready, leave it just as it is until the customer engineer can check it. If possible, run jobs that do not require this disk drive.

Data Check—The data check message will be printed on the console typewriter. At this time, the operator may attempt the operation again by typing in RETRY. If the data check occurs again, either the drive or the pack may be at fault.

Stop the job and mount the pack on a different drive, alternately assigning the device, and rerun the job.

If the failure occurs again under these conditions, it indicates that the pack is bad. *DO NOT MOVE THE PACK AGAIN.* The pack is damaged and may be damaging the drives.

Call the customer engineer and let him examine the drives and the pack. Do not use the drives until the customer engineer has checked them.

If possible, run other jobs not requiring these drives.

Seek Check—In the event of a seek check the procedure for the data check should be followed.

Missing Address Marker—Attempts to recover from a missing address marker are the same for a data check.

If the problem continues after changing drives it may indicate that the pack should be re-initialized. Reinitialize the pack and try the job again.

No Record Found—No record found is a programming error. The message indicates that the information being sought is not on the disk. The operator should retry once but if the message prints out again, the job should be cancelled and the dump given to the programmer. Do not attempt to restart the job.

5.2 ERROR RECOVERY USING INTERNAL SYSTEM DOS HARDWARE AND SOFTWARE AIDS

This section will deal with error recovery methods using available system hardware and software aids. In order to fully understand these methods it is first necessary to describe the characteristics of system-to-operator messages. These were discussed briefly in Chapter 4 and are presented in much greater detail in the ensuing paragraphs.

The DOS system is so designed that the operator will be informed via the console typewriter if the system is having a problem processing the run due to difficulties. The portion of the system that handles these messages is called the operating system monitor. The components of the monitor that the operator is associated with are the System Supervisor, Job Control, and IBM Routines.

The System Supervisor controls the operation of the CPU, the input-output operation, error recovery, and job interruptions. Job Control prepares the system logically for each job that is run. It works in conjunction with the first few statements of the job which identify the function of the job and the specific devices to be used. It also logically repositions the system after each job to ease job-to-job transition.

IBM Routines are DOS routines supplied by IBM to enable the program to communicate with the supervisor. These routines are concerned with input-output modules, DOS subroutines, and computer language compilers.

It is important to remember these three groups since the *DOS Operating Guide* is divided into sections by this grouping. It will be, at times, necessary for the operator to consult this guide.

5.2.1 System-to-Operator Messages

Before going any further, it is important to identify the format of the system DOS messages to the operator. All messages from the monitor to the operator are typed out on the console typewriter. The format consists of a two-character program identifier, followed by a four-character message code, and comments. The comments can extend to more than one line but the program identifier and message code are printed only at the beginning of the first line.

The following program identifiers are found in the DOS system:

Identifier	*Program*
BG	Background Program
F1	Foreground-one Program
F2	Foreground-two Program
AR	Attention Routine
SP	Supervisor

The first character of the four-character message code indicates the portion of the job that was being processed at the time the run was interrupted. These first portion indicators are as follow:

Code	*Originator*
0	Supervisor or IPL
1	Job Control
2	Linkage Editor
3	Librarian
4	Logical IOCS
5	PL-1
6	RPG
7	Sort-Merge
8	Utilities
9	Autotest
A	Assembler
B	Fortran
C	Cobol

The second, third, and fourth characters of the message code are the message number. The action indicator (I, A, or D) following the message code tells what action is required of the operator.

Action Indicator	*Meaning*
A–Action	The operator must perform a manual action before continuing.
D–Decision	The operator must make a choice between alternate courses of action.
I–Information	The message does not require immediate operation action.

Now that the format of the message has been explained, let us look at two examples. The most common type of message is:

BG 0P08A INTERV REQ

The "BG" indicates the message came from the background program. The "0" indicates that it is a Supervisor originated message; "P08" is the message number; and "A" indicates that the operator must perform a manual action. In this case it might be necessary to mount a tape or otherwise ready an input-output device.

Another example of a message that might appear is:

1A80D SYSTEM FILE OPEN FAILURE

The "1" indicates that the message originated in the Job Control routine of the monitor; "A80" is the message number; and "D" indicates that the operator must make a decision. In this case a logical unit has not been assigned by the IBM Supervisor and the operator must either type a new assignment through the console typewriter or key end-of-block on the console typewriter, or type CANCEL to cancel the job, or submit new label information through the card reader to correct the failure.

All messages with which the operator is not familiar or messages which call for a decision should be checked in the *DOS Operating Guide*. To respond properly to messages it is necessary to know the cause and response.

The following procedure should be followed when a system message is printed on the console typewriter:

1. Determine the source of the message.
2. Read the interpretation of the message.
3. Examine the courses of action.
4. Perform required action.
5. Respond to the message.

In 5.1 we dealt with errors on devices and it was pointed out that these errors would sometimes be indicated by lights, and at other times no physical indication would be given. The Disk Operating System, however, provides the operator with indications of all error situations in the form of messages. We have dealt with the format of these messages and listed the steps an operator should take in responding to these messages. We will now deal more specifically with the way an operator should follow through when responding to messages from the system.

The five steps given for proper operation action are again:

1. Determine source of message.
2. Read message interpretation.
3. Examine courses of action.
4. Perform required action.
5. Respond to the message.

The message example to be used in describing operator actions using the available system aids will be:

0P11D DATA CHECK

Determine Source of Message:

By analyzing the message identifier the source of the error can be determined. The 0 indicates it is a supervisor message; P11 indicates the message number; and D signifies that a decision is to be made. The message on the console typewriter tells which device has failed.

Read the Message Interpretation:

Look up the message in the Operating Guide and read everything that pertains to the message. The message write-up in the Operating Guide will tell the operator what may have caused the error and what courses of action are available.

In the case of a DATA CHECK the operator has the decision of typing on the 1052 keyboard CANCEL, IGNORE, or RETRY. The result of each response should be known by the operator. The Operating Guide will explain what course of action can be taken.

Examine the Courses of Action:

Typing CANCEL terminates the job without any further attempt to process the run.

Typing IGNORE causes the information in error to be processed in spite of the fact that an error occurred. Processing will continue if the information in error does not destroy the job's logical run procedure.

Typing RETRY results in the re-execution of the operation in which the error occurred. If this is successful the job will be processed as if the error had never occurred. If the RETRY fails the message will be printed out again on the 1052 typewriter. This will give the operator the opportunity either to RETRY again or try some other means of recovery.

Typing BYPASS results in the error being ignored. The routine will be terminated and the operation which was processing at the time of the error will be bypassed. No input or output information will be transferred.

Typing DELETE will destroy the file being overlapped and a new file will be created. This action refers only to disk packs and should not be typed without consent of the shift supervisor.

Typing DSPLYV results in another message which asks whether the operator wishes to display on either SYSLST or SYSLOG. Under normal conditions the response

a. *Analyze the Message*

Each message in the Operating Guide gives the cause for the error. Read the message thoroughly. This will determine the system logical unit to be listed.

b. *Prepare and Cancel*

Before responding to the message, remove the /& from the back of the deck to ensure that the job is not completely terminated. After removing the /& card, reply CANCEL to the message. This will cancel the job but retain the assignments made up to this point.

c. *List the I/O Assignments*

By typing into the console keyboard one of the LISTIO commands, found in the Operating Guide, the type of list desired will be printed out on the console typewriter.

d. *Examine the List*

An I/O list of all the standard assignments for the computer should be near the console. This list contains all of the I/O units and the devices to which they are assigned. If a UA in the device address appears, this indicates that the device is unassigned.

If the device or I/O unit is in error, this list will aid in determining the operator's course of action.

e. *Correct the Condition*

If the operator is able to spot the error after checking the LISTIO with the standard assignment list, he should do so. If the operator cannot see the error, the shift supervisor should be notified; the information gathered should be sufficient to correct the error.

f. *Rerun the Job*

After the error is corrected, rerun the job by positioning all devices and restarting in the manner requested on the run sheet. Be sure to replace the /& card at the end of the input deck to ensure proper job delimiting.

If multiple assignments are read as job control statements, only the last one read is valid.

// ASSGN SYS001, X '180' followed in the job stream by // ASSGN SYS001, X '182' would assign SYS001 to device 182. Remove the card in error in any situation of this type.

5.3 CONSOLE BYPASS

At times it might be necessary to operate when the 1052 printer-keyboard is not available to the system. If this situation does occur, the following steps may be taken to run a job:

1. *Assign a Printer to SYSLOG*

Messages to the operator are printed on SYSLOG, after which an *assumed* operator response, where applicable, is undertaken. In most cases, the assumed response results in the termination of the job.

2. *Assign a Printer to SYSLST*

If the same printer is assigned to both SYSLOG and SYSLST, system-to-operator messages may be embedded in the program's output.

3. *Assign a Card Reader to SYSRDR and SYSIPT*

This may be the same card reader or two different card readers.

4. *Assign a Card Punch to SYSPCH*

When a 1052 printer-keyboard is not available, jobs containing job step errors or improper I/O assignments will be cancelled. In many cases such errors could be corrected by the operator using the 1052 printer-keyboard.

The operator cannot communicate with the system except to respond to certain I/O messages. The message is printed on the printer assigned to SYSLOG and the system enters the wait state. The printed message also appears in main storage bytes 0–3. The operator response is entered into byte 4 of main storage and then the interrupt key is pressed to continue processing.

The following instructions will deal with errors that may occur during IPL procedure and Device Error Recovery.

5.3.1 IPL Error Messages

If the machine enters the wait state during an IPL procedure, the operator should display the first five bytes of low core (bytes 0–4). The IPL error message number and action code are displayed in hexadecimal in these bytes. For example:

Message 0I11A appears in low core bytes 0–4 as
F0C9F1C1

The operator should interpret the error message, look up the message in the Operating Guide, and perform the required action.

5.3.2 Device Error Recovery

If byte 0 contains a binary number 08–60, it indicates a 2(OP) device error recovery message. If the 1052 is inoperable when an error recovery message is issued the system immediately enters the wait state until the operator replies. The operator should display the contents of byte 1 to obtain the action code, in BCD.

If byte 1 contains an A(C1):

The operator should refer to System-to-Operator Messages in the Operating Guide. If the operator decides to try to continue operations it will be necessary to display bytes 2 and 3 of low core to obtain the channel and unit number of the device. The operator then should perform any manual recovery procedures implied by the error condition. It might merely be necessary to ready a device. If the operator wishes to cancel the job, this can be done by entering X '03' in byte 4 and pressing INTERRUPT.

If byte 1 contains a D(C4):

A trial-and-error procedure must be performed. The operator should first store X '01' (RETRY) in byte 4, then press INTERRUPT. If the system accepts this replay, the machine exits from the wait state. If not, store one of the following in byte 4:

X '02' (IGNORE)
X '03' (CANCEL)

Then press INTERRUPT. When the replay is accepted by the system, the machine will exit from the wait state.

5.4 RESTART AND SAVE PROCEDURES

If a job is to be very long the program may include checkpoints in order to analyze and record the status of the program up to that point. As the job is running, checkpoints are being taken at preselected intervals and the following message will print out on SYSLOG:

0C00I CHKPT NO. XXXX WAS TAKEN ON SYSXXX = CUU

The reason for checkpointing is to allow the operator to recover from an error that cancels the job (i.e. save the job) without having to rerun the job from the beginning. If a job with checkpoints is running and a failure occurs, the recovery procedure is as follows:

1. *Examine the Console Typewriter Messages*

By examining the messages the operator can determine how many checkpoints have been taken and the number of the last checkpoint taken. The message will also tell on which system unit the last checkpoint was taken.

2. *Determine the Last Checkpoint Taken*

To determine on which device the last checkpoint was written, examine the job control statements for a // ASSGN that is equal to the SYSXXX of the last checkpoint message. From this assignment card it is possible to determine the file name from the job run information sheet or the file name can be found by examining the // DLBL following the // ASSGN card.

3. *Position the Job to Initial Start*

Reposition all files as they were at the beginning of the job. All data cards used in the job should be returned to the reader, preceded by the job control statements. One job control card must be altered.

4. *Substitute RSTRT for EXEC*

The // EXEC card for the job being executed should be removed and replaced with a // RSTRT card of the form:

// RSTRT SYSXXX,nnnn,filename

The XXX portion is the system number printed in the last checkpoint message. This is the checkpoint from which the restart is being attempted. The nnnn is the four position number of the checkpoint record to be used for restarting. The filename is the name on the file on which the checkpoint was taken.

Once this card has been punched, place it in the job control statements in place of the // EXEC card. The job control deck and data deck are now ready to be placed in the reader.

5. *Restart*

The restart is accomplished in the same manner as a normal start of a job run.

The only two checkpoint-restart error messages from which the operator should attempt to recover are 0R00I and 0R03I. The former states that either the SYSXXX portion or the filename portion of the RSTRT card is invalid. The card should be examined and corrected. The 0R03I message is the result of not having the proper nnnn field in the RSTRT card or not giving all of the job control statements at restart time. Re-examine the original job control statements to be sure they are in sequence.

If the checkpoint is error free, restart will be at the position the checkpoint record was taken. All devices will reposition to that point and go.

EXERCISES

1. When checking a card which has caused a Validity Check, what should the operator check?
2. When restarting a job with checkpoints, which card is removed from the program deck? What replaces it?
3. What are the most common job-control card errors? How can the operator help guard against them?
4. What action should the operator take if he notices the printer forms sliding from one form to the next without stopping or printing?

ANSWERS

1. Rows 1 through 7 of the card in error should be checked for more than one punch in a column.
2. The // EXEC card should be removed and replaced by a // RSTRT card.
3. The most common job control card errors are improper job end or beginning and improper assignments. Many errors of this type can be avoided if the operator eye-checks the deck before running the job.
4. Press the *Carriage Stop* button immediately.

6

GENERAL OPERATOR PRACTICE

This chapter concerns itself with the rules and procedures to be followed by the computer operator to insure that jobs run by him under DOS are processed efficiently and accurately. These good operator practices are important since they will establish at an early stage the type of habits that result in operator proficiency. The items covered in this chapter include: the use of LOGS to aid the operator in recovering from situations where the programmer has omitted some control instructions; rules for establishing priorities with respect to the running of jobs and the effecting of error recovery procedures and the saving of data; methods and procedures for the running of jobs under DOS; and other special topics of interest in establishing good operator practices.

6.1 LOGS

Occasionally when running a job the operator will have the opportunity to recover from an error made by the programmer. This might be anything from the changing of an assignment statement to the changing of a control card to accept a tape at the proper mode setting. With the aid of this manual the operator should be able to handle any of these situations that arise.

For example:

Assume the following message is printed on the console typewriter during the processing of a job:

// ASSGN SYSPCH, X '042'
1A54D DEVICE NOT-DEFINED

Here the programmer has stated in his program that the SYSPCH would be assigned to 042, a device that was not defined in the PUB table. An operator could, under normal conditions, cancel the job at this point and have the programmer resubmit the run. But the good operator could save the job by seeing what the programmer is trying to do and altering the assignment. If the operator types into the console typewriter:

ASSGN SYSPCH, X '00D'

the job would then assign the SYSPCH to 00D, an address which is supported by the system, and the job will probably run to completion.

Another problem that may occur is as follows: The operator sees that the job run is to read a certain tape on a nine-track drive, but glancing at the tape label he notices that the tape in actuality was created on a seven-track drive. Again the operator could cancel the job, but the good operator can change the control cards pertaining to that tape and run the job.

Another good operator practice is always to check and recheck the log or run sheet supplied by the programmer with his job, to be sure that you have complied with his instructions and that he hasn't made any slight errors. If he has, correct them if possible. *The most important thing is to get the job done.*

There are many such situations that will arise where the operator will have the choice between abandoning the job because of some small error or making a change and allowing the job to run. *The rule for a good operator should be: if you can do anything to get the program to run, do it.*

6.2 PRIORITIES

Before beginning his shift, each operator should copy the schedule for the computer he is to operate. Certain jobs are scheduled for certain times and the operator should adhere to the schedule as much as possible. There will be times when the shift supervisor will alter the schedule. There also will be times when either no jobs are scheduled for a certain time or a job will not run as long as the programmer had anticipated. At these times it is important that the operator know just what job to run.

First the operator should check with the shift supervisor to see if there is any job that he knows must be run. If the shift supervisor is not available or there are no jobs pending, the operator should run jobs which have not been scheduled, in the following order:

1. Clean jobs (those jobs which do not require any tapes or special disks).
2. Jobs requiring a limited amount of tapes and/or disks.
3. If it was necessary to mount a special disk, run all other jobs requiring that disk.

The main thought of the operator should be to run as many nonscheduled jobs as possible before the next scheduled job is due.

In general, the single most important function an operator performs is collecting information from a job run and saving all necessary information. The programmer is responsible for the necessary input and job control information to run the job. The operator is responsible for the successful processing of the job and the information (tape reels, cards, paper, disk packs) that the job required.

If the job fails, the operator must collect all pertinent data for correcting the situation so that the job will run successfully the next time. If possible, the operator should do everything he can to prevent non-productive job runs.

6.3 JOB RUNNING

The normal run procedures of any job require initial setup, the run itself, and maintaining a good run status. The initial setup in DOS is to IPL the system resident disk pack. The runs will include all of the jobs that the operator will run under the control of the Disk Operating System. Application of what has been covered in the preceding chapters will help the operator maintain a good run status.

The Initial Program Load (IPL) procedure of the Disk Operating System can be initiated from the console typewriter or from cards in the card reader.

The procedure for IPL'ing is as follows:

1. *Ready DOS System Pack*

The procedure for readying the DOS pack is the same for readying any disk pack on a drive. But there is one restriction. The DOS pack must be loaded on the particular drive that the installation has specified. This is normally 190 or 290. If the normal drive is not available the operator will have to make changes using ADD, DEL, and ASSGN commands. Once the drive is ready the operator should set up the CPU controls.

2. *Set CPU Controls*

The CPU must now be readied with the switches set properly. Set the CHECK CONTROL switch to STOP, and all other control switches to PROCESS. The INTERVAL TIMER switch, if present, should be turned off. The LOAD UNIT switches should be set to the address of the disk drive on which the DOS pack is mounted. The CPU, KEYBOARD, and PRINTER switches should be turned to the ON position on the keyboard switch panel.

3. *Load*

Press the LOAD key on the console. The LOAD, SYSTEM, and TEST lights will come on; then the LOAD light will go out, followed by the SYSTEM light. Next, the WAIT light will come on.

The system is now in a wait state awaiting an interrupt of any kind. At this time if any device attached to the system is made ready, DOS will attempt to handle the situation by going to that device to receive instructions. For this reason the first device to interrupt this wait state *must* be the valid device from which the operator will complete the IPL.

4. *Request*

Pressing the REQUEST key on the console typewriter will initiate an interrupt to the system and will cause an appropriate action to be taken. If the operator was IPL'ing from the card reader he would ready the reader with the ADD, DEL, and SET commands rather than keying REQUEST on the console typewriter.

At this time a message will print out on the console typewriter. The message will be:

0I10A GIVE IPL CONTROL COMMANDS

This message indicates that the system is awaiting the operator's control commands.

The PROCEED light on the console typewriter is now on and the keyboard is ready.

5. *Set Date and Clock*

After the PROCEED light comes on, the keyboard is unlocked. If it is necessary to change the PUB table with ADD or DEL commands it should be done at this time. The system will not be ready to run until the SET command is given. The normal form of the SET command is:

SET DATE = mm/dd/yy, CLOCK = hh/mm/ss

This indicates the date in the form of month, day, and year, and the time in the form of hour, minute, and second. If there is no interval timer on the machine the CLOCK section would be omitted.

At this time two messages should print out on the console typewriter:

0I20I DOS IPL COMPLETE
1I00A READY FOR COMMUNICATIONS

This tells the operator that job control is now ready to handle jobs. The system will enter a wait state and the PROCEED light on the console typewriter will come on.

The input device should be readied and EOB keyed in and the jobs will begin processing.

Job Run Procedure

The first message that prints out on the console typewriter when jobs begin processing is:

BG // JOB NAME (name specifies the place the job name will appear)

In the process of running the jobs, the operator will be loading and readying input and output devices, responding to messages, and setting up jobs which are scheduled to be run.

The normal setup of a job run, as far as I/O devices are concerned, may vary. The device required for the next job may be in use during the job that is presently running. To maintain maximum efficiency, the operator should set up as many devices as possible for the next job during the present job run.

When a job is completed, the console typewriter will print out a message that states the job is done. The message will be:

BG EOJ Name
BG 1C00A ATTN CUU (address of system reader)

At this time the PROCEED light will come on. This indicates that the computer is waiting to process another job. The operator should begin the final setup of all devices necessary for the job to be run. The setup information is on the run sheet supplied by the programmer. Remove, record, and store all pertinent data from any previous run.

Figure 6-1 illustrates various job control statements to further familiarize the operator with the manner and sequence of assignment statements in a job control deck.

6.4 OTHER GOOD OPERATOR PRACTICES

This section will deal with a few of the fundamental aids at the operator's disposal to facilitate good operations. The proper use of these programs will supplement the environment of the installation.

Magnetic Tape Control

The magnetic tape control (MTC) is a command that may be used by the operator to initiate magnetic tape control operations on 2400 series magnetic tape units. The MTC can be issued only between job steps. The form of the MTC command is:

$$\text{MTC opcode, } \left\{\begin{array}{c} \text{X `CUU'} \\ \text{or} \\ \text{SYSXXX} \end{array}\right\} \text{ [,nn]}$$

The opcode can be: BSF (backspace file)
BSR (backspace record)
ERG (erase gap)
FSF (forward space file)
FSR (forward space record)
REW (rewind)
RUN (rewind and unload)
WTM (write tape mark)

X 'CUU' = the address of the tape drive upon which the operation is to be performed.

SYSXXX = any logical unit assigned to tape.

nn = decimal number (01–99)

Assume the operator wished to place 12 tape marks on the tape mounted on drive 181 which is assigned to SYS013; he could enter into the console typewriter either of the following MTC commands:

MTC WTM, X '181', 12
MTC WTM, SYS013, 12

If he wished to forward space three files on the same drive the command would be:

MTC FSF, X '181', 3
MTC FSF, SYS013, 3

```
                 SAMPLE JOB CONTROL STATEMENTS
// JOB SEQ DISK
// LOG
// ASSGN SYS005,X'191'
// OPTION DUMP
// DLBL DISKOUT,'SEQUENTIAL DISK FILE.',68/001,SD
// EXTENT SYS005,111111,1,1,1600,1
// EXTENT SYS005,111111,1,2,1601,9
// DLBL DISKIN,'SEQUENTIAL DISK FILE.',68/001,SD
// EXTENT SYS005,111111,1,1,1600,1
// EXTENT SYS005,111111,1,2,1601,9
// EXEC SEQDISK
         DATA CARDS
         DATA CARDS
         DATA CARDS
         DATA CARDS
/*
/&

// JOB LOAD A DIRECT ACCESS FILE
// UPSI 0
// UPSI 1
// ASSGN SYS004,X'191'
// DLBL DISK,'DA FILE,LOAD,ADD OR PROCESS',68/001,DA
// EXTENT SYS004 111111,1,0,1700,99
// EXEC LOADDA
         DATA CARDS
         DATA CARDS
         DATA CARDS
         DATA CARDS
/*
/&

// JOB INDEX SEQUENTIAL TO LOAD A FILE.
// ASSGN SYS005,X'191'
// DLBL LOAD,'INDEX SEQUENTIAL EXAMPLE FILE',68/001,ISC
// EXTENT SYS005,111111,4,1,1520,9  CYLINDER EXTENT
// EXTENT SYS005,111111,1,2,1530,80 PRIME DATA EXTENT
// EXEC LOADIS
         DATA CARDS
         DATA CARDS
         DATA CARDS
         DATA CARDS
/*
/&
```

Figure 6-1. Sample Job Streams

```
// JOB RETRIEVE THE PREVIOUSLY LOADED INDEXED FILE
// ASSGN SYS005,X'191'
// DLBL DISK,'INDEX SEQUENTIAL EXAMPLE FILE',68/001,ISE
// EXTENT SYS005,111111,4,1,1520,9  CYLINDER EXTENT
// EXTENT SYS005,111111,1,2,1530,80 PRIME DATA EXTENT
// UPSI 1
// UPSI 01
// UPSI 00
// EXEC ISADRTR
/*
/&

// JOB LOAD AND RETRIEVE A TAPE FILE
// ASSGN SYS004,X'180',X'C0'
// ASSGN SYS004,X'182',ALT
// ASSGN SYS007,X'184',X'C0'
// ASSGN SYS007,X'183',ALT
// ASSGN SYS009,X'00E'
// ASSGN SYS008,X'00C'
// OPTION DUMP
// TLBL SYS004,'SAMPLE TAPE LABEL',68/300,111111
// TLBL SYS007,'SECOND TAPE LABEL',68/300,111111
// EXEC TAPELDRT
         DATA CARDS
         DATA CARDS
         DATA CARDS
         DATA CARDS
/*
/&

// JOB SORT USING SORT PROGRAM
// ASSGN SYS000,X'190'
// DLBL SORTCKP,'SORT CHECKPOINT',65/207,SD
// EXTENT SYS000,111111,,,1900,80
// VOL SYS006,SORTIN1
//       DLAB  'DISK SORT INPUT FILE                     1111111'    X
               0001,66123,69123,'THIS IS FUN  ',SD
//       XTENT 1,0,000137000,000139005,'111111',SYS002
// DLBL SORTWK1,'DISK WORK FILERK FILE                 ',65/207,SD
// EXTENT SYS003,111111,,,1600,50
// VOL SYS005,SORTOUT
// DLAB '16 K DOS OUTPUT FILE                     1111111',       C
               0001,66280,66280,'0000000000000',SD
// XTENT 1,001,000180000,000185000,'111111',SYS001
// EXEC SORT
 SORT FIELDS=(1,9,A),FORMAT=ZD,SIZE=1000,WORK=1,CHKPT
 RECORD TYPE=F,LENGTH=(80,80,80)
 INPFIL BLKSIZE=(800)
 OUTFIL BLKSIZE=80
 OPTION STORAGE=11000
 END
/*
/&
```

Figure 6-1. (Continued)

```
// JOB DOS UTILITY DISK TO PRINT
// ASSGN SYS004,X'191'
// ASSGN SYS005,X'00E'
// UPSI 00000000
// DLBL UIN,'DISK FILE',67/365
// EXTENT SYS004,333333,1,0,01220,0020
// EXEC DKPR
// UDP TL,FF,A=(80,400),B=(132),S1
// END
/*
/&
// JOB UTILITY TAPE TO PRINT
// ASSGN SYS004,X'181'
// ASSGN SYS005,X'00E'
// UPSI 0010         STANDARD LABEL ON TAPE
// TLBL UIN,'SECOND TAPE LABEL',68/300,111111
// EXEC TPPR
// UPT TLF,FV,A=(37,98),B=(40,132),PN,OC,S2
// FS 1,37,1/CV
// END
/*
/&
```

Figure 6-1. (Continued)

The operator will find the MTC command helpful in many situations where it is necessary to perform magnetic tape operations.

Utility Programs

The function of these utility programs is to perform certain tasks that supplement the operating system environment. All of the utilities are in card deck form and must be loaded by the reader. Each time a utility program is run, the operator must IPL from the address of the reader into which it is loaded.

DEBE

DEBE is an IBM-supplied utility program, either on cards or on tape. The DEBE program is used to do many jobs, such as card-to-tape, card-to-card, tape-to-punch, tape-to-print, tape dumps, etc., that an operator might be required to run.

Storage Print

The storage print utility is also referred to as a core dump, EBCDIC dump, or BPS dump, but all these names refer to the same program.

This program will print, in EBCDIC, the contents of all the storage registers and all of core storage.

SEREP (System Error Report)

Occasionally, while running a job a message will print out SEREP on the console typewriter to load. IPL'ing the SEREP deck will cause an error report to be printed out. If the operator is requested to run SEREP, he should do so and give the output to the customer engineer.

UCS (Universal Character Set) Buffer Load

Any time a print chain is changed or the system has just been powered up, the UCS program should be loaded. It must be loaded to ensure that proper characters are printed for printer output. Be sure to load the UCS buffer program that goes with the print chain in the printer.

OS (Operating System) Dump Restore

It might be necessary at times to copy information from disk to tape, disk or cards; these operations can be performed with the OS Dump Restore utility.

Unlike the other utility programs, on this one the operator will be required to punch some control cards to run the OS Dump Restore. Figure 6-2 gives examples of decks for both the dump and restore control cards.

```
DUMP      JOB
          MSG      TODEV = 1052, TOADDR = 01F
          DUMP     FROMDEV = 2311, FROMADDR = 190,
                   TODEV = 2400, TOADDR = 180
          END
```

```
RESTORE   JOB
          MSG      TODEV = 1052, TOADDR = 01F
          RESTORE  FROMDEV = 2400, FROMADDR = 180,
                   TODEV = 2311, TOADDR = 190,
                   VOLID = SYSRES
          END
```

Figure 6-2. OS Dump Restore

The control cards are placed behind the utility program and loaded into the reader. After the reader is IPL'ed, all but the last four cards will be read and the system will drop into a wait state.

At this point press the REQUEST key twice and a message will print out asking the operator to define the input device. The reply to this message should be:

$$\text{INPUT} - \begin{Bmatrix} 2540 \\ \text{or} \\ 1442 \end{Bmatrix}, \text{CUU}$$

After entering this reply the utility should begin. If there were any errors in the control cards, the card in error would be flagged. Do not disturb the CPU while this job is running.

6.5 DOS OPERATOR TRAINING SYSTEM EXERCISE

Objective

The System Exercise is intended to give a new operator a controlled experience in working with the System/360 Disk Operating System (DOS).

Materials

The following information should be available when running the System Exercise:

1. IBM System/360 Disk Operating System Operating Guide (C24-5022).
2. IBM System/360 Model 30 Operating Guide (A24-3373).
3. System Exercise card deck (V25-6465-2).
4. Printed listing of System Exercise card deck.
5. IBM System/360 Reference Data DOS Job Control Language Operator Communication Reference Card (X20-1770).

Description

This card deck contains two programs that are assembled and cataloged in the core image library. Because the programs are retained, the exercise can be interrupted at any point after card number 338 and then continued at the operator's convenience. When the entire exercise is completed, the control cards for deleting the programs are processed.

The total run time of this exercise is approximately one hour. It is suggested that the operator be allowed to repeat the exercise.

Operating Considerations

1. Review IPL procedure.
2. Review procedures for readying I/O devices.
3. Your system residence pack must be able to execute Assembler and Linkage Editor programs.
4. The special instructions on the I/O Setup Sheet (Figure 71 in C20-8093) should be reviewed before going on the system. Note from the I/O Setup Sheet the logical units required and the types of physical devices that must be assigned to them. If the

necessary assignments are not present, you may enter them as ASSGN commands when the system pauses because of the // PAUSE statement (card 22 of deck).

5. A person familiar with the system and with the job stream and I/O Setup Sheet for this exercise should be available to provide guidance while running the System Exercise. The lead operator or systems programmer must note the following considerations before guiding the student through the exercise:

– If your system has no timer, discard all cards punched TIME in columns 73–76. If your system has a timer, discard the card punched NOTM in columns 73–76.

– The cards that are punched MULT in columns 73–76 should be removed if no multiprogramming exercises are to be done.

– Cards that are punched SNGL in columns 73–76 should be removed if multiprogramming is being done.

– The card that is punched ALTR in columns 73–76 should be adjusted if no multiprogramming exercises are to be done. The +0 should be changed to an S.

– The card that is punched EROR in columns 73–76 should be inserted in a keypunch and a character T over-punched on the / in column 3 to create an error card.

– The tape exercise cards are punched TAPE in columns 73–76. They should be removed if no tape drives are on your system.

– Some cards in the deck should have blank cards placed behind them to simulate data files. These cards are punched PLACE ABOUT 250 CARDS AFTER THIS CARD. This comment is punched in card numbers 379, 393, 403, 428, 513, 528, and 553.

6. Set margin stops on console typewriter for maximum width line.

```
SET DATE=06/02/69 THIS CARD COMPLETES THE IPL PROCEDURE.                  NOTH0001
SET DATE=06/02/69,CLOCK=08/30/00 THIS CARD COMPLETES THE IPL PROCEDURE. TIME0002
* DOS OPERATOR'S EXERCISE PROBLEM DECK, IBM FORM NUMBER V25-6465-2.           0003
ALLOC F1=0K,F2=0K YOU WILL MANUALLY ALLOCATE PARTITION SPACE LATER.       MULT0004
*                                                                             0005
* WHEN JOB CONTROL IS THE ACTIVE PROGRAM YOU MAY ENTER JOB CONTROL            0006
* STATEMENTS AND/OR JOB CONTROL COMMANDS. THE // PAUSE STATEMENT IS           0007
* USED TO UNLOCK THE KEYBOARD. EACH STATEMENT OR COMMAND SHOULD BE            0008
* FOLLOWED BY AN ALTERNATE CODE 5 (EOB).                                      0009
* AN EXTRA EOB IS REQUIRED TO TERMINATE COMMUNICATION                         0010
* IF YOU MAKE A MISTAKE WHILE YOU ARE TYPING A MESSAGE,                       0011
* YOU MAY USE THE 'MESSAGE CANCEL KEY'. (IT IS AN ALTERNATE CODE KEY)         0012
* YOU MAY THEN RETYPE YOUR MESSAGE.                                           0013
*                                                                             0014
* BE SURE TO READ ALL INSTRUCTIONS AND ERROR MESSAGES CAREFULLY.              0015
* ERRORS ARE INTENTIONALLY INTRODUCED TO GIVE YOU EXPERIENCE IN THE USE       0016
* OF YOUR OPERATOR GUIDE.                                                     0017
*                                                                             0018
* TURN THE TIMER ON.                                                      TIME0019
* YOU WILL NOW USE THE  SET  COMMAND TO ENTER THE CORRECT DATE                0020
* AND CLOCK TIME.                                                         TIME0021
// PAUSE                                                                      0022
*                                                                             0023
*                                                                             0024
// JOB TIMER TEST                                                         TIME0025
/& &         DOES THE TIMER INDICATE THE CORRECT TIME OF DAY?             TIME0026
*                                                                             0027
*                                                                             0028
* THE DATE WILL PRINT ON THE ASSEMBLY LISTING THAT WILL FOLLOW.               0029
*                                                                             0030
*                                                                             0031
// JOB CREATE A PROGRAM FOR CONSOLE EXERCISES.                                0032
* YOU WILL SEE A TYPICAL PROGRAM ASSEMBLY PROCESS (SIMILAR TO A               0033
* PROGRAM COMPILATION PROCESS).                                               0034
// OPTION CATAL,NODECK,XREF,LISTX,ERRS,DUMP                                    0035
// EXEC ASSEMBLY                                                              0036
         ISEQ  77,80                                                          0037
* LINKAGE EDITOR CONTROL CARDS. ***************************************       0038
 PUNCH ' ACTION NOAUTO '                                                      0039
 PUNCH ' ACTION MAP '                                                         0040
 PUNCH ' PHASE EXERCISE,S OPERATOR EXERCISE PROGRAM'                          0041
 PUNCH ' INCLUDE  ,(CSECT1)  MAIN PROGRAM '                                   0042
 PUNCH ' INCLUDE  ,(IJ2L0036)  LOGIC FOR THE CONSOLE TYPEWRITER. '            0043
 PUNCH ' INCLUDE IJFFZZZZ MTMOD FIXUNB,FORWARD,NO-CKPTREC,NOWORKA '           0044
 PUNCH ' INCLUDE IJGFOZZZ SDMOD NO-HOLD,OUTPUT,NO-ERROPT,NO-TRUNCS,NOC'       0045
 PUNCH ' PHASE    PUNCH,+0 SELF RELOCATING PROGRAM '                      ALTR0046
 PUNCH ' INCLUDE ,(FGPCSECT) MAIN PROGRAM '                                   0047
 PUNCH ' INCLUDE IJCFZOI4 CDMOD FIXUNB,NO-CTL,OUTPUT,IOAREA2,2540 '           0048
CSECT1   START 0                                                              0049
         PRINT ON,GEN,DATA                                                    0050
EXER     TITLE 'OPERATOR EXERCISE PROGRAM                               '     0051
               '                                                              0052
* EQUATES FOR THE PROGRAM MASK BITS.. *******************************         0053
SIGNIFCN EQU   1 SIGNIFICANCE, FLOATING POINT FEATURE.                        0054
EXPUNFLO EQU   2 EXPONENT UNDERFLOW, FLOATING POINT FEATURE.                  0055
DECOVFLO EQU   4 DECIMAL OVERFLOW, DECIMAL FEATURE.                           0056
PAGE   1
```

```
BINOVFLO EQU   8 BINARY OVERFLOW, STANDARD FIXED POINT INSTRUCTIONS.
         SPACE 3
* EQUATES FOR UPSI BIT POSITIONS. BIT LABELS HAVE A 'B' PREFIX. *******
BIT0     EQU   128
BIT1     EQU   64
BIT2     EQU   32
BIT3     EQU   16
BIT4     EQU   8
BIT5     EQU   4
BIT6     EQU   2
BIT7     EQU   1
         SPACE 3
* REGISTER ASSIGNMENTS ARE LABELS WITH AN 'R' PREFIX. ****************
RMVCBASE EQU   1 USE REG. 1 FOR A TEMPORY BASE REGISTER
RLENGTH  EQU   2 IS THE  RECSIZE=(2)  FOR VARIOUS DTF'S.
RCONSOLE EQU   3 POINTS TO THE DTFCN FOR THIS FILE.
RDISKFIL EQU   4 POINTS TO THE DTFSD FOR DISK LABEL EXERCISE.
RCARDFIL EQU   4 POINTER TO THE DTFCD FOR THE PUNCH PROGRAM
RBIGLOOP EQU   5 CONTAINS LOOP CONTROL COUNTER.
RTAPEOUT EQU   6  POINTS TO THE DTFMT FOR THE TAPE OUTPUT FILE.
RTAPEIN  EQU   7  POINTS TO THE DTFMT FOR THE TAPE INPUT FILE.
RTAPEIO  EQU   11 CONTAINS THE POINTER TO THE CURRENT I/O AREA FOR TAPE
RDISKIO  EQU   11
RELOFCTR EQU   12 CONTAINS THE PROGRAM BASE (AND THE RELOCATION FACTOR)
         SPACE 3
COMMUREG DSECT
ENTIRREG DS    0CL46 COMMUNICATION REGION DESCRIPTION.
DATE     DS    CL8  0-7
         DS    CL2  8-9
         DS    CL2  10-11
USERAREA DS    CL11 12-22
UPSI     DS    CL1  23-23
JOBNAME  DS    CL8  24-31
         DS    CL4  32-35
         DS    CL4  36-39
         DS    CL4  40-43
         DS    CL4  44-45
         EJECT
CSECT1   CSECT
         USING *,RELOFCTR REGISTER WILL CONTAIN THE RELOCATION FACTOR.
         BALR  RELOFCTR,0       ADJUST BASE REGISTER-
         BCTR  RELOFCTR,0       -TO AGREE WITH-
         BCTR  RELOFCTR,0       -THE USING STATEMENT.
         LA    RELOFCTR,0(,RELOFCTR)    CLEAR THE HIGH ORDER BYTE
         LA    11,BINOVFLO+DECOVFLO+EXPUNFLO+SIGNIFCN
         SPM   11   SET PROGRAM MASK
         LA    RCONSOLE,CONSOLE
         OPENR (RCONSOLE)
         COMRG
         USING COMMUREG,1
         TM    UPSI,BIT0
         BO    EXERCIS0   CANCEL MACRO PROGRAM
         TM    UPSI,BIT1
         BO    EXERCIS1  LOOP, OPERATOR MUST CANCEL TO TERMINATE
         TM    UPSI,BIT2
         BO    EXERCIS2  PROGRAM CHECK WILL BE FORCED
 PAGE  2
```

```
         TM    UPSI,BIT3
         BO    EXERCIS3  PROGRAM WILL ASK THE OPERATOR FOR HIS NAME.
         TM    UPSI,BIT4
         BO    EXERCIS4  OPERATOR WILL USE THE INTERRUPT KEY
         TM    UPSI,BIT5
         BO    EXERCIS5 ERROR
         TM    UPSI,BIT6
         BO    EXERCIS6  DISK FILE LABEL EXERCISE
         TM    UPSI,BIT7
         BO    EXERCIS7  TAPE FILE LABEL EXERCISE
EXERCIS5 LR    1,RCONSOLE
         LA    0,UPSIMESS
         LA    RLENGTH,L'UPSIMESS
         PUT   (1),(0)
         EOJ
UPSIMESS DC    C'YOU DID NOT HAVE A VALID UPSI CARD '
         SPACE 3
EXERCIS0 LR    1,RCONSOLE          POINT TO DTF
         LA    0,CANCELMS          POINT TO MESSAGE
         LA    RLENGTH,L'CANCELMS  INDICATE MESSAGE LENGTH
         PUT   (1),(0)
         LR    1,RCONSOLE
         LA    0,NOTEASTR    POINT TO MESSAGE
         LA    RLENGTH,L'NOTEASTR  INDICATE MESSAGE LENGTH.
         PUT   (1),(0)
         CANCEL
CANCELMS DC    C'UPSI BIT 0 IS ON, THIS PROGRAM WILL CANCEL'
NCTEASTR DC    C'DID YOU NOTICE THAT THERE IS NO  *  ON A PROGRAM GENER-
               ATED MESSAGE?'
         SPACE 3
EXERCIS1 LR    1,RCONSOLE          POINT TO DTF
         LA    0,LOOPMSG
         LA    RLENGTH,L'LOOPMSG
         PUT   (1),(0)
         L     RBIGLOOP,=F'2147483647'
         BCT   RBIGLOOP,*          LOOP ON MYSELF
         EOJ
* OBTAIN THE ATTENTION ROUTINE AND CANCEL THIS PROGRAM .
LOOPMSG  DC    C'NOTE THE SYSTEM LIGHT AND OTHER LIGHTS. A TIGHT LOOP IS
               IN PROGRESS'
         SPACE 3
EXERCIS2 LR    1,RCONSOLE
         LA    0,PRGCKMSG     PROGRAM CHECK MESSAGE
         LA    RLENGTH,L'PRGCKMSG
         PUT   (1),(0)
         LA    1,1
         ST    1,0(RELOFCTR,1)
PRGCKMSG DC    C'THIS PROGRAM WILL ATTEMPT AN INCORRECT OPERATION'
         SPACE 3
EXERCIS3 LR    1,RCONSOLE
         LA    0,QUESTION
         LA    RLENGTH,L'QUESTION
         PUT   (1),(0)
         LR    1,RCONSOLE
         LA    0,ANSWER
         GET   (1),(0)
PAGE   3
```

```
* COMPOSE RETURN MESSAGE.**********************************************
         LA    RMVCBASE,ANSWER(RLENGTH) ADD LENGTH TO BASE-DISPLACEMENT
         MVC   0(L'COMPLIMN,RMVCBASE),COMPLIMN
* MESSAGE LENGTH = L'RETURN+L'COMPLIMN+L'OPERATORSNAME  ***************
         LA    RLENGTH,L'RETURN+L'COMPLIMN(0,RLENGTH)
* SET UP THE 'PUT' TO THE CONSOLE.*************************************
         LR    1,RCONSOLE
         LA    0,RETURN
         PUT   (1),(0)
         EOJ
QUESTION DC    C'TYPE YOUR COMPLETE NAME (NO MORE THAN 34 CHARACTERS).'
RETURN   DC    C'THIS COMPUTER SYSTEM IS RUN BY '
ANSWER   DC    120C' '
COMPLIMN DC    C', THE GREATEST CONSOLE OPERATOR IN THE WORLD '
         SPACE 3
EXERCIS4 LR    1,RCONSOLE
         LA    0,WARNING
         LA    RLENGTH,L'WARNING
         PUT   (1),(0)
         LA    0,OCROUTIN
         LA    1,OCSAVE
         STXIT OC,(0),(1)
         L     RBIGLOOP,=F'2147483647'
         BCT   RBIGLOOP,*
         EOJ
OCSAVE   DC    9D'0'
OCROUTIN LR    1,RCONSOLE
         LA    0,CONFIRM
         LA    RLENGTH,L'CONFIRM
         PUT   (1),(0)
         LR    1,RCONSOLE
         LA    0,CONVERSE
         LA    RLENGTH,L'CONVERSE
         PUT   (1),(0)
         EXIT  OC
WARNING  DC    C'YOUR DOS SUPERVISOR MAY IGNORE THE INTERRUPT KEY.'
CONFIRM  DC    C'YOUR DOS SUPERVISOR ALLOWS THE INTERRUPT KEY FUNCTION -
               ''OPERATOR COMMUNICATION'''
CONVERSE DC    C'PLEASE OBTAIN THE ATTENTION ROUTINE, TYPE  PAUSE  THEN-
                CANCEL THIS PROGRAM.'
         SPACE 3
EXERCIS6 LR    1,RCONSOLE
         LA    0,RESPONSI BILITY MESSAGE
         LA    RLENGTH,L'RESPONSI BILITY MESSAGE.
         PUT   (1),(0)
         LR    1,RCONSOLE
         LA    0,BILITY
         LA    RLENGTH,L'BILITY MESSAGE
         PUT   (1),(0)
         LR    1,RCONSOLE
         LA    0,OPENTRY
         LA    RLENGTH,L'OPENTRY
         PUT   (1),(0)
         LA    RDISKFIL,DISKFIL
         OPENR (RDISKFIL)
         LA    RBIGLOOP,475
 PAGE  4
```

```
PUTDISK  PUT    (RDISKFIL)
         BCT    RBIGLOOP,PUTDISK
         CLOSER (RDISKFIL)
         EOJ
RESPONSI DC     C'YOU ARE RESPONSIBLE FOR VITAL DATA FILES. YOU MUST N-E-
                -V-E-R DELETE A FILE'
BILITY   DC     C'UNLESS YOU HAVE PERMISSION. ALWAYS USE LISTVTOC WHEN YOU
                NEED TO IDENTIFY A PACK'
OPENTRY  DC     C'THE PROGRAM WILL NOW TRY TO USE A DISK AREA FOR AN OUT
                PUT FILE'
         SPACE 3
EXERCIS7 LR     1,RCONSOLE
         LA     0,CHECKTAP
         LA     RLENGTH,L'CHECKTAP
         PUT    (1),(0)
         LA     RTAPEOUT,TAPEOUT
         OPENR  (RTAPEOUT)
         LA     RBIGLOOP,20
PUT      PUT    (RTAPEOUT)
         BCT    RBIGLOOP,PUT
         CLOSER (RTAPEOUT)
         LA     RTAPEIN,TAPEIN
         OPENR  (RTAPEIN)
         LA     RBIGLOOP,25
GET      GET    (RTAPEIN)
         BCT    RBIGLOOP,GET
EOFTAPE  CLOSER (RTAPEIN)
         EOJ
CHECKTAP DC     C'THE PROGRAM WILL TEST FOR STANDARD LABELS AND PRINT THE
                EM ON THE CONSOLE.'
         LTORG
CONSOLE  DTFCN                                                         X
                DEVADDR=SYSLOG,                                        X
                TYPEFLE=INPUT,                                         X
                RECFORM=UNDEF,                                         X
                BLKSIZE=120,                                           X
                IOAREA1=CNSLEOUT,                                      X
                WORKA=YES,                                             X
                RECSIZE=(2) MESSAGE SIZE, RLENGTH
CNSLEOUT DC     CL120' '
DISKFIL  DTFSD                                                         X
                DEVADDR=SYS005,                                        X
                DEVICE=2311,                                           X
                TYPEFLE=OUTPUT,                                        X
                RECFORM=FIXUNB,                                        X
                BLKSIZE=15,                                            X
                IOAREA1=DISKOUT1,                                      X
                IOAREA2=DISKOUT2,                                      X
                IOREG=(11),                                            X
                VERIFY=YES
DISKOUT1 DC     C'DUMMY DISK RCRD'
DISKOUT2 DC     C'DUMMY DISK RCRD'
TAPEOUT  DTFMT                                                         X
                DEVADDR=SYS001,                                        X
                TYPEFLE=OUTPUT,                                        X
                RECFORM=FIXUNB,                                        X
PAGE   5
```

```
                 BLKSIZE=15,                                                   X
                 IOAREA1=TAPEOUT1,                                             X
                 IOAREA2=TAPEOUT2,                                             X
                 FILABL=STD,                                                   X
                 HDRINFO=YES,                                                  X
                 IOREG=(11) THIS REGISTER IS NAMED RTAPFIO.
TAPEOUT1 DC      C'DUMMY TAPE RCRD'
TAPEOUT2 DC      C'DUMMY TAPE RCRD'
TAPEIN   DTFMT                                                                 X
                 DEVADDR=SYS001,                                               X
                 TYPEFLE=INPUT,                                                X
                 READ=FORWARD,                                                 X
                 EOFADDR=EOFTAPE,                                              X
                 RECFORM=FIXUNB,                                               X
                 BLKSIZE=15,                                                   X
                 IOAREA1=TAPEIN1,                                              X
                 IOAREA2=TAPEIN2,                                              X
                 FILABL=STD,                                                   X
                 HDRINFO=YES,                                                  X
                 IOREG=(11) THIS REGISTER IS NAMED RTAPEIO.
TAPEIN1  DC      CL15' '
TAPEIN2  DC      CL15' '
         LTORG
         EJECT
FGPCSECT CSECT
         USING *,RELOFCTR REGISTER WILL CONTAIN THE RELOCATION FACTOR.
         BALR    RELOFCTR,0      ADJUST BASE REGISTER-
         BCTR    RELOFCTR,0      -TO AGREE WITH-
         BCTR    RELOFCTR,0      -THE USING STATEMENT.
         LA      RELOFCTR,0(,RELOFCTR)     CLEAR THE HIGH ORDER BYTE
         LA      11,BINOVFLO+DECOVFLO+EXPUNFLO+SIGNIFCN
         SPM     11    SET PROGRAM MASK
         LA      RCARDFIL,CARDFIL
         OPENR   (RCARDFIL)
         LA      RBIGLOOP,3000
PUTCARDS PUT     (RCARDFIL)
         BCT     RBIGLOOP,PUTCARDS
         CLOSER  (RCARDFIL)
         EOJ
         LTORG
CARDFIL  DTFCD                                                                 X
                 DEVADDR=SYS001,                                               X
                 DEVICE=2540,                                                  X
                 RECFORM=FIXUNB,                                               X
                 BLKSIZE=80,                                                   X
                 IOAREA1=PUNCH1,                                               X
                 IOAREA2=PUNCH2,                                               X
                 IOREG=(11),                                                   X
                 SSELECT=2,                                                    X
                 CRDERR=RETRY,                                                 X
                 TYPEFLE=OUTPUT
PUNCH1   DC      CL80' '
PUNCH2   DC      CL80' '
         LTORG
         END
/* *
PAGE  6
```

```
// EXEC LNKEDT
/& &
*
*
// JOB CONSOLE LOGGING CONCEPTS.
* NOTE THAT THE JOB CARD (// JOB NAME), THE PAUSE CARD (// PAUSE ),
* THE END OF JOB CARD (/& )AND THE COMMENTS CARD (* ) WILL ALWAYS BE
* LOGGED ON SYSLOG.
// PAUSE  TYPE THE  LOG  COMMAND TO PRINT ALL STATEMENTS ON SYSLOG.
/& &
*
*
// JOB ERROR HANDLING EXERCISE.
* AN ERROR MESSAGE WILL PRINT IF A CONTROL CARD HAS AN ERROR.
* IN THIS EXERCISE YOU WILL BE ASKED TO CORRECT AN ERROR IN A CONTROL
* CARD. A-L-W-A-Y-S CONSULT YOUR OPERATOR'S GUIDE FOR APPROPRIATE
* ACTION.
*
* THE '04' IN THE ERROR MESSAGE '1S04D' MEANS THAT THE 4TH ELEMENT OF
* THE FOLLOWING STATEMENT HAS AN ERROR. THE  //  IS THE 1ST ELEMENT.
// ASSGN SYS009,X190 ASSIGN LOGICAL UNIT 9 TO A PHYSICAL DISK DRIVE .
*
*
* THE NEXT CARD TO FALL IN THE READ STACKER IS MULTIPUNCHED IN COLUMN 3
* TO CORRECT THE BAD CARD, KEY PUNCH A  T  IN COLUMN 3 AND RESUBMIT IT.
* THIS CARD WAS LACED IN COLUMN 3, IT IS NOW CORRECT.                 ERGR
* SAVE THE ERROR CARD FOR A LATER EXERCISE.
/& &
*
*
// JOB ONE. THIS PROGRAM ROUTINE WILL DEMONSTRATE A PROGRAMMED CANCEL.
// UPSI 10000000 BIT 0, CANCEL MESSAGE DEMONSTRATION.
*
* THE  UPSI  CARD IS USED TO SELECT A PARTICULAR FUNCTION (ROUTINE) IN
* ANY GIVEN PROGRAM, IN THIS CASE THE PROGRAM IS CALLED  EXERCISE  .
*
* AFTER THE CANCELLATION
*  OTHER CARDS MAY STILL BE IN THE CARD READER.
*  YOU NEED NOT REMOVE THEM. THE SYSTEM WILL AUTOMATICALLY PASS
*  ALL THESE CARDS TO CLEAR THE READER. AS A MATTER OF FACT THE SYSTEM
*  WILL PASS AS MANY CARDS AS NECESSARY UNTIL IT ENCOUNTERS A /& CARD.
// EXEC EXERCISE ONE.
* PLACE ABOUT 250 CARDS AFTER THIS CARD. ******************************
/& &
*
*
// JOB TWO. THIS EXERCISE WILL ALLOW YOU TO CANCEL AN ERROR PROGRAM.
// OPTION DUMP (THIS SETS THE CORE MEMORY DUMP OPTION ON).
// UPSI 01000000 BIT 1, PROCESS LOOP, THE OPERATOR MUST CANCEL THE PROG
*  NOTE THAT UPON CANCELLATION OF THE PROGRAM A MESSAGE IS PRINTED
*  ON THE CONSOLE TYPEWRITER.
*  IN ADDITION TO THE CONSOLE TYPEWRITER MESSAGE A CORE DUMP IS
*  PRINTED ON THE PRINTER. THE MESSAGE ON THE CONSOLE TYPEWRITER ALSO
*  APPEARS ON THE CORE DUMP. THIS IS TO DOCUMENT ALL ERRORS
*  AUTOMATICALLY FOR THE PROGRAMMER.
// EXEC EXERCISE TWO.
PAGE  7
```

```
* PLACE ABOUT 250 CARDS AFTER THIS CARD. *******************************     0393
/& &                                                                         0394
*                                                                            0395
*                                                                            0396
// JOB THREE. THIS PROGRAM ROUTINE WILL DEMONSTRATE WHAT HAPPENS WHEN        0397
* A PROGRAM CEVELOP S AN ERROR. (THIS USUALLY HAPPENS IN UNTESTED            0398
* PROGRAMS)                                                                  0399
// OPTION DUMP (THIS SETS THE CORE MEMORY DUMP OPTION ON).                   0400
// UPSI 00100000 BIT 2, A PROGRAM CHECK WILL BE FORCED CAUSING A DUMP.       0401
// EXEC EXERCISE THREE.                                                      0402
* PLACE ABOUT 250 CARDS AFTER THIS CARD. *******************************     0403
/& &                                                                         0404
*                                                                            0405
*                                                                            0406
// JOB FOUR. THIS EXERCISE DEMONSTRATES HOW THE PROGRAM CAN REQUEST          0407
* OPERATOR ACTION.                                                           0408
// UPSI 00010000 BIT 3, THE OPERATOR WILL BE ASKED TO ENTER HIS NAME.        0409
// EXEC EXERCISE FOUR.                                                       0410
/& &                                                                         0411
*                                                                            0412
*                                                                            0413
// JOB FIVE. INTERRUPT KEY EXERCISE.                                         0414
// OPTION NODUMP                                                             0415
// UPSI 00001000 BIT 4, YOU WILL BE ASKED TO USE THE INTERRUPT KEY.          0416
*                                                                            0417
*                                                                            0418
* IF YOUR INTERRUPT KEY DOES NOT CAUSE A MESSAGE TO BE PRINTED ON THE        0419
* CONSOLE, YOU MUST CANCEL THE PROGRAM.                                      0420
*                                                                            0421
* (A PROCESS LOOP WILL BE USED TO SIMULATE A RUNNING PROGRAM)                0422
*                                                                            0423
* IF YOU ISSUE A  PAUSE  BEFORE A  CANCEL  , YOU WILL REGAIN CONTROL OF      0424
* THE SYSTEM. THIS IS CONV ENIENT IF CARDS NEED TO BE REARRANGED ETC.        0425
*                                                                            0426
// EXEC EXERCISE FOUR.                                                       0427
* PLACE ABOUT 250 CARDS AFTER THIS CARD. *******************************     0428
/& &                                                                         0429
*                                                                            0430
*                                                                            0431
// JOB SIX. DISPLAY SYSTEM RESOURCES                                         0432
* IN ORDER TO DETERMINE THE POSSIBILITY OF RUNNING MORE THAN ONE PROGRM MULT0433
*                                                                   MULT0434
// PAUSE TYPE IN THE  MAP  COMMAND.                                 MULT0435
* NOTE THE CURRENT JOB NAME RUNNING IN THE BG,IT CAME FROM THE JOB CARD MULT0436
* NOTE THE FOUR PARTITION NAMES, NOTE THE TIMER CODE 'T',           MULT0437
* AND THE PARTITION THAT OWNS THE TIMER. USE THE ATTENTION ROUTINE TO   MULT0438
* ASSIGN THE TIMER TO ANOTHER PARTITION.                            MULT0439
* WHEN THE BACKGROUND IS STOPPED DEPRESS THE REQUEST KEY ON THE CONSOLE MULT0440
* CHANGE THE TIMER ASSIGNMENT AND ISSUE THE  START BG  COMMAND.     MULT0441
*                                                                   MULT0442
STOP YOU WILL NOTE THAT THE WAIT LIGHT IS ON AND THE KEYBOARD IS LOCKED MULT0443
// PAUSE  YOU SHOULD NOW TRY TO ALLOCATE F1 TO CONTAIN  2K OF STORAGE.  MULT0444
// PAUSE REQUEST A MAP TO VERIFY THAT YOU HAVE MADE THE CORRECT CHANGES MULT0445
* A LISTIO COMMAND WILL USE THE SYSLOG AS AN OUTPUT DEVICE.                  0446
// PAUSE REQUEST. A  LISTIO X'CUU'  FOR THE ADDRESS  01F  .                  0447
*                                                                            0448
PAGE   8
```

```
* THE // LISTIO  STATEMENT SHOULD BE USED IF MANY UNITS ARE TO BE          0449
* PRINTED OUT. THE LIST WILL PRINT ON SYSLST.                              0450
// PAUSE REQUEST A LISTING OF ALL I/O UNITS.                               0451
/& &                                                                       0452
*                                                                          0453
*                                                                          0454
// JOB SEVEN. THIS EXERCISE PROVIDES EXPERIENCE IN STARTING A          MULT0455
* FOREGROUND PROGRAM. YOU WILL NEED A PUNCH UNIT, SO YOU SHOULD        MULT0456
* VERIFY THAT A PUNCH IS AVAILABLE (UNASSIGNED).                       MULT0457
// PAUSE DO A LISTIO FOR YOUR PUNCH ADDRESS, X'CUU'.                   MULT0458
// PAUSE TYPE  ASSGN SYSNNN,UA  FOR EACH STANDING ASSIGNMENT.          MULT0459
// PAUSE DO ONE MORE LISTIO JUST TO BE SURE THAT THE PUNCH IS AVAILABLE MULT0460
*                                                                      MULT0461
* YOU WILL NOW START A FOREGROUND PARTITION. IT WILL PUNCH 3000 CARDS  MULT0462
* ONLY BLANKS WILL BE PUNCHED TO SAVE YOUR CARDS.                      MULT0463
*                                                                      MULT0464
* BE SURE THAT THE PUNCH HOPPER IS LOADED WITH BLANK CARDS. YOU SHOULD MULT0465
* INSERT THE MULTIPUNCHED CARD INTO THE MIDDLE OF THE CARD PUNCH HOPPER MULT0466
* YOU WILL GET A PUNCH CHECK FROM THIS CARD, TYPE  IGNORE  SO THE PUNCH MULT0467
* OPERATION WILL BE RETRIED. THE ERROR CARD WILL BE IN THE NORMAL      MULT0468
* PUNCH POCKET, P1.                                                    MULT0469
*                                                                      MULT0470
*    REQUEST THE ATTENTION ROUTINE AND START F1, ASSGN SYS001,X'00D',  MULT0471
* THEN  EXEC PUNCH , THEN REQUEST THE ATTENTION ROUTINE AND  START BG . MULT0472
*                                                                      MULT0473
STOP (DO NOT FORGET TO START BG AFTER THE FOREGROUND IS RUNNING)       MULT0474
/&          THE FOREGROUND PROGRAM SHOULD BE PUNCHING CARDS NOW.       MULT0475
*                                                                      MULT0476
*                                                                      MULT0477
* BEFORE YOU RESPOND TO ANY CONSOLE MESSAGES, BE SURE THAT YOU KNOW    MULT0478
* WHICH PARTITION THAT YOU ARE COMMUNICATING WITH.                     MULT0479
*                                                                          0480
*                                                                          0481
// JOB SEVEN. THIS EXERCISE WILL PERMIT YOU TO MANUALLY START A PROGRAM SNGL0482
* YOU WILL NEED A PUNCH UNIT SO YOU SHOULD VERIFY THAT A PUNCH IS      SNGL0483
* AVAILABLE (UNASSIGNED).                                              SNGL0484
// PAUSE DO A LISTIO FOR YOUR PUNCH ADDRESS.                           SNGL0485
// PAUSE TYPE  ASSGN SYSNNN,UA  FOR EACH STANDING ASSIGNMENT.          SNGL0486
// PAUSE DO ONE MORE LISTIO JUST TO BE SURE THAT THE PUNCH IS AVAILABLE SNGL0487
*                                                                      SNGL0488
* YOU WILL NOW START A PROGRAM THAT WILL PUNCH CARDS.                  SNGL0489
* IT WILL PUNCH 3000 CARDS WITH BLANKS TO SAVE YOUR CARDS.             SNGL0490
*                                                                      SNGL0491
* INSERT THE MULTIPUNCHED CARD INTO THE MIDDLE OF THE CARD PUNCH HOPPER SNGL0492
* YOU WILL GET A PUNCH CHECK FROM THIS CARD, TYPE  IGNORE  SO THE PUNCH SNGL0493
* OPERATION WILL BE RETRIED. THE ERROR CARD WILL BE IN THE NORMAL      SNGL0494
* PUNCH POCKET, P1.                                                    SNGL0495
*                                                                      SNGL0496
// PAUSE PLEASE ASSIGN SYS001 TO YOUR PUNCH ADDRESS.                   SNGL0497
// PAUSE NOW TYPE  // EXEC PUNCH  .                                    SNGL0498
/& &                                                                   SNGL0499
*                                                                          0500
*                                                                          0501
// JOB EIGHT. THIS ROUTINE DEMONSTRATES A VOLUME MOUNTING MESSAGE.         0502
* NORMALLY YOU WOULD BE REQUIRED TO CHANGE DISK PACKS, BUT YOU SHOULD      0503
* NOT DO THIS SINCE THE REQUIRED DISK PACK WILL NOT BE AVAILABLE TO YOU    0504
PAGE   9
```

```
// OPTION NODUMP
* THE FOLLOWING STATEMENT SHOULD BE RETYPED.
ASSGN SYS005,X'    ' A 'SCRATCH' DISK PACK IS READY AT THIS ADDRESS.
// DLBL DISKFIL,'OPERATORS PRACTICE EXTENT',1 ONE DAY RETENTION CYCLE.
// EXTENT SYS005,$$$$$$,,,1990,10
* USE THE  CANCELV  OPTION.
// UPSI 00000010 BIT 6, DISK FILE VOLUME AND STANDARD LABEL EXERCISES.
// EXEC EXERCISE EIGHT.
* PLACE ABOUT 250 CARDS AFTER THIS CARD. ********************************
/& &
*
*
// JOB NINE. THIS ROUTINE DEMONSTRATES OVERLAPPING DISK AREAS.
*  OVERLAPPING FILES CAUSE DATA TO BE LOST, THE PROGRAMMER WILL HAVE TO
* DEFINE A DIFFERENT DISK AREA WITH A NEW  // EXTENT  CARD.
*
* DISPLAY THE VTOC FOR HIM TO HELP HIM DETERMINE AVAILABLE SPACE.
* USE  DSPLYV  ON THE SYSLST DEVICE, THEN YOU SHOULD TYPE  CANCELV
// OPTION NODUMP
// DLBL DISKFIL,'OPERATORS PRACTICE EXTENT',1 ONE DAY RETENTION CYCLE
// EXTENT SYS005,,,,1,1998 THE ENTIRE PACK (EXCEPT THE IPL-VOL TRACK).
// UPSI 00000010 BIT 6, DISK FILE VOLUME AND STANDARD LABEL EXERCISES.
// EXEC EXERCISE NINE.
* PLACE ABOUT 250 CARDS AFTER THIS CARD. ********************************
/& &
*
*
// PAUSE NOTE THE 2 DIFFERENT FORMATS OF THE VTOC ON THE SYSLST.
*
*
// JOB DISPLAY AN EDITED VOLUME TABLE OF CONTENTS OF ANY DISK PACK.
* THE  LISTVTOC  PROGRAM GIVES THE BEST VTOC PRINTOUT.
// ASSGN SYS005,X'00E' ASSIGN A PRINTER.
// ASSGN SYS004,X'    ' THE SAME DISK ADDRESS THAT IS ASSIGNED TO SYS005
// EXEC LISTVTOC
/& &
*
*
// PAUSE NOTE THE 3RD KIND OF VTOC PRINTOUT.
*
*
// JOB DISK THIS ROUTINE DEMONSTRATES A 'NO LABEL INFORMATION' MESSAGE.
* IF THE  FILE NAME  (ON A DLBL CARD) IS MISSPELLED, AN ERROR IS CAUSED.
// OPTION NODUMP
// DLBL MISPELL,'OPERATORS PRACTICE EXTENT',1 ONE DAY RETENTION CYCLE
// EXTENT SYS005,,,,10,10
// UPSI 00000010 BIT 6, DISK FILE VOLUME AND STANDARD LABEL EXERCISES.
// EXEC EXERCISE
* PLACE ABOUT 250 CARDS AFTER THIS CARD. ********************************
/& &
*
*
// JOB YOURXTNT
// OPTION NODUMP
// UPSI 00000010 BIT 6, DISK FILE VOLUME AND STANDARD LABEL EXERCISES.
*
PAGE 10
```

```
* YOUR DUTIES DO NOT NORMALLY REQUIRE YOU TO DETERMINE WHAT DISK SPACE
* IS AVAILABLE FOR DATA FILES. HOWEVER, IN ORDER TO RUN THE NEXT
* EXERCISE YOU MUST COMPLETE AN EXTENT CARD.
*
* USE THE LISTVTOC PRINTOUT TO LOCATE AN AVAILABLE CYLINDER OF SPACE.
*
* COMPLETE THE  EXTENT  STATEMENT FOR SYS005.
// DLBL DISKFIL,'OPERATORS PRACTICE EXTENT',1 ONE DAY RETENTION CYCLE
// EXTENT SYS
// EXEC EXERCISE
* CONGRATULATIONS                           ---------------- WELL DONE
/& &
*
*
// JOB JOBNAME
* YOU SHOULD NOW PRINTOUT THE VTOC AND INSPECT YOUR NEW EXTENT.
// ASSGN SYS005,X'00E' PRINTER.
* TELL THE SYSTEM WHAT ADDRESS YOUR EXTENT IS ON.
// ASSGN SYS004,X'   ' DISK ADDRESS OF PERMANT SYS005.
// EXEC LISTVTOC
/& &
*
*
// PAUSE  TIME OUT FOR INSPECTION.
*
*
// JOB TAPE EXERCISES
// OPTION NODUMP
* MOUNT A SCRATCH TAPE AND ASSIGN IT TO SYS001. LEAVE THE FILE PROTECT
* RING OUT, AFTER YOU HAVE A CHANCE TO SEE THE MESSAGE, PUT THE RING
* BACK IN.
* THE TAPE SHOULD BE AN UNLABELED TAPE.
* COMPLETE THE FOLLOWING STATEMENT.
// ASSGN SYS001,X'   ',X'C0' TAPE ADDRESS AND TRACK SPECIFICATIONS.
*
* USE THE  MTC  COMMAND FOR THE NEXT EXERCISE. (// MTC WTM,SYS001,06)
// PAUSE TYPE THE COMMAND TO WRITE 6 TAPE MARKS ON THE REEL.
* YOU WILL NEED TO SUPPLY A VOLUME SERIAL NUMBER IN THIS EXERCISE.
* YOU SHOULD OBTAIN THE VOLUME SERIAL NUMBER FROM THE TAPE LIBRARIAN.
*
// TLBL TAPEOUT,'OPERATOR EXERCISE',1,,1,1,1,1
// TLBL  TAPEIN,'OPERATOR EXERCISE'
// UPSI 00000001 BIT 7, TAPE FILE VOLUME AND STANDARD LABEL EXERCISES.
// EXEC EXERCISE
/& &
*
*
// JOB DELETE THE EXERCISE PROGRAM FROM THE CORE IMAGE LIBRARY.
// PAUSE CANCEL THIS JOB IF YOU WANT TO KEEP THE EXERCISE PROGRAM.
// EXEC MAINT
 DELETC EXERCISE,PUNCH
/* *
/& &
// PAUSE  ENTER THE COMMAND TO STOP LOGGING CONTROL CARDS ON SYSLOG
*  THIS IS THE END OF THE EXERCISE. ALTHOUGH YOU HAVE NOT
*  ENCOUNTERED ALL THE POSSIBLE CONDITIONS WHICH MAY OCCUR WHILE
PAGE 11
```

```
*  YOU ARE OPERATING THE SYSTEM, YOU NOW HAVE EXPERIENCED SOME OF THE        0617
*  OPERATING CHARACTERISTICS OF THE DISK OPERATING SYSTEM.                   0618
*  YOUR ABILITY TO RECOGNIZE WHAT IS HAPPENING WITHIN THE SYSTEM             0619
*  WILL INCREASE AS YOU GAIN EXPERIENCE ......                               0620
STOP (THE END OF THE BACKGROUND JOB STREAM)                              MULT0621
 END OF DATA
PAGE 12
```

INDEX

A

ADDRESS COMPARE switch, 69
Adjusting printing controls, 63
Aligning forms, 60-63
  lateral, steps for, 60
    illustration, 62
  ribbon shield, use of, illustration, 61
  vertical, 60, 63
American Standard Code for Information Interchange, 103
Appearance of Model 30, 17-25
ASCII, 103
Assembler language for System/360, 34
Assignment statement, definition of, 124
ATTN commands to DOS console, four basic, 121

B

BAL as assembler language for System/360, 34
Base-displacement principle for addressing with System/360, 96-97
Basic structure and data flow of System/360 Model 30, illustration of, 18
Batch processing in foreground area, 108-109
BCD, 94
Binary numbering system, 93
  DECIMAL system, 94
BPI, definition of, 132
BPS dump, operator and, 151
Buffered device, meaning of, 27
"Burst mode," 104
BYPASS, avoiding use of without supervisor's permission, 138-139
Byte, meaning of, 25, 35, 92
"Byte Interleaf" mode, 104

C

CANCEL command, 121
Card punch errors, 128-129
  Chip Box, 128
  Punch checks, 128
  Punch Stop, 129
Card reader, procedure for readying for job run, 75
Card reader errors, 126-128
  feed stop, 126
  read checks, 126-127

Card reader errors (*cont.*)
stacker, 127
transport checks, 127-128
validity checks, 127
Card reader-punch, 41-44
procedure for preparing for operation, 44
2540 card reader-punch, illustration of, 43
2821 as connecting control unit, 42
Carriage control tape, definition of, 71
mounting, 64-65
tape tension, testing, 64
CCW, 105
Channels employed with System/360, 104-105
overlapping, 104-105
"flip-flop" technique, 104
program, 105
Characteristics of equipment compatible for use with Model 30, 18-25
CHECK CONTROL switch, 69
CHECK RESET key, 68
COBOL, 34, 72, 112
Command statement, definition of, 124
COmmon Business Oriented Language, 34, 72, 112
Communication with DOS, 115-116
reasons for, 116
Compiler program, meaning of, 34
Console bypass for error recovery, 140-142
device error recovery, 141-142
IPL error messages, 141
Console control panel, mechanical operation of, 66-71
illustration, 67
keys, names and functions of, 66-68
switches, names and functions of, 66-70
Console typewriter, 31-33
control panel, illustration of, 32-33
illustration, 32
mechanical aspects of, 36-41
keys, 40-41
1052 keyboard-printer, illustration of, 37-39
set-up procedures, 41
switches, 40
1052 Printer-Keyboard, illustration of, 31
Control panel, 31-33, 35
illustration, 32-33
1052 Printer-Keyboard, illustration of, 31
and operator's console, illustration of, 32
CORE dump, operator and, 151
Core storage organization in Disk Operating System, illustration of, 113
C.P.U. organization, 95-100
fixed field length operations, 97-99
floating point operations, 99-100
variable field length operations, 96-97
instructions, three forms of, 96
CSW, 105

D

Data storage, manual, operational procedure for, 78
DATA switches, 70
DEBE, operator and, 151
Decimal numbering system, 92
DELETE, avoiding use of without supervisor's permission, 138-139
Density of magnetic tape, 29, 132
Device error recovery, 125-134
card punch errors, 128-129
Chip Box, 128
Punch Checks, 128
Punch Stop, 129
card reader errors, 126-128
feed stop, 126
read checks, 126-127
stacker, 127
transport checks, 127-128
validity checks, 127
disk errors, 133-134
data check, 134
missing address marker, 134
no record found, 134
seek check, 134
select lock, 133-134
line printer errors, 129-131
end of forms, 130-131
form checks, 130
improperly placed output, 131
no form movement, 131
print checks, 130
runaway forms, 131
sync checks, 131
tape errors, 132-133
data check, 132
load-point errors, 132-133 (see also "Tape errors")
Disk errors, 133-134
data check, 134
missing address marker, 134
no record found, 134
seek check, 134

Disk errors (*cont.*)
select lock, 133-134
Disk files, magnetic, 49-53
disk pack, loading *onto* disk storage drive, 52-53
disk pack, removing from disk storage drive, steps for, 51
Disk Operating System, introduction to System/360, 91-124
channels, 104-105
overlapping, 104-105
program, 105
communicating with, 115-116
reasons for, 116
C.P.U. organization, 95-100
fixed field length operations, 97-99
floating point operations, 99-100
variable field length operations, 96-97
definitions, 123-124
exercises, 122-124
I/O devices, allocating, 116-119
assignment statements and command statements, 118-119
programmer logical units, 117-118
system logical units, 117-118
job entry, 114-115
job control language, 114-115
job step, meaning of, 114
job stream, illustration of, 115
job stream, meaning of, 91
multiprogramming, 105-111
batch processing in foreground area, initiating, 108
multiprogramming operation–batch processing in one foreground, 108-109
single program initiation, 109-111
single program termination, 111
numbering systems, 92-95
BCD, 94
binary, 93
decimal, bases and exponents, 92
EBCDIC, 95
hexadecimal, 92, 93-94
packed decimal, 95
operating procedures, 119-122
ATTN commands, four basic, 121
functions, five, of operator, 119
OPTION statement, 120
operating systems, 91
program execution, 100-104
status, 100-101
status word, 101-104

Disk Operating system (*cont.*)
supervisor program, 100
structure, 112-113
core storage organization, illustration of, 113
SUPERVISOR, 91
Disk storage drive, 29-31
meaning of, 35, 49-50
most popular form of auxiliary storage, 80
Disk storage and magnetic tape major storage devices, 28
DISPLAY key, 68
DOS Operating Guide, 135

E

EBCDIC dump, operator and, 151
EBCDIC system, 95
EOB character, 121
Equipments compatible for use with Model 30, 18-25
Error detection and recovery procedures, 125-143
console bypass, 140-142
device error recovery, 141-142
IPL error messages, 141
device error recovery, 125-134
card punch errors, 128-129 (see also "Card punch errors")
card reader errors, 126-128 (see also "Device error recovery")
disk errors, 133-134 (see also "Disk errors")
line printer errors, 129-131 (see also "Line printer errors")
tape errors, 132-133 (see also "Tape errors")
exercises, 143
restart and save procedures, 142-143
using internal system DOS hardware and software aids, 134-140
job control errors, 139-140 (see also "Hardware and software aids....")
monitor, 134-135
program identifiers, 135
system-to-operator messages, 135-139
Extended Binary Coded Decimal Interchange Code, 95
"External interrupt," 101

F

Feed clutch, definition of, 71
control on line printer, illustration of, 57

File-feed, definition of, 71
Fixed field length operations of System/360, 97-99
Fixed Partitioned Multiprogramming, 105-111
batch processing in foreground area, initiating, 108
multiprogramming operation—batch processing in one foreground, 108-109
single program initiation, 109-111
termination, 111
"Flip-flop" technique of buffering data, 104
Floating point operations of System/360, 99-100
Floating Point Register, meaning of, 35
Forms stacker mechanism of line printer, adjusting, 65
FORmula TRANslation, 34, 72, 112
Functions, five, of operator of DOS, 119

G

General Register, meaning of, 35

H

Hardware and software aids, Internal system DOS, error recovery using, 134-140
job control errors, 139-140
assignments, improper, 139-140
job end or beginning, improper, 139
monitor, 134-135
program identifiers, 135
system-to-operator messages, 135-139
action, proper, five steps for, 137-139
Hexadecimal numbering system, 92, 93-94
Higher-order language, 72-73
Hollerith zoned data, 95
Hopper, definitions of, 71
Human-oriented language, 72-73

I

IBM System/360 Disk Operating System: Operating Guide (C24-5022), 116
IBM 2030 as processing unit for System/360 Model 30, 25-26
IBM 2401 Magnetic Tape Unit, illustration of, 29
IBM 2540 Model 1 card read-punch unit, illustration of, 27
IGNORE, avoiding use of without supervisor's permission, 138-139
Initialization of job, 73-76
card reader, procedure for readying, 75
line printer, readying, 75-76
system power, procedure for turning on, 74
Tape Unit Caution, 74
Input data, meaning of, 72
Input-output devices, 26-28
Input tape, definition of, 132
INSTRUCTION ADDRESS switches, 69
Instruction set, meaning of, 35
INTERRUPT key, 68
Interval timer on console control panel, 68
Introduction to System/360, 17-35
appearance of Model 30, 17-25
basic system structure and data flow, illustration of, 18
control panel and operator's console, 31-33
control panel, illustration of, 32-33
illustration, 32
1052 Printer-Keyboard, illustration of, 31
equipments compatible for use with the Model 30, 18-25
input-output devices, 26-28
1403 Model N1 printer, 28
2540 Model 1 card read-punch unit, 27
processor, 25-26
byte, meaning of, 25
multiplexer channel, 25
selector channel, 25
reason for choice of Model 30, 17
registers, 34
compiler program, 34
universal set of instructions, meaning of, 34
storage media, 28-31
2311 Disk Storage Drive, illustration of, 30
2401 Magnetic Tape Unit, illustration of, 29
I/O devices in Model 30 DOS, allocating, 116-119
assignment statements and command statements, 118-119
programmer logical units, 117-118
system logical units, 117-118
"I/O interrupts," 101
IPL, definition of, 123
commands, 120

J

Job Control, definition of, 123-124
commands, 120
Job entry with DOS, 114-115
job control language, 114-115
job step, meaning of, 114

Job entry with DOS (*cont.*)
job stream, illustration of, 115
Job stream, meaning of, 91
sample, operator and, 149-151
Joggler gate, definition of, 71

K

Keyboard printer, mechanical aspects of, 36-41
keys, 40-41
1052 keyboard-printer, illustration of, 37-39
set-up procedures, 41
switches, 40
Keys of console control panel, names and functions of, 66-68

L

LAMP TEST key, 68
Lateral alignments of forms in the printer, 60
illustration, 62
Line printer, mechanical operation of, 53-66
aligning forms, 60-63
lateral, 60
lateral adjustment, illustration of, 62
ribbon shield, use of, illustration of, 61
vertical, steps for, 60, 63
vertical adjustment, illustration of, 63
carriage control tape, mounting, 64-65
tape tension, testing, 64
checking and running, final, 65-66
feed clutch control, illustration of, 57
forms stacker mechanism, adjusting, 65
illustrations of, 54, 55
loading forms, 56-60
mechanism, becoming familiar with, 54-56
print unit, illustration, 58
printing controls, adjusting, 63
tractors, illustration of, 59
Line printer, readying for job run, 75-76
Line printer errors, 129-131
end of forms, 130-131
form checks, 130
improperly placed output, 131
no form movement, 131
print checks, 130
runaway forms, 131
sync checks, 131
LOAD key, 68
LOAD UNIT switches, 70
Loading forms into line printer, 56-60
Loading forms into line printer (*cont.*)
feed clutch control, illustration of, 57
print unit, illustration of, 58
tractors, illustration of, 59
LOCAL/REMOTE switch on console control panel, 66
LOG command, 121
Logs, operators and, 144-145

M

"Machine check interrupt," 101
Machine language, meaning of, 34
Magnetic disk files, 49-53
disk pack, loading *onto* disk storage drive, steps for, 52-53
disk pack, removing from disk storage drive, steps for, 51
Magnetic tape control, operator and, 148-151
Magnetic tape and disk storage major storage devices, 28
Magnetic tape unit, mechanical aspects of operation of, 45-49
leader, stripping off, 46, 47
manual control of, steps for, 49
meaning of, 35
mounting, 46-49
threading, 47, 48
2401 tape feed, illustration of, 45
MAIN STORAGE ADDRESS switch, 69
Manual control of tape station, steps for, 49
Mechanical aspects of operation, 36-71
card reader-punch, 41-44
procedure for preparing for operation, 44
2540 card reader-punch, illustration of, 43
2821 as connecting control unit, 42
console control panel, 66-71
illustration, 67
keys, names and functions of, 66-68
switches, names and functions of, 66-70
console-typewriter, 36-41
keys, 40-41
1052 keyboard printer, illustration of, 37-39
set-up procedures, 41
switches, 40
definitions, 71
line printer, 53-66
aligning forms, 60-63 (see also "Aligning forms")
carriage control tape, mounting, 64-65
checking and running, final, 65-66

Mechanical aspects of operation (*cont.*)
feed clutch control, illustration of, 57
forms stacker mechanism, adjusting, 65
illustrations of, 54, 55
loading forms, 56-60
mechanism, becoming familiar with, 54-56
print unit, illustration of, 58
printing controls, adjusting, 63
tractors, illustration of, 59
magnetic disk files, 49-53
disk pack, loading *onto* disk storage drive, steps for, 52-53
disk pack, removing from disk storage drive, steps for, 51
magnetic tape unit, 45-49
leader, stripping off, 46, 47
manual control of, steps for, 49
mounting, 46-49
threading, 47, 48
2401 tape feed, illustration of, 45
Mnemonics, meaning of, 34
Model 30, reason for choice of as model for instruction, 17
Mounting tape, procedure for, 46-49
Multiplexer channel, 25, 85-86, 104-105
meaning of, 35
Multiprogramming with System/360, 105-111
batch processing in foreground area, initiating, 108
multiprogramming operation—batch processing in foreground, 108-109
single program initiation, 109-111
termination, 111

N

Nine-track tape drives, 29
"No ring—no write" as rule for using magnetic tape, 45
NOLOG command, 121
Nomenclature of equipments compatible for use with Model 30, 18-25
Non-process run out, meaning of, 44, 71
Numbering systems of System/360, 92-95
BCD, 94
binary, 93
decimal, bases and exponents, 92
EBCDIC, 95
hexadecimal, 92, 93-94
packed decimal, 95

O

Object program, meaning of, 73
Off-line, definition of, 71
On-line, meaning of, 32, 36, 71
1052 keyboard printer, 31, 36-39
illustration, 31, 37-39
1403 Model N1 printer, illustration of, 28
OpCode, definition of, 123
Operating procedures of DOS, 119-122
ATTN commands, four basic, 121
functions, five, of operator, 118-119
OPTION statement, 120
Operating System Dump Restore, operator and, 152-153
Operating systems, 91
Operational procedures for running job, 76-79
data storage, manual, procedure for, 78
operator, function of, 77
summary of, 79-80
Operator, five steps for actions of in error recovery, 137-139
Operator practice, general, 144-154
DEBE, 151
job running, 146-148
job streams, sample, 149-151
logs, 144-145
magnetic tape control, 148-151
OS Dump Restore, 152-153
priorities, 145-146
SEREP, 152
storage print, 151
system exercise, operator-training, 153-154
UCS buffer load, 130, 152
utility programs, 151
Operator's console, 31-33
control panel, illustration of, 32-33
illustration, 32
1052 Printer-Keyboard, illustration of, 31, 37-39
OPTION statement, 120
Organization, preliminary, for running job, 72-73
OS Dump Restore, operator and, 152-153
Output tape, definition of, 132
Overlapping of processing, 104-105
Over-punching card, validity checks caused by, 127

P

Packed decimal system, 95
Parity byte, meaning of, 78, 92

PAUSE command, 121
Payroll checks, tape programs to print, sample, 83-90
device addresses, typical, 87
readying tape unit, 85-87
run sheet, 86
schedule, 85
Preliminary organization for running job, 72-73
Print density, definition of, 71
adjustment, 63
Print unit of line printer, illustration of, 58
Printer-keyboard, 31-33
control panel, illustration of, 32-33
illustration, 32
meaning of, 35
1052 Printer-Keyboard, illustration of, 31, 37-39
Printing controls, adjusting, 63
Priorities, operator and, 145-146
Processor, 25-26
Program, meaning of, 72
"Program check interrupt," 101
Program execution on System/360, 100-104
status, 100-101
"I/O interrupts," 101
status word, 101-104
supervisor program, 100
Program identifiers in DOS system, 135
Program Status Word, definition of, 123
PSW, 101-104

R

RATE switch, 68-69
Records, meaning of, 28
Recovery procedures, error detection and, 125-143 (see also " Error detection and recovery procedures")
Register forms, tape programs to print, sample, 83-90
device addresses, typical, 87
readying tape unit, 85-87
run sheet, 89
schedule, 85
Registers, 34
Restart for error recovery, 142-143
Ribbon shield, use of for alignment, 60, 61
ROS ADDRESS switches, 69
ROS CONTROL switch, 69
Run book, definition of, 44, 73
Running a job, 72-90
Running a job (*cont.*)
initialization, 73-76
card reader, procedure for readying, 75
line printer, readying, 75-76
system power, procedure for turning on, 74
Tape Unit Caution, 74
operational procedures, 76-79
data storage, manual, procedure for, 78
operator, function of, 77
operator and, 146-148
preliminary organization, 72-73
input data, meaning of, 72
program, meaning of, 72
run book, 73
sample jobs for practice, 80-90
payroll checks and register forms, tape programs to print, 83-90
sales report, card program to print, 80-83
summary of console operator's role, 79-80

S

Sales report, card program to print, sample, 80-83
run sheet, 82
schedule for card program, 81
Scratch tape, definition of, 132
SELECT STORAGE switch, 69
Selector channel, 25, 85-86, 104-105
meaning of, 35
SEREP, operator and, 152
Set-up procedures for console-typewriter, 41
Seven-track tape drives, 29
Single program initiation with System/360, 109-111
termination, 111
Source program, 73
START key, 67-68
Status switching, 100-101
STOP key, 67
Storage media, 28-31
Storage print, operator and, 151
STORE key, 68
Structure of Disk System, 112-113
core storage organization, illustration of, 113
"Supervisor call interrupt," 101
Supervisor program, 91, 100
Switches of console control panel, names and functions of, 66-70

SYSRDR, definition of, 124
System Error Report, operator and, 152
System exercise, operator-training, 153-154
System Logical Units, definition of, 124
SYSTEM RESET key, 67

T

Tape errors, 132-133
    data check, 132
    load- point errors, 132-133
        pinched tape, 133
        runaway tape, 133
        tape that will not load, 133
        wrong density or mode, attempting to load at, 133
Tape leader, stripping off, 46, 47
Tape mark, definition of, 132
Tape Operating System, 92
Tape programs to print payroll checks and register forms, sample of, 83-90
    device addresses, typical, 87
    readying tape unit, 85-87
    run sheet of payroll check program, 86
    run sheet of register program, 89
    schedule, 85
Tape reels, 45-49
    leader, stripping off, 46, 47
    manual control, steps for, 49
    mounting, 46-49
    threading, 47, 48
    2401 tape feed, illustration of, 45
Tape tension, testing, illustration, 64
Tape Unit Caution, 74
Threading tape, 47, 48
360 MOD 30 360 DOS Training Manual, 80*n*, 83*n*
TOS, 92
Tractors on line printer, illustration of, 59
2311 Disk Storage Drive, 30, 50, 112
    illustration of, 30
2401 Magnetic Tape Unit, illustration of, 29
2401 tape feed, illustration of 45
2540 card reader-punch, 41-44
    illustration of, 43
    procedure for operating, 44
    2821 as connecting control unit, 42

U

UCS buffer load, 130
    operator and, 152
Universal Character Set, operator and, 130, 152
Universal set of instructions, meaning of, 34
Utility programs, operator and, 151

V

Variable field length operations of System/360, 96-97
    instruction, three forms of, 96
Vertical alignments of forms in printer, steps for, 60, 63
    illustration, 63

W

Word of hexidecimal characters, 94
Work tape, definition of, 132